

A Semi-Analytical Model for Gravitational Microlensing

by

Paul Robin Brian Chote

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the requirements for the degree of
Master of Science in Physics.

Victoria University of Wellington
2011

Abstract

This thesis describes the theory and implementation of a semi-analytical model for gravitational microlensing. Gravitational microlensing is observed when a distant background ‘source’ star comes into close alignment with an intermediate ‘lens’ star. The gravitational field of the lens deflects the paths of light emitted from the source, which causes an increase in its observed brightness. As the alignment of the two stars changes with time, the apparent magnification of the source follows a well defined ‘lightcurve’. A companion body (such as a planet) orbiting the lens star can introduce large deviations from the standard lightcurve, which can be modelled to determine a mass ratio and separation for the companion(s). This provides a means to detect extrasolar planets orbiting the lens star.

We show, from basic principles, the development of the standard model of a microlensing event, including the effect of multiple lens masses and orbital motion. We discuss the two, distinctly different, numerical approaches that are used to calculate theoretical lightcurves using this model. The ‘ray shooting’ approaches are discussed with reference to the previously developed modelling code (MLENS), which implemented them. This is followed by a comprehensive description of the ‘semi-analytical’ approaches used in the new software (mlens2) developed during this thesis programme; a key feature of these techniques is the determination of the source magnification from the roots of a high order polynomial. We also discuss the process of finding the best-fit model for an observed microlensing event, with respect to the mlens2 software package. Finally, we demonstrate the capabilities of our semi-analytical model by generating theoretical lightcurves for the microlensing events OGLE-2005-BLG-390 and OGLE-2006-BLG-109 and comparing them to the observational data and published models.

Contents

Preface	vii
Acknowledgements	xi
Introduction	xiii
1 Microlensing Theory	1
1.1 The Einstein Ring	2
1.2 Single Lens Microlensing	5
1.3 Source Motion	8
1.4 Compound Lenses	11
1.4.1 Lens Equation	11
1.4.2 Image Magnification	13
1.4.3 Image Topology	15
1.5 Finite Source Effects	20
1.6 Orbital Motion	23
1.6.1 Orbital Parallax	23
1.6.2 Xallarap	31
1.6.3 Lens Motion	32
2 Numerical Techniques	35
2.1 Inverse Ray-Tracing	36
2.1.1 Ray Shooting	36
2.1.2 Magnification Maps	40
2.2 Semi-analytical techniques	42
2.2.1 The Lens Polynomial	42
2.2.2 Critical Curves	44
2.3 Hexadecapole Approximation	45
2.4 Polygon Method	46
2.5 Hybrid techniques	56

3	Modelling Procedures	57
3.1	mlens2	58
3.2	Algorithm Selection	59
3.2.1	Caustic Distance Algorithm	60
3.2.2	Magnification Curvature Algorithm	61
4	Modelling Selected Events	65
4.1	OGLE-2005-BLG-390	65
4.2	OGLE-2006-BLG-109	68
5	Discussion and Conclusions	73
A	The Jacobian	75
B	Lens Polynomial Coefficients	79
C	Critical Curve Polynomial Coefficients	85
D	mljob Program Structure	89
E	Control Files	93

List of Figures

1.1	Simple microlensing geometry	3
1.2	Geometry forming an Einstein ring	4
1.3	Images of a single lens	6
1.4	Area distortion by a single lens	7
1.5	Source motion geometry	9
1.6	MOA-2010-BLG-407 lightcurve	10
1.7	MOA-2009-BLG-387 lightcurve	11
1.8	Jacobian topology for a single lens	16
1.9	Jacobian topology for perfect source and lens alignment	17
1.10	Jacobian topology for a binary lens	18
1.11	Magnification of finite sources	20
1.12	Limb darkening geometry	21
1.13	Limb darkening intensity profiles	22
1.14	Simple parallax geometry	23
1.15	Effect of parallax on the Einstein ring	24
1.16	Lens, source and observer planes	25
1.17	Geometry of an elliptical orbit	27
1.18	Orbital and observer planes	28
1.19	Orbital position angles	29
1.20	Single-lens parallax lightcurves	31
1.21	Geometry of lens motion	33
2.1	Ray shooting a single lens	37
2.2	Even-odd rule	38
2.3	Example of pixel reduction algorithm	39
2.4	MOA-2009-BLG-387 magnification map	41
2.5	Hexadecapole point placement	46
2.6	Finite source caustic crossing magnifications	47
2.7	Polygon area calculation	48
2.8	Polygons constructing an Einstein ring	49
2.9	Polygon vertex placement	51

2.10	Polygon curve segments	53
2.11	Polygon curve vertices	53
2.12	Limb darkened polygon images	54
2.13	Polygon limb annulus profiles	55
3.1	OGLE-2005-BLG-390 calculation regions (caustic distance)	61
3.2	OGLE-2006-BLG-109 calculation regions (caustic distance)	62
3.3	OGLE-2006-BLG-109 lightcurve error	62
3.4	OGLE-2006-BLG-109 calculation regions (magnification curvature)	64
3.5	OGLE-2006-BLG-109 lightcurve (calculation types)	64
4.1	OGLE-2005-BLG-390 lightcurve	66
4.2	OGLE-2005-BLG-390 lightcurve comparison	67
4.3	OGLE-2006-BLG-109 lightcurve	69
4.4	OGLE-2006-BLG-109 source track	70
4.5	OGLE-2006-BLG-109 lightcurve (no orbital motion)	70
4.6	OGLE-2006-BLG-109 lightcurve (VUW model)	71
4.7	OGLE-2006-BLG-109 source track (VUW model)	71
D.1	Public interface of SourceTarget class	90
D.2	Public interface of GlobalMemory class	91
E.1	OGLE-2005-BLG-390 Control file	94
E.2	OGLE-2005-BLG-390 MLENS Control file	95
E.3	OGLE-2006-BLG-109 Control file	96
E.4	OGLE-2006-BLG-109 MLENS Control file	97

Preface

My first introduction to microlensing was in the summer of 2007-2008, when I took part in a university funded research project with my (now) supervisor, Prof. Denis J Sullivan (DJS). At the time, the microlensing group at VUW consisted of DJS and Dr Aarno Korpela, who had recently finished a PhD programme where he developed a set of modelling codes called MLENS.

Massively parallel computing is an important part of modelling gravitational microlensing events. The huge parameter space involved requires many millions or billions of separate calculations to be made. If run sequentially on a single computer, these calculations could easily take months or years. However, each calculation is independent, which allows the workload to be divided into smaller chunks and run in parallel over many computing nodes. This can reduce the overall calculation time to the order of hours or days.

At the time, the microlensing group at VUW had access to three parallel computing facilities. The *SGE grid* (running the Sun Grid Engine software) was a ‘cycle stealing’ grid within the School of Mathematics and Computer Science. Software running in the background of (approximately) 100 NetBSD workstation computers made them available for computations whenever a physical user is absent. Due to restructuring, this grid is now known as the *ECS grid*, managed by the School of Engineering and Computer Science. The second available facility was the *Condor grid*, managed by the university’s Internet Technology Services. It is now known as the *SCS grid*, also managed by the School of Engineering and Computer Science. Condor is another cycle stealing grid, running on 1000 dual core student computers running Windows XP. The third available facility was the *Blue Fern* supercomputer at Canterbury University, consisting of a two rack Blue Gene/L system with 4096 cores, plus an additional IBM p575 cluster for testing and calculations in its own right.

One goal of the summer project was to investigate whether MLENS (which was designed to use the SGE grid exclusively) could be modified to make use of the extra capabilities available with Blue Fern or Condor. We determined fairly quickly

that Blue Fern was unsuitable. Supercomputers like the IBM Blue Gene series focus on fast communication between relatively weak computing nodes. This makes them excellent for calculations that require frequent communication between the nodes, such as distributed differential equations, but is unnecessary for our purposes. MLENS would have required significant changes to efficiently use this architecture, so we decided to focus our efforts on the Condor grid instead.

Condor operates similarly to SGE but with an order of magnitude more computers. The main difference between Condor and SGE is in the operating systems and file sharing. On SGE, every computer in the network is equivalent. Tasks can be issued and run on any computer, with a common file system available across the network for data and communication. Condor, on the other hand, has a master server running Red Hat Linux, which sends a self contained task (including all the files required) to the computation node running Windows XP. Communication back to the master server only occurs once the task has been completed. We determined that with a moderate amount of effort, MLENS could be modified to run well on Condor. The work required would be far beyond the scope of the summer project, so the idea was put on hold.

Over this summer period, the event OGLE-2006-BLG-109 was being actively modelled by the international microlensing community. This event was particularly challenging to model and required capabilities that the MLENS code did not support: three lens masses, plus orbital motion in the lens and observer. My own interest in the underlying numerical techniques started with discussions of how the ray-tracing code worked, and what would be required to make it support the third lens. By the end of the project, Dr Korpela had added support for the third lens, and we were able to run basic models of the event.

I followed my interest in the numerical ray-tracing technique by participating in an Honours project with DJS, developing a simple ray-tracing model similar to the one used in MLENS. This was used to illustrate how the individual images changed during the OGLE-2006-BLG-109 event, causing the dramatic changes in light intensity seen over the course of the event.

This was followed by a year long break from microlensing as I finished my BSc Honours degree. My efforts in this endeavour were rewarded with a VUW Masters by Thesis Scholarship, allowing me to embark on this thesis.

This thesis work started in July 2009. The original direction of the project was to follow up on the work from the summer scholarship; changing the MLENS code to run on the Condor grid and then to be involved with active modelling of new microlensing events. The first 2 or 3 months were spent understanding the MLENS

code, commenting and reorganising it to make it clearer to myself and others. This turned into a restructuring of the entire MLENS code-base, which took several months. While focusing on perfecting the code, I lost focus of the larger picture which was to create something that worked and then to use it to model events.

During this time, it became increasingly obvious that we needed to be able to include orbital motion effects in our models, particularly the parallax introduced by the Earth's motion around the Sun. These effects are being increasingly modelled on new events, as they can provide the extra information required to constrain the mass of the lens star and any planets that may be orbiting it. The theory that describes these effects is tricky, and progress on understanding and developing this theory was slow. Additionally, the MLENS code made several core assumptions on the source motion which made it difficult to include these effects in the code. The central ray-tracing calculations would have needed rewriting to include parallax, and so I was reluctant to do this.

As such, when we attended the 14th workshop on Gravitational Microlensing held in Auckland in late January 2010, our progress had essentially stalled and we had no results of any significance to discuss. This conference marked a turning point in the thesis. After discussions with other modelling groups, we decided to abandon our existing ray-tracing approach in favour of the semi-analytical methods that several modelling groups had used with great success. This approach is radically different to ray-tracing, and required us to understand new theory that we had previously avoided. The following months were spent furiously reading papers, understanding the theory involved, and extending and developing it into the techniques discussed within this thesis. This meant abandoning the original goal of modelling events; the time and effort required to build a fully functional modelling code, and then to model events with it, is more within the scope of a PhD than a Masters programme.

Now, sixteen months after embarking on this thesis, I have ended up at a far more satisfying position than the original topic direction would have given. The new modelling code (called `mlens2`) that I have built uses several new semi-analytical techniques to model lightcurves for a set of model parameters, including the extra parameters used to model orbital motion within the observer or lens. The functionality from the original MLENS to search parameter space for the best set of parameters has not yet been reimplemented. In the first stage of the project, I had developed a general wrapper that allows the code to be run efficiently on SGE, Condor, or a single computer with multiple processors. This needs to be integrated with the new modelling code, plus new code to control the stochastic search of parameter space, and would be within the scope of a summer or Honours project for a student with a strong programming background.

Acknowledgements

First and foremost I would like to thank my supervisor Denis Sullivan (DJS) for his supervision, guidance and unique wit in the time we have spent together. Also to VUW for the Masters by Thesis scholarship, without which I could not have done this work. To the other members of the VUW microlensing group, my thanks go to Aarno Korpela for the original MLENS software which I have spent so much time improving, and to Pauline Harris for the discussions and many editorial comments on my thesis. I would like to acknowledge and thank Michael Miller for his contributions (detailed in Appendix B) towards the lens polynomial coefficient expressions. Also, the School of Engineering and Computer Science, and in particular Kevin Buckley for access to, and support for the grid computing systems at VUW. Last, but not least, I would like to thank my friends and family for their support over the past 16 months. It has been a long, but rewarding, journey.

Introduction

Gravitational microlensing uses the slight deflection that light undergoes as it passes near a massive body to probe for planets around distant stars within our galaxy. The idea of being able to detect planets like this is recent, but the idea that gravity itself can affect light is not.

The deflection of light by gravity was first proposed by Newton as one of the queries at the conclusion of his treatise *Opticks* (Newton et al. [1718]). The very first of these queries asks

“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?”

The phrasing of this question (“Do not...”) reads as if this was the obvious conclusion, and that Newton was daring someone to prove him wrong.

The first derivation of an equation for this deflection was a century later in 1804, by Johann Soldner. Soldner used Newton’s corpuscular model of light to show that a light ray grazing the limb of our Sun should undergo a deflection of

$$\alpha = \frac{2GM}{c^2 R} \approx 0.87 \text{ arcseconds.}$$

It was not long after this result that Young’s famous double slit experiment showed that light acted, not as a particle, but as a wave. Newton’s corpuscular model for light was discarded, and this idea of deflection was seemingly invalidated.

A further century passed before Einstein revisited this idea with his general theory of relativity. His first attempt, based on his equivalence principle between gravity and acceleration, predicted the same result that Soldner derived. Under his completed theory, however, he predicted a deflection of *twice* the earlier result

$$\alpha = \frac{4GM}{c^2 R} .$$

This prediction was confirmed by two separate expeditions to observe the 1919 solar eclipse, and became one of the key confirmations of Einstein’s new theory.

The idea that this bending could lead to a lens-like effect was begrudgingly published by Einstein in 1936 (Einstein [1936]), at the request of R. W. Mandl. He showed that a distant background star would appear as a “luminous ring” when perfectly aligned with an intermediate star. He commented that

“Of course, there is no hope of observing this phenomenon directly”

and that, instead, the effect will manifest itself as an increase in the apparent brightness of the background star.

This effect is now known as gravitational lensing, and the luminous ring that Einstein describes is termed the *Einstein ring*. Einstein’s calculations, however, did not account for the extremely large distances and masses involved with galaxies. At the very large scale, entire galaxies can act as gravitational lenses, causing multiple distorted, *visible*, images of the background to be seen. The position and size of these images can be used to measure the mass of the intermediate galaxy and to infer the presence of dark matter.

At the other end of the scale, gravitational *microlensing* occurs within our galaxy, with individual stars like Einstein first described. In particular, when looking towards the Galactic bulge (where the density of stars is greatest), these distant background stars are now routinely observed to be lensed by intermediate stars in the Galactic disk, measured by their characteristic change in brightness with time. The timescale of these variations is of the order of days to months, making it practical to observe many events in parallel with wide angle surveys. The theory behind this is discussed in detail in Chapter 1.

The first application of gravitational microlensing to a physical scenario was the search for dark matter in the form of MACHOs (Massive Compact Halo Objects) in the halo of our galaxy. By observing a large number of stars in the large and small Magellanic clouds, these objects (should they exist) would act as gravitational lenses when they come into alignment with the background stars, leading to a characteristic change in brightness with time (Paczynski [1986]). The rate at which these events occur will be proportional to the density of MACHOs in the Galactic halo. After extensive observations, it was found that a significant proportion of the dark matter in our galaxy *cannot* be attributed to MACHOs (see e.g. Alcock et al. [1997], Tisserand et al. [2007]).

The idea of using microlensing to search for extrasolar planets was first suggested by Mao and Paczynski [1991]. If the lens star has a planet near the Einstein ring (typically 2 – 4 AU for events within our galaxy) the gravitational influence of the planet can cause significant deviations from the standard microlensing lightcurve (the measure of brightness with time). These deviations can be very dramatic,

causing the brightness to change by multiple orders of magnitude over a timescale of a few hours, in a manner characteristic of the mass ratio and orbital radius of the planet (among other parameters). The deviations can thus be modelled to extract these parameters.

Current microlensing surveys continuously monitor large areas of the Galactic plane, and hundreds of events are observed per year. For example, the MOA (Microlensing Observations in Astrophysics) group observes 50 Million stars towards the Galactic bulge, and a further 55 million towards the Magellanic clouds (Sumi [2010]). From these observations, approximately 500 microlensing events are seen per year towards the Galactic bulge, and two per year towards the Magellanic clouds.

To date, ten planets have been discovered from nine observed planetary events (Bond et al. [2004], Udalski et al. [2005], Beaulieu et al. [2006], Gould et al. [2006], Gaudi et al. [2008], Bennett et al. [2008], Sumi et al. [2010], Dong et al. [2009], Janczak et al. [2010]).

Microlensing has several advantages as a planetary detection technique, which make it complementary to other methods. In contrast to most other techniques, it doesn't rely on measuring light from the planet or its host star. This makes it possible to detect planets around distant stars, and faint or non-luminous objects such as brown dwarfs or even free-floating planets. It also provides an instantaneous snapshot of the mass distribution within the lens system, which means it can be sensitive to planets orbiting far from the host star without requiring observations over a full orbital period.

The biggest downside of microlensing is that it is difficult to extract key physical information from observed events. For example, a planet to star mass ratio can be confidently deduced from an appropriately perturbed light curve, but the individual masses can normally only be inferred from statistical values. Furthermore, the one-off nature of events means that they cannot be verified with repeat observations. This places a large emphasis on collecting as much high quality observational data as possible over the course of the event.

This is achieved by the two-tier nature of the microlensing observing groups. Two survey groups, MOA and OGLE (Optical Gravitational Lensing Experiment, recently inactive), regularly survey a large number of stars, and have automated systems in place to detect potential microlensing candidates. Candidate events are alerted to the follow-up groups, μ FUN-PLANET, Robonet-II and MiNDSTEp, which have many $\sim 1\text{m}$ or smaller aperture telescopes to intensively monitor the individual events.

The observed events must then be modelled. The simplest single lens models are very

simple, and can be determined with simple automated algorithms. More complex events, such as those with composite lenses with multiple bodies, are much more complicated and require an intensive numerical search of the huge parameter space involved. This is only feasible with the advent of readily available high performance parallel computing resources.

This thesis is concerned with building a model that generates theoretical lightcurves from a specified set of model parameters. This lightcurve is then compared with the observed data as part of the process of identifying the optimum model to fit the data. The higher level structure required to search the parameter space for the best fitting model (using the χ^2 fit statistic) has not been specifically included in this thesis work, and remains as an element of required future work.

Chapter 1 describes the background theory required to model microlensing events. The simplest single lens model is derived, and used as a basis for describing the complex scenarios that arise for lenses with multiple components. The modifications required to model stellar limb darkening in the source, and orbital motion of the bodies involved are then worked through; orbital motion is a particular focus in this thesis, as it was not available in the previous VUW modelling code (MLENS).

Chapter 2 focuses on the numerical techniques required to solve the models produced in Chapter 1. A brief overview of the ray-shooting and magnification map techniques as implemented in the original MLENS code is provided, and contrasted with the semi-analytical techniques developed as part of this thesis programme.

Chapter 3 gives a brief overview of the new modelling code (mlens2). The current state of the code is discussed with reference to the original goals, along with the requirements to be a usable and useful addition to the VUW microlensing programme.

Chapter 4 demonstrates the capabilities of the mlens2 modelling code. Lightcurves and χ^2 fits are calculated for the planetary events OGLE-2005-BLG-390 and OGLE-2006-BLG-109. The results from mlens2 are compared with the original MLENS results, and to the observational data.

The five appendices cover a collection of sub-topics which do not fit into the main flow of the thesis, but provide useful information to those interested in understanding the details of the mathematics or creating their own modelling code.

Appendix A gives a brief overview on the Jacobian matrix and its determinant, which plays a key role in obtaining the source magnification in the semi-analytical models.

Appendices B and C contain a list of analytical expressions for calculating the poly-

nomial coefficients required to solve for the image and critical curve positions.

Appendix D gives a brief overview to the *mljob* executable within *mlens2*, which calculate the model fits and lightcurves for a specified *control file* containing event parameters.

Appendix E contains a selection of sample control files used to generate model lightcurves from both *mlens2* and the original MLENS.

Chapter 1

Microlensing Theory

At first glance, the theory behind gravitational lensing may seem hopelessly complex. A large number of directly unobservable parameters conspire to produce a given lightcurve, with no guarantee that any given solution is correct or unique. Thankfully, all this complex behaviour that can be related back to one simple fact: the path that a ray of light follows through space is affected by a gravitational field like a physical body with mass. After several appropriate approximations are made, the underlying relationships can be understood with simple vector algebra. This does not mean that the theory can be dismissed as trivial however; the fundamentals may be simple, but the devil is in the details.

The purpose of this chapter is to give the reader an overview of the many layers of detail involved. We start by introducing the Einstein ring, observed when a background ‘source’ star is perfectly aligned with a foreground ‘lens’ object. This forms a basis to discuss the consequences of the more common case, where the source and lens are offset. A model for the relative motion of the source and lens is introduced, which provides an explanation for how the observed magnification changes with time.

Having covered the basic theory, the focus moves to the deviations that are of particular significance for modelling planetary microlensing events. First, the underlying theory is expanded to cover lenses containing multiple bodies, such as a star with an orbiting planet. This adds significant complexity to the theoretical and modelling process. Next, we relax the assumption of the source as being a point source of light. Finally, the effects of orbital motion within the source, lens, and the observer are dealt with.

1.1 The Einstein Ring

The strength of the gravitational field described by general relativity in the space outside an object can be characterised by its Schwarzschild radius R_s . This radius defines the event horizon of a (non-rotating, uncharged) black hole with the same mass as the object being considered and depends only on the mass of the object, providing a convenient scale with which to compare distances with mass.

The Schwarzschild radius for an object with mass M is

$$R_s = \frac{2GM}{c^2} , \quad (1.1)$$

where G is the Newtonian gravitational constant, and c is the speed of light.

Very dense objects like neutron stars can have a physical size comparable to its Schwarzschild radius. The full general relativistic field equations are required to describe the gravitational field immediately outside these sorts of objects. However, far from these, the gravitational field is much weaker and the simpler weak-field equations of general relativity can be used.

Typical stars have a much weaker gravitational field. The Schwarzschild radius of the Sun, for example, is only 3 km, while its physical radius R_\odot is approximately 10^6 km. This large difference means that the weak-field equations are valid everywhere outside it, and most other stars.

It can be shown using these weak-field equations that a photon travelling past a spherically symmetric body with a Schwarzschild radius R_s will be deflected from its original path by an angle

$$\alpha = \frac{2R_s}{b} , \quad (1.2)$$

where b is the impact parameter of the photon passing the body. The Schwarzschild radius is many orders of magnitude smaller than the inter-solar distances we are interested in, so we can assume that this bending occurs at a single point, when the photon is closest to the deflecting body. This is known as the thin-lens approximation.

Figure 1.1 shows this bending from a side-on view. A distant source star S a distance D_S from the observer O emits a photon at some angle ψ . An intermediate body L lies on the line between O and S , a distance D_L from the observer. The photon passes L with an impact parameter b , and is deflected by an angle α towards O by the gravitational field of L . The observer thus sees the photon arriving from an angle θ , apparently emitted from the offset position I .

Because O , L and S lie on the same line, the deflection behaviour will be symmetric around this common axis. The observer will see the source as a ring made up of

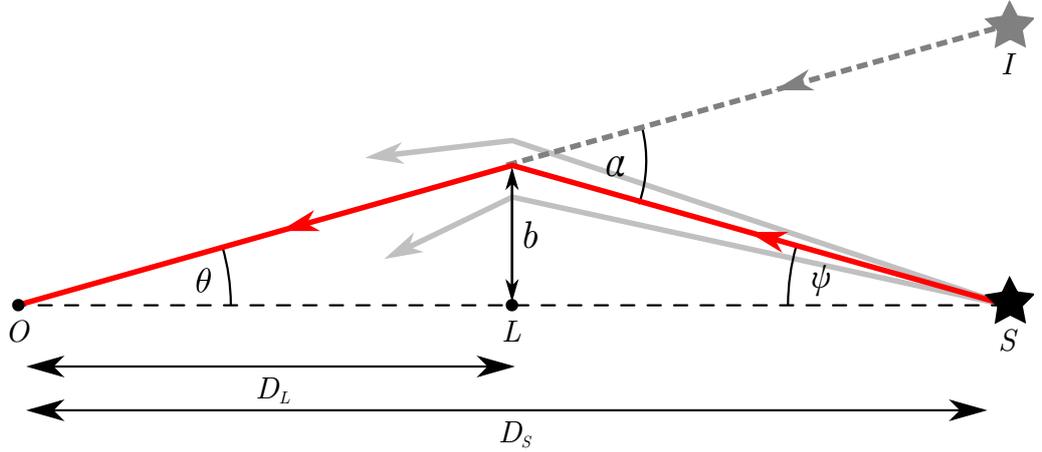


Figure 1.1: In the simplest microlensing geometry the observer, lens, and source are perfectly aligned. Photons emitted (red arrow) from the source at an angle ψ are deflected by the gravitational field of the lens, causing the observer to see a displaced and distorted image at an angle θ from the true source position. Unlike a true lens geometry, photons emitted at slightly different angles (light grey arrows) are not focused towards the observer.

the points I rotated around the line of sight. This action (shown in Figure 1.2) is similar to a lens focusing diverging light paths of an object at its focus, so L is commonly called the *lens*. This is a cute analogy, but a misnomer. A true lens would focus *all* light from the source impacting it, not just the rays passing with an impact parameter b .

The values of ψ , b , and θ for which a photon is deflected towards the observer are determined by the distances D_L , D_S , and the mass of L . Photons emitted from the source at any other angle will still be deflected by the lens, but not towards the observer.

The size of θ is found by relating the angles and distances in Figure 1.1 to give the relationships

$$\begin{aligned} \theta D_L &= (D_S - D_L) \psi \\ \alpha &= \frac{2R_s}{D_L \theta} \\ \psi &= \alpha - \theta . \end{aligned} \tag{1.3}$$

Additionally, a photon emitted from the source at an angle ψ will pass the lens with an impact parameter $b = (D_S - D_L) \psi$. By Equation 1.2, this produces a deflection of

$$\alpha = \frac{2R_s}{(D_S - D_L) \psi} .$$

The maximum possible deflection α will occur for a photon with an impact parameter

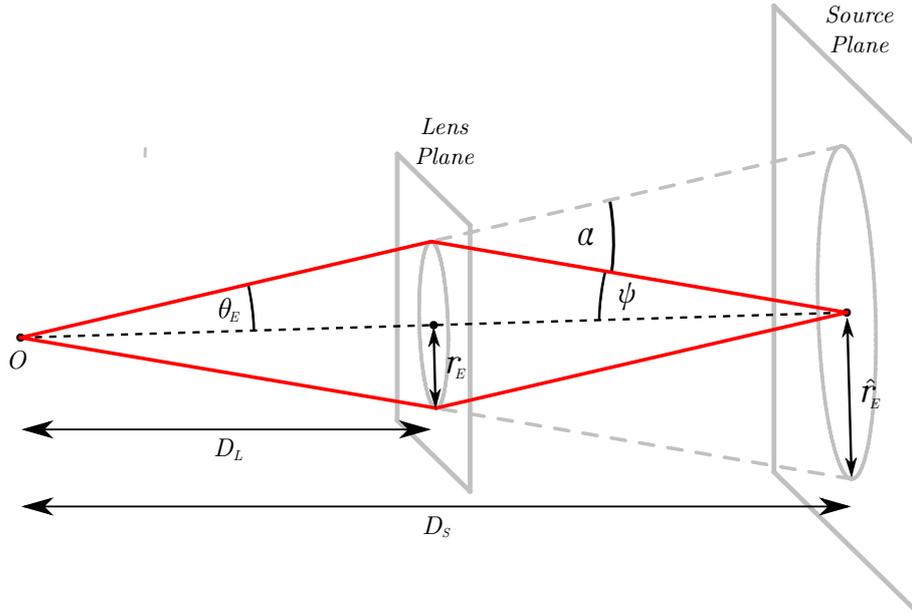


Figure 1.2: A lens object in perfect alignment with a background source focuses light emitted at an angle ψ towards the observer. The observer sees an Einstein ring of angular size θ_E .

equal to the physical radius of the lens. For the Sun, a typical star, the ratio of Schwarzschild radius to physical radius is approximately 10^{-6} . This corresponds to a deflection of $\alpha \approx 2$ arcseconds.

Combining the above equations gives an expression for θ

$$\theta^2 = 2R_s \frac{D_S - D_L}{D_S D_L} .$$

This result was first published by Einstein [1936], so the solution θ_E is known as the *Einstein angle*

$$\theta_E = \sqrt{2R_s \frac{D_S - D_L}{D_S D_L}} . \quad (1.4)$$

Similarly, the resulting image is known as an *Einstein ring*.

For a typical event within our galaxy, we can assume a source in the Galactic bulge, $D_S \approx 8$ kpc, with an intermediate lens at $D_L \approx 4$ kpc. Assuming a lens like our Sun, $R_S \approx 3$ km gives an Einstein angle of $\theta_E \approx 1$ milliarcseconds.

The Rayleigh criterion gives the (diffraction limited) minimum aperture size required to resolve the Einstein ring as

$$D = \frac{1.22\lambda}{\sin(2\theta_E)} . \quad (1.5)$$

For visible light ($\lambda \approx 550$ nm), this yields an aperture size of $D \approx 60$ m, which is too large to feasibly observe the Einstein ring. The only (potentially) detectable

result of this microlensing effect will be the change in brightness of the source due to the focusing of light by the lens.

The angular size of the Einstein ring is not observable, therefore angular units are not directly useful. We will see in the following sections that projected distances or vectors are much more convenient. We obtain the *Einstein radius* by projecting the Einstein angle towards the source, into a plane perpendicular to the line of sight

$$\begin{aligned} r_E &= \theta_E D_L \text{ (Lens plane)} \\ \widehat{r}_E &= \theta_E D_S \text{ (Source plane)} . \end{aligned} \tag{1.6}$$

The *lens plane* is located at the position of the lens. Under the thin lens approximation, all bending occurs when the photons pass through this plane. Similarly, the *source plane* is located at the position of the source. Any apparent motion of the source will occur within this plane. Angles projected into these planes will have different physical lengths, but can be considered equivalent if they are measured relative to the Einstein radius within the respective plane.

1.2 Single Lens Microlensing

The continual motion of stars in the sky means that, even if two stars are perfectly aligned at one time, they will not remain so. We should therefore consider what happens when the lens and source are misaligned by some angle β , as shown in Figure 1.3. For the photons from the source to be significantly affected by the gravity of the lens, β is necessarily small.

We saw earlier that α is also small, so we can apply the paraxial (small angle) approximation to the geometry in Figure 1.3 to produce a modified form of Equations 1.3.

$$\begin{aligned} (\theta - \beta)D_L &\approx (D_S - D_L)\psi \\ \alpha &= \frac{2R_s}{\theta D_L} \\ \psi &= \alpha - \theta + \beta . \end{aligned}$$

Combining these relationships gives a new expression for θ

$$\theta(\theta - \beta) = 2R_s \frac{D_S - D_L}{D_S D_L} .$$

Applying the definition of the Einstein angle and rearranging gives the *lens equation*

$$\beta = \theta - \frac{\theta_E^2}{\theta} . \tag{1.7}$$

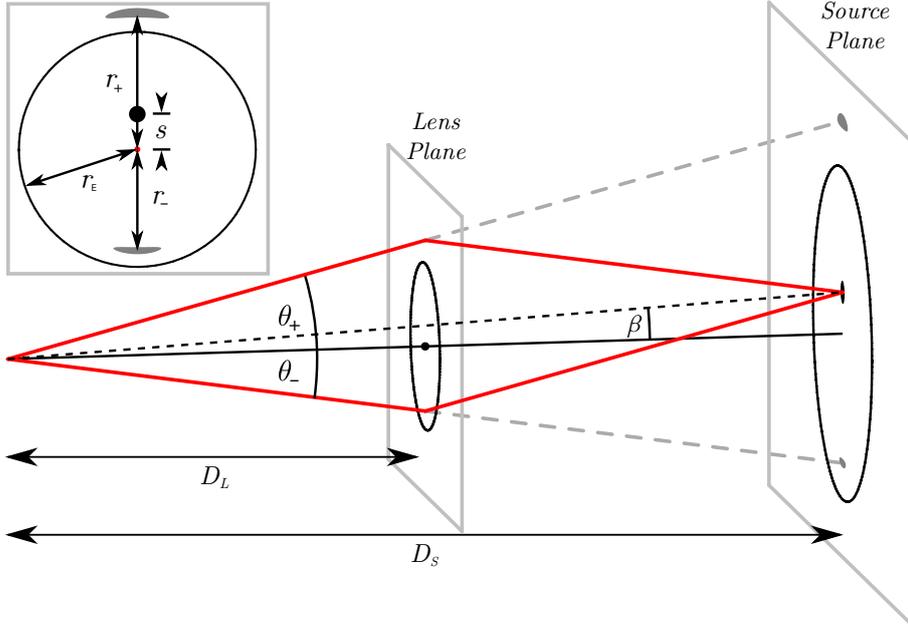


Figure 1.3: A source offset from the lens by an angle β will be seen as two distorted images on opposite sides of the lens. The relative positions of the source and images can be measured by angles β , θ_+ , θ_- , or as positions s , r_+ , r_- relative to the Einstein ring.

There are two distinct solutions for the image positions, given by

$$\theta_{\pm} = \frac{\beta}{2} \pm \theta_E \sqrt{\frac{\beta^2}{4\theta_E^2} + 1}. \quad (1.8)$$

The ‘+’ image appears on the same side of the lens as the source, outside the Einstein ring. The ‘-’ image appears on the opposite side of the lens as the source, inside the Einstein ring. As the symmetry from the perfect alignment is broken, these two images are distinct.

The angular positions can be projected into the lens or source plane, measured relative to the Einstein radius in their respective planes. If we define s to be the source position

$$s = \frac{\beta}{\theta_E} = \frac{\beta D_L}{r_E} = \frac{\beta D_S}{\hat{r}_E},$$

and r to be an image position

$$r = \frac{\theta}{\theta_E} = \frac{\theta D_L}{r_E} = \frac{\theta D_S}{\hat{r}_E},$$

these values will remain constant when moving between coordinate systems.

Applying these definitions to Equation 1.7 produces a simpler form of the lens equation

$$s = r - \frac{1}{r}. \quad (1.9)$$

The image positions r_{\pm} for a given source position s are thus

$$r_{\pm} = \frac{1}{2} \left[s \pm \sqrt{s^2 + 4} \right] . \quad (1.10)$$

These positions are shown in the inset of Figure 1.3, where they can be compared to the related angular quantities θ and β .

The two images contribute differently to the total amount of light received by the observer, proportional to their relative area. The deflection of photons by the gravitational field of the lens does not change the intrinsic brightness of the source, and so the intensity per unit area in each image is the same as in the source. The deflection of source photons instead causes the size of each ‘unit area’ to vary with position.

The magnification, M , defines the ratio of observed light to the amount of light we would see if there was no lensing. If the source disk is uniformly bright, this will be equal to the ratio of image area to source area. If the source disk shows significant stellar limb darkening (discussed in Section 1.5), then each element of area in the source must first be weighted by the relative flux at that point.

The contribution of each image to the total magnification is calculated as follows: In polar coordinates, an elemental area unit is an arc. An arc at a radius r will have an infinitesimal width dr , and an infinitesimal angular width $r d\theta$. Its area is thus $dA = r dr d\theta$. Likewise, the elemental source area is given by $s ds d\theta$, shown in Figure 1.4.

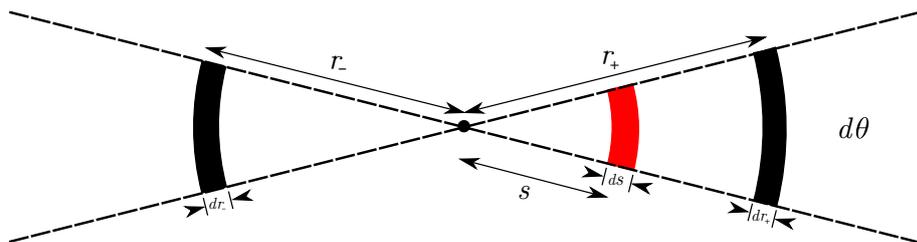


Figure 1.4: The magnification of a point source is given by the ratio of areas of infinitesimal arcs. The red arc at s is an element of area inside the source, the black arcs at r_{\pm} are the corresponding elements of area in the two images.

The magnification of the each elemental image is thus

$$M_{\pm} = \frac{r_{\pm} dr_{\pm} d\theta}{s ds d\theta} = \frac{r_{\pm}}{s} \frac{dr_{\pm}}{ds} .$$

The ratio dr_{\pm}/ds is found by differentiating Equation 1.10. Substituting for r_{\pm} and

its derivative gives the expression

$$\begin{aligned} M_{\pm} &= \frac{r_{\pm}}{s} \frac{dr_{\pm}}{ds} \\ &= \frac{1}{4} \left[1 \pm \frac{\sqrt{s^2 + 4}}{s} \right] \left[1 \pm \frac{s}{\sqrt{s^2 + 4}} \right] \\ &= \frac{1}{2} \left[1 \pm \frac{s^2 + 2}{s\sqrt{s^2 + 4}} \right]. \end{aligned}$$

The second term is always greater than 1, which implies that $M_+ > 1$ and $M_- < 0$ always. This negative magnification occurs due to the mirroring of the ‘-’ image relative to the source. The magnitude of this magnification is correct, so the absolute value is taken. The significance of the sign of each magnification term is discussed in Section 1.4.3.

The total magnification of the elemental area is thus given by the sum

$$\begin{aligned} M &= |M_+| + |M_-| \\ &= M_+ - M_- \\ &= \frac{s^2 + 2}{s\sqrt{s^2 + 4}}. \end{aligned} \tag{1.11}$$

If the source can be treated as a point-source of light, then this also describes the total magnification of the source. This *point-source* approximation will be valid whenever the elemental unit-area magnification varies linearly across the source disk, as the average magnification will be equal to the magnification at the centre. The finite source effects that become important when the source comes into close alignment with the lens are discussed in Section 1.5.

1.3 Source Motion

As previously discussed, the observational data for a microlensing event comes in the form of a *lightcurve*, which is a measure of intensity with time. In order to model a theoretical lightcurve, we need a model for the source position as a function of time.

The simplest model for describing the relative motion of the observer, lens, and source is depicted in Figure 1.5. The observer sees the source move along a straight track (red) at a constant speed past the lens at the centre of the Einstein ring. The time parameter t_0 defines when the source is closest to the lens, where the separation between the lens and source is given by the impact parameter u_0 . The crossing time t_E (in days) is a convenient measure of the relative velocity, and is the time it takes for the source to move one Einstein radius.

As time increases, the vector position $\mathbf{s}(t)$ of the source will sweep around the lens in a clockwise or anti-clockwise sense. This is shown in Figure 1.5 for $\mathbf{s}(t_0 - t_E)$ and $\mathbf{s}(t_0 + t_E)$. To remove ambiguity between clockwise and anti-clockwise motion, we define positive u_0 to signify anti-clockwise motion, and negative u_0 to define clockwise motion.

A single lens system is circularly symmetric, so the source track orientation has no impact on the light curve. However, more complicated lens geometries can break this symmetry (e.g. via multiple lens components or a curved source track), requiring at least one additional parameter to unambiguously define the track origination. Typically, the angle ϕ is used to define the source track angle relative to the binary lens axis.

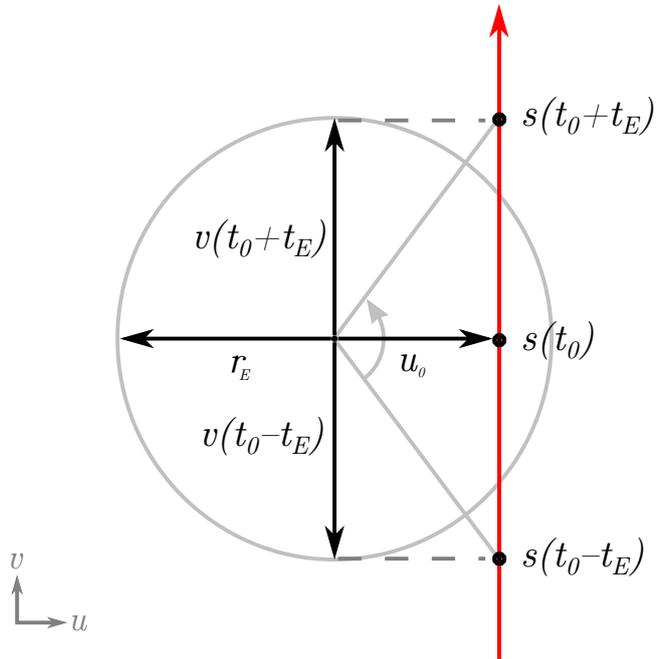


Figure 1.5: Relative motion between the observer, lens, and source is modelled by moving the source along a straight line past the (fixed) lens. This motion is described by the parameters u_0 , t_0 and t_E .

For the vertical track shown in Figure 1.5, the position $\mathbf{s}(t)$ of the source is defined by its components

$$\begin{aligned} u(t) &= u_0 \\ v(t) &= \frac{t - t_0}{t_E} . \end{aligned} \quad (1.12)$$

The source-lens separation is given by the magnitude of this vector, unaffected by any rotation about the lens

$$s(t) = |\mathbf{s}(t)| = \sqrt{u_0^2 + \left[\frac{t - t_0}{t_E} \right]^2} . \quad (1.13)$$

This model defines a standard analytical lightcurve (Paczynski [1986]) which accurately models many observed microlensing events. Figure 1.6 shows one such example.

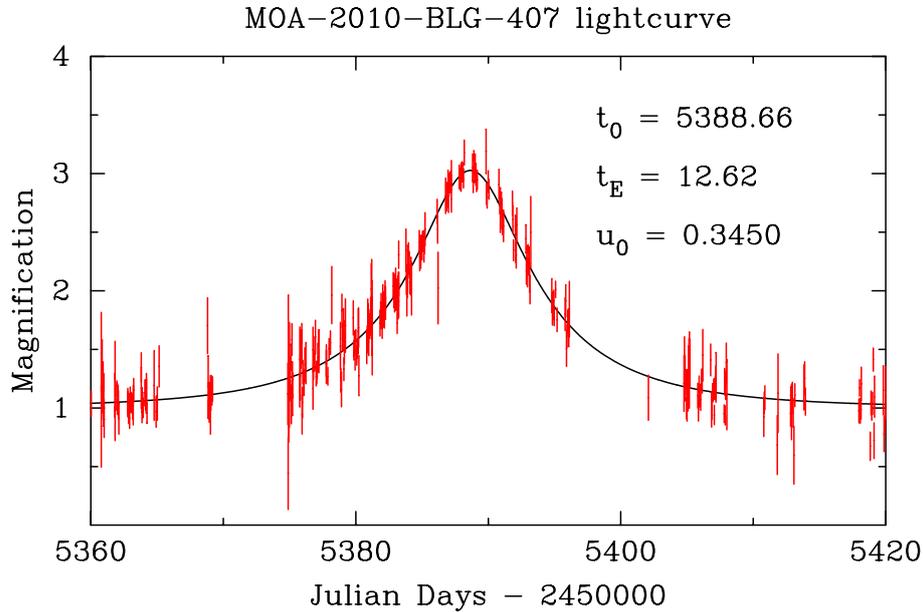


Figure 1.6: An example of one of the hundreds of microlensing events that are observed each year. The Paczynski model, with a point-source moving along a straight track, is a good model for most of these events. Data and model parameters obtained from (moa)

These simple events hide the physically interesting characteristics (the lens mass, relative velocities, and distances between observer lens and source) degenerately within the crossing time t_E . It is not possible to uniquely determine any of these parameters without additional information; they must be inferred from statistical models.

1.4 Compound Lenses

The primary goal of modelling microlensing events is to discover lens systems that contain planets. This necessarily requires a model that can describe a lens containing multiple discrete masses. The gravitational field of these compound lenses can be derived simply, but determining the subsequent magnification of a source star involves a difficult mathematical inversion. The resulting lightcurves can show dramatic deviations from the single-lens lightcurves we saw previously; Figure 1.7 shows one such example.

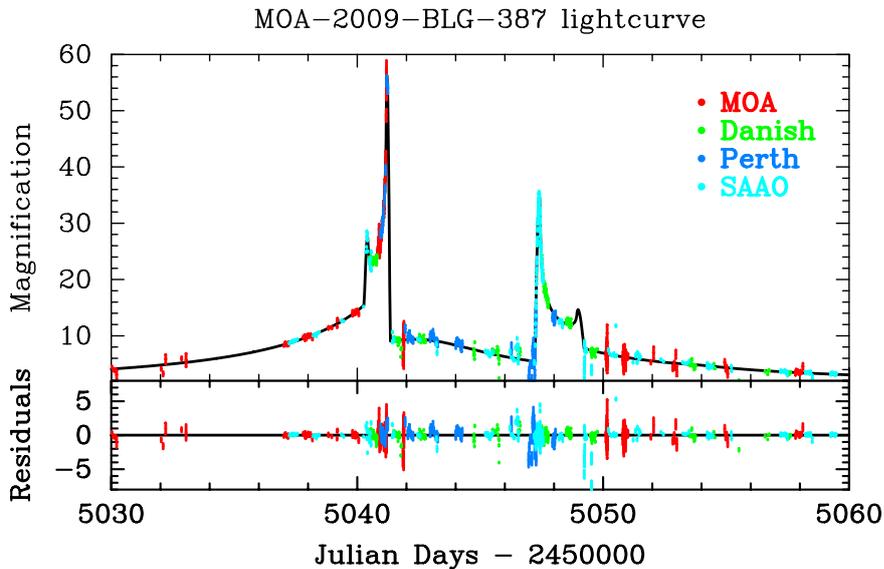


Figure 1.7: Lightcurve of the binary-lens microlensing event MOA-2009-BLG-387 (unpublished VUW model).

The nature of these deviations strongly depend on the internal structure of the lens, and so we can probe the internal lens details by fitting model lightcurves to the observed data. In particular, the presence of companion bodies near the Einstein ring of the primary lens can strongly perturb the light deflection. This makes it very likely (and in many cases, guaranteed) that their presence will cause a noticeable deviation from the standard single lens lightcurve. This has an interesting consequence; not only can microlensing confirm the presence of planets around a star, it can also confirm the *absence* of planets.

1.4.1 Lens Equation

A general form of the lens equation must be developed before we can understand the deflection of a photon by the gravitational field of a composite lens. The simple symmetry of the single lens meant that the deflection could be considered as

a 1-dimensional problem, and concisely described using angles or distances. The addition of extra lens components breaks this symmetry, and so requires the use of 2-dimensional vectors within the lens or source planes.

The lens equation for a single mass (Equation 1.9) can be generalised by defining \mathbf{s} and \mathbf{r} to be the *vector* positions of the source and images. For a lens at the origin of the lens plane, the source-lens separation defines the direction of \mathbf{s} . The deflection occurs along the line between the source and lens, and so \mathbf{r} will be parallel to \mathbf{s} . We can define the unit vector $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}| = \mathbf{r}/r$ to be the direction of this deflection, giving

$$\begin{aligned}\mathbf{s} &= \mathbf{r} - \frac{1}{r}\hat{\mathbf{r}} \\ &= \mathbf{r} - \frac{\mathbf{r}}{|\mathbf{r}|^2}.\end{aligned}\tag{1.14}$$

The deflection depends only on the source-lens separation, and so it should be clear that the equivalent statement for a lens mass at an arbitrary position \mathbf{r}_i is just

$$\mathbf{s} = \mathbf{r} - \frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^2}.$$

Using the thin lens approximation, we can treat all of the lens components as if they lie within the lens plane. The additive nature of gravity means that the total deflection due to the lens will be given by the (vector) sum of deflections produced by each component separately. The lens equation for a lens with N components therefore has N deflection terms

$$\mathbf{s} = \mathbf{r} - \sum_{i=1}^N \varepsilon_i \frac{\mathbf{r} - \mathbf{r}_i}{|\mathbf{r} - \mathbf{r}_i|^2}.\tag{1.15}$$

The deflection of each component must be scaled by a factor ε_i which depends on its mass. Keeping definitions similar to those introduced with the single lens description, the Einstein ring radius is determined by the total mass of the lens, while the light deflection produced by a particular component depends on the Einstein radius, r_{E_i} , corresponding to that component. By comparing this situation with the original lens equation (Equation 1.7), we see that this factor is equal to the squared ratio of Einstein ring radii, which corresponds to the *mass fraction* of the component

$$\varepsilon_i = \frac{r_{E_i}^2}{r_E^2} = \frac{\theta_{E_i}^2}{\theta_E^2} = \frac{M_i}{\sum_{j=1}^N M_j}.\tag{1.16}$$

Equation 1.15 is the general lens equation that applies to an arbitrary number and distribution of point masses within the lens. It relates a given (point) source position

to multiple image positions. It looks simple enough but, in general, it cannot be solved analytically to give an expression for each image position \mathbf{r} as a function of source position \mathbf{s} .

There are two basic approaches to the inversion of Equation 1.15. The first is a ‘brute force’ technique called *inverse ray-tracing* which involves substituting ‘all’ potential image positions \mathbf{r} in Equation 1.15 to see which ones correspond to the given source position \mathbf{s} . Thus, all of the lens plane’s possible positions are searched in order to find the actual images. Given that fitting procedures require the generation of many light curves, the practicality of this method depends on the availability of significant computing resources, such as those provided by grid computing configurations. It should be clear that any actual implementation of this method will involve searching for images corresponding to sources with a finite area. The image to source area ratio, and therefore the magnification of each image can be estimated as part of the procedure. This method and algorithms to improve the efficiency of the search are discussed in the next chapter.

The second approach to solving Equation 1.15 is to perform the actual inversion, which produces (with mounting difficulty as the number of lenses increase) an $(N^2 + 1)^{th}$ order polynomial in complex numbers for the multiple image positions. Unfortunately, even the simplest multiple lens system (two point mass lenses) results in a 5^{th} order complex polynomial which can only be solved numerically to give the five possible image positions. This technique is discussed at length in the next chapter and forms the basis of the techniques implemented in the `mlens2` modelling code produced during this thesis work.

1.4.2 Image Magnification

It is relatively straightforward to obtain an expression for the infinitesimal area change introduced by Equation 1.15. This gives an analytic expression for a composite lens that defines the point-source magnification for a given set of image positions. When combined with the lens polynomial (discussed in Section 2.2.1) for determining image positions, we have a method we will refer to as *semi-analytical* for calculating magnification. It is not fully analytical, as the polynomial roots (image positions) must be found numerically.

This semi-analytical approach is not only more satisfying from a theoretical standpoint; in many cases is computationally faster than a purely numerical calculation via the inverse ray-tracing method.

The geometric argument of ‘arc areas’ applied earlier to the single lens was an

example of a *Jacobian* calculation. The Jacobian is the determinant of the matrix of partial derivatives, the *Jacobian matrix*, of an equation. A brief background to the Jacobian is given in Appendix A.

The Jacobian specifies the change in an infinitesimal area unit introduced by a transformation. This is exactly the quantity that we wish to derive. The magnification of an elemental area at some point in the source is simply the Jacobian of the source plane to lens plane transformation, evaluated at the point of interest in the source. Of course, this cannot be evaluated directly because we lack an analytic expression that links a point in the source plane to points in the lens plane (if we did, we could use that expression to find the images directly, and this approach would be unnecessary!).

Our interpretation of the Jacobian as the ratio of areas gives a simple alternative. We can evaluate the Jacobian of the inverse problem, which gives the area change in going from the lens plane into the source plane. The area change of the reverse transformation is simply the inverse of this value.

The magnification can thus be found from the Jacobian of the lens equation (Equation 1.15) evaluated at each image position \mathbf{r}_j ,

$$M = \sum_j \frac{1}{|J(\mathbf{r}_j)|} . \quad (1.17)$$

The *parity* of an image is given by the sign of the Jacobian at the image position, and is related to the topology of the images. As for the single-lens, the absolute value of the Jacobian is taken to ensure that the correct (positive) magnification is obtained when summing over the images.

As discussed in Appendix A, the Jacobian is evaluated from the partial derivatives of the lens equation

$$J = \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \quad (1.18)$$

where x and y are the components of \mathbf{r} , and u and v are the components of \mathbf{s} . Recall, that because \mathbf{r} and \mathbf{s} are measured relative to the Einstein ring, the values are independent of the plane they are considered in.

Expressions for u and v in terms of x and y can be found by separating the lens equation into its components

$$\begin{aligned} u(x, y) &= x - \sum_i \varepsilon_i \frac{x - x_i}{|r - r_i|^2} \\ v(x, y) &= y - \sum_i \varepsilon_i \frac{y - y_i}{|r - r_i|^2} \end{aligned} \quad (1.19)$$

where $|r - r_i|$ is the distance between the lens and source,

$$|r - r_i| = \sqrt{(x - x_i)^2 + (y - y_i)^2} . \quad (1.20)$$

The partial derivatives are thus

$$\begin{aligned} \frac{\partial u}{\partial x} &= 1 + \sum_i \frac{\varepsilon_i}{|r - r_i|^2} \left[\frac{2(x - x_i)^2}{|r - r_i|^2} - 1 \right] \\ \frac{\partial v}{\partial y} &= 1 + \sum_i \frac{\varepsilon_i}{|r - r_i|^2} \left[\frac{2(y - y_i)^2}{|r - r_i|^2} - 1 \right] \\ \frac{\partial u}{\partial y} = \frac{\partial v}{\partial x} &= 2 \sum_i \frac{\varepsilon_i (x - x_i)(y - y_i)}{|r - r_i|^4} . \end{aligned}$$

The values obtained from these expressions can then be substituted back into Equations 1.18 and 1.17 to obtain the point-source magnification.

1.4.3 Image Topology

We can understand some of the results that we derived earlier by looking at the topology that the lens system imposes on the lens and source planes. The gross features of an observed lightcurve can be linked back to this topology, which can allow an experienced eye to make an educated guess towards potential lensing geometries from the observed lightcurve data alone.

The lens equation allows us to trace the path of a photon through a point \mathbf{r} in the lens plane back to its origin \mathbf{s} in the source plane. Recall that for a single lens

$$\mathbf{s} = \mathbf{r} - \frac{\mathbf{r}}{|\mathbf{r}|^2} .$$

Notice that there is a degeneracy between \mathbf{r} and $1/\mathbf{r}$ in this equation, which essentially divides the lens plane into two regions. Every point \mathbf{r} outside the Einstein ring has a corresponding point $1/\mathbf{r}$ inside the Einstein ring that maps to the same point in the source plane. The Einstein ring defines the boundary between the two regions, with *all* points on the ring mapping to the projected lens position at the origin of the source plane. The lens plane is thus mapped twice over the source plane, leading to the two distinct images of the source (except when the lens and source are perfectly aligned, where there is only one).

Now consider the Jacobian of the lens equation. Appendix A shows that the Jacobian for a single lens is given by

$$J = 1 - \frac{1}{|\mathbf{r}|^4} . \quad (1.21)$$

From this, we can see that all points outside the Einstein ring have positive parity, and that all inside the Einstein ring have negative parity. The Einstein ring itself is defined by the points $J = 0$ and so have an undefined parity.

These so far are just a restatement of what we saw earlier. Together however, they motivate a useful graphical representation.

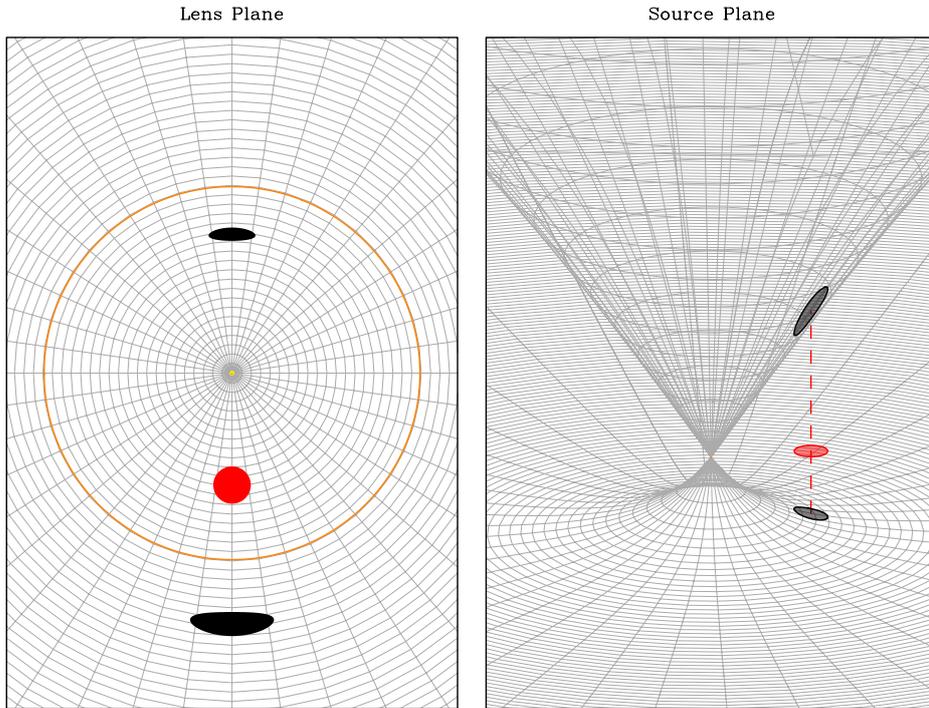


Figure 1.8: The topology of the single lens is visualised by mapping points in the lens plane (left panel) into the source plane (right panel), plotted with a height proportional to the value of the Jacobian (using an arbitrary negative log scale). The topology of the single lens is a double-cone structure where the Einstein ring (orange) in the lens plane is transformed to a single point at the origin of the source plane. The images (black) are found by projecting the source disk (red) vertically and finding where it intersects with the transformed lens plane. See text for details.

Figure 1.8 plots the Jacobian of points in the lens plane (using an arbitrary log scale, flipped vertically for the sake of visual clarity) against the 2-dimensional position of the corresponding source-plane points. The images occur wherever the transformed lens plane overlaps the source, and so the total number of images at a given point is equal to the number of times that a vertical line through that point intersects the lens-plane structure. We can also immediately see which images will contribute significantly to the overall magnification based on their height; Equation 1.17 states that images with smaller Jacobian (ignoring sign), $|J|$, are magnified more than

those with larger $|J|$.

Figure 1.8 contains enough information to qualitatively describe the behavior of the images for any source position. Far from the lens, the lower plane is flat with $J \approx 1$ and the upper cone has an extremely large value of $|J|$. This tells us that a source far from the lens will have a dominant image that closely approximates the source outside the Einstein ring, and a vanishingly small image inside the Einstein ring. The flat nature of the lower plane tells us that this outer image will vary little as the source approaches the lens. The upper cone shape (plotted with a log scale) tells us that the inner image will increase exponentially in size. When the source comes into closer alignment with the lens, $r < 1r_E$, the lower plane begins to curve towards $|J| = 0$ and the upper cone has a comparable $|J|$ to the lower plane. This indicates that the outer image will begin to increase in size, while the inner image now contributes significantly to the total magnification. When the source overlaps the projected lens position, we can see that the two images will appear to merge across the Einstein ring, as shown in Figure 1.9.

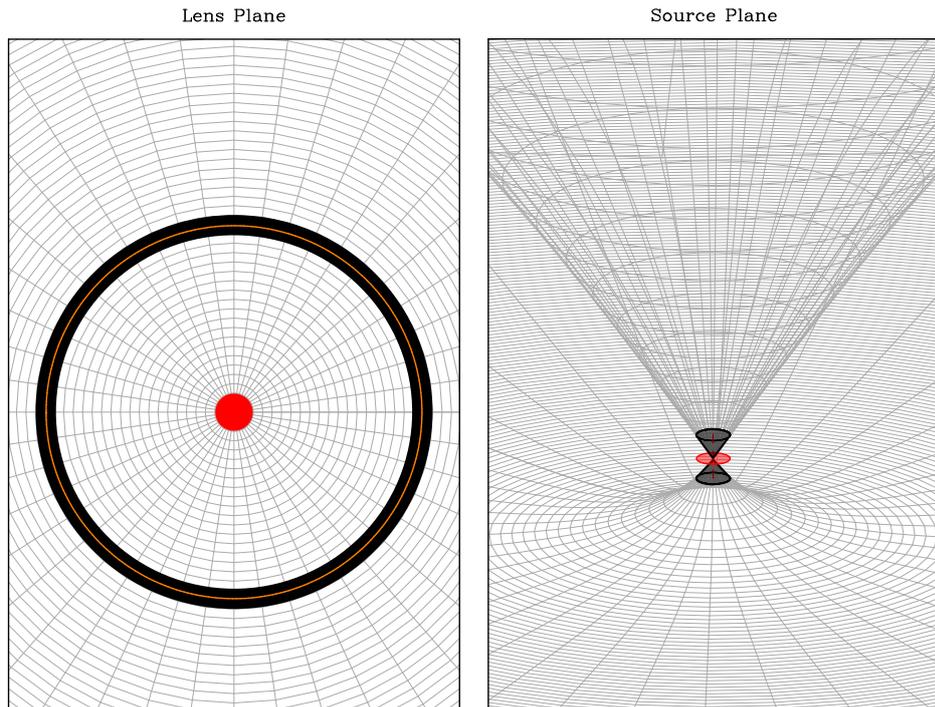


Figure 1.9: When the source covers the point where the two planes meet, the resulting images merge together at the Einstein ring, appearing as a single image. See text for details.

In general, the lens plane is divided into regions with opposite parity by *critical curves*, defined by $J = 0$. These curves map into *caustics* in the source plane. The critical curves and caustics are central provide a powerful tool for qualitatively

understanding the behavior of the images and corresponding magnification behavior.

The 3-dimensional visualisations are more interesting when applied to composite lenses, which have more complex mappings. As one might expect, each additional lens component introduces a feature where photons passing near the lens component are deflected by a large amount. What may be surprising is that this will not necessarily be located at the lens position, due to the gravitational influence of the other lens components. This causes a much more complex mapping from the lens plane into the source plane, shown for a specific binary lens in Figure 1.10.

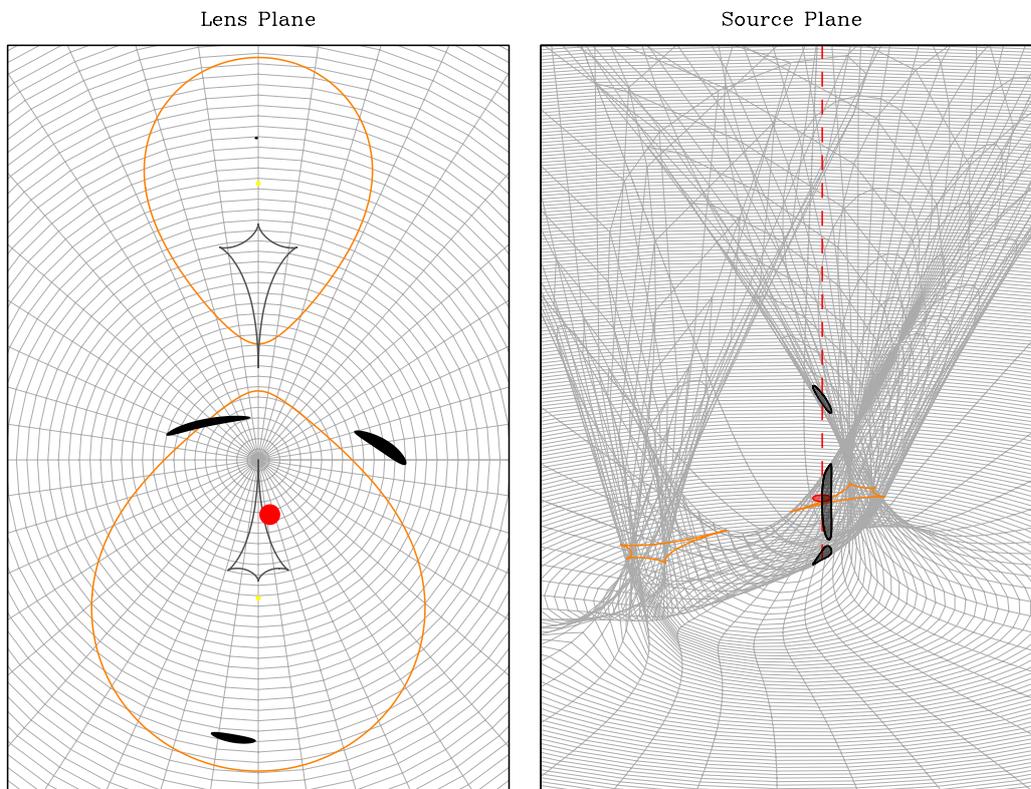


Figure 1.10: Topology of a binary lens as the source (red) crosses a caustic (left panel, grey curve). An extra pair of images of the source region within the caustic (black in the left panel, orange in the right) appear in the lens plane, joined at the critical curve (left panel, orange curve). The right panel shows that this is due to the source intersecting an extra pair of layers twisted within the shape defined by the caustic. See text for details.

Like the single-lens case, far from the lenses (more correctly, far from the *caustics*), there is a flat plane (with positive parity) plus a cone structure (with negative parity) for each lens component. In general, a source outside of the caustics will have $N + 1$ images, with a total parity of $1 - N$. This result was first shown mathematically by

Witt [1990].

The transformed lens plane twists around itself, overlapping additional times in the regions bounded by the caustics. This leads to the source having an additional pair of images when it lies inside one of these caustic regions. If the source lies partially over a caustic, these two images will appear to merge into a single image across a critical curve, like we saw with the single lens case. These merging images will usually dominate the overall magnification, as they necessarily have $|J| \approx 0$. This can be seen for two of the images in Figure 1.10.

When the source passes inside a caustic, the two additional images that form have opposite parity, and so the total parity remains fixed, independent of the source position. This provides a useful method for algorithmically checking the correctness in any numerical calculation of the images.

1.5 Finite Source Effects

We saw in the previous section that the Jacobian (and thus the magnification) behaves smoothly far away from the caustics. A source in this region can be treated as an infinitesimal point, as any variation over the source disk will be linear and the total magnification will average to its central value. This assumption allows much of the observed lightcurve to be efficiently modelled, but breaks down when the source is near a caustic where the non-linear changes in magnification causes the total magnification to differ from the magnification at the center of the source disk. The finite size of the source becomes important. It is important to have an accurate model for the source magnification near caustics, because the sharp changes in magnification that occur in these regions place a strong constraint on the source size.

The first step in correctly modelling these finite-source effects is to treat the source as a uniformly bright disk. The magnification can be calculated numerically by evaluating the image and source areas separately and taking the ratio, or semi-analytically by considering a 2D expansion of the magnification across the disk.

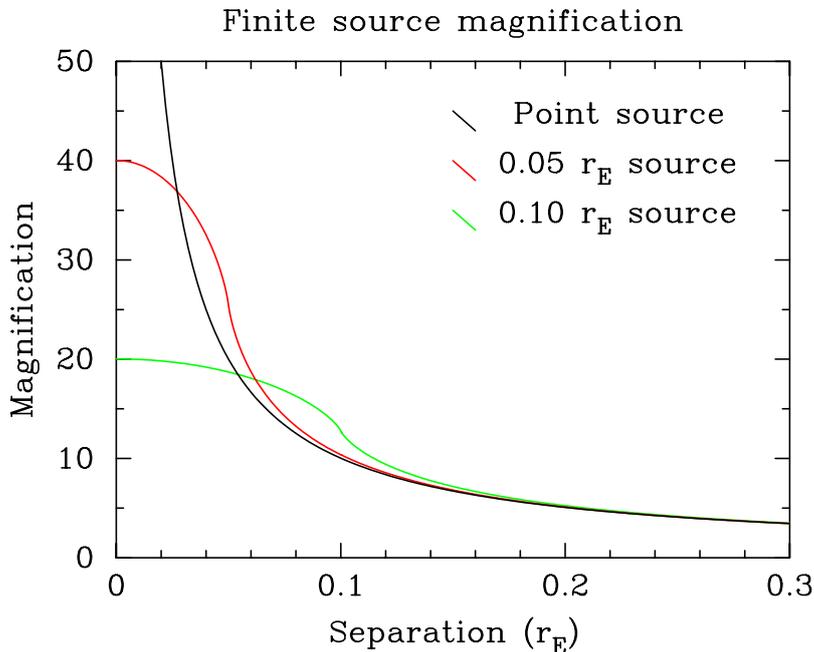


Figure 1.11: The magnification of a source near a caustic strongly depends on the size of the source. Total magnification for a point-source and two finite source sizes are shown for a range of source-lens separations.

Figure 1.11 shows how the calculated magnification differs near a caustic for a point-source and two finite sources with different radii (calculated using the ‘polygon method’ discussed in Section 2.4).

As the source approaches the caustic, we see that larger sources has a broader and flatter magnification change than small sources. This should be intuitive, as since a larger source covers a greater range of magnifications, the contribution of any single point contributes less to the whole. It is apparent that the that the peak value of the magnification and, to a lesser extent, the width of the peak strongly depend on the source size. Observational data of a caustic crossing can thus provide a strong constraint on the source size.

The next step is to consider the effect of non-uniform source brightness: *limb darkening*. The photons that we receive from a star originate from some average depth below its surface, governed by the optical depth of the stellar material. The temperature within the star increases with depth, and the apparent brightness increases with temperature via the Stefan-Boltzmann law. The apparent brightness of a point on the stellar disk thus depends on the depth from which the photons were emitted. Figure 1.12 demonstrates how changing the line of sight across the spherical star changes the depth from which the visible photons originate. This leads to the center of the disk appearing brighter than the outer limb, hence the term ‘limb darkening’.

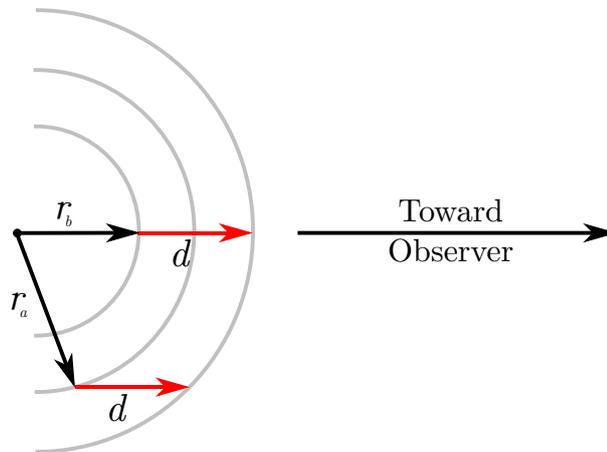


Figure 1.12: The photons that are received by an observer originate from some constant depth, d , along the line of sight within a star. When looking at the outer limb of the stellar disk, this depth corresponds to a radius r_a , which is closer to the surface (and thus cooler) than when looking at the center of the stellar disk, where photons are visible from a deeper (hotter) radius r_b .

Limb darkening is a relatively minor effect in microlensing and is sufficiently described by a simple linear model. To ensure compatibility between numerical methods, we take the definition used by Gould [2008], which is integrated into the hex-

adecapole approximation model described in Section 2.3

$$\frac{I(r)}{I_0} = 1 - \Gamma \left(1 - \frac{3}{2} \sqrt{1 - \left(\frac{r}{r_s} \right)^2} \right). \quad (1.22)$$

Here, I_0 defines the average flux per unit area across the source, r_s the source radius, and Γ the shape of the intensity profile (which may be wavelength dependent). Figure 1.13 shows the range of intensity profiles that can be achieved.

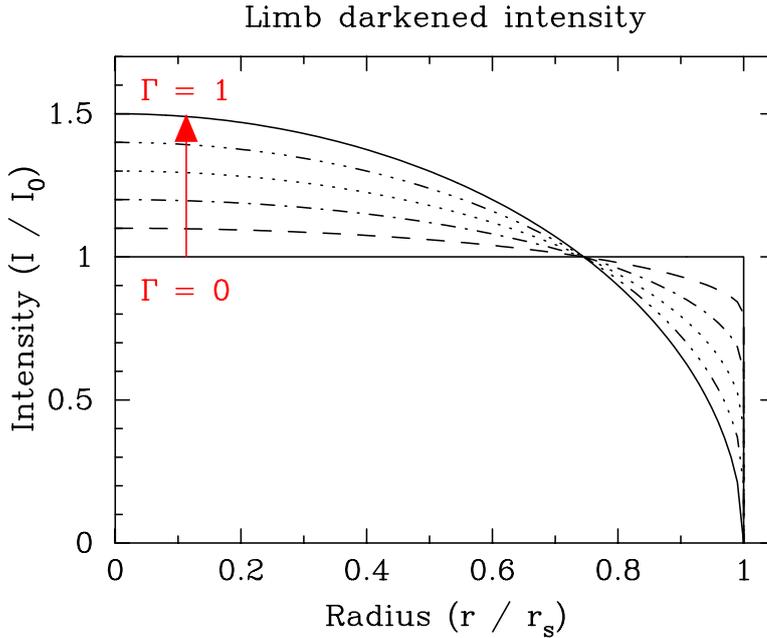


Figure 1.13: The limb darkening parameter Γ defines the radial intensity profile $I(r)$ for a limb darkened source, as defined by Equation 1.22.

The value for Γ can be calculated from standard stellar models using colour-magnitude relationships. The specific implementation of these finite source effects limb darkening into each numerical model is covered in the relevant sections of Chapter 2.

1.6 Orbital Motion

An observer on Earth orbits around the Sun once per year. Likewise, the source star may be a part of a binary star system, or the lens star may have orbiting planets. These orbital motion effects introduce additional non-rectilinear relative motion between the observer, lens, and source.

In Section 1.3 we modelled this relative motion by moving the source along a straight track past the lens. Orbital motion in the observer or source will change this track, while motion of the lens changes the underlying magnification pattern that the source travels over. These effects have the potential to significantly alter the resulting lightcurve, but in most cases they will provide a minor perturbation that becomes apparent when comparing the χ^2 fit of different models.

1.6.1 Orbital Parallax

Parallax is a geometric phenomenon seen when an observer moves relative to two stationary bodies. The apparent shift of the background object when the observer moves a known distance can be related to the ratio of distances between the observer and the two bodies. This is shown in Figure 1.14.

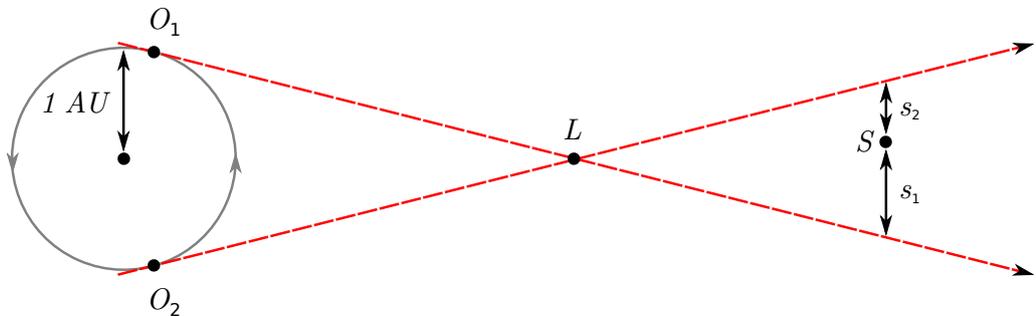


Figure 1.14: The apparent position, s , of the source relative to the lens changes as the observer moves from O_1 to O_2 . The ratio of distances between the bodies can be calculated by measuring the ratio of the offsets.

This is particularly useful in the case of microlensing because knowledge of the relative distances between the observer, lens, and source can allow the mass of the lens to be extracted unambiguously from a measurement of the Einstein angle.

Parallax can have significant effects on the lightcurve of events that occur over a period of months, where the Earth moves through a significant fraction of its orbit. No significant effects will be visible for short timescale events, and so the straight track model will be sufficient for these events. The first observed microlensing event

that required a parallax model was observed in 1995 (Alcock et al. [1995]); it is now routinely included in the modelling of new events.

Figure 1.15 relates the motion of the observer to the apparent motion of the source. If the observer is offset from its original position by a vector $\Delta\mathbf{o}$, perpendicular to the observer-lens line, the origin of the source plane will be offset in the opposite direction by

$$\Delta\mathbf{s} = -\frac{D_S - D_L}{D_L} \Delta\mathbf{o} . \quad (1.23)$$

We see that the source position in this new coordinate system is just

$$\mathbf{s}' = \mathbf{s} - \Delta\mathbf{s} . \quad (1.24)$$

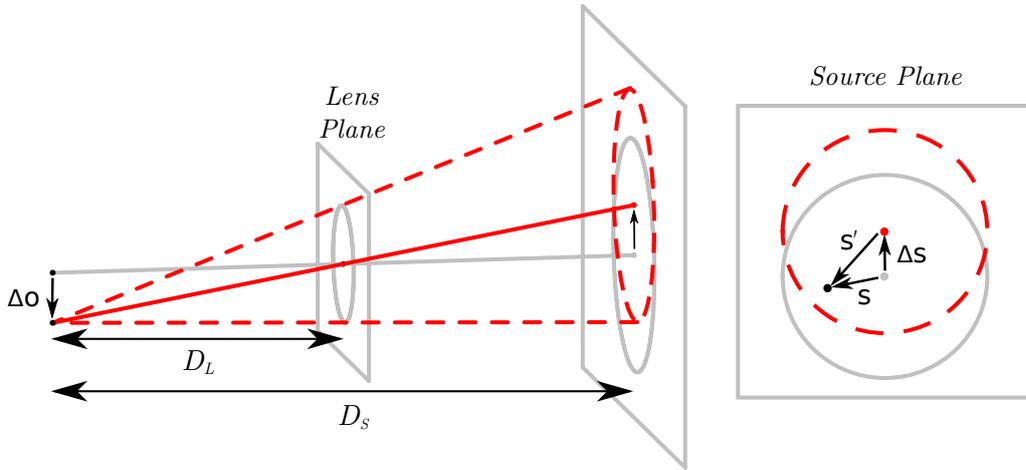


Figure 1.15: Motion of the observer changes the origin of the source plane by $\Delta\mathbf{s}$. The physical position of the source is unchanged; its position in the new coordinate system (dotted red Einstein ring) is found by subtracting $\Delta\mathbf{s}$ from the original position (grey Einstein ring).

The distances D_S and D_L can be eliminated by introducing the *observer plane*, shown in Figure 1.16.

The Einstein radius in the observer plane, \tilde{r}_E , is found by projecting the r_E in the lens plane back towards the observer

$$\begin{aligned} \tilde{r}_E &= D_S \psi \\ &= \frac{D_S}{D_S - D_L} r_E \\ &= \frac{D_L}{D_S - D_L} \hat{r}_E . \end{aligned} \quad (1.25)$$

Equation 1.23 thus simplifies to

$$\begin{aligned} \frac{\Delta\mathbf{s}}{\hat{r}_E} &= -\frac{D_S - D_L}{D_L} \frac{\Delta\mathbf{o}}{\hat{r}_E} \\ &= -\frac{\Delta\mathbf{o}}{\tilde{r}_E} . \end{aligned}$$

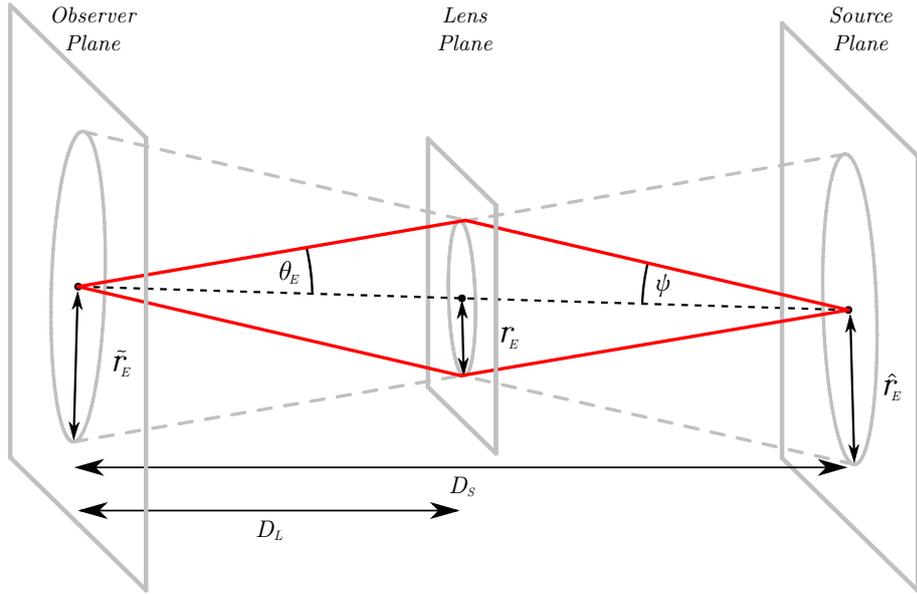


Figure 1.16: The size of the Einstein ring in each plane is related to the distances D_L and D_S between the planes by finding the angles θ_E and ψ .

If we define $\Delta \mathbf{s}$ and $\Delta \mathbf{o}$ to be measured relative to the Einstein radius in the appropriate plane, the relationship simplifies further to

$$\Delta \mathbf{s} = -\Delta \mathbf{o} . \quad (1.26)$$

Substituting this relationship into Equation 1.12 gives the source position

$$\begin{aligned} u(t) &= u_0 + \Delta \mathbf{o}(t) \cdot \hat{\mathbf{u}} \\ v(t) &= \frac{t - t_0}{t_E} + \Delta \mathbf{o}(t) \cdot \hat{\mathbf{v}} . \end{aligned} \quad (1.27)$$

The *Microlensing Parallax*, π_E , relates distances in the observer plane (measured relative to the Einstein ring) to physical distances (measured in AU)

$$\pi_E = \frac{1 \text{ AU}}{\tilde{r}_E} . \quad (1.28)$$

The main motivation of modelling parallax comes from the ability to determine this parameter. If an independent measurement of the source size can be made (e.g. by comparing the angular source size with a stellar radius found from a colour-magnitude relationship), a physical value for the lens mass can be calculated by combining Equations 1.25 and 1.6 to produce

$$\theta_E \pi_E = \frac{D_S - D_L}{D_S D_L} ,$$

which can be substituted back into the definition of the Einstein angle (Equation 1.4) to reveal the Schwarzschild radius of the combined lens mass as

$$R_s = \frac{\theta_E}{2\pi_E} .$$

In this thesis, we are concerned only with the orbital parallax due to the orbit of the Earth. There are other types of parallax documented in the literature which are (currently) of less practical use. For example, if an event was observed in tandem from the Earth and from a satellite in solar orbit, the differences in observed magnification between the two measured lightcurves could be used to determine a lens mass in a similar manner to orbital parallax. Some discussion of this can be seen in Gould [1994] and Graff and Gould [2002].

For very high magnification events, such as those caused by a star crossing a central caustic close to the primary lens, a similar effect can be observed between telescopes on opposite sides of the Earth. This *terrestrial parallax* is discussed in Gould [1997] and Yee et al. [2009].

Calculating the Earth Offset

The vector $\Delta\mathbf{o}$ gives the positional offset of the Earth within the observer plane. Calculating this offset is a two step process: first the Earth's position, $\Delta\mathbf{p}$, must be calculated within its orbital plane, and then this offset must be projected into the observer plane. This depends on the choice of coordinate system where we specify $\Delta\mathbf{p} = \Delta\mathbf{o} = 0$.

Early models for orbital parallax (e.g. Alcock et al. [1995], Dominik [1998]) chose to specify $\Delta\mathbf{p}$ relative to the Sun. This is an obvious choice, but means that the parallax offset will always be large. The more sensible choice, used in more recent models (e.g. Gould [2004]) and described here is to define $\Delta\mathbf{p} = 0$ as the position of the Earth at $t = t_0$. This choice minimises the effect of parallax over the peak of the event and means that parallax can be added to a model without requiring significant changes in other parameters.

The angular position of the Earth within its orbit can be described by its *true anomaly* or its *eccentric anomaly*, illustrated in Figure 1.17. The true anomaly is measured with respect to the sun (S), and defines the 'true' angular position in the orbit relative to the orbit perihelion (P). An ellipse can be thought of as a circle whose plane has been rotated away from the observer; the eccentric anomaly describes the position of the Earth on this circle. The true elliptical orbit is obtained by scaling the 'x' and 'y' axes by different factors a and b , corresponding to the semi-major and semi-minor axes of the orbit:

$$\begin{aligned}\mathbf{p}(t)_x &= a \cos(E(t)) \\ \mathbf{p}(t)_y &= b \sin(E(t)) .\end{aligned}$$

If we define the origin to be the position of the sun and substitute the orbital eccen-

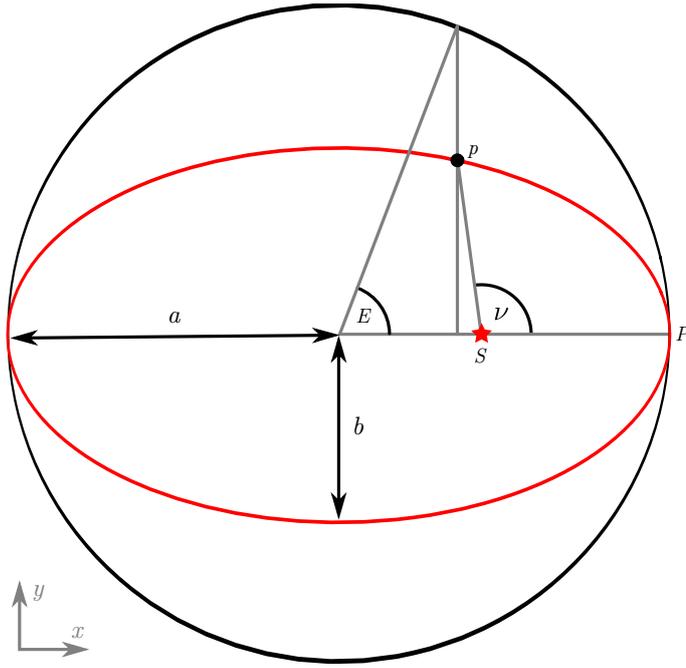


Figure 1.17: The position \mathbf{p} of a body in an elliptical orbit can be described using the true anomaly, ν , or eccentric anomaly, E .

tricity ϵ for the semi-minor axis, we obtain equivalent but more useful expressions for the earth's orbit

$$\begin{aligned}\mathbf{p}(t)_x &= a(\cos(E(t)) - \epsilon) \\ \mathbf{p}(t)_y &= a\sqrt{1 - \epsilon^2} \sin(E(t)) .\end{aligned}\tag{1.29}$$

In order to calculate the eccentric anomaly, we need to introduce a third angle, the *mean anomaly*, M . The mean anomaly increases linearly from 0 to 2π in one orbital period T

$$M(t) = \frac{2\pi}{T} (t - \tau) .$$

The reference time τ defines when the Earth is at the perihelion. It can be shown (see e.g. Green [1985]) that the eccentric and mean anomalies are related using Kepler's equation

$$M(t) = E(t) - \epsilon \sin(E(t)) .\tag{1.30}$$

Kepler's equation cannot be solved analytically for E . However, because the Earth's orbit is nearly circular ($\epsilon \approx 0.017$) the approximation $\epsilon \sin E \approx \epsilon \sin M$ is sufficiently accurate for our purposes, so

$$E \approx M + \epsilon \sin M .\tag{1.31}$$

To calculate the offset vector $\Delta\mathbf{p}(t)$, we take the difference in position between times t and t_0 . An extra term is required to account for the constant velocity of the

coordinate system (given by the instantaneous velocity at t_0)

$$\Delta \mathbf{p}(t) = \mathbf{p}(t) - \mathbf{p}(t_0) - (t - t_0) \left. \frac{d\mathbf{p}}{dt} \right|_{t=t_0} \quad (1.32)$$

where the velocity is calculated from the derivatives of Equations 1.29 and 1.31

$$\begin{aligned} \frac{dE}{dt} &= \frac{2\pi}{T} \left[1 + 2\epsilon \cos \left(\frac{2\pi(t - t_p)}{T} \right) \right] \\ \frac{d\mathbf{p}}{dt}_x &= -a \sin(E) \frac{dE}{dt} \\ \frac{d\mathbf{p}}{dt}_y &= a\sqrt{1 - \epsilon^2} \cos(E) \frac{dE}{dt} . \end{aligned}$$

Calculating the Observer Plane

The offset $\Delta \mathbf{o}$ in the observer plane is found by projecting $\Delta \mathbf{p}$ in the direction of the lens. We thus need to find the relative orientation of the orbital plane and the observer plane (which is normal to the vector pointing towards the lens). Figure 1.18 shows the three angles defining this relationship. φ and χ are fixed by the known position of the lens in the sky, while ϕ_E is an unknown parameter that is varied during the modelling process to find the best fit.

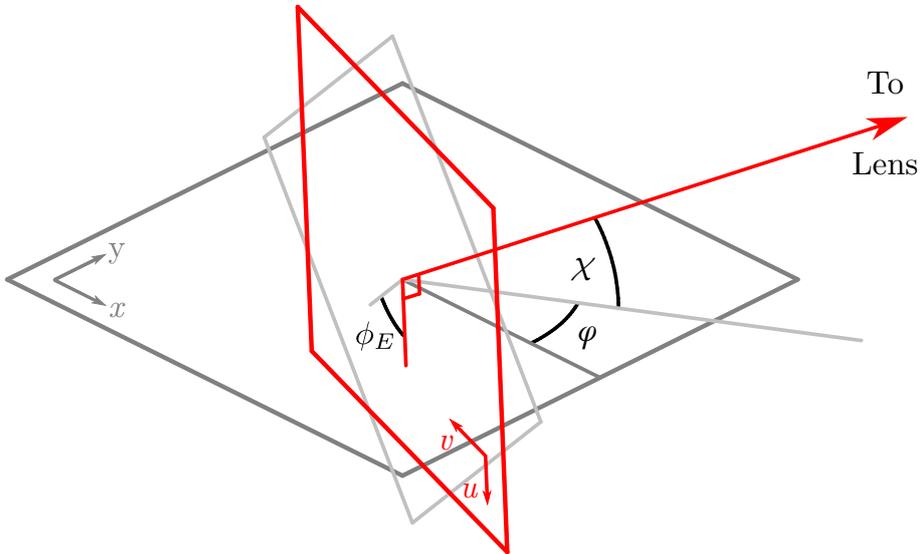


Figure 1.18: The relative orientation of the observer plane (red) and orbital plane (dark grey) is defined by the known angles φ and χ , and the unknown angle ϕ_E .

The standard coordinates for giving the position of a star are its right ascension, α , and declination, δ . These coordinates follow the same lines as the latitude and longitude of the Earth, with the zero of right ascension defined by the position of the sun in the sky at the vernal equinox near mid March. A more useful set of

coordinates are the ecliptic latitude, β , and ecliptic longitude, λ , which are measured relative to the orbital plane.

The relationship between these angles is given by Green [1985]

$$\beta = \arcsin(\sin(\delta) \cos(23.5^\circ) - \cos(\delta) \sin(\alpha) \sin(23.5^\circ))$$

$$\lambda = \arctan\left(\frac{\sin(\delta) \sin(23.5^\circ) + \cos(\delta) \sin(\alpha) \cos(23.5^\circ)}{\cos(\delta) \cos(\alpha)}\right).$$

The relationship between these ecliptic coordinates and the angles φ and χ toward the lens are given by

$$\chi = \beta$$

$$\varphi = \lambda + \gamma$$

where φ and λ measure the same physical angle, but relative to different origins, as shown in Figure 1.19. φ is measured relative to the perihelion, while λ is measured relative to the Sun's position at the vernal equinox.

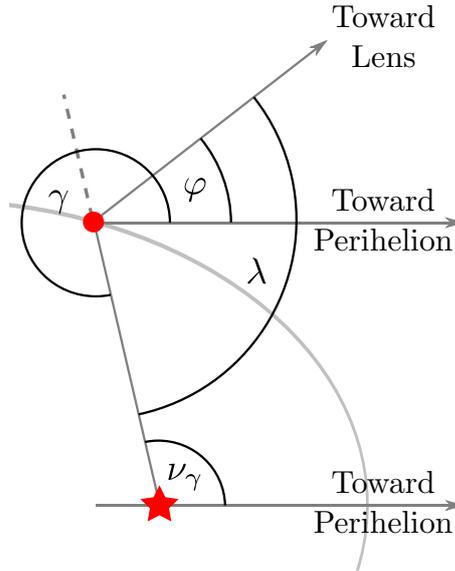


Figure 1.19: The Earth (red circle), and Sun (red star) at the vernal equinox. The angle φ between the perihelion and and lens can be constructed from the true anomaly ν_γ and ecliptic longitude λ .

The angle ν_γ is defined as the true anomaly of the Earth at the vernal equinox. It can be calculated from the eccentric anomaly of the Earth, E_γ , using the relationship (given by Green [1985])

$$\nu_\gamma = 2 \arctan \left[\sqrt{\frac{1+\epsilon}{1-\epsilon}} \tan \left(\frac{E_\gamma}{2} \right) \right]. \quad (1.33)$$

Thus $\gamma = \nu_\gamma + \pi$, and so

$$\varphi = \lambda + \nu_\gamma + \pi . \quad (1.34)$$

The simplest relationship between $\Delta \mathbf{p}$ and $\Delta \mathbf{o}$ occurs when $\varphi = \chi = \phi_E = 0$. Figure 1.18 shows that in this case the coordinate vectors are related by

$$\begin{aligned} u &= -y \\ v &= 0 \end{aligned}$$

and so

$$\begin{aligned} \Delta \mathbf{o}_u &= -\Delta \mathbf{p}_y \\ \Delta \mathbf{o}_v &= 0 . \end{aligned}$$

If $\varphi \neq 0$, the orbital plane can always be rotated to align the u and $-y$ coordinate directions. This is achieved by rotating $\Delta \mathbf{p}$ by $-\varphi$.

When $\chi \neq 0$, the x component of $\Delta \mathbf{p}$ is projected into the $-v$ component of $-\Delta \mathbf{o}$ so that

$$\begin{aligned} \Delta \mathbf{o}_u &= -\Delta \mathbf{p}_y \\ \Delta \mathbf{o}_v &= -\Delta \mathbf{p}_x \sin(-\chi) . \end{aligned} \quad (1.35)$$

After this projection, a rotation about the lens direction by $-\phi_E$ followed by a scaling by π_E gives the final $\Delta \mathbf{o}$ vector. The composition of these operations gives the general relationship between the two vectors (using matrix notation)

$$\Delta \mathbf{o}(t) = \pi_E \begin{bmatrix} \cos \phi_E & \sin \phi_E \\ -\sin \phi_E & \cos \phi_E \end{bmatrix} \begin{bmatrix} 0 & -1 \\ \sin \chi & 0 \end{bmatrix} \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \Delta \mathbf{p}(t) . \quad (1.36)$$

Many published models combine π_E and ϕ_E into a single *Microensing parallax vector*, $\pi_{\mathbf{E}}$. This vector takes the obvious definition

$$\pi_{\mathbf{E}} = \pi_E \begin{bmatrix} \cos(\phi_E) \\ \sin(\phi_E) \end{bmatrix} . \quad (1.37)$$

Effects of Parallax

For t close to t_0 , the effects of parallax on the source track can be approximated as a constant acceleration. We can thus characterise the general effects of parallax on a lightcurve by applying a constant acceleration to the source, which defines a parabolic source track

$$\begin{aligned} u(t) &= u_0 + \frac{a \sin \theta}{2} \left(\frac{t - t_0}{t_E} \right)^2 \\ v(t) &= \frac{t - t_0}{t_E} + \frac{a \cos \theta}{2} \left(\frac{t - t_0}{t_E} \right)^2 . \end{aligned}$$

An acceleration of $a = 0.2 r_E/\text{day}^2$ is applied to the source in parallel and perpendicular directions. The resulting single lens lightcurves are shown in Figure 1.20.

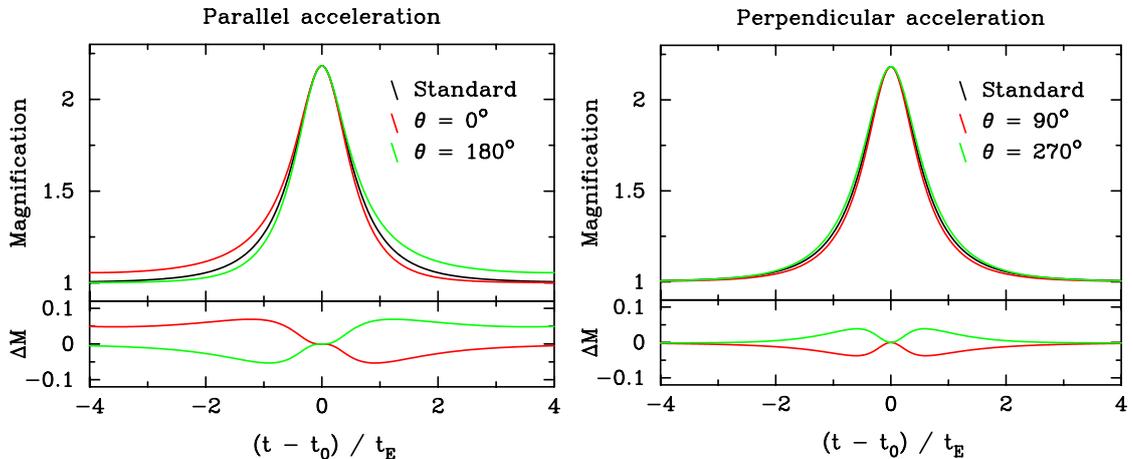


Figure 1.20: An acceleration parallel to the source velocity causes the lightcurve to become asymmetric (left panel). An acceleration perpendicular to the source velocity changes the lightcurve shape symmetrically about the peak of the event (right panel). The differences between the acceleration and non-acceleration models are shown in the lower panels of each plot.

An acceleration parallel to the source track ($\theta = 0^\circ, 180^\circ$) leaves the track unmodified, but changes the speed of the source with time. This produces a long-term systematic asymmetry in the lightcurve, which is easy to detect in the observational data.

An acceleration perpendicular to the source track ($\theta = 90^\circ, 270^\circ$) curves the source symmetrically away from its original path. This subtly changes the shape of the lightcurve, and is much more difficult to constrain from the observational data. For further details on parameter degeneracies and modelling implications for parallax, see Smith et al. [2003], Gould [2004], Poindexter et al. [2005].

1.6.2 Xallarap

A similar effect to parallax can occur in the source if it is part of a binary system. This ‘backwards parallax’ effect has been termed *Xallarap* in the microlensing literature. The geometry behind this effect is identical to parallax, but is relative to the source, instead of the observer.

Xallarap differs from parallax in that none of the orbital parameters of the source are known, and so there are many more free parameters that can be varied to model a wider range of curved source tracks. Further perturbations can arise if the source is

orbiting around a luminous companion, as both bodies can contribute light towards the observed lightcurve.

1.6.3 Lens Motion

Orbital motion of the components within the lens affects the lightcurve in a completely different way than observer or source motion. The origin of the microlensing coordinate system is defined as the centre of mass of the lens. True orbital motion within the lens does not change the centre of mass, and so the apparent position of the source is unaffected. Instead, this motion changes the underlying caustics and magnification pattern the source passes over, thus affecting the final light curve.

Like with Xallarap, there is little physical information that can be used to constrain orbital motion within the lens. Attempting to model a general orbit for the lens components adds too many free parameters to reasonably model, so a simpler model of linear or circular motion is adopted for each component.

The position vector \mathbf{r}_i of the i 'th component defines the position at $t = t_0$. For linear lens motion, the position is given by defining a velocity vector \mathbf{v}_i ,

$$\mathbf{r}_i(t) = \mathbf{r}_i + (t - t_0)\mathbf{v}_i . \quad (1.38)$$

Circular motion can be modelled by adding an extra parameter for the period of the orbit, T .

The instantaneous velocity of a body undergoing circular motion with a known radius r and period T is tangential to the circular motion, with magnitude

$$v = \frac{2\pi r}{T} .$$

The radius of this circle (denoted a to avoid confusion with the lens position) is thus

$$a = \frac{T |\mathbf{v}_i|}{2\pi} .$$

The geometry of this orbit is shown in Figure 1.21.

We see that the position of the lens at some time t is given by

$$\mathbf{r}_i(t) = \mathbf{r}_i - \mathbf{a} + \mathbf{a}(t) \quad (1.39)$$

where \mathbf{a} and $\mathbf{a}(t)$ are lens positions relative to the origin of motion, \mathbf{o} at t_0 and t respectively.

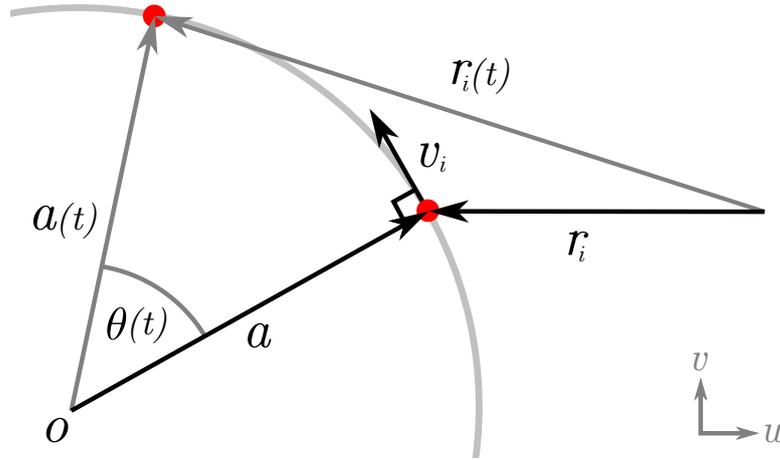


Figure 1.21: A lens moving in a circular orbit has a position $\mathbf{r}_i(t)$, calculated from a reference position \mathbf{r}_i , velocity \mathbf{v}_i , and period T .

The vector \mathbf{a} has length a , and is perpendicular to \mathbf{v} , so

$$\mathbf{a} = \frac{T}{2\pi} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{v}_i .$$

The vector $\mathbf{a}(t)$ is found by rotating \mathbf{a} by an angle corresponding to the time difference between t and t_0 . This angle can be positive for anticlockwise orbits, or negative for clockwise orbits

$$\theta = \pm \frac{2\pi(t - t_0)}{T} . \quad (1.40)$$

Equation 1.39 can thus be written as

$$\begin{aligned} \mathbf{r}_i(t) &= \mathbf{r}_i + \left(\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} - 1 \right) \mathbf{a} \\ &= \mathbf{r}_i + \frac{T}{2\pi} \begin{bmatrix} \sin \theta & \cos \theta - 1 \\ 1 - \cos \theta & \sin \theta \end{bmatrix} \mathbf{v}_i . \end{aligned} \quad (1.41)$$

This formulation assumes that the relative velocities are chosen for each component such that the centre of mass does not change with time.

Chapter 2

Numerical Techniques

Chapter 1 presented the analytical theory required to develop microlensing models. We saw that only the simplest case involving a single lens and a point source yielded a straight-forward analytical model. More complex systems, containing composite lenses or finite sources necessitated the introduction of numerical techniques. This chapter discusses two quite different approaches to modelling these more complex cases.

The first is the inverse ray-tracing method, which takes the brute force approach of repeatedly solving the lens equation (Equation 1.15) to directly find the images and critical curves for a given source-lens configuration. The *ray shooting* and *magnification map* techniques as implemented by Korpela [2007] in the original MLENS software are variations of this basic approach, and they are summarised in Section 2.1.

The second method, which we refer to as a semi-analytical technique, inverts the lens equation to form a complex polynomial, whose solutions for a given source position gives the image positions directly. We will show that this ‘lens polynomial’ is of degree $N^2 + 1$, which means that even for a binary lens, the image positions must be solved numerically. These resulting image positions can then be used with the theory from Section 1.4.2 to find the magnification of a point source, and the critical curves. The new MLENS software (mlens2) developed as part of this thesis work implements these semi-analytical techniques.

Two techniques to extend this *polynomial method* to include finite sources are then developed.

The *polygon method* represents the finite source using a network of m points around the source perimeter. The lens polynomial is then solved m times to yield the corresponding points in the images, which are grouped into separate polygons. The

effective image and source areas can then be estimated using standard geometrical techniques to produce a magnification. This approach can be further extended to handle source disk non-uniformity (limb-darkening) by using multiple concentric polygons to cover the source disk.

The *hexadecapole approximation* provides a faster approximation for finite source corrections by combining multiple point-source magnifications to represent a two-dimensional expansion of the source magnification around a particular point.

2.1 Inverse Ray-Tracing

Inverse ray-tracing techniques avoid the complication of inverting the lens equation by solving the inverse problem. Instead of trying to solve for the image positions for a given source position, the source position is calculated for a large number of potential image positions (using Equation 1.15), and those that correspond to the specified source position(s) are recorded.

This process works because the path of light is reversible. Any photon path from the source to the observer is an equally valid path from the observer to the source. By directing *inverse* light rays towards a point in the lens plane (which is then the direction of a potential image point) and determining if they are deflected towards the source position(s), one can build up a picture of the image positions in relation to various source positions.

The ray shooting and magnification map techniques are two implementations of this approach, applying different optimisations to make the process computationally feasible.

2.1.1 Ray Shooting

Ray shooting is a direct implementation of the method above. The lens plane is divided into a grid of square pixels with some defined size, and the lens equation is evaluated at the centre of each. If the resulting source plane position is inside the source disk, the pixel in the lens plane is marked as being contained within an image. This is shown in Figure 2.1.

Limb darkening can be modelled by weighting each pixel with an intensity given by Equation 1.22. The brightness, I , of a pixel which traces to a position \mathbf{x} in the

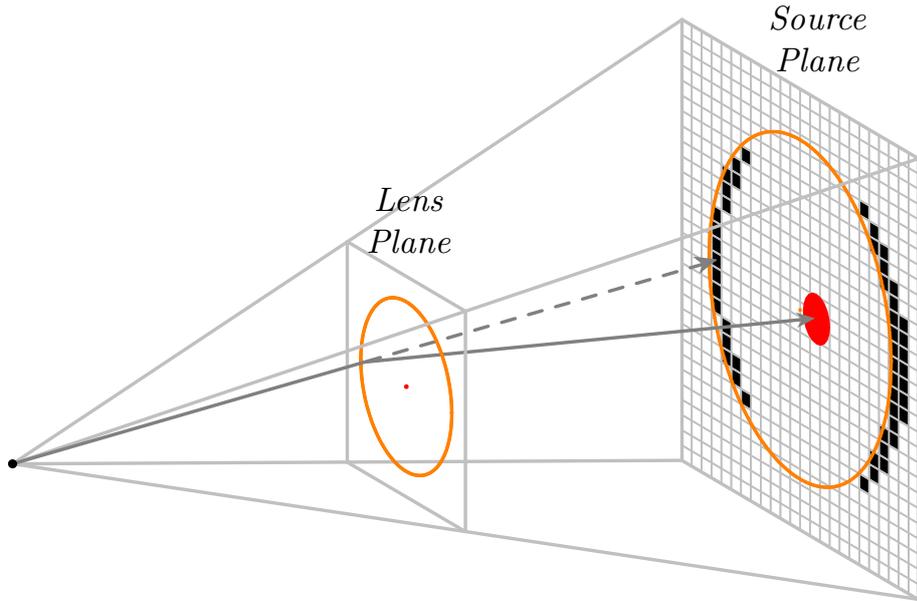


Figure 2.1: Ray shooting tiles the lens plane with pixels (shown here projected into the source plane for visibility) and calculates the source plane position of each using the lens equation. Pixels that are mapped inside the source disk (red) are marked as being in an image (black).

source plane, is thus

$$I(\mathbf{r}) = \begin{cases} 0 & |\mathbf{x} - \mathbf{s}| > r_s \\ 1 - \Gamma \left(1 - \frac{3}{2} \sqrt{1 - \left(\frac{|\mathbf{x} - \mathbf{s}|}{r_s} \right)^2} \right) & |\mathbf{x} - \mathbf{s}| \leq r_s \end{cases} . \quad (2.1)$$

After evaluating each pixel in the lens plane, the source plane is likewise divided into pixels, and the source brightness evaluated directly in each. The magnification is then given as the ratio of the summed image pixel brightness to summed source pixel brightness.

The numerical accuracy of the technique depends strongly on the number and size of the pixels covering the images and source. An accurate calculation of magnification requires a pixel size much smaller than the source, such that the finite source is represented by many pixels. Without knowing where to search for images, the entire lens plane must be covered with these pixels. Consequently, the number of pixels involved in the calculation (and thus the total calculation time) increases quadratically with increasing resolution. It is impractical and very inefficient to perform the calculation for every pixel in the lens plane for each source position, even if the search is restricted to a few Einstein radii.

The core idea in the ray shooting technique is thus to minimise the number of pixels that must be evaluated using Equation 2.1. The implementation in MLENS had two distinct methods for doing this.

The first and most obvious technique, also used in the magnification map method, is to calculate all times in the lightcurve simultaneously. By making the assumption that the lens is static, the mapping between the lens and source plane becomes independent of time. Multiple source ‘targets’ are then placed in the source plane, and the destinations of the directed rays for all pixels are compared with these source positions.

This assumption cannot be made if there is orbital motion within the lens, which makes these techniques a poor choice for modelling events that include lens motion.

The second technique is more subtle and makes use of the lens plane topology discussed in Section 1.4.3. We saw that, provided it does not contain a lens or critical curve, an arbitrary bounded region R in the lens plane is transformed continuously into the bounded region S in the source plane where the boundary of S is given by the transformation of the boundary of R .

If the boundary of S is approximated by a polygon, we can determine whether the source lies inside or outside S using the *even-odd rule* (e.g. Foley [1990]), shown in Figure 2.2.

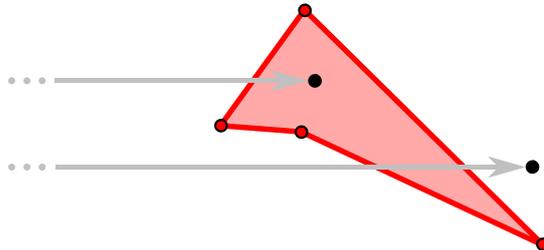
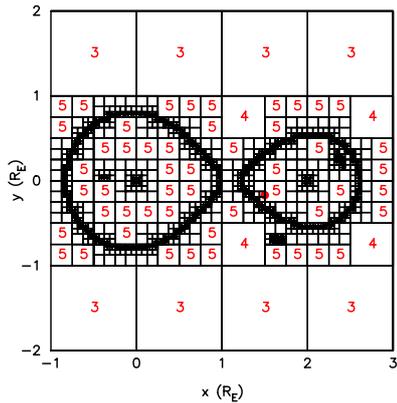


Figure 2.2: A simple method for determining whether a point is inside a (non self-intersecting) polygon is to project a line from infinity to the point. The point is inside the polygon if and only if the line crosses an odd number of edges.

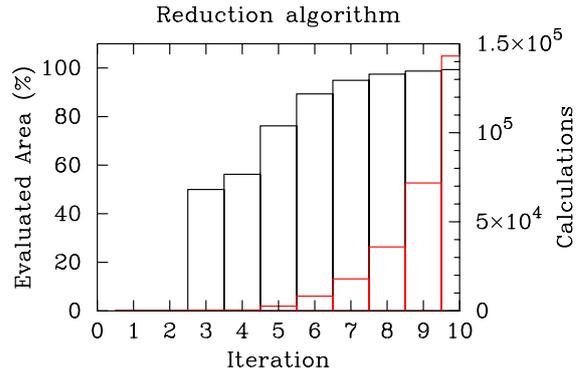
This is complicated slightly by the finite source size; in practice one also needs to check for any edges passing within r_s of the source centre.

This test allows an arbitrarily large region to be checked for the presence of images without having to calculate every pixel within the region. Only a subset of points around the border of the region need to be examined.

Hence regions of the lens plane that do not contain images can be rapidly discarded using an appropriate iterative algorithm; a list of regions to search is maintained, and this initially consists of a single square region that covers the lens plane out to a few r_E .



(a) Lens plane showing the iteration that each region was discarded.



(b) Area (black, left axis) and cumulative number of calculations (red, right axis) for each iteration of the algorithm.

Figure 2.3: The pixel reduction algorithm is applied to a binary lens configuration. The search quickly converges to the regions containing images, lenses, or critical curves with relatively few calculations. Efficiency decreases after covering 90% of the lens plane due to the many small regions to be tested.

Each region is then compared with the lens positions to make sure that none are contained within. Critical curves are searched for by calculating the Jacobian at points around the border. A sign change in the Jacobian indicates that a critical curve has been crossed. If the region contains a lens or critical curve, it is divided into four smaller regions which replace the original region.

A number (dependent on the size of the region, and desired resolution) of points around the edge of the region are selected, and transformed into the source plane. These form a polygon, that is checked for the presence of the source using the even-odd rule. If the source is outside the polygon, and sufficiently far from every vertex, the region is discarded. Otherwise, it is divided into smaller regions to be reevaluated. This is continued until the regions reach a suitable threshold size, where they are marked as requiring ray shooting and then discarded.

This algorithm quickly converges to find the critical curves and image locations, with orders of magnitude less pixels to evaluate. Figure 2.3 shows an example of the algorithm applied to a binary lens. A similar algorithm is also used by the GRAVLENS software (Keeton [2001]) to calculate the critical curves for large distributed gravitational lenses.

Further improvements to the ray shooting method are possible, but were not used in the original MLENS software. For example the lens plane can be tiled with hexagonal pixels, or polar-coordinates (Bennett [2010]), to provide a better representation

of the curved images than square pixels.

2.1.2 Magnification Maps

The ray shooting method is optimised in order to minimise the number of calculations required to evaluate a single model light curve. The magnification map technique takes a different approach. While searching parameter space for the best-fit model light curve, a vast number of lightcurves need to be calculated. The magnification map technique aims to reduce the overall calculation time for the computation of multiple model light curves, so in spite of being less efficient than ray shooting for a single model light curve, it is more efficient when one considers the entire modelling process.

A magnification map provides a numerical approximation to the point-source magnification at each point in the source plane. Thus, the source can be placed anywhere on the map, and its magnification found by averaging the pixels that it overlaps. Limb darkening can be included by weighting each of these pixels by an appropriate intensity.

The map is calculated by directing rays from the observer with a fixed (area) density over a large region of the lens plane. Each ray is traced into the source plane using the lens equation, and binned into the nearest pixel. The magnification associated with a given pixel is simply the ratio of the number of rays that impact that pixel to the number that would hit it if there was no lens system (which is proportional to the density of rays directed towards the lens plane). An example magnification map is shown in Figure 2.4.

The calculation of the map itself can take tens of minutes, and each map applies only to a single combination of mass ratio and lens separation. This requires many different maps to be created. Once the map is generated however, the calculation of lightcurves corresponding to the remaining parameters is extremely fast. The maps can also be stored for future use, but there is obviously a trade-off here between storage capacity and computation time.

The effects of parallax and xallarap can be included by using a curved source track, but lens motion would require a new map to be created for each time within the calculation. This makes it prohibitive to use magnification maps for modelling such events.

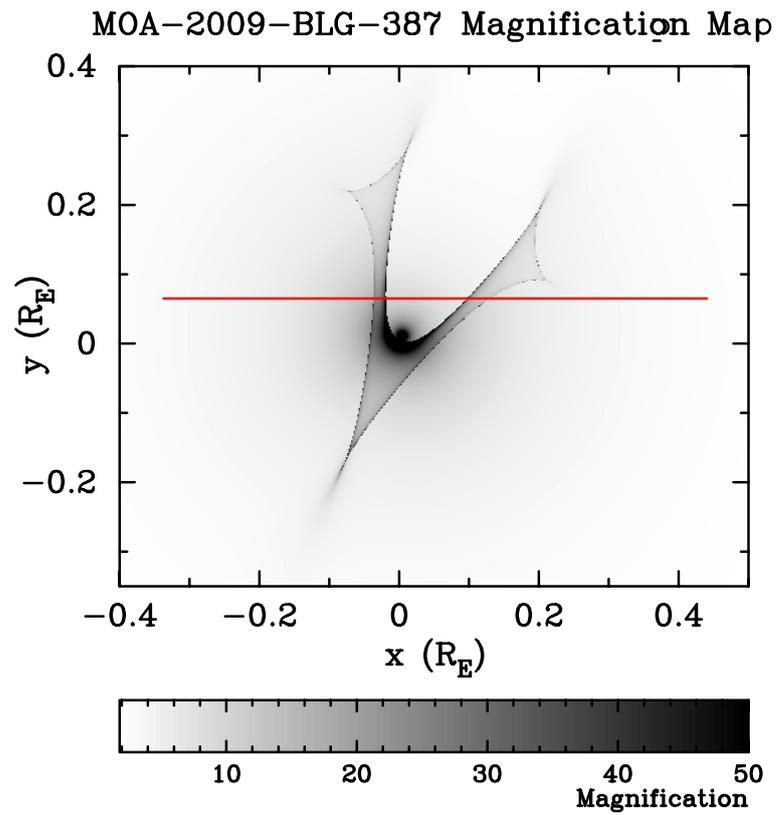


Figure 2.4: Magnification map of an unpublished VUW model for the MOA-2009-BLG-387 event. The resulting lightcurve, shown in Figure 1.7, is produced by moving the source from left to right along the (red) track, averaging the pixel intensities below the source.

2.2 Semi-analytical techniques

Semi-analytical techniques make use of the fact that it is possible to invert the lens equation (Equation 1.15) and produce an equation whose solutions directly yield the image positions for a given source-lens configuration. This is most easily accomplished by using complex variables to represent the two-dimensional vector positions in the image and source planes. The inversion procedure for an N component lens then yields a complex polynomial of degree $N^2 + 1$. There is no general analytical formula for solving the roots of a degree five or higher polynomial, which means that even the simplest multiple lens system must be solved numerically. However, polynomial root finding is a mature area of numerical mathematics so algorithms exist to perform the task. A subset of the polynomial roots then give the image positions in the lens plane.

Having found the image locations, the point-source magnification can be calculated using Equation 1.17. This procedure can be extended to cover finite sources by using algorithms that utilise a selection of nearby multiple point-source magnification values.

Although the lens equation looks relatively simple, the inversion procedure becomes increasingly complex as the number of lens components increases. For this reason, a number of modellers continue to use the ‘brute force’ inverse ray-tracing techniques.

2.2.1 The Lens Polynomial

The first step in converting the lens equation (Equation 1.15) into a polynomial is to represent the vector positions using complex numbers. This reduces the lens equation from a system of two linearly independent equations to a single equation. Making the substitution of $u, x \rightarrow$ real axes, $v, y \rightarrow$ imaginary axes, we define the complex positions z and w by

$$\begin{aligned} \mathbf{r}(x, y) &\rightarrow z = x + iy; & \bar{z} &= x - iy \\ \mathbf{s}(u, v) &\rightarrow w = u + iv; & \bar{w} &= u - iv \end{aligned} \tag{2.2}$$

where z or \bar{z} define a position in the lens plane, and w or \bar{w} define a position in the source plane.

Substituting these definitions into Equation 1.15 gives an equation relating the complex number coordinates of the source (w) to the complex number coordinates of

the images (z and \bar{z}); that is, the lens equation using complex numbers

$$\begin{aligned} w &= z - \sum_{i=1}^N \varepsilon_i \frac{z - z_i}{|z - z_i|^2} \\ &= z - \sum_{i=1}^N \frac{\varepsilon_i}{\bar{z} - \bar{z}_i} . \end{aligned} \quad (2.3)$$

Here, N defines the number of components in the lens, and z_i defines the position of each component using complex number coordinates.

These equations contain both z and \bar{z} . In order to directly relate an image position to a source position without separating the equation into real and imaginary parts, one of these must be eliminated. Rearranging to find an expression for z and conjugating produces

$$\bar{z} = \bar{w} + \sum_{i=1}^N \frac{\varepsilon_i}{z - z_i} . \quad (2.4)$$

On substituting back into Equation 2.3, we obtain an equation in one unknown, z .

$$w = z - \sum_{i=1}^N \frac{\varepsilon_i}{\bar{w} - \bar{z}_i + \sum_{j=1}^N \frac{\varepsilon_j}{z - z_j}} .$$

This equation is not a polynomial, as it contains both positive and negative powers of z . To form a polynomial, we multiply both sides of the expression by factors of z , and rearrange to produce a polynomial in z .

$$\begin{aligned} &\prod_i \left[(\bar{w} - \bar{z}_i) \prod_j (z - z_j) + \sum_j \varepsilon_j \prod_{k \neq j} (z - z_k) \right] (z - w) - \\ &\sum_l \varepsilon_l \prod_{i \neq l} \left[(\bar{w} - \bar{z}_i) \prod_{j \neq i} (z - z_j) + \sum_j \varepsilon_j \prod_{k \neq j} (z - z_k) \right] = 0 . \end{aligned} \quad (2.5)$$

The terms in each sum or product vary over each component in the lens, except where specified.

This equation, while technically a polynomial, is not useful in this form. It must be rearranged into the form $p(z) = \sum_i a_i z^i$ before it can be solved by a numerical algorithm. The coefficients a_i uniquely determine the polynomial, and thus its roots.

The highest order term in the polynomial can be determined from a direct examination of Equation 2.5, and is given by

$$a_{N^2+1} = \prod_i (\bar{w} - \bar{z}_i) .$$

The lower order terms, however, become rapidly more complicated with increasing N , making symbolic manipulation tools like Maple extremely useful (even essential). All the coefficients for the polynomials corresponding to N up to 4 have been determined by a combination of Maple manipulations with subsequent manual rearrangement and are provided in Appendix B. It is worth noting that the $N = 3$ and $N = 4$ cases required extensive manual manipulation to obtain efficient expressions for the resulting polynomial coefficients.

The polynomial coefficients are calculated for the particular source-lens configuration, and then these are passed to a two-stage root finding algorithm. The first stage uses the Jenkins-Traub algorithm (ACM Algorithm 419. Jenkins and Traub [1972, 1970]), which converges rapidly on the roots. Occasional large numerical errors were found to cause problems, such as placing images on the wrong side of a critical curve. This required the addition of a second stage, in which each root is ‘polished’ using the Laguerre method (Riley et al. [2006]) to determine a more accurate result.

The fundamental theorem of algebra states that the number of complex roots of any polynomial is equal to its degree. The lens polynomial is degree $N^2 + 1$, but we saw in Section 1.4.3 that far from the lens there are only $N + 1$ images; only a subset of the determined roots correspond to physical images. The non-physical roots are identified by substituting all root values in the complex number lens equation (Equation 2.3). Those that do not recover the original source position do not correspond to physical images and are discarded.

The remaining complex roots are converted back to 2D image positions by applying the inverse substitution to Equation 2.2. These points can then be substituted directly into Equation 1.17 to produce the point-source magnification.

2.2.2 Critical Curves

A similar approach can be applied to calculate the critical curves. Appendix A shows that the Jacobian in complex coordinates is given by

$$J = \frac{\partial(u, v)}{\partial(x, y)} = \frac{\partial(w, \bar{w})}{\partial(z, \bar{z})} .$$

Using the symmetry between Equations 2.3 and 2.4, the Jacobian can be written as

$$\begin{aligned} J &= \frac{\partial w}{\partial z} \frac{\partial \bar{w}}{\partial \bar{z}} - \frac{\partial w}{\partial \bar{z}} \frac{\partial \bar{w}}{\partial z} \\ &= 1 - \left| \frac{\partial \bar{w}}{\partial z} \right|^2 . \end{aligned}$$

The critical curves (given by $J = 0$) are thus the solutions to

$$\frac{\partial \bar{w}}{\partial z} = \sum_{i=1}^N \frac{\varepsilon_i}{(z_i - z)^2} = e^{i\varphi}$$

where $e^{i\varphi}$ ($\varphi \in [0, 2\pi)$) defines the unit circle on the complex plane.

As before, we can convert this into a polynomial by multiplying through by the denominator of each term to produce

$$\sum_i \left[\varepsilon_i \prod_{j \neq i} (z_j - z)^2 \right] - e^{-i\varphi} \prod_i (z_i - z)^2 = 0. \quad (2.6)$$

This result was first published by Witt [1990].

Examining the second term, we see that the polynomial is of order $2N$, with the highest coefficient given by $a_{2N} = -e^{-i\varphi}$. All $2N$ roots for each φ correspond to valid points in the critical curve.

The remaining coefficients for N up to 4 are given in Appendix C.

2.3 Hexadecapole Approximation

Finite source effects are often important in modelling microlensing light curves, particularly when the source is near a caustic where the magnification changes rapidly with source position. We saw in Section 1.5 that the point-source approximation behaves poorly in these regions, making it inappropriate by itself.

The Hexadecapole approximation (Gould [2008]) employs multiple point-source values to estimate the magnification resulting from a finite source, including limb darkening effects if required. While this approximation provides an accurate model over a much larger region than the point-source model, it fails when the source crosses a caustic. The technique is computationally much faster than the polygon method discussed next, so this method provides a useful bridge between the polygon and point-source techniques.

The point-source magnification is evaluated at 13 points on the source disk, and used to interpolate (up to the hexadecapole term) the underlying magnification field covered by the source. The magnification is then found by using these values to approximate the analytical integral of the magnification field over the source disk, weighted by the specified limb darkening profile. The spacing of these 13 points are shown in Figure 2.5.

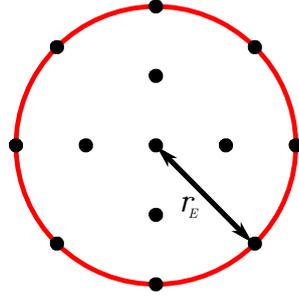


Figure 2.5: The hexadecapole approximation is constructed from 13 point-source magnifications sampled across the source disk.

Gould shows that this technique provides a simple approximation for the magnification,

$$A_{\text{finite}} = A_0 + \frac{A_2 r_s^2}{2} \left(1 - \frac{1}{5}\Gamma\right) + \frac{A_4 r_s^4}{3} \left(1 - \frac{11}{35}\Gamma\right). \quad (2.7)$$

The three terms correspond to the monopole, quadrupole, and hexadecapole terms in the expansion, and r_s gives the radius of the source disk.

The monopole term A_0 is simply the point-source magnification evaluated at the centre of the source disk. A_2 and A_4 are defined by

$$A_2 r_s^2 = \frac{16A_{r_s/2,+} - A_{r_s,+}}{3} \quad A_4 r_s^4 = \frac{A_{r_s,+} + A_{r_s,\times}}{2} - A_2 r_s^2.$$

where $A_{r_s,+}$ and $A_{r_s,\times}$ are the average magnifications calculated at four points on the edge of the source disk, in a ‘+’ and ‘×’ configuration. Similarly, $A_{r_s/2,+}$ is the average of the four ‘+’ points located at half the radius of the source disk.

The approximation fails near caustics, where the magnification field cannot be accurately represented by the hexadecapole expansion. This is demonstrated in Figure 2.6, where a $0.005 r_E$ source approaches crossing a ‘fold’ caustic (far from any cusps). The hexadecapole approximation remains within 0.01% to within $1.5 r_s$ of the caustic, far better than the point-source approximation. The sharp changes in magnification occur as each of the point-sources that are used in the hexadecapole calculation cross the caustic.

2.4 Polygon Method

The large changes in magnification that occur when the source crosses a caustic provide a strong constraint on model parameters. This also makes them difficult to model; the diverging Jacobian means that neither the point-source nor hexadecapole approximations can be applied. An entirely different approach is needed.

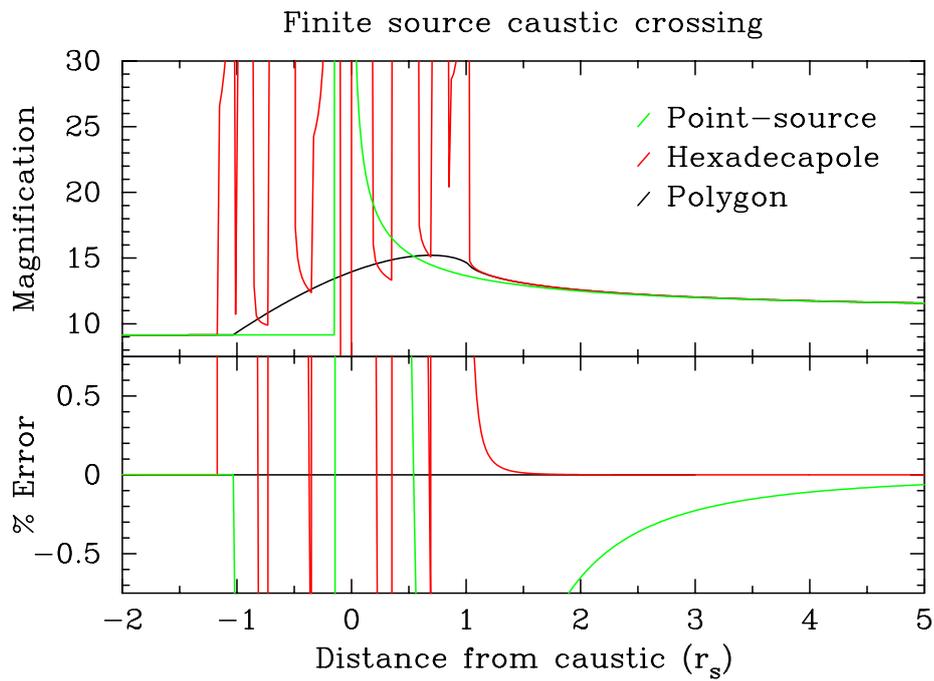


Figure 2.6: Magnification of a $0.005 r_E$ source with varying distance from the caustic (measured in source radii). The hexadecapole approximation provides a more accurate magnification than the point-source approximation close to the caustic. Both approximations provide an incorrect magnification when the source is within $1 r_s$ of the caustic.

The polygon method is based on the ‘Stokes theorem’ algorithm by Gould and Gaucherel [1997], extended to support any number of lenses for which the lens polynomial can be solved.

In summary, the source disk is approximated by a polygon with N vertices around its perimeter. The lens polynomial is then solved for each of these vertices to produce corresponding polygons, also with N vertices, for each of the images. The areas of these polygons can be evaluated analytically, and therefore the total magnification can be estimated from the ratio of image area to source area. Limb darkening effects can be included by using multiple concentric polygons to represent the source disk and weighting the annular areas accordingly.

Area of a Polygon

A polygon is defined by an ordered list of vertices, connected by edges. The area enclosed within a polygon can be found by considering the signed area beneath each edge.

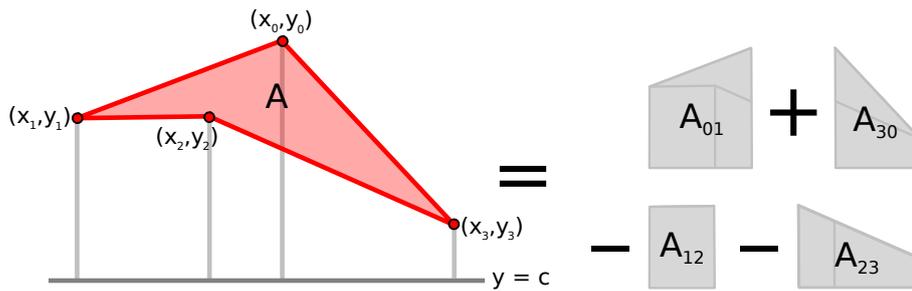


Figure 2.7: The area inside by a (non self-intersecting) polygon is given by the sum of (signed) areas below each edge. $A = A_{01} - A_{12} - A_{23} + A_{30}$.

Consider the 4-sided polygon in Figure 2.7. A_{01} defines the area of the the parallelogram formed between the edge connecting the first two vertices, and the line $y = c$

$$\begin{aligned} A_{01} &= \frac{1}{2} (x_0 - x_1) (y_0 + y_1 - 2c) \\ &= \frac{1}{2} (x_0 y_1 - x_1 y_0) + \frac{x_0 (y_0 - 2c)}{2} - \frac{x_1 (y_1 - 2c)}{2}. \end{aligned}$$

Similarly, A_{12} is found by considering the edge between the second and third vertices

$$A_{12} = \frac{1}{2} (x_1 y_2 - x_2 y_1) + \frac{x_1 (y_1 - 2c)}{2} - \frac{x_0 (y_0 - 2c)}{2}.$$

Notice that the second term of A_{12} cancels the third term of A_{01} . When all edges are considered, all terms containing c cancel. Notice that $x_2 > x_1$ and $y_2 = y_1$,

making A_{12} negative. Provided the vertices are ordered in an anticlockwise fashion, the resulting areas will have the required sign for the sum to give the enclosed area directly.

It is important to include the final edge connecting the last vertex back to the first. This is achieved by inserting an extra vertex, $(x_M, y_M) = (x_0, y_0)$, which allows the area to be calculated by

$$A = \frac{1}{2} \sum_{i=1}^M (x_{i-1}y_i - x_i y_{i-1}) .$$

The sign of the area for a given image will depend on its parity (discussed in Section 1.4.3). Images with negative parity can be thought of as a mirrored image of the source, which reverses the order of the vertices and gives a negative area.

A more subtle effect occurs when the source entirely covers a caustic, producing an image with holes. The simplest example of this is shown in Figure 2.8, where a (single) lens and source are aligned.

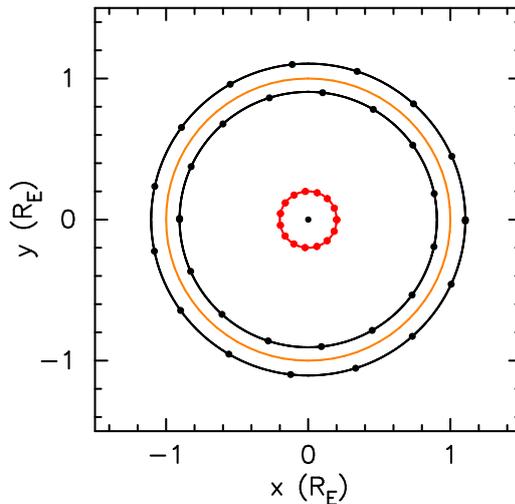


Figure 2.8: The two images of a source that transits behind the lens merge across the Einstein ring to form a single continuous ring. The ring area is given by the difference between the two image polygon areas.

The vertices in the two polygons are ordered in the same sense, and so both have positive area. The inner polygon (with negative parity) should define the hole in the image, and so its sign is again reversed from its desired value. Gould and Gaucherel [1997] showed that these incorrect signs can always be corrected by multiplying each polygon's area by its parity. This is extremely fortunate, as it allows the global area to be calculated by considering each polygon locally, without any information of the overall image topology.

Polygon Construction

The vertices in the image polygons are calculated by first choosing points around the edge of the source disk and then solving the lens polynomial roots for each position. These image points must then be grouped into the corresponding physical images.

The position of a vertex on the edge of the source disk can be parametrised using an angle θ . For a source centred at (x, y) with radius r_s , the vertex at angle θ is given by

$$\mathbf{v}(\theta) = \begin{bmatrix} x \\ y \end{bmatrix} + r_s \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} .$$

The task then involves determining the set of angles $\theta \in \{\theta_i\}$ that give the best distribution of vertices around the images, which in high magnification situations are extremely distorted from the circular source disk shape. If the image vertices are spaced too far apart, the numerical accuracy of the method will degrade; if they're spaced too closely then computing time is wasted calculating unnecessary vertices.

As a first approximation, we can space N vertices evenly around the source disk

$$\theta_i = \frac{2\pi i}{N} \quad \text{for } i = 0..(N - 1) . \quad (2.8)$$

The resulting distribution of vertices in the images (shown for a single lens in Figure 2.9a) can be highly irregular. The high magnification region close to the lens spreads adjacent vertices far apart leading to inadequate numerical accuracy, while those in lower magnification regions are close enough to yield good accuracy.

A better approach is to dynamically space the vertices around the source such that spacing of vertices around the images remains roughly constant. An algorithm was created to obtain this result, which required the introduction of two resolution parameters

- \mathbf{N}_{\min} Minimum number of source vertices.
- \mathbf{d}_{\max} Maximum allowed spacing between image vertices,

where \mathbf{N}_{\min} defines the maximum angular separation of two vertices by

$$\Delta\theta_{\max} = \frac{2\pi}{\mathbf{N}_{\min}} . \quad (2.9)$$

The first source vertex is placed at $\theta_0 = 0$, and the corresponding image vertices calculated. The next vertex is chosen as $\theta_1 = \theta_0 + \Delta\theta$, where $\Delta\theta$ is initially set to $\Delta\theta_{\max}$. The corresponding image vertices are calculated and compared with the vertices from the previous step. If the maximum distance s_i between vertices in any

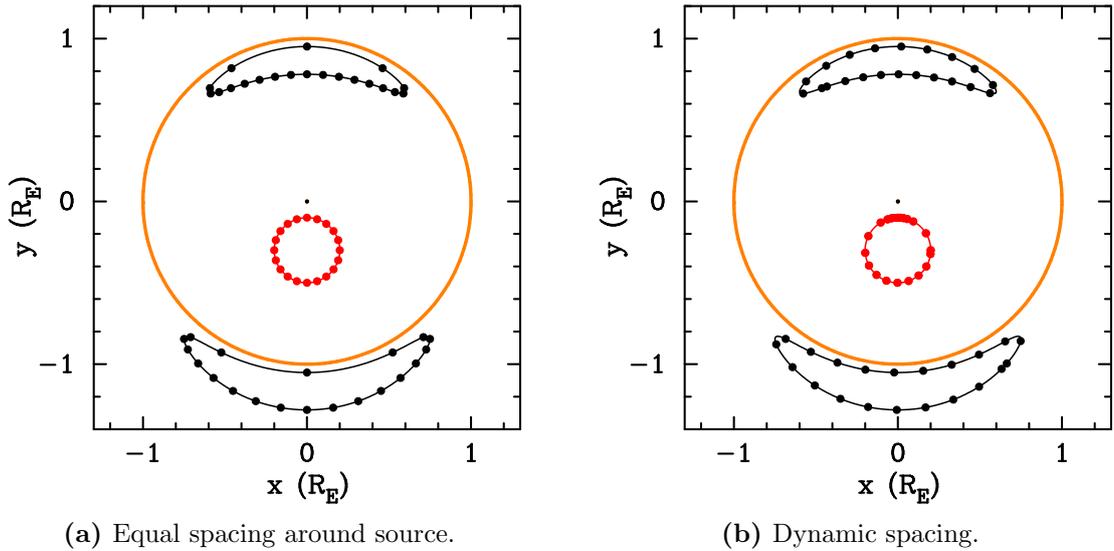


Figure 2.9: Two algorithms for vertex placement. Filled circles represent vertices in the source (red) and images (black). Solid curves represent the actual image and source borders.

image exceeds d_{\max} , the current vertices are discarded and the calculation repeated with $\Delta\theta$ halved.

Once the step is successful, the $\Delta\theta$ for the next step is calculated by linearly estimating the source-plane spacing required for a lens plane spacing of $0.75d_{\max}$

$$\Delta\theta_{i+1} = \min \left(\Delta\theta_{\max}, \frac{3d_{\max}}{4s_i} \Delta\theta_i \right). \quad (2.10)$$

These steps are repeated until the entire source has been covered ($\theta \geq 2\pi$). An extra vertex is calculated with $\theta > 2\pi$, and then discarded, in order to ensure that the distance between the last and first vertex in each polygon are within d_{\max} . The improved distribution of image vertices that this method produces are shown in Figure 2.9b.

Vertices are grouped into polygons incrementally by matching the vertices obtained at each step with those from the previous step. For a single lens, the analytical expression for the image vertices makes this assignment simple. All of the ‘-’ vertices go together, and all of the ‘+’ vertices go together.

In the general case, the matching must be done numerically. The separation of vertices within the same image are (with one exception) much smaller than the separation between different images. The matching vertex from the previous step is thus the closest. This may not hold if the source is near a caustic, due to a pair of images elongating towards the critical curve. In this case, the opposite parity of the images is sufficient to distinguish vertices between the two. The most complex

case occurs when the source straddles a caustic, as the region inside the caustic appears in more images than the region outside. When stepping around the edge of the source, and crossing a caustic, a pair of vertices will appear or disappear in the images. Special care must be taken to ensure that they are combined into a polygon that correctly crosses the critical curve.

The approach taken here is to group the vertices first into curves. Each curve records its parity, a reference to adjacent curves at either end (initially undefined), and a list of vertices. Only after the entire source has been evaluated are these curves merged into a set of polygons.

An initial set of curves are created for the first set of image vertices. At each step, vertices are matched against the set of curves that do not have an end-neighbour defined. When a caustic is entered, the matching procedure will leave two vertices unpaired. The vertices will be on opposite sides of a critical curve, with opposite parity. If necessary, the step size is adjusted so that their separation is less than d_{\max} , then two new curves are created, one for each vertex, with each start-neighbour set to the other curve. This assignment allows the curve to be followed across the critical curve in order to create a composite polygon for the two partial images. On stepping out of a caustic, two curves will be left with unpaired vertices. The final vertices in the curves again lie on opposite sides of a critical curve, with opposite parity. The step size is adjusted if necessary so that these final vertices are within d_{\max} , and the end neighbour of each curve set to the other.

An example of the resulting curve segments are shown in Figure 2.10a. The segments will all start and end at a critical curve, except for those truncated by the calculation starting and finishing at $\theta = 0$. The curves that have neither a start-neighbour or end-neighbour defined specify an image that doesn't cross a critical curve. The first and last vertices are checked to ensure that they are within d_{\max} , and polygons created for these images. The remaining curves are joined to form polygons that span multiple partial images. The curves that form these polygons will have at least one of the start-neighbour or end-neighbour curves defined. Any curve with a single neighbour must start or end at $\theta = 0$, and is matched with the closest unmatched end of the remaining curves with equal parity.

These remaining curves (if any) define at least one polygon that spans a critical curve. The curve segments on opposite sides of the critical curve have opposite parity, and thus an opposite vertex ordering. This can be seen in Figure 2.11, where the vertices converge or diverge at the critical curve. Swapping the vertex ordering of one of these creates a consistent ordering within the grouping. An 'effective parity' is chosen for the polygon, and the vertex ordering flipped for the curves that have the opposite parity. The curves are then combined to form the polygon.

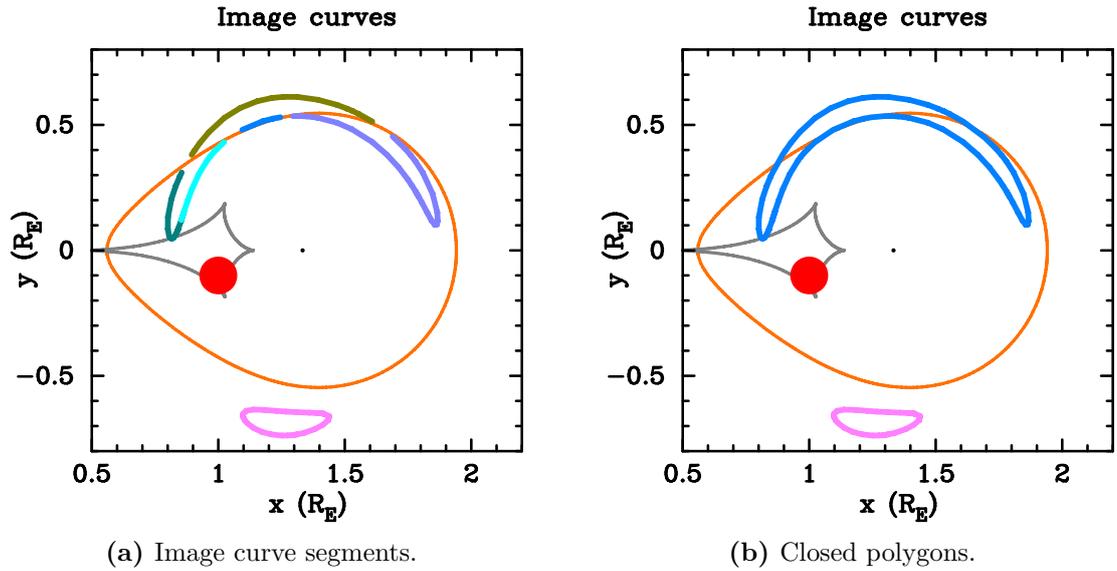


Figure 2.10: The polygon algorithm first finds a series of curve segments defining the edges of the images. They are then connected into a set of closed polygons.

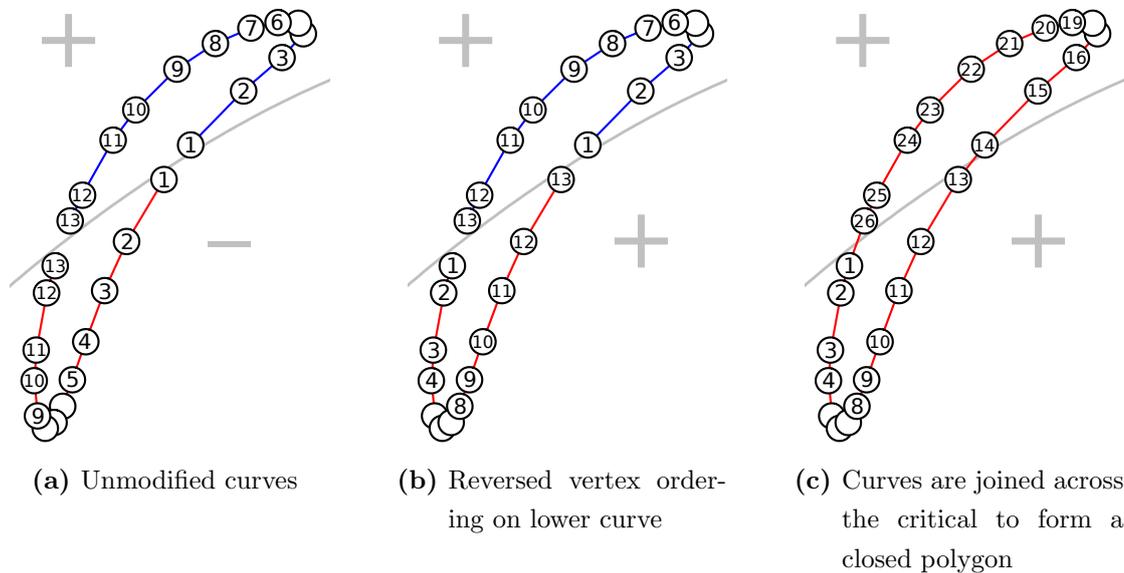


Figure 2.11: Two image curves meeting at a critical curve have opposite parity and vertex ordering. Reversing the vertex ordering of one of the curves changes its effective parity, and allows the curves to be joined to form a single polygon that covers both images. The '+' and '-' show the effective parity of the two curves at each step.

The end result of this curve matching is a set of closed polygons that define the spatially distinct images, shown in Figure 2.10b.

Limb Darkening

Using the polygon areas as above to determine the magnification is fundamentally incompatible with any limb darkening model, as it effectively considers the source and image brightness to be constant. However, limb darkening effects can be included by dividing the source disk into concentric polygons, each of which *can* then be treated as having a constant brightness. The individual images corresponding to each region are found, and then added together with a weighting determined from Equation 1.22 to construct a composite image containing relative brightness information. Given the (normally assumed) circular symmetry of the source, the natural choice is to divide the source into a series of successively larger annuli, which produce approximately radial contours in the resulting images, as shown in Figure 2.12.

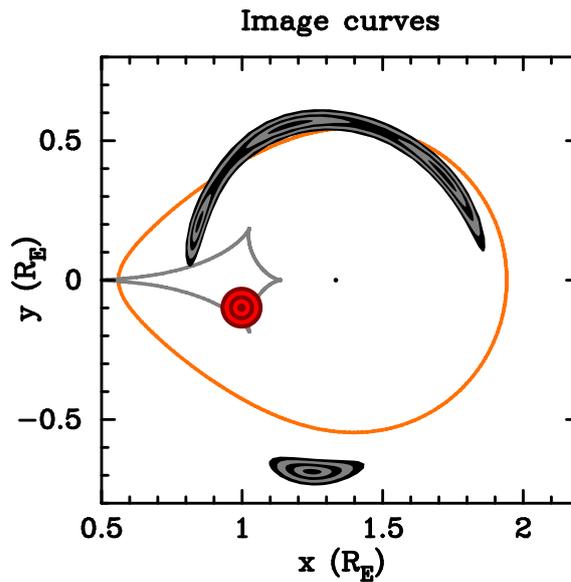


Figure 2.12: Limb darkening effects are modelled by dividing the source into a set of annuli and finding the images of each. The area of each partial image is weighted by a brightness factor and added to find the total image flux.

The radial width of each annulus can be made constant, or chosen so that the change in intensity is constant across the width of each.

For a constant radius, the outer edge of the i 'th annulus is simply

$$r_i = \frac{iI(r_s)}{N} . \quad (2.11)$$

The spacing for a constant intensity change is found by inverting Equation 1.22

$$\frac{r_i}{r_s} = \sqrt{1 - \frac{4}{9} \left(1 - \frac{1 - I_i}{\Gamma}\right)^2}. \quad (2.12)$$

The intensity at the outer edge of the i 'th annulus is found by interpolating the intensity at the centre and edge of the source disk (found using Equation 1.22) across the N annuli

$$I_i = \frac{I(0)}{I_0} + \frac{i}{N} \left(\frac{I(r_s)}{I_0} - \frac{I(0)}{I_0} \right). \quad (2.13)$$

Substituting I_i into Equation 2.12 then gives the outer radius.

The spacing of the resulting annuli for the two methods are shown in Figure 2.13. More analysis on the numerical accuracy of the two methods when applied to real models is required before a decision can be made as to which method is the better.

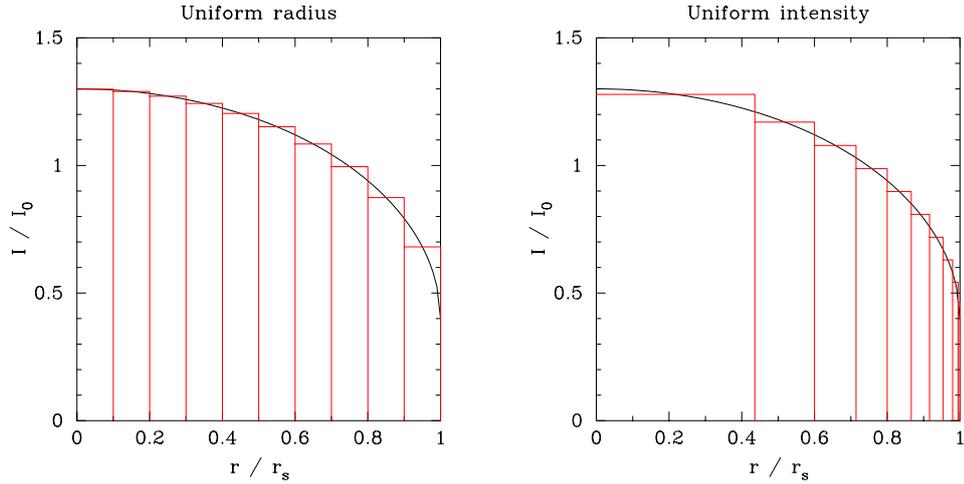


Figure 2.13: The intensity profile of a limb-darkened source can be approximated by annuli with uniform changes in radius or uniform changes in intensity.

The image polygon for an annulus requires roughly twice the number of vertices (and thus, calculations) as a solid disk of the same size. Half of these vertices will be redundant, as the outer edge of one annulus is shared with the inner edge of the next. It is more efficient, therefore, to calculate the image polygons for a collection of increasingly larger disks, and to find the area of each annulus by taking differences.

If A_i describes the area of the i 'th disk image polygons, the weighted flux of the composite image is obtained via

$$F = \frac{I(0)}{I_0} A_0 + \sum_{i=1}^{N-1} \left[\frac{I\left(\frac{r_i+r_{i-1}}{2}\right)}{I_0} (A_i - A_{i-1}) \right]. \quad (2.14)$$

The same procedure is applied to calculate the source flux from the source annulus areas.

The annulus image areas can in principle be stored and reused to efficiently calculate the magnification for a range of increasing source sizes with limb darkening. If the image and source areas are calculated for a source of a given size, only a single, larger, annulus area needs calculated in order to efficiently find the magnification of a larger source.

2.5 Hybrid techniques

A third class of methods exists, which combines elements of both inverse raytracing and semi-analytical techniques. Typically, the lens polynomial is solved to find the location of the images, which restricts the regions which must be targeted with rays. The ‘loop linking’ approach of Dong et al. [2006] is similar to the polygon method, but the resulting polygon is ray traced which avoids the difficulty involved with modelling limb darkening. Bennett [2010] gives a summary of different ‘flood-fill’ approaches, where the image positions found from the lens polynomial are used as seeds for an algorithm that expands outward to cover the images with rays. Due to time constraints, these techniques were not studied in any further detail during this thesis programme.

Chapter 3

Modelling Procedures

Generating theoretical lightcurves for a specified set of model parameters is an interesting theoretical task in itself, but only really becomes useful once these models are applied to real observational data. There is no simple direct relationship between the model parameters and the resulting theoretical lightcurve, which means that there is no straightforward method for extracting parameters from any observational data.

The task of generating a theoretical model to match the observed data thus becomes one of searching for a needle in an N dimensional haystack. This haystack contains a wide range of models with different parameters, each producing a different theoretical lightcurve which must be compared with the observed data. Those that do not match the data to within a specified fit criteria are discarded, and new models are generated. This is repeated over many iterations until the best fitting theoretical lightcurve is found.

The two common methods for searching parameter space for models are to regularly sample the space with a lattice of points, iteratively honing in on the best fitting region, or to apply stochastic techniques based on Monte Carlo Markov Chains. These searching algorithms are beyond the scope of this thesis, but many references are available in the literature, e.g. Bennett [2010]. Likewise, the relationships between model parameters and physical quantities are not discussed.

A point to note about these searches is that a huge number of theoretical lightcurves must be generated before the best fitting model is found. This necessitates the use of massively parallel computing grids, and requires the lightcurve generation to be as efficient as possible.

The quality of fit between the theoretical lightcurve and observed data is quantified

via the χ^2 fit

$$\chi^2 = \sum_i \left(\frac{O_i - M(t_i)}{\sigma_i} \right)^2 \quad (3.1)$$

where O_i and σ_i are the observed magnification and uncertainty at the time t_i , and $M(t_i)$ is the theoretical magnification at that time. If the uncertainties are correctly estimated and the model curve describes the data well, we should expect that each term of the sum is of the order of 1. It can be shown that $\chi^2/\text{DOF} \sim 1$ for a good model, where DOF is the number of degrees of freedom (the number of data points minus the number of model parameters). After a ‘good’ model has been found for an event, the errors in each data set are often rescaled to give $\chi^2/\text{DOF} = 1$. This allows comparable models to be compared by their relative $\Delta\chi^2$. In any case, a smaller χ^2 indicates a better fit. For further discussion on comparing and fitting the model and observational data, see Korpela [2007].

3.1 mlens2

The creation of a new software tool that uses the semi-analytical models developed and presented in this thesis was a significant effort, and the main output of this thesis programme. Despite its beginnings as an incremental update to the original MLENS code developed by Korpela [2007], the replacement of the inverse ray shooting calculation techniques with the semi-analytical approach means that little of the original code remains and that mlens2 stands alone as a distinctly different code.

The original software used the magnification map and ray shooting techniques discussed in Chapter 2, and was tightly integrated with the ECS grid for searching parameter space. The new software features an entirely new set of calculation routines, and a general parallel computing layer which allows it to run on both computing grids available at VUW, as well as on a standard workstation computer (at a much reduced capacity).

The core software tasks in mlens2 are assembled into two executables, *mljob* and *mlserver*, which overall involved approximately 13000 lines of C++ code. It has been tested to run on Mac OS X, Windows, NetBSD, and several distributions of Linux using the gcc compiler. Several utility scripts also exist for converting raw observational data into the format required by mlens2.

The *mlserver* executable controls the global parameter space search to determine the best model for an observed event. Given an appropriate set of initial conditions, it divides the initial calculation workload into batches that are submitted to the un-

derlying grid software to be calculated in parallel. The server will track the returned results and assign further calculations until the best fitting model is obtained. At the culmination of the thesis work, the server is able to communicate with the ECS and SCS grids, or emulate a small grid when running on a single computer, and read the returned χ^2 values from each job. It currently lacks the logic required to generate the jobs to be calculated and to interpret the returned χ^2 values to determine the best fit. An important element of future work will be to complete this logic, which will enable *mlens2* to be used for searching parameter space for models for new events.

The *mljob* executable contains the semi-analytical methods presented in this thesis, plus the logic required to compare the model lightcurve with observational data to calculate a χ^2 fit value. It also contains routines to plot model lightcurves with fitted data, plus an interactive visualisation mode which was used to generate many of the figures in this thesis. A brief overview on the structure and operation of *mljob* is provided in Appendix D.

3.2 Algorithm Selection

The lightcurve calculation has two requirements. The first is to be accurate, the second to be efficient. The numerical root finding of the lens polynomial is a major computational bottleneck, and so the best way to reduce the calculation time is to minimise the number of times these roots must be found, while retaining numerical accuracy to an acceptable level.

Of the three semi-analytical techniques, the point-source approximation requires only one set of roots to be found. Far from the caustics, where the approximation is valid, this is the best choice. Closer to the caustics where the single-lens approximation does not apply, the hexadecapole approximation requires an additional 12 sets of roots to be found. This is an order of magnitude slower than the point-source approximation, but still much faster than the polygon method, which can require thousands of sets of roots to be found for a high magnification event with limb darkening. Thus, a mechanism is needed in which to specify which calculation to use for each magnification calculation.

The simplest method is to allow the user to specify the calculation regions manually, by generating a lightcurve using each technique and noting the regions where the approximations provide accurate results. Provided the gross lightcurve features are well constrained, these regions can be applied to subsequent model calculations. This provides a sufficient solution that works well when refining an already good

model, which forms a significant fraction of the overall modelling.

This will not work for general parameter space searches, as the model lightcurve features will not be in any way constrained. An algorithm is needed to calculate these regions automatically for each individual model. Gould [2008] suggests that with current-generation telescopes, an accuracy of 0.1% in the data is possible. This sets a limit on the acceptable numerical error for the model lightcurve.

In the following analysis, the numerical error of the polygon technique is assumed to be zero, justified by specifying that a large number of vertices should be calculated per polygon. The numerical error in the hexadecapole and point-source approximations are then given as the difference in calculated magnification with respect to the polygon calculation.

3.2.1 Caustic Distance Algorithm

Gould [2008] states that an upper bound on the error in the hexadecapole approximation is given by

$$\frac{\Delta A(z)}{A} = \frac{25}{2^{10}} z^{-6}$$

where z is the tangential distance between the source and caustic. This suggests that *distance to nearest caustic* may be an effective means of separating the calculation regions.

A simple algorithm was created that calculated the caustics for the event (using the method in Section 2.2.2), and the distance between each source position and the nearest caustic found. Sources closer than $5 r_s$ to a caustic were calculated with the polygon method, those between $5r_s$ and $10r_s$ were calculated with the hexadecapole approximation, otherwise the point-source approximation was used.

Figure 3.1 shows a magnification map and caustics for the OGLE-2005-BLG-390 event, superimposed with the source track and calculation regions.

This algorithm worked well on several events that were tested, but major problems were encountered with the event OGLE-2006-BLG-109, which is discussed in more detail in Chapter 4.

The first problem arises when there is significant orbital motion, which cause the caustics to change with time. The basic algorithm required the caustics to be recalculated for each unique time, and as the caustics are found by finding the roots of a polynomial, this adds a significant calculation overhead across the thousands of points that were calculated. In practise, this will not be a large problem as orbital motion is generally not modelled until the gross lightcurve features have been well

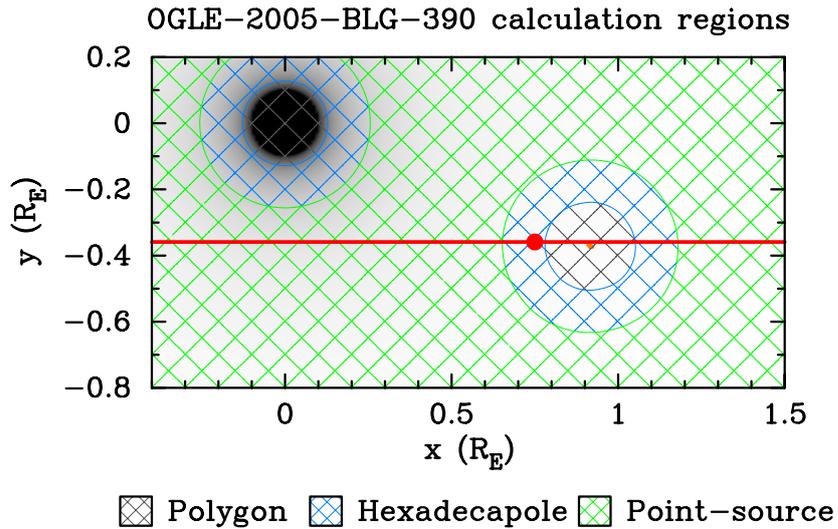


Figure 3.1: Caustic distance calculation regions for OGLE-2005-BLG-390. The caustic (orange) that the source passes over is much smaller than the source itself (red disk, track).

constrained. This means that the calculation type regions can be specified manually, but is still a limitation of the method.

The second problem occurs when the source track passes outside a caustic cusp, where a long extension of high magnification can extend beyond the point of the cusp, into the point-source approximation region. This was an issue with OGLE-2006-BLG-109, as one of the lightcurve features is caused by one of these cusp passages. Figure 3.2 shows the calculation regions near the cusp, while Figure 3.3 shows the error that the incorrect use of the point-source approximation introduces into the lightcurve.

Although this problem could be worked around by adding a special check for caustic cusps, it was decided to abandon this method in favour of an approach that treated the cause of the failures directly.

3.2.2 Magnification Curvature Algorithm

A more intuitive approach is to target the cause of the approximation failure; large nonlinear changes in the magnification field across the source disk. This non-linearity can be approximated analytically by considering the derivatives of the point-source magnification, calculated from derivatives of the Jacobian.

Consider the point-source magnification, M , calculated at the centre of the source using Equation 1.17. The maximum rate of magnification change at this point is

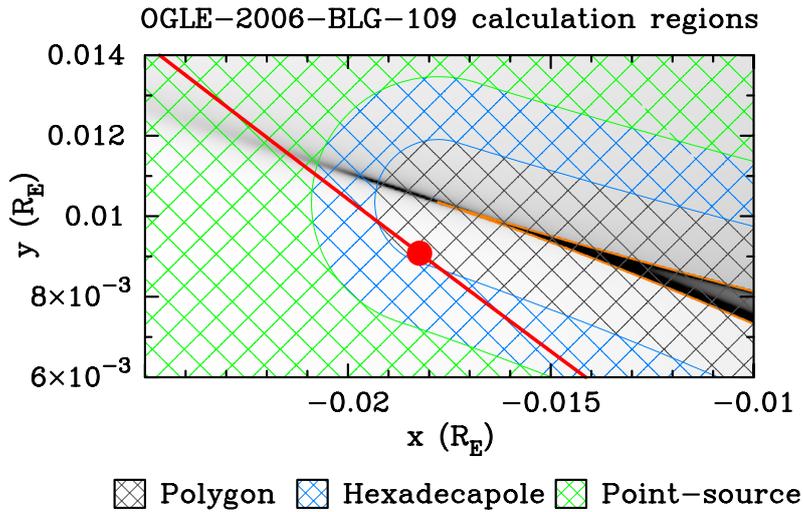


Figure 3.2: A magnification map of the the source plane around the final OGLE-2006-BLG-109 cusp crossing, showing the calculation type regions. A long finger of high magnification extends beyond the point of the cusp into the point-source calculation region.

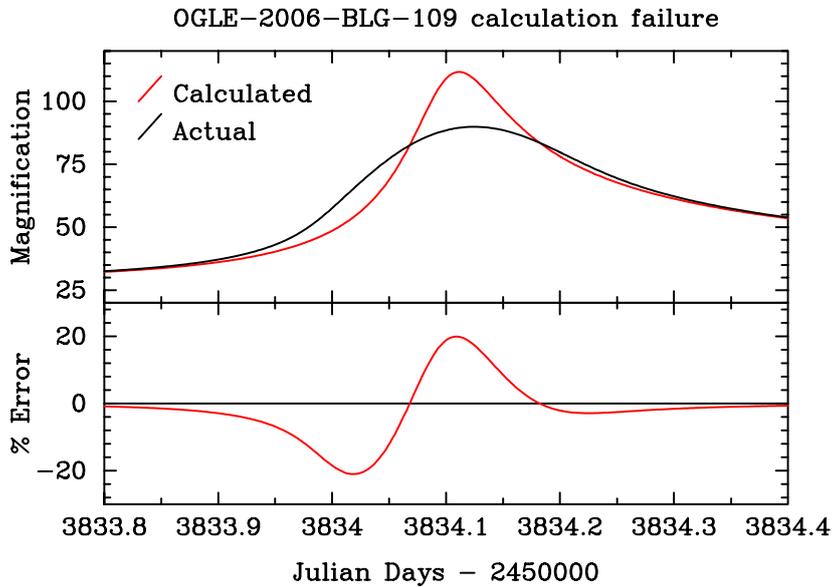


Figure 3.3: The calculated magnification (using the point-source approximation) produces an error of up to 20% from the true magnification as the source crosses the extension of high magnification from the caustic cusp.

given by the length of the gradient vector

$$G = |\nabla M| = \frac{\sqrt{\left(\frac{\partial J}{\partial x}\right)^2 + \left(\frac{\partial J}{\partial y}\right)^2}}{J^2}.$$

This gradient approximates the linear change in magnification, which does not affect the accuracy of the approximations. To find the nonlinear changes, we must consider the gradient of this gradient, which can be thought of as a curvature

$$C = |\nabla G| = \sqrt{\left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2}.$$

The full expression for the curvature involves the 12 unique first, second, and third order derivatives of the lens equation, making it too long to present here. Despite its apparent complexity, the curvature can be constructed efficiently at a known image position. As discussed earlier, the time consuming calculation is in solving these image positions from the lens polynomial. A different value for curvature will be obtained at each image position, of which the maximum is taken. The curvature gives a value in terms of magnification/ r_E^2 , so multiplying by the squared radius of the source gives an estimate of the nonlinear change in magnification across the source disk.

The calculation type is chosen by comparing this value to the central magnification by

$$\delta = \frac{r_s^2 C}{M}.$$

Numerical investigation of OGLE-2006-BLG-109 found that the point-source approximation was suitably accurate for $\delta < 10^{-3}$, and the hexadecapole approximation for $10^{-3} < \delta < 10^{-2}$. The polygon method is used otherwise.

Figure 3.4 shows the results of applying this algorithm to the region presented in Figure 3.2. The region covering the magnification extension is (correctly) selected as requiring the polygon method, while the majority of the remaining area is calculated using the hexadecapole approximation.

A problem was encountered with sharp changes in the magnification curvature itself near caustics. It was possible for the centre of the source to have C small enough to select the point-source or hexadecapole approximation, while the edge of the source lay over a caustic, giving an incorrect magnification calculation. This was solved by calculating the curvature at the 8 points on the edge of the source used for the hexadecapole magnification calculation, and then falling back to the point-source approximation if the curvature is low. This causes the overall calculation to be at

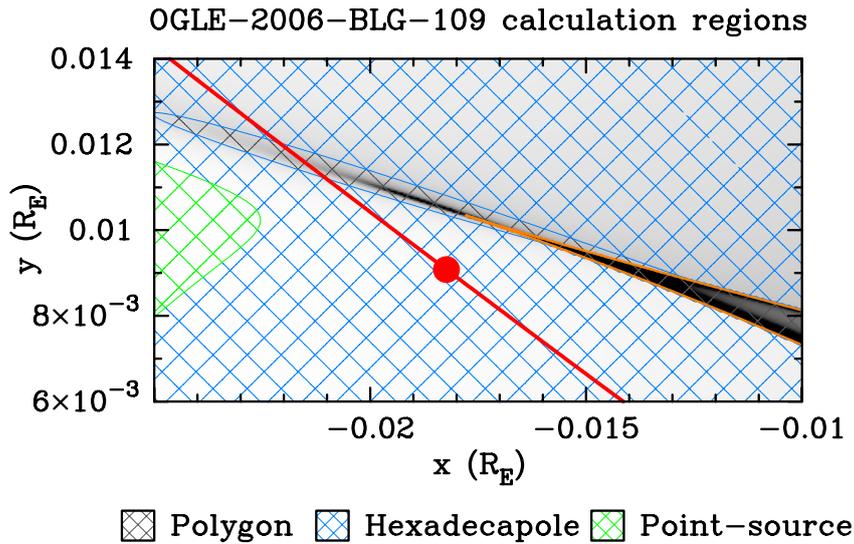


Figure 3.4: The curvature algorithm correctly selects the polygon method calculation in the region around the magnification extension.

best 9 times slower than the normal point-source calculation, but ensures that the correct calculation type is chosen.

The resulting lightcurve for the entire OGLE-2006-BLG-109 event is shown in Figure 3.5. The error remains below 0.1% as required. The δ threshold values can be adjusted if necessary to suit the desired trade-off between calculation speed and accuracy.

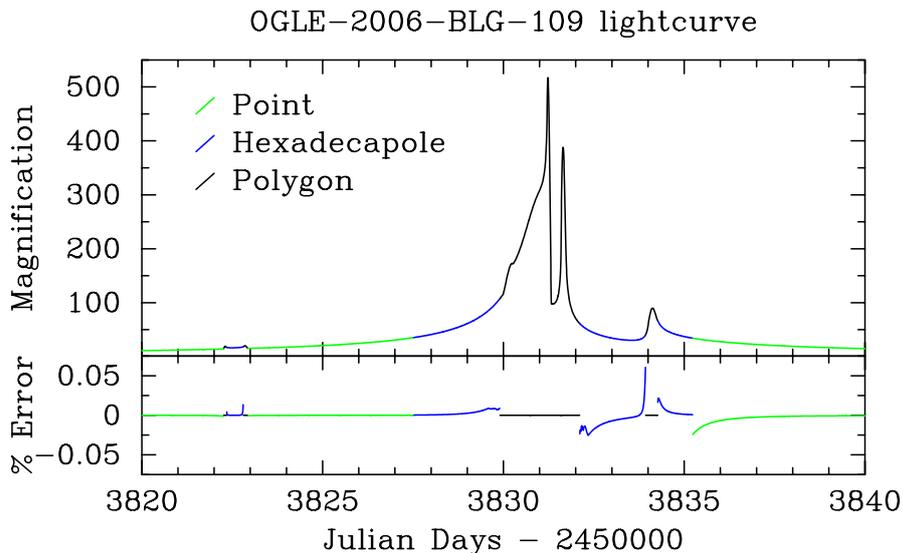


Figure 3.5: The OGLE-2006-BLG-109 lightcurve shows the calculation methods selected by the magnification curvature algorithm at each time.

Chapter 4

Modelling Selected Events

As discussed in the previous chapter, the current version of the `mlens2` software package does not include code to efficiently search model parameter space in order to find the “best” model for any given data set. This higher level of coding is not conceptually difficult, and a framework for doing so has been partially completed. This remaining coding will need to be completed in order to make `mlens2` a general purpose package for modelling microlensing events, but was considered beyond the scope of the thesis programme primarily due to time constraints. This was deemed acceptable as all the interesting physics is contained in the lightcurve modelling; the remaining effort is mostly related to effectively using the parallel computing facilities we have access to at VUW.

This chapter demonstrates the capabilities of the new `mlens2` code by comparing models generated from published parameters with the published results, and results produced by the previous `MLENs` code.

4.1 OGLE-2005-BLG-390

OGLE-2005-BLG-390 was the third published microlensing event with a planetary companion (Beaulieu et al. [2006]), and contains one of the least massive planets that has been discovered to date. The event was announced by OGLE as a microlensing candidate on July 11, 2005. Following the alert, it was intensely monitored by OGLE, MOA, and PLANET. A deviation from the standard single-lens lightcurve was observed on August 9 and lasted for approximately a day. Subsequent modelling showed that this deviation was caused by a companion with a mass ratio of 7.6×10^{-5} at a distance of $1.61 r_E$ from the primary lens. It was determined by applying Bayesian analysis that this corresponded to a ~ 5.5 Earth mass planet orbiting at ~ 2.6 AU around a ~ 0.22 solar mass M-dwarf star.

The lightcurve shows an anomaly when the source crosses a small star shaped caustic induced by the companion planet. An interesting feature of the model is the fact that the source is much larger than this planetary caustic, and covers the entire caustic as it passes over. This results in the potentially sharp caustic crossing features being smoothed out into the small subsidiary peak observed at ~ 3593 days. The resulting lightcurve, shown in Figure 4.1, agrees well with the data and published model fit.

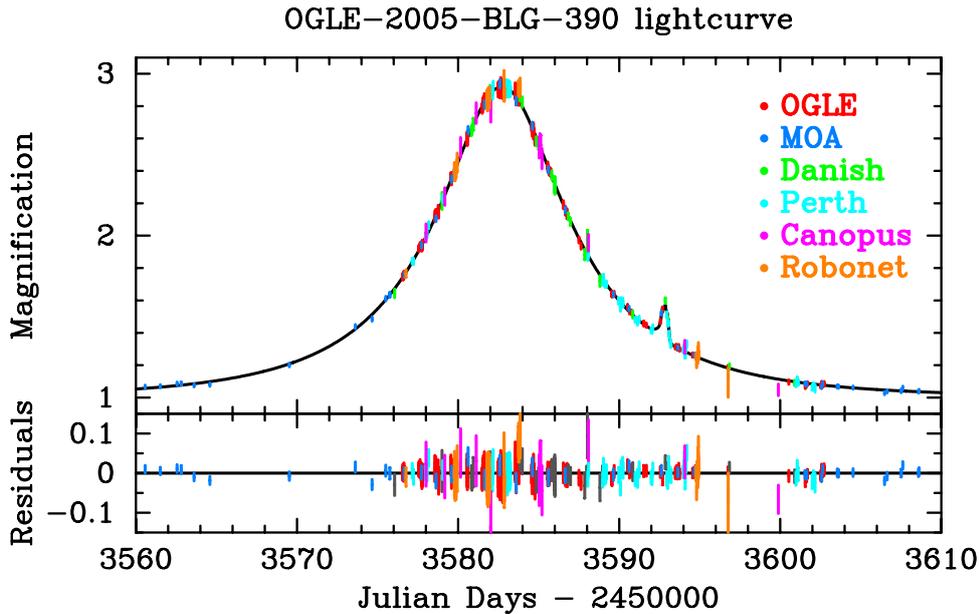


Figure 4.1: Lightcurve for the planetary event OGLE-2005-BLG-390, using parameters specified in Beaulieu et al. [2006]. The `mlens2` control file for this lightcurve is included as an example in Appendix E.

A χ^2 fit of 568 for 650 data points was calculated in 3 seconds using `mlens2`. This is close to the published fit value of 562. The original MLENs software required 194 seconds to calculate a χ^2 of 569 for the same data. This is particularly (and unfairly) slow; the full data set contains a large number of baseline observations which do not strongly constrain the model parameters. These baseline points span a range of 1500 days, which requires a significant area of the lens plane to be populated with rays, and so causes the MLENs ray shooting calculation to be particularly slow. A restricted fit was calculated using the data points between 3560 and 3610 days, producing a χ^2 of 264.5 in 20 seconds for MLENs, compared with 263.7 in 2 seconds using `mlens2`. Even with this restricted data set, `mlens2` is significantly more efficient than MLENs for this calculation. This can be attributed to the efficiency of the point-source approximation, which can be applied to the majority of the lightcurve (the calculation method regions were shown earlier in Figure 3.1).

A comparison of the model lightcurves generated using MLENs and `mlens2` is shown

in Figure 4.2.

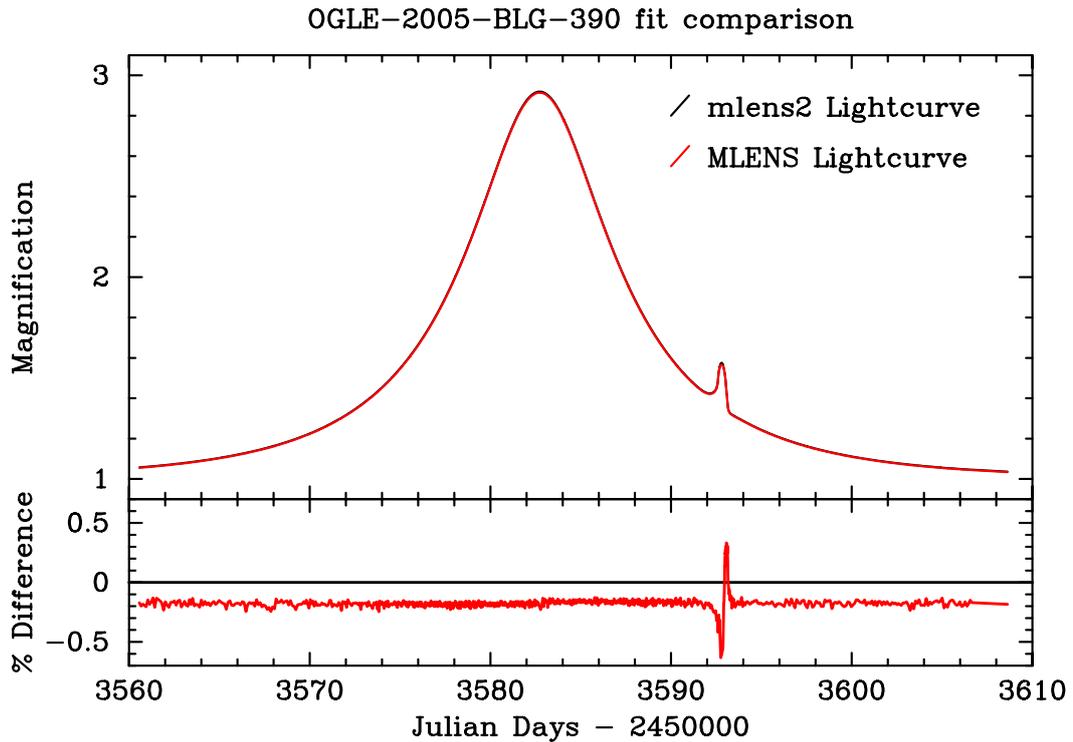


Figure 4.2: A comparison of the ray shooting lightcurve generated by MLENs, and the semi-analytical lightcurve generated by mlens2 for OGLE-2005-BLG-390. The two curves are in good agreement, but show elements of numerical error.

The ‘chi by eye’ fit is very good, but calculation of the percentage difference shows a systematic difference of 0.2% between the two models. This can be explained by a numerical error in the MLENs ray shooting code. The ray shooting magnification is calculated by taking the ratio of the total image flux to the unlensed source flux. The source flux remains constant (ignoring numerical errors) over the event, so any systematic error in its calculation will introduce a constant-percentage error such as the one seen in Figure 4.2. This type of error will have no impact on the calculated χ^2 , as each data set is scaled to the model before the fit value is calculated. This negates any constant multiplicative or additive errors in the model.

The ‘s’ shaped difference across the planetary perturbation is caused by a small change in the calculated position of the planetary mass due to different coordinate system definitions. This shifts the caustic, causing the source to cross it slightly earlier in the mlens2 calculation. The (relatively) large magnitude of the difference is caused by the rapid change in magnification when the source hits the caustic.

Ignoring these two effects, the two model lightcurves agree to within 0.05%, which is smaller than the required numerical accuracy of 0.1%. This indicates that both models are accurate representations of the correct theoretical lightcurve, or that

both are consistently wrong (which is unlikely, given that they employ completely different methods to obtain their respective lightcurves).

4.2 OGLE-2006-BLG-109

OGLE-2006-BLG-109 was the 5th published planetary microlensing event, and remains the most complex event to be successfully modelled to date. The details of the event are given by Gaudi et al. [2008]

“On 28 March 2006 ($HJD \sim 3822$), the OGLE Early Warning System (EWS) announced OGLE-2006-BLG-109 as a non-standard microlensing event possibly indicative of a planet. This immediately triggered followup observations by μ FUN and RoboNet, which gained intensity as the event approached high-magnification. On 5 April, the event underwent a deviation from the single-lens form indicative of a binary lens. Within 12 hours of this deviation, a preliminary model indicated a jovian-class planet, which was predicted to generate an additional peak on 8 April. The 8 April peak occurred as predicted, but in the meantime, there was an additional peak on 5/6 April, which turned out to be due to a second Jovian-class planet.”

The model parameters of the published model were taken from a subsequent, more detailed, paper (Bennett et al. [2010]), but required converting into the coordinate system used by `mlens2`.

Figure 4.3 shows the model light curve generated by `mlens2`, with a calculated χ^2 fit of 2512 for 1193 observations. This χ^2 value cannot be compared with the published value, as the final data sets used by Gaudi et al. [2008] were not available. The lightcurves and χ^2 fits presented here use a preliminary data set from the initial modelling in late 2007.

Without applying a direct comparison, Figure 4.3 qualitatively agrees well with the observational data and published model. Some small systematic errors introduced from the parameter conversions can be seen on the trailing edge of the lightcurve at ~ 3832 days, and in the initial caustic cusp crossing at ~ 3823 days. An optimised fit to correct these small errors will be possible once the parameter search algorithms are implemented in `mlens2`.

The combined effects of parallax and lens motion are visible in the source track, shown in Figure 4.4. Parallax curves the track, causing the source (moving right to left) to pass over the caustic cusp at ~ 3823 days. The lens motion lengthens

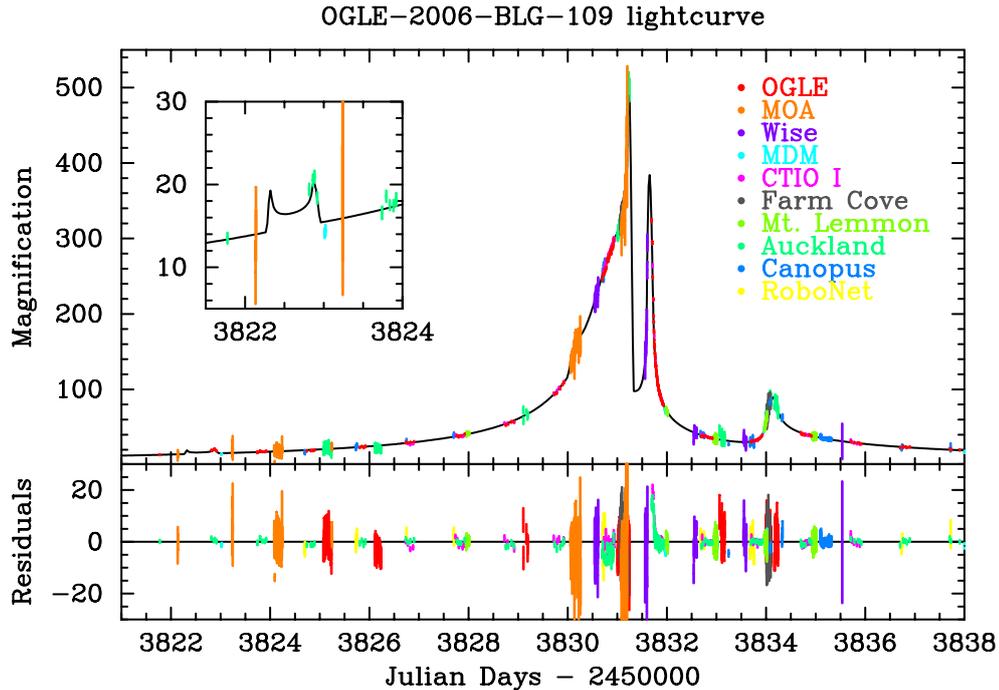


Figure 4.3: Lightcurve for the planetary event OGLE-2006-BLG-109 using model parameters from Bennett et al. [2010].

the caustic with time, so that the cusp is closer to the primary lens when the source crosses it. Without these two effects, the first caustic crossing cannot be modelled.

Figure 4.5 shows the effect of these orbital motion parameters on the lightcurve, by comparing the model lightcurves with and without parallax and lens motion (keeping the other parameters fixed). Without these effects, the first caustic crossing feature is completely absent, and the timing of the last feature is delayed. There is only a minimal effect on the lightcurve features near the peak of the event.

No attempt has been made to optimise the remaining parameters to produce the best ‘static’ model, i.e. without orbital motion. Instead, we use an earlier model produced by the VUW microlensing group in 2007 with the MLENs code. Figures 4.6 and 4.7 show the lightcurve and source track for the model, and its control file is included in Appendix E. The χ^2 fit for this model is 5744 for the 1193 observations. This is significantly worse than model that includes orbital motion.

This event highlights the relatively poor performance of the polygon method for calculating the source magnification. The χ^2 calculation for the model presented in Figure 4.3 took 88 seconds, which dropped to 28 seconds when the limb darkening calculation was disabled. The event is very high magnification, and so requires a large number of vertices to be calculated around the (relatively large) images.

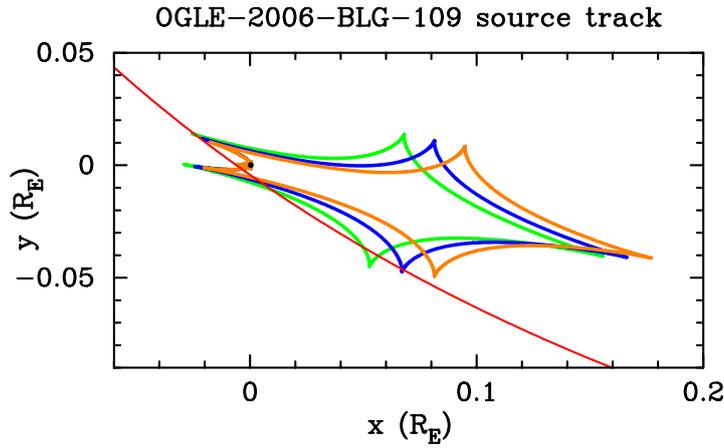


Figure 4.4: The source moves in a clockwise sense along its track (red), passing over the moving caustics. The caustics are shown at 3822 days (green), 3828 days (blue), and 3834 days (orange).

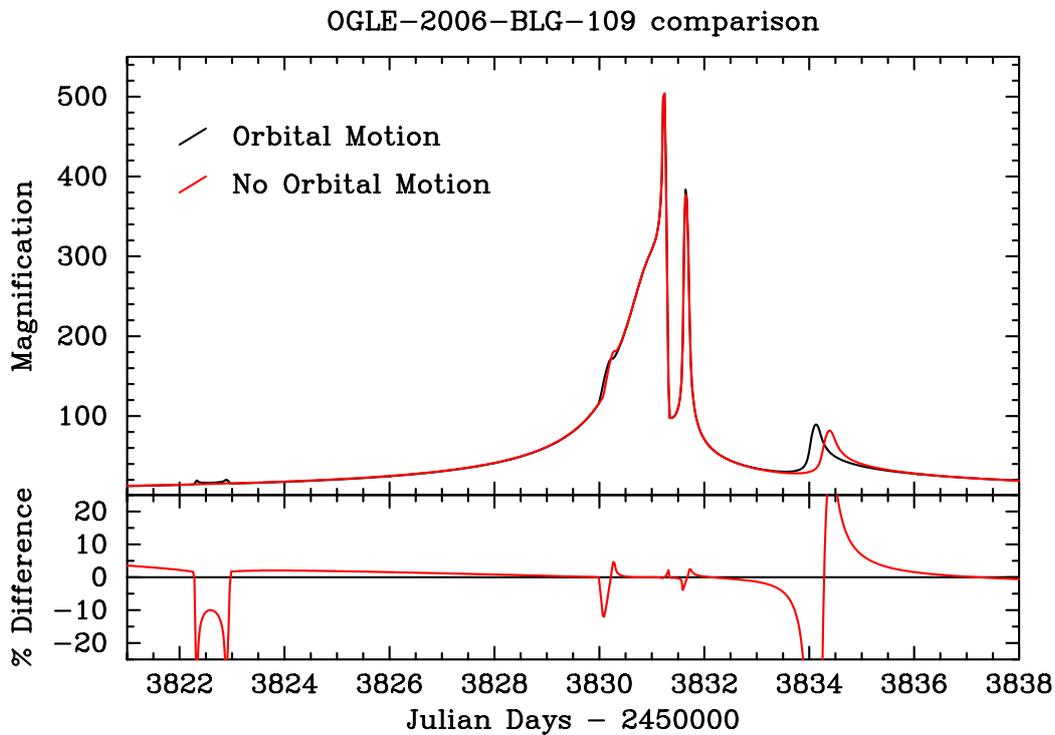


Figure 4.5: Lightcurves with and without parallax and lens motion (leaving the other parameters unchanged).

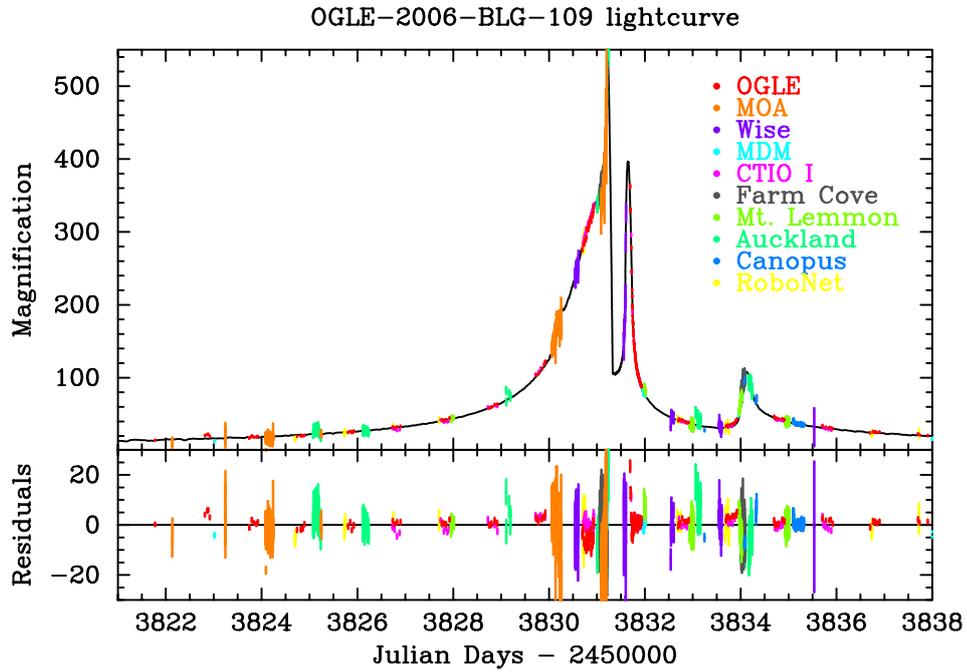


Figure 4.6: An unpublished alternative model for OGLE-2006-BLG-109 produced using MLENS. This model does not include parallax or lens motion effects.

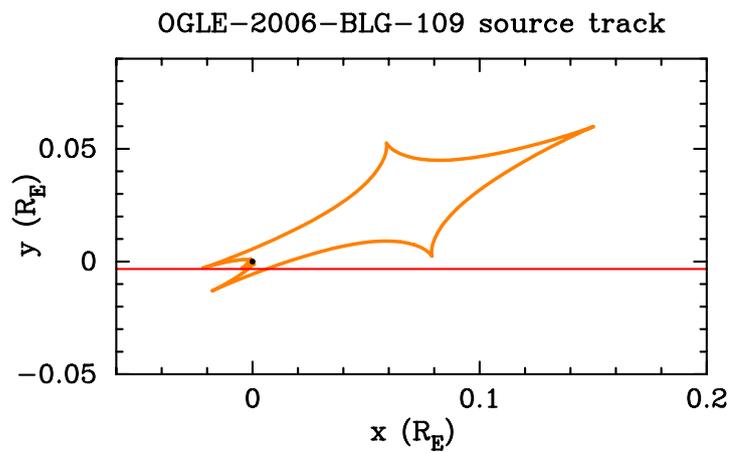


Figure 4.7: Source track for the VUW OGLE-2006-BLG-109 model. The source travels from right to left.

Each of these vertex calculations is slower than the corresponding binary lens vertex calculation, as twice the number of roots must be found for the three component lens than for two. As such, this event presents a worst-case measure of the performance of `mlens2`. These calculation times are an order of magnitude longer than what is required to efficiently search the parameter space for the best-fit model for an event. Further work is required to optimise or replace the polygon method to improve this performance.

Chapter 5

Discussion and Conclusions

This thesis describes the development from first principles of a software package (mlens2) that calculates model lightcurves and χ^2 fits for analysing gravitational microlensing events involving multiple point mass lenses and finite sources. Several semi-analytical procedures were employed in this work, in contrast to the inverse ray-tracing method used in the previous generation of modelling code at VUW (MLENS) which was developed as part of an earlier PhD programme. The theoretical development presented in this thesis also serves as an introduction to future researchers at VUW and elsewhere to the analytical and numerical techniques employed in modelling these events.

The semi-analytical methods used in mlens2 enable up to four lensing masses to be included, whereas inverse ray-tracing, in theory, can handle any number of point mass lenses. This is no real limitation, as the increasing dimensionality of the parameter space makes it very unlikely that even a four mass lens (e.g. star plus three planets) system, if discovered, could be usefully modelled.

The mlens2 code was found to be more efficient and robust than the previous MLENS code. In particular, the point-source and hexadecapole approximations were found to be extremely efficient in their regions of applicability. The results of this were seen in the lightcurve and fit calculations for the OGLE-2005-BLG-390 event; mlens2 completes these calculations an order of magnitude faster than MLENS. Furthermore, the introduction of parallax and lens motion into the model allowed all of the published model parameters for OGLE-2006-BLG-109 event to be used in the fit calculations. This yielded a significantly improved fit to the data than the previous static (without lens motion) model found using MLENS.

The major element of future work will be to complete the overlying parameter space search that was only partially completed. This code will interact with the parallel grid computing systems at VUW, running many candidate model calculations to

determine the best fitting model for an observed event. Another area of interest with potential future work is in investigating the hybrid techniques described in Section 2.5. One of these techniques may provide a more efficient, robust calculation to supercede the polygon method implemented in this thesis work. This would make it more feasible to model complex events such as OGLE-2006-BLG-109 in ‘real time’ as they are being observed in the future.

Appendix A

The Jacobian

The Jacobian matrix and its determinant (simply called the Jacobian for brevity) generalise the concept of the gradient operator ∇ , defining the tangent plane at a point to a vector-valued function. It is particularly useful for microlensing as the determinant of the Jacobian matrix can be related to the area change induced on an element of area by the lens equation, and thus give an approximation of magnification.

Consider an integral over an N-dimensional surface

$$\int_S f(x_1, \dots, x_N) dx_1 \dots dx_N \quad (\text{A.1})$$

If we want to change the coordinates that we integrate over, we apply a transformation to the coordinate basis of the function

$$\begin{aligned} x_1 &= g_1(u_1, u_2, \dots, u_N) \\ &\vdots \\ x_N &= g_N(u_1, u_2, \dots, u_N) . \end{aligned}$$

By differentiating the above, we find

$$\begin{aligned} dx_1 &= \frac{\partial g_1}{\partial u_1} du_1 + \dots + \frac{\partial g_1}{\partial u_N} du_N \\ &\vdots \\ dx_N &= \frac{\partial g_N}{\partial u_1} du_1 + \dots + \frac{\partial g_N}{\partial u_N} du_N . \end{aligned}$$

Substituting these into Equation A.1 we see that it is equivalent to the integral

$$\int_S f(g_1, \dots, g_N) J du_1 \dots du_N \quad (\text{A.2})$$

where J is the determinant of the matrix of partial derivatives

$$J = \left| \frac{\partial(u_1, \dots, u_N)}{\partial(x_1, \dots, x_N)} \right| = \begin{vmatrix} \frac{\delta u_1}{\delta x_1} & \dots & \frac{\delta u_1}{\delta x_N} \\ \vdots & \ddots & \vdots \\ \frac{\delta u_N}{\delta x_1} & \dots & \frac{\delta u_N}{\delta x_N} \end{vmatrix}$$

Considering Equations A.1 and A.2, we see that J is simply the ratio of the new unit area to old unit area

$$J = \frac{dx_1 \dots dx_N}{du_1 \dots du_N}.$$

Single-lens Jacobian calculation

As an example of the Jacobian calculation, we present the derivation of the single lens Jacobian

$$J = 1 - \frac{1}{|\mathbf{r}|^4}.$$

Following the method given in Section 1.4.2, the partial derivatives of the lens equation for the single lens are given by

$$\begin{aligned} \frac{\partial u}{\partial x} &= 1 + \frac{2x^2}{|\mathbf{r}|^4} - \frac{1}{|\mathbf{r}|^2} \\ \frac{\partial v}{\partial y} &= 1 + \frac{2y^2}{|\mathbf{r}|^4} - \frac{1}{|\mathbf{r}|^2} \\ \frac{\partial u}{\partial y} = \frac{\partial v}{\partial x} &= 2 \frac{xy}{|\mathbf{r}|^4} \end{aligned}$$

where $|\mathbf{r}|$ is the source-lens separation.

The Jacobian is found by expanding Equation 1.18 for these terms

$$\begin{aligned} J &= \left[1 + \frac{2x^2}{|\mathbf{r}|^4} - \frac{1}{|\mathbf{r}|^2} \right] \left[1 + \frac{2y^2}{|\mathbf{r}|^4} - \frac{1}{|\mathbf{r}|^2} \right] - 4 \frac{x^2 y^2}{|\mathbf{r}|^8} \\ &= 1 - \frac{2}{|\mathbf{r}|^2} + \frac{2(x^2 + y^2) + 1}{|\mathbf{r}|^4} - \frac{2(x^2 + y^2)}{|\mathbf{r}|^6}. \end{aligned}$$

Using the fact that $(x^2 + y^2) = |\mathbf{r}|^2$, this simplifies to the final result

$$J = 1 - \frac{1}{|\mathbf{r}|^4}. \quad (\text{A.3})$$

Complex Coordinates

In Section 2.2.2, we state that the Jacobian is unchanged by the move from vector coordinates to complex coordinates

$$J = \frac{\partial(u, v)}{\partial(x, y)} = \frac{\partial(w, \bar{w})}{\partial(z, \bar{z})}. \quad (\text{A.4})$$

This is straightforward to show by applying the chain rule to the relationship between the two coordinate systems. Rearranging Equation 2.2, we obtain

$$\begin{aligned} u &= \frac{w + \bar{w}}{2}; & v &= \frac{w - \bar{w}}{2i} \\ z &= x + iy; & \bar{z} &= x - iy \end{aligned}$$

with partial derivatives

$$\begin{aligned} \frac{\partial u}{\partial w} &= \frac{\partial u}{\partial \bar{w}} = \frac{1}{2} & \frac{\partial v}{\partial w} &= \frac{1}{2i} & \frac{\partial v}{\partial \bar{w}} &= -\frac{1}{2i} \\ \frac{\partial z}{\partial x} &= \frac{\partial \bar{z}}{\partial x} = 1 & \frac{\partial z}{\partial y} &= i & \frac{\partial \bar{z}}{\partial y} &= -i \end{aligned}$$

Applying the chain rule to the u, v coordinates of Equation A.4, we obtain

$$\begin{aligned} \frac{\partial(u, v)}{\partial(x, y)} &= \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \\ &= \left(\frac{\partial u}{\partial w} \frac{\partial w}{\partial x} + \frac{\partial u}{\partial \bar{w}} \frac{\partial \bar{w}}{\partial x} \right) \left(\frac{\partial v}{\partial w} \frac{\partial w}{\partial y} + \frac{\partial v}{\partial \bar{w}} \frac{\partial \bar{w}}{\partial y} \right) \\ &\quad - \left(\frac{\partial u}{\partial w} \frac{\partial w}{\partial y} + \frac{\partial u}{\partial \bar{w}} \frac{\partial \bar{w}}{\partial y} \right) \left(\frac{\partial v}{\partial w} \frac{\partial w}{\partial x} + \frac{\partial v}{\partial \bar{w}} \frac{\partial \bar{w}}{\partial x} \right) \\ &= \frac{1}{2i} \left(\frac{\partial \bar{w}}{\partial x} \frac{\partial w}{\partial y} - \frac{\partial w}{\partial x} \frac{\partial \bar{w}}{\partial y} \right). \end{aligned}$$

Repeating for x, y

$$\begin{aligned} \frac{\partial(u, v)}{\partial(x, y)} &= \left(\frac{\partial \bar{w}}{\partial z} \frac{\partial z}{\partial x} + \frac{\partial \bar{w}}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial x} \right) \left(\frac{\partial w}{\partial z} \frac{\partial z}{\partial y} + \frac{\partial w}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial y} \right) \\ &\quad - \left(\frac{\partial w}{\partial z} \frac{\partial z}{\partial x} + \frac{\partial w}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial x} \right) \left(\frac{\partial \bar{w}}{\partial z} \frac{\partial z}{\partial y} + \frac{\partial \bar{w}}{\partial \bar{z}} \frac{\partial \bar{z}}{\partial y} \right) \\ &= \frac{\partial w}{\partial z} \frac{\partial \bar{w}}{\partial \bar{z}} - \frac{\partial w}{\partial \bar{z}} \frac{\partial \bar{w}}{\partial z} \\ &= \frac{\partial(w, \bar{w})}{\partial(z, \bar{z})} \end{aligned}$$

as required.

Appendix B

Lens Polynomial Coefficients

Section 2.2.1 introduced the lens polynomial, and showed that a subset of its roots are solutions to the inverse lens equation.

While analytical solutions exist for the roots of polynomials with degree ≤ 4 , the general lens equation results in a polynomial of degree $N^2 + 1$. This corresponds to a degree 5, 10, or 17 polynomial for lenses with 2, 3 or 4 components respectively. This necessitates the use of a numerical algorithm. Many numerical techniques exist for finding the roots of an arbitrary polynomial, each with different characteristics for efficiency, convergence, and the types of roots that they can find.

The first numerical implementation used in `mlens2` was the Laguerre algorithm (Riley et al. [2006]). This had the advantage of being a straightforward algorithm, and allowed initial guesses to be inserted for the roots to speed up the computation, but suffered sporadic problems where not all roots would converge on a solution.

The next implementation used the Jenkins-Traub algorithm (Jenkins and Traub [1972, 1970]), which always converged on a solution, but had issues with successively larger numerical error in each due to the deflation process used.

The current implementation used in `mlens2` uses Jenkins-Traub to find the approximate root positions, which are then given as initial guesses to the Laguerre algorithm. This was found to overcome the convergence and numerical error problems associated with each algorithm, at the expense of computational efficiency. A future improvement could be to use Laguerre by default, falling back to Jenkins-Traub only when a solution does not converge.

These numerical algorithms operate on a vector of the polynomial coefficients, which are obtained by analytically rearranging Equation 2.5 into the form

$$\sum_i^{N^2+1} a_i z^i = 0 .$$

The expressions to calculate each a_i for $N \leq 4$ lens components are included below. Note that they are transcribed from the mlens2 source code (available by request); the reader is encouraged to verify the calculations themselves.

Two Lens Components

Starting with the standard definitions given in Section 2.2.1,

$$\begin{aligned} w &= (\text{complex}) \text{ Source position.} \\ \varepsilon_i &= \text{Mass fraction of } i\text{'th lens.} \\ z_i &= (\text{complex}) \text{ Position of the } i\text{'th lens.} \end{aligned}$$

We define some intermediate factors

$$\begin{aligned} A &= \bar{w} - \bar{z}_1 & B &= \bar{w} - \bar{z}_2 \\ z_s &= z_1 + z_2 & z_p &= z_1 z_2 \\ U_r &= \varepsilon_1 z_2 + \varepsilon_2 z_1 & U_e &= A \varepsilon_2 + B \varepsilon_1 \\ A_{rs} &= A z_s - 1 & B_{rs} &= B z_s - 1 \\ A_{rp} &= A z_p - U_r & B_{rp} &= B z_p - U_r \end{aligned}$$

The 6 coefficients are then given by

$$\begin{aligned} a_5 &= AB \\ a_4 &= -wAB - AB_{rs} - BA_{rs} - U_e \\ a_3 &= (sA + \varepsilon_1) B_{rs} + (wB + \varepsilon_2) A_{rs} + A_{rs} B_{rs} + AB_{rp} + BA_{rp} + z_s U_e \\ a_2 &= -w(AB_{rp} + BA_{rp} + A_{rs} B_{rs}) - \varepsilon_2 (A_{rp} + z_s A_{rs}) - \varepsilon_1 (B_{rp} + z_s B_{rs}) \\ &\quad - (A_{rs} B_{rp} + B_{rs} A_{rp}) - z_p U_e \\ a_1 &= w(A_{rp} B_{rs} + B_{rp} A_{rs}) + A_{rp} B_{rp} + \varepsilon_2 (z_p A_{rs} + z_s A_{rp}) + \varepsilon_1 (z_p B_{rs} + z_s B_{rp}) \\ a_0 &= -s A_{rp} B_{rp} - z_p (A_{rp} \varepsilon_2 + B_{rp} \varepsilon_1) \end{aligned}$$

Three Lens Components

Like the two lens case, we start with the same basic model parameters, and define a (much larger) set of intermediate factors

$$\begin{aligned}
 z_s &= z_1 + z_2 + z_3 \\
 z_q &= z_2z_3 + z_1z_3 + z_1z_2 \\
 z_p &= z_1z_2z_3 \\
 \varepsilon_q &= \varepsilon_1(z_2 + z_3) + \varepsilon_2(z_1 + z_3) + \varepsilon_3(z_1 + z_2) \\
 \varepsilon_p &= \varepsilon_1(z_2z_3) + \varepsilon_2(z_1z_3) + \varepsilon_3(z_1z_2)
 \end{aligned}$$

$$\begin{array}{lll}
 A = \bar{w} - \bar{z}_1 & B = \bar{w} - \bar{z}_2 & C = \bar{w} - \bar{z}_3 \\
 A_s = Az_s - 1 & A_q = Az_q - \varepsilon_q & A_p = Az_p - \varepsilon_p \\
 B_s = Bz_s - 1 & B_q = Bz_q - \varepsilon_q & B_p = Bz_p - \varepsilon_p \\
 C_s = Cz_s - 1 & C_q = Cz_q - \varepsilon_q & C_p = Cz_p - \varepsilon_p
 \end{array}$$

$$\begin{array}{ll}
 t_1 = ABC_s + AB_sC + A_sBC & t_2 = ABC_q + AB_qC + A_qBC \\
 t_3 = AB_sC_s + A_sBC_s + A_sB_sC & t_4 = ABC_p + AB_pC + A_pBC \\
 t_5 = A_sB_sC_q + A_sB_qC_s + A_qB_sC_s & t_6 = A_sB_sC_p + A_sB_pC_s + A_pB_sC_s \\
 t_7 = AB_qC_q + A_qBC_q + A_qB_qC & t_8 = A_sB_qC_q + A_qB_sC_q + A_qB_qC_s \\
 t_9 = AB_pC_p + A_pBC_p + A_pB_pC & t_{10} = A_qB_qC_p + A_qB_pC_q + A_pB_qC_q \\
 t_{11} = AB_pC_p + A_pBC_p + A_pB_pC & t_{12} = A_sB_pC_p + A_pB_sC_p + A_pB_pC_s
 \end{array}$$

$$\begin{array}{ll}
 o_1 = \varepsilon_1BC + \varepsilon_2AC + \varepsilon_3AB & o_2 = \varepsilon_1B_sC_s + \varepsilon_2A_sC_s + \varepsilon_3A_sB_s \\
 o_3 = \varepsilon_1B_qC_q + \varepsilon_2A_qC_q + \varepsilon_3A_qB_q & o_4 = \varepsilon_1B_pC_p + \varepsilon_2A_pC_p + \varepsilon_3A_pB_p
 \end{array}$$

$$\begin{aligned}
m_1 &= \varepsilon_1 (BC_q + B_qC) + \varepsilon_2 (AC_q + A_qC) + \varepsilon_3 (AB_q + A_qB) \\
m_2 &= \varepsilon_1 (BC_p + B_pC) + \varepsilon_2 (AC_p + A_pC) + \varepsilon_3 (AB_p + A_pB) \\
m_3 &= \varepsilon_1 (BC_s + B_sC) + \varepsilon_2 (AC_s + A_sC) + \varepsilon_3 (AB_s + A_sB) \\
m_4 &= \varepsilon_1 (B_sC_q + B_qC_s) + \varepsilon_2 (A_sC_q + A_qC_s) + \varepsilon_3 (A_sB_q + A_qB_s) \\
m_5 &= \varepsilon_1 (B_sC_p + B_pC_s) + \varepsilon_2 (A_sC_p + A_pC_s) + \varepsilon_3 (A_sB_p + A_pB_s) \\
m_6 &= \varepsilon_1 (B_qC_p + B_pC_q) + \varepsilon_2 (A_qC_p + A_pC_q) + \varepsilon_3 (A_qB_p + A_pB_q)
\end{aligned}$$

$$\begin{aligned}
n_1 &= AB_sC_q + AB_qC_s + A_sBC_q + A_qBC_s + A_sB_qC + A_qB_sC \\
n_2 &= AB_sC_p + AB_pC_s + A_sBC_p + A_pBC_s + A_sB_pC + A_pB_sC \\
n_3 &= AB_qC_p + AB_pC_q + A_qBC_p + A_pBC_q + A_qB_pC + A_pB_qC \\
n_4 &= A_sB_qC_p + A_sB_pC_q + A_qB_sC_p + A_pB_sC_q + A_qB_pC_s + A_pB_qC_s
\end{aligned}$$

The 11 polynomial coefficients are then

$$\begin{aligned}
c_{10} &= ABC \\
c_9 &= -sABC - t_1 - o_1 \\
c_8 &= st_1 + t_2 + t_3 + z_s o_1 + m_3 \\
c_7 &= -s(t_2 + t_3) - A_s B_s C_s - z_q o_1 - z_s m_3 - t_4 - n_1 - o_2 - m_1 \\
c_6 &= s(A_s B_s C_s + t_4 + n_1) + z_s(o_2 + m_1) + z_p o_1 + z_q m_3 + t_5 + t_7 \\
&\quad + n_2 + m_4 + m_2 \\
c_5 &= -s(t_5 + t_7 + n_2) - z_p m_3 - z_q(o_2 + m_1) \\
&\quad - z_s(m_4 + m_2) - t_6 - t_8 - n_3 - o_3 - m_5 \\
c_4 &= A_q B_q C_q + s(t_6 + t_8 + n_3) + z_p(o_2 + m_1) + z_q(m_4 + m_2) \\
&\quad + z_s(o_3 + m_5) + n_4 + t_9 + m_6 \\
c_3 &= -s(A_q B_q C_q + n_4 + t_{11}) - z_p(m_2 + m_4) - z_q(o_3 + m_5) \\
&\quad - z_s m_6 - t_{10} - t_{12} - o_4 \\
c_2 &= s(t_{10} + t_{12}) + z_p(o_3 + m_5) + z_q m_6 + z_s o_4 + t_{13} \\
c_1 &= -st_{13} - A_p B_p C_p - z_p m_6 - z_q o_4 \\
c_0 &= sA_p B_p C_p + z_p o_4
\end{aligned}$$

We note that these coefficients had previously been derived by Rhie [2002], but our derivation was completed independently as an extension of the two-component lens coefficients, for which there does not appear to be any published results in the literature. The algebraic manipulation of these terms into the form presented here was aided by Michael Miller.

Four Lens Components

A similar procedure is applied to calculate the 18 coefficients for the 4 component lens. The list of (approximately 130) intermediate factors is far too long to transcribe here, but the source code to evaluate the final coefficients is available by request. The algebraic manipulation and initial code implementation for this case was completed by Michael Miller.

Appendix C

Critical Curve Polynomial Coefficients

Section 2.2.2 showed that, like the images, the critical curves of a given lensing configuration can be found by solving the roots of a polynomial. The caveats from Appendix B with respect to solving for these roots apply, but the situation is more tractable due to the simpler form of the critical curve polynomial.

The critical curves are found by recasting Equation 2.6 into a set of coefficient equations of the form

$$\sum_{i=0}^{2N} a_i(\varphi) z^i = 0 . \quad (\text{C.1})$$

These a_i are then given to the numerical root finder to find the critical curve points. This is repeated for a range of $\varphi \in (0, 2\pi]$ in order to define the entire curve.

The coefficient equations for $N = 2, 3, 4$ lenses are given below, using the same lens parameters as Appendix B.

ε_i = Mass fraction of i 'th lens.

z_i = (complex) Position of the i 'th lens.

The critical curve for a single lens is simply a circle of radius $1/r_E$, which does not require any special calculations.

Two Lens Components

The 5 coefficients for the two-component lens are

$$\begin{aligned}
 a_4 &= -e^{-i\varphi} \\
 a_3 &= 2(z_1 + z_2) e^{-i\varphi} \\
 a_2 &= 1 - (z_1^2 + z_2^2 + 4z_1z_2) e^{-i\varphi} \\
 a_1 &= -2(\varepsilon_1z_2 + \varepsilon_2z_1) + 2z_1z_2(z_1 + z_2) e^{-i\varphi} \\
 a_0 &= \varepsilon_1z_2^2 + \varepsilon_2z_1^2 - z_1^2z_2^2 e^{-i\varphi}
 \end{aligned}$$

Three Lens Components

Similarly, the 7 coefficients for the three-component lens are

$$\begin{aligned}
 a_6 &= -e^{-i\varphi} \\
 a_5 &= 2(z_1 + z_2 + z_3) e^{-i\varphi} \\
 a_4 &= 1 - (z_1^2 + z_2^2 + z_3^2 + 4(z_1z_3 + z_1z_2 + z_2z_3)) e^{-i\varphi} \\
 a_3 &= -2[\varepsilon_1(z_2 + z_3) + \varepsilon_2(z_1 + z_3) + \varepsilon_3(z_1 + z_2)] \\
 &\quad + 2[z_1(z_2^2 + z_3^2) + z_2(z_1^2 + z_3^2) + z_3(z_1^2 + z_2^2)] e^{-i\varphi} \\
 &\quad + 8z_1z_2z_3 e^{-i\varphi} \\
 a_2 &= 4(\varepsilon_1z_2z_3 + \varepsilon_2z_1z_3 + \varepsilon_3z_1z_2) \\
 &\quad + \varepsilon_1(z_2^2 + z_3^2) + \varepsilon_2(z_1^2 + z_3^2) + \varepsilon_3(z_1^2 + z_2^2) \\
 &\quad - [z_1z_2(z_1z_2 + 4z_3^2) + z_2z_3(z_2z_3 + 4z_1^2) + z_1z_3(z_1z_3 + 4z_2^2)] e^{-i\varphi} \\
 a_1 &= -2[\varepsilon_1(z_2z_3^2 + z_2^2z_3) + \varepsilon_2(z_1z_3^2 + z_1^2z_3) + \varepsilon_3(z_1z_2^2 + z_1^2z_2)] \\
 &\quad + 2(z_1z_2^2z_3^2 + z_1^2z_2z_3^2 + z_1^2z_2^2z_3) e^{-i\varphi} \\
 a_0 &= \varepsilon_1z_2^2z_3^2 + \varepsilon_2z_1^2z_3^2 + \varepsilon_3z_1^2z_2^2 - z_1^2z_2^2z_3^2 e^{-i\varphi}
 \end{aligned}$$

Four Lens Components

Finally, the 9 coefficients for the four-component lens are

$$a_8 = -e^{-i\varphi}$$

$$a_7 = 2(z_1 + z_2 + z_3 + z_4) e^{-i\varphi}$$

$$a_6 = 1 - \left((z_1^2 + z_2^2 + z_3^2 + z_4^2) + 4(z_1z_2 + z_1z_3 + z_2z_3 + z_1z_4 + z_2z_4 + z_3z_4) \right) e^{-i\varphi}$$

$$a_5 = -2 \left[\varepsilon_1(z_2 + z_3 + z_4) + \varepsilon_2(z_1 + z_3 + z_4) + \varepsilon_3(z_1 + z_2 + z_4) + \varepsilon_4(z_1 + z_2 + z_3) \right] \\ + 2 \left[(z_2 + z_3 + z_4)z_1^2 + (z_1 + z_3 + z_4)z_2^2 + (z_1 + z_2 + z_4)z_3^2 + (z_1 + z_2 + z_3)z_4^2 \right] e^{-i\varphi} \\ + 8(z_1z_2z_3 + z_1z_2z_4 + z_1z_3z_4 + z_2z_3z_4) e^{-i\varphi}$$

$$a_4 = \varepsilon_1 \left[z_2^2 + z_3^2 + z_4^2 + 4(z_2z_3 + z_2z_4 + z_3z_4) \right] + \varepsilon_2 \left[z_1^2 + z_3^2 + z_4^2 + 4(z_1z_3 + z_1z_4 + z_3z_4) \right] \\ + \varepsilon_3 \left[z_1^2 + z_2^2 + z_4^2 + 4(z_1z_2 + z_1z_4 + z_2z_4) \right] + \varepsilon_4 \left[z_1^2 + z_2^2 + z_3^2 + 4(z_1z_2 + z_1z_3 + z_2z_3) \right] \\ - (z_1^2z_2^2 + z_1^2z_3^2 + z_1^2z_4^2 + z_2^2z_3^2 + z_2^2z_4^2 + z_3^2z_4^2) e^{-i\varphi} \\ - 4 \left[(z_2z_3 + z_2z_4z_1 + z_3z_4)z_1^2 + (z_1z_3 + z_1z_4 + z_3z_4)z_2^2 \right] e^{-i\varphi} \\ - 4 \left[(z_1z_2 + z_1z_4 + z_2z_4)z_3^2 + (z_1z_2 + z_1z_3 + z_2z_3)z_4^2 \right] e^{-i\varphi} \\ - 16z_1z_2z_3z_4 e^{-i\varphi}$$

$$a_3 = -2\varepsilon_1 \left[z_2^2(z_3 + z_4) + z_3^2(z_2 + z_4) + z_4^2(z_2 + z_3) + 4z_2z_3z_4 \right] \\ - 2\varepsilon_2 \left[z_1^2(z_3 + z_4) + z_3^2(z_1 + z_4) + z_4^2(z_1 + z_3) + 4z_1z_3z_4 \right] \\ - 2\varepsilon_3 \left[z_1^2(z_2 + z_4) + z_2^2(z_1 + z_4) + z_4^2(z_1 + z_2) + 4z_1z_2z_4 \right] \\ - 2\varepsilon_4 \left[z_1^2(z_2 + z_3) + z_2^2(z_1 + z_3) + z_3^2(z_1 + z_2) + 4z_1z_2z_3 \right] \\ + 2 \left[z_1(z_2^2z_3^2 + z_2^2z_4^2 + z_3^2z_4^2) + z_2(z_1^2z_3^2 + z_1^2z_4^2 + z_3^2z_4^2) \right] e^{-i\varphi} \\ + 2 \left[z_3(z_1^2z_2^2 + z_1^2z_4^2 + z_2^2z_4^2) + z_4(z_1^2z_2^2 + z_1^2z_3^2 + z_2^2z_3^2) \right] e^{-i\varphi} \\ + 8(z_1^2z_2z_3z_4 + z_1z_2^2z_3z_4 + z_1z_2z_3^2z_4 + z_1z_2z_3z_4^2) e^{-i\varphi}$$

$$a_2 = \varepsilon_1 \left[z_2^2z_3^2 + z_2^2z_4^2 + z_3^2z_4^2 + 4(z_2^2z_3z_4 + z_2z_3^2z_4 + z_2z_3z_4^2) \right] \\ + \varepsilon_2 \left[z_1^2z_3^2 + z_1^2z_4^2 + z_3^2z_4^2 + 4(z_1^2z_3z_4 + z_1z_3^2z_4 + z_1z_3z_4^2) \right] \\ + \varepsilon_3 \left[z_1^2z_2^2 + z_1^2z_4^2 + z_2^2z_4^2 + 4(z_1^2z_2z_4 + z_1z_2^2z_4 + z_1z_2z_4^2) \right] \\ + \varepsilon_4 \left[z_1^2z_2^2 + z_1^2z_3^2 + z_2^2z_3^2 + 4(z_1^2z_2z_3 + z_1z_2^2z_3 + z_1z_2z_3^2) \right] \\ - (z_1^2z_2^2z_3^2 + z_1^2z_2^2z_4^2 + z_2^2z_3^2z_4^2 + z_1^2z_3^2z_4^2) e^{-i\varphi} \\ - 4(z_1^2z_2^2z_3z_4 + z_1^2z_2z_3^2z_4 + z_1z_2^2z_3^2z_4 + z_1^2z_2z_3z_4^2 + z_1z_2^2z_3z_4^2 + z_1z_2z_3^2z_4^2) e^{-i\varphi}$$

$$a_1 = -2 \left[\varepsilon_1(z_2^2z_3^2z_4 + z_2^2z_3z_4^2 + z_2z_3^2z_4^2) + \varepsilon_2(z_1^2z_3^2z_4 + z_1^2z_3z_4^2 + z_1z_3^2z_4^2) \right] \\ - 2 \left[\varepsilon_3(z_1^2z_2^2z_4 + z_1^2z_2z_4^2 + z_1z_2^2z_4^2) + \varepsilon_4(z_1^2z_2^2z_3 + z_1^2z_2z_3^2 + z_1z_2^2z_3^2) \right] \\ + 2(z_1^2z_2^2z_3^2z_4 + z_1^2z_2^2z_3z_4^2 + z_1^2z_2z_3^2z_4^2 + z_1z_2^2z_3^2z_4^2) e^{-i\varphi}$$

$$a_0 = \varepsilon_1 z_2^2 z_3^2 z_4^2 + \varepsilon_2 z_1^2 z_3^2 z_4^2 + \varepsilon_3 z_1^2 z_2^2 z_4^2 + \varepsilon_4 z_1^2 z_2^2 z_3^2 - z_1^2 z_2^2 z_3^2 z_4^2 e^{-i\varphi}$$

Appendix D

mljob Program Structure

The mljob application implements the microlensing model described in this thesis. It is operated from the command line by passing arguments, one of which specifies a control file which contains the details of the model to evaluate.

The command line arguments can include

- ctl** Path to a control file specifying the model parameters.
Some example control files are presented in Appendix E.
- lightcurve** Generate a lightcurve for the given model parameters.
- evaluate** Evaluate the χ^2 for the given model parameters.
- evaluategrid** Evaluates the χ^2 for a lattice of models evaluated within the specified range of model parameters.
- interactive** Run in an interactive graphical mode to visualise event geometry and other figures.
- v, -vv, -vvv** Print debug information with increasing verbosity.

The main calculation routines are contained within the GlobalMemory and SourceTarget classes, which are designed to be easily understandable and accessible to future users who may want to understand and modify the code. GlobalMemory defines the ‘global’ model information and actions, such as the lens components’ position and motion, loading and saving data files, etc. It also contains a collection of the observational data, and a list of ‘source targets’. The SourceTarget class encapsulates a single magnification calculation, including the time and source position, and the magnification calculation itself.

Some insight into the structure of these classes are given via their public interfaces, included below.

```

SourceTarget(Point position, double time, SourceType type, double radius=0, double limbCoefficient=0, \
             int limbCount=1, LimbSpacingType limbType=LimbSpacingRadius);

void setObservationKey(std::pair<int,int> observationKey);
std::pair<int,int> getObservationKey() const;

SourceType getCalculationType();
void calculateAmplification();
double getAmplification() const throw (InvalidDataException);
double getChiSquaredContribution() const;
Point getPosition() const;
double getTime() const;

void PGDrawSource();
void PGDrawImages();
void PGDrawLimbImage(double radius);
std::vector<Curve> getImagePolygons();
void getPointSourceImages(std::vector<Point> &images);
void getVertexImages(std::vector<Point> &images, double angle);
bool isValidMagnification();

```

Figure D.1: Public interface for the SourceTarget class.

```

// Initialization and observations
void SetModel(std::string ctrlFile);
void loadDataFiles();
DataPoint* getObservation(std::pair<int,int> observationKey);
void scaleDataToModel();

// Source manipulation
void placeSourcesOnObservations();
void placeSourcesEvenly(int number);
int placeSource(const SourceTarget &source);
void removeSources();
SourceTarget* getSource(int sourceKey);

void calculateSourceAmplifications();
double calculateChiSquared();
void saveLightcurve(bool logScale, bool usingObservations);
void getCausticPoints(double t, double dx, std::vector<Point> &points);
void getCriticalCurves(double t, double dx, std::vector<Curve> &criticalPolys, \
    std::vector<Curve> &causticPolys);

// Lens properties
double getLensDistance(int lensID);
void setLensDistance(int lensID, double value);
double getLensAngle(int lensID);
void setLensAngle(int lensID, const double value);
double getLensMassFraction(int lensID);
double getLensMass(int lensID);
void setLensMass(int lensID, const double value);
Point getLensPositionCOM(int lensID, double t);
Point getLensCOM();
Point getLensCOM(double t);
int getLensCount();

// Calculation routines
double JacobianDeterminantAtPoint(Point p, double t);
double JacobianDeterminantGradientAtPoint(Point p, double t);
Point MagnificationCurvatureAtPoint(Point p, double t);
Point InvRayTrace(Point pos, double t);

// Point-magnification calculations
double getPointMagnification(Point p, double t);
double getPointMagnificationGradient(Point p, double t);
double getPointMagnificationCurvature(Point p, double t);

// Image coefficients
void getOneLensPolynomialCoefficients(Point sp, double t, std::complex<double> c[]);
void getTwoLensPolynomialCoefficients(Point sp, double t, std::complex<double> c[]);
void getThreeLensPolynomialCoefficients(Point sp, double t, std::complex<double> c[]);
void getFourLensPolynomialCoefficients(Point sp, double t, std::complex<double> c[]);

// Critical curve coefficients
void getOneLensCriticalCurvePolynomial(double phi, double t, std::complex<double> coeff[]);
void getTwoLensCriticalCurvePolynomial(double phi, double t, std::complex<double> coeff[]);
void getThreeLensCriticalCurvePolynomial(double phi, double t, std::complex<double> coeff[]);
void getFourLensCriticalCurvePolynomial(double phi, double t, std::complex<double> coeff[]);

// Source position and Parallax calculations
Point getSourcePosition(double t);
Point projectTowardsLensPlane(Point p, double phi, double chi);
double getEccentricAnomaly(double t, double tp);
double getEccentricChange(double t, double tp);
Point getEarthPosition(double t, double tp);
Point getEarthVelocity(double t, double tp);
Point getParallaxOffset(double t);

```

Figure D.2: Public interface for the GlobalMemory class.

Appendix E

Control Files

The control file specifies the model parameters and other factors that are used by `mlens2` to calculate a model lightcurve or χ^2 fit. It consists of a plain text file (generally with the file extension `.ctl`, but this is a convention, not a requirement) beginning with the line `#CTLFILEV3`, followed by a list of parameters and values grouped into sections, one parameter per line. There are 51 recognised parameters, split into 6 categories

#CTLFILEV3	Denotes the start of the control file, and contains non-model parameters such as event name, celestial coordinates of the lens (for parallax calculations), and numerical resolution parameters.
PARAMETERS	Contains the model parameters describing the event.
PASSBANDS	Defines the linear limb darkening parameters for each wavelength range in the observational data.
OBSERVATIONS	Defines a list of the observational data files and the passband used for each.
INTERACTIVE	Defines parameters specific to the interactive mode.
LIGHTCURVE	Defines parameters specific to lightcurve generation, including whether to display the data in magnitude or magnification units, and the number of uniform steps used to define the continuous model lightcurve.

Lens masses are specified as a mass ratio relative to the primary lens (which by definition has a mass of 1). These mass ratios are internally converted into a mass fraction. All separations and lengths are given in units of the total-mass Einstein ring, different to the original `MLEN`S software which measures these distances relative to the primary-mass Einstein ring.

Some example control files, used to generate figures in Chapter 4, are shown.

```

#CTLFILV3
EventName          OGLE-2005-BLG-390
MinVertexCount     50
MaxVertexSeparation 0.002
Limbrings          5
PARAMETERS
ImpactAngle        0
ImpactParameter    -0.359
SourceRadius       0.02556663645
Companion1Angle    337.9071683
Companion1Distance 1.610
Companion1Mass     0.000076
PeakTime           3582.731
CrossingTime       11.03
StartTime          3560
EndTime           3610
LIGHTCURVE
UseObservations    true
UniformSteps       1000
PASSBANDS
1 0.538
2 0.626
OBSERVATIONS
phot-ogle390-dand  1
phot-ogle390-moa2r 2
phot-ogle390-oglem 1
phot-ogle390-perth 1
phot-ogle390-tas   1
phot-ogle390-fham  2

```

Figure E.1: Control file used to generate Figure 4.1 for OGLE-2005-BLG-390.

```

0
EventName          OGLE390
LensCount           2
ChipRadius         1.2
ITresolution       0.01
IPresolution       0
PlaneRadius        -99999
RequiredMag        2.5
PeakBase           0.5
ExpectedRays       25          0.001
ExtendTrack        true
LimbDarken         true
PARAMETERS
SecondaryA         337.9071683
SecondaryR         1.610122360
SecondaryM         0.000076
ImpactRadius       -0.359027284
SourceRadius       0.02556857951
TrackPassing       3582.731
TrackCrossing      11.03
TrackStart         3560          0
TrackEnd           3610          0
PASSBANDS
1 0.538
2 0.626
OBSERVATIONS
phot-ogle390-dand  1
phot-ogle390-moa2r 2
phot-ogle390-oglem 1
phot-ogle390-perth 1
phot-ogle390-tas   1
phot-ogle390-fham  2

```

Figure E.2: Original MLENs control file used to generate the OGLE-2005-BLG-390 comparison in Figure 4.2. This control file uses the original MLENs syntax, and will not work with `mlens2`.

```

#CTLFILV3
EventName          06ob-109
MinVertexCount     50
LimbRings          5
LimbRingsSpaceRadius true
MaxVertexSeparation 0.001
RightAscension     17 52 34.51
Declination        -30 05 16.0
PerihelionTime     3737.12
EquinoxTime        3815.27
PARAMETERS
ImpactParameter    0.003479
ImpactAngle        144.5555328
SourceRadius       0.0003120188531
Companion1Angle    180.0112307
Companion1Distance 0.6263753290
Companion1Mass     0.001358729084
Companion2Angle    -13.49888565
Companion2Distance 1.04185
Companion2Orbit    0.00169 0.00181
Companion2OrbitPeriod 4901.960784
Companion2Mass     0.0005061020381
PeakTime           3831.0197
CrossingTime       127.300
StartTime          3820
EndTime            3840
ParallaxScale      0.3620
ParallaxAngle      156.3945597
INTERACTIVE
Viewport           -0.06 0.18 -0.1 0.12
Steps 500
LIGHTCURVE
UseObservations    true
LogScale           false
UniformSteps       1000
UniformPassband    3
PASSBANDS
# V band
1 0.6630
# R band
2 0.5887
# I band
3 0.5090
# H band
4 0.3292
# Farm Cove, unfiltered
5 0.5413
# Auck, unfiltered
6 0.5490
OBSERVATIONS
06ob-109-auck     6
06ob-109-fcov    5
06ob-109-liv     2
06ob-109-mdm     3
06ob-109-mfun    3
06ob-109-moa     3
06ob-109-mtlem   3
06ob-109-ogle    3
06ob-109-tas     3
06ob-109-wise    2

```

Figure E.3: Control file used to generate Figure 4.3 for OGLE-2006-BLG-109.

```

0
EventName          06ob-109
LensCount          3
AnomalyStart       3820
AnomalyEnd         3836
ChipRadius         0.25
ITresolution       0.0001
IPresolution       0.0001
LensPlaneU         -99999
LensPlaneV         -99999
ViewCenter         0 0
ViewRadius         0.03
RequiredMag        400
PeakBase           0.4
ExpectedRays       100
ExtendTrack        true
ChiSquare          8634.324206
PARAMETERS
ImpactRadius       -0.003341
SourceRadius       0.000303
LensA              326.69
LensS              0.632
LensQ              0.00124
LensA              158.24
LensS              1.0381
LensQ              0.00041
TrackPassing       3830.98696
TrackCrossing      134.1870411
TrackStart         3800
TrackEnd           3840
PASSBANDS
1 0.6630
2 0.5887
3 0.5090
4 0.3292
5 0.5413
6 0.5490
OBSERVATIONS
06ob-109-auck     6
06ob-109-fcov     5
06ob-109-liv      2
06ob-109-mdm      3
06ob-109-mfun     3
06ob-109-moa      3
06ob-109-mtlem    3
06ob-109-ogle     3
06ob-109-tas      3
06ob-109-wise     2

```

Figure E.4: Original MLENS control file used to generate the OGLE-2006-BLG-109 model and calculations in Figure 4.6. This control file uses the original MLENS syntax, and will not work with mlens2.

Bibliography

Moa microlensing alert page. URL <https://it019909.massey.ac.nz/moa/alert/alert2010.html>.

C. Alcock, R. A. Allsman, D. Alves, T. S. Axelrod, D. P. Bennett, K. H. Cook, K. C. Freeman, K. Griest, J. Guern, M. J. Lehner, S. L. Marshall, B. A. Peterson, M. R. Pratt, P. J. Quinn, A. W. Rodgers, C. W. Stubbs, and W. Sutherland. First Observation of Parallax in a Gravitational Microlensing Event. *ApJ*, 454: L125+, December 1995. doi: 10.1086/309783.

C. Alcock, R. A. Allsman, D. Alves, T. S. Axelrod, A. C. Becker, D. P. Bennett, K. H. Cook, K. C. Freeman, K. Griest, J. Guern, M. J. Lehner, S. L. Marshall, B. A. Peterson, M. R. Pratt, P. J. Quinn, A. W. Rodgers, C. W. Stubbs, W. Sutherland, D. L. Welch, and The MACHO Collaboration. The MACHO Project Large Magellanic Cloud Microlensing Results from the First Two Years and the Nature of the Galactic Dark Halo. *ApJ*, 486:697–+, September 1997. doi: 10.1086/304535.

J.-P. Beaulieu, D. P. Bennett, P. Fouqué, A. Williams, M. Dominik, U. G. Jørgensen, D. Kubas, A. Cassan, C. Coutures, J. Greenhill, K. Hill, J. Menzies, P. D. Sackett, M. Albrow, S. Brilliant, J. A. R. Caldwell, J. J. Calitz, K. H. Cook, E. Corrales, M. Desort, S. Dieters, D. Dominis, J. Donatowicz, M. Hoffman, S. Kane, J.-B. Marquette, R. Martin, P. Meintjes, K. Pollard, K. Sahu, C. Vinter, J. Wambsganss, K. Woller, K. Horne, I. Steele, D. M. Bramich, M. Burgdorf, C. Snodgrass, M. Bode, A. Udalski, M. K. Szymański, M. Kubiak, T. Więckowski, G. Pietrzyński, I. Soszyński, O. Szewczyk, L. Wyrzykowski, B. Paczyński, F. Abe, I. A. Bond, T. R. Britton, A. C. Gilmore, J. B. Hearnshaw, Y. Itow, K. Kamiya, P. M. Kilmartin, A. V. Korpela, K. Masuda, Y. Matsubara, M. Motomura, Y. Muraki, S. Nakamura, C. Okada, K. Ohnishi, N. J. Rattenbury, T. Sako, S. Sato, M. Sasaki, T. Sekiguchi, D. J. Sullivan, P. J. Tristram, P. C. M. Yock, and T. Yoshioka. Discovery of a cool planet of 5.5 Earth masses through gravitational microlensing. , 439:437–440, January 2006. doi: 10.1038/nature04441.

D. P. Bennett. An Efficient Method for Modeling High-magnification Planetary

- Microensing Events. *ApJ*, 716:1408–1422, June 2010. doi: 10.1088/0004-637X/716/2/1408.
- D. P. Bennett, I. A. Bond, A. Udalski, T. Sumi, F. Abe, A. Fukui, K. Furusawa, J. B. Hearnshaw, S. Holderness, Y. Itow, K. Kamiya, A. V. Korpela, P. M. Kilmartin, W. Lin, C. H. Ling, K. Masuda, Y. Matsubara, N. Miyake, Y. Muraki, M. Nagaya, T. Okumura, K. Ohnishi, Y. C. Perrott, N. J. Rattenbury, T. Sako, T. Saito, S. Sato, L. Skuljan, D. J. Sullivan, W. L. Sweatman, P. J. Tristram, P. C. M. Yock, M. Kubiak, M. K. Szymański, G. Pietrzyński, I. Soszyński, O. Szewczyk, L. Wyrzykowski, K. Ulaczyk, V. Batista, J. P. Beaulieu, S. Brilliant, A. Cassan, P. Fouqué, P. Kervella, D. Kubas, and J. B. Marquette. A Low-Mass Planet with a Possible Sub-Stellar-Mass Host in Microlensing Event MOA-2007-BLG-192. *ApJ*, 684:663–683, September 2008. doi: 10.1086/589940.
- 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, L. Wyrzykowski, The OGLE Collaboration, D. L. DePoy, C. Han, S. Kaspi, C.-U. Lee, F. Mallia, T. Natusch, B.-G. Park, R. W. Pogge, D. Polishook, The μ FUN 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, The 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*, 713:837–855, April 2010. doi: 10.1088/0004-637X/713/2/837.
- I. A. Bond, A. Udalski, M. Jaroszyński, N. J. Rattenbury, B. Paczyński, I. Soszyński, L. Wyrzykowski, M. K. Szymański, M. Kubiak, O. Szewczyk, K. Żebruń, G. Pietrzyński, F. Abe, D. P. Bennett, S. Eguchi, Y. Furuta, J. B. Hearnshaw, K. Kamiya, P. M. Kilmartin, Y. Kurata, K. Masuda, Y. Matsubara, Y. Muraki, S. Noda, K. Okajima, T. Sako, T. Sekiguchi, D. J. Sullivan, T. Sumi, P. J. Tristram, T. Yanagisawa, and P. C. M. Yock. OGLE 2003-BLG-235/MOA 2003-BLG-53: A Planetary Microlensing Event. *ApJ*, 606:L155–L158, May 2004. doi: 10.1086/420928.
- M. Dominik. Galactic microlensing with rotating binaries. *A&A*, 329:361–374, January 1998.

- S. Dong, D. L. DePoy, B. S. Gaudi, A. Gould, C. Han, B.-G. Park, R. W. Pogge, A. Udalski, O. Szewczyk, M. Kubiak, M. K. Szymański, G. Pietrzyński, I. Soszyński, Ł. Wyrzykowski, and K. Żebruń. Planetary Detection Efficiency of the Magnification 3000 Microlensing Event OGLE-2004-BLG-343. *ApJ*, 642: 842–860, May 2006. doi: 10.1086/501224.
- S. Dong, I. A. Bond, A. Gould, S. Kozłowski, N. Miyake, B. S. Gaudi, D. P. Bennett, F. Abe, A. C. Gilmore, A. Fukui, K. Furusawa, J. B. Hearnshaw, Y. Itow, K. Kamiya, P. M. Kilmartin, A. Korpela, W. Lin, C. H. Ling, K. Masuda, Y. Matsubara, Y. Muraki, M. Nagaya, K. Ohnishi, T. Okumura, Y. C. Perrott, N. Rattenbury, T. Saito, T. Sako, S. Sato, L. Skuljan, D. J. Sullivan, T. Sumi, W. Sweatman, P. J. Tristram, P. C. M. Yock, The MOA Collaboration, G. Bolt, G. W. Christie, D. L. DePoy, C. Han, J. Janczak, C.-U. Lee, F. Mallia, J. McCormick, B. Monard, A. Maury, T. Natusch, B.-G. Park, R. W. Pogge, R. Santallo, K. Z. Stanek, The μ FUN Collaboration, A. Udalski, M. Kubiak, M. K. Szymański, G. Pietrzyński, I. Soszyński, O. Szewczyk, Ł. Wyrzykowski, K. Ulaczyk, and The OGLE Collaboration. Microlensing Event MOA-2007-BLG-400: Exhuming the Buried Signature of a Cool, Jovian-Mass Planet. *ApJ*, 698:1826–1837, June 2009. doi: 10.1088/0004-637X/698/2/1826.
- A Einstein. Lens-like action of a star by the deviation of light in the gravitational field. *Science*, 84(2188):506–507, 1936. ISSN 00368075. URL <http://www.jstor.org/stable/1663250>.
- James D Foley. *Computer graphics :principles and practic*. Addison-Wesley systems programming series. Addison-Wesley, 2nd edition, 1990.
- 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, D. L. DePoy, C. Han, S. Kaspi, C.-U. Lee, F. Mallia, T. Natusch, R. W. Pogge, B.-G. Park, 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, 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, B. Chaboyer, A. Crocker, S. Frank, and B. Macintosh. Discovery of a Jupiter/Saturn Analog with Gravitational Microlensing. *Science*, 319:927–, 2008. doi: 10.1126/science.1151947.
- A. Gould. MACHO velocities from satellite-based parallaxes. *ApJ*, 421:L75–L78,

- February 1994. doi: 10.1086/187191.
- A. Gould. Extreme Microlensing toward the Galactic Bulge. *ApJ*, 480:188–+, May 1997. doi: 10.1086/303942.
- A. Gould. Resolution of the MACHO-LMC-5 Puzzle: The Jerk-Parallax Microlens Degeneracy. *ApJ*, 606:319–325, May 2004. doi: 10.1086/382782.
- A. Gould. Hexadecapole Approximation in Planetary Microlensing. *ApJ*, 681:1593–1598, July 2008. doi: 10.1086/588601.
- A. Gould and C. Gaucherel. Stokes’s Theorem Applied to Microlensing of Finite Sources. *ApJ*, 477:580–+, March 1997. doi: 10.1086/303751.
- A. Gould, A. Udalski, D. An, D. P. Bennett, A.-Y. Zhou, S. Dong, N. J. Rattenbury, B. S. Gaudi, P. C. M. Yock, I. A. Bond, G. W. Christie, K. Horne, J. Anderson, K. Z. Stanek, D. L. DePoy, C. Han, J. McCormick, B.-G. Park, R. W. Pogge, S. D. Poindexter, I. Soszyński, M. K. Szymański, M. Kubiak, G. Pietrzyński, O. Szewczyk, L. Wyrzykowski, K. Ulaczyk, B. Paczyński, D. M. Bramich, C. Snodgrass, I. A. Steele, M. J. Burgdorf, M. F. Bode, C. S. Botzler, S. Mao, and S. C. Swaving. Microlens OGLE-2005-BLG-169 Implies That Cool Neptune-like Planets Are Common. *ApJ*, 644:L37–L40, June 2006. doi: 10.1086/505421.
- D. S. Graff and A. Gould. Microlens Parallaxes of Binary Lenses Measured from a Satellite. *ApJ*, 580:253–260, November 2002. doi: 10.1086/343128.
- Robin M. Green. *Spherical Astronomy*. Cambridge University Press, 1985.
- J. Janczak, A. Fukui, S. Dong, L. A. G. Monard, S. Kozłowski, A. Gould, J. P. Beaulieu, D. Kubas, J. B. Marquette, T. Sumi, I. A. Bond, D. P. Bennett, F. Abe, K. Furusawa, J. B. Hearnshaw, S. Hosaka, Y. Itow, K. Kamiya, A. V. Korpela, P. M. Kilmartin, W. Lin, C. H. Ling, S. Makita, K. Masuda, Y. Matsubara, N. Miyake, Y. Muraki, M. Nagaya, T. Nagayama, K. Nishimoto, K. Ohnishi, Y. C. Perrott, N. J. Rattenbury, T. Sako, T. Saito, L. Skuljan, D. J. Sullivan, W. L. Sweatman, P. J. Tristram, P. C. M. Yock, The MOA Collaboration, J. H. An, G. W. Christie, S.-J. Chung, D. L. DePoy, B. S. Gaudi, C. Han, C.-U. Lee, F. Mallia, T. Natusch, B.-G. Park, R. W. Pogge, The μ FUN Collaboration, T. Anguita, S. Calchi Novati, M. Dominik, U. G. Jørgensen, G. Masi, M. Mathiasen, The MiNDSTEp Collaboration, V. Batista, S. Brilliant, A. Cassan, A. Cole, E. Corrales, C. Coutures, S. Dieters, P. Fouqué, J. Greenhill, and The PLANET Collaboration. Sub-Saturn Planet MOA-2008-BLG-310Lb: Likely to be in the Galactic Bulge. *ApJ*, 711:731–743, March 2010. doi: 10.1088/0004-637X/711/2/731.
- M. Jenkins and J. Traub. A three-stage variable-shift iteration for polynomial zeros

- and its relation to generalized rayleigh iteration. *Numerische Mathematik*, 14(3): 252–263, 02 1970. URL <http://dx.doi.org/10.1007/BF02163334>.
- M. A. Jenkins and J. F. Traub. Algorithm 419: zeros of a complex polynomial [c2]. *Commun. ACM*, 15(2):97–99, 1972. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/361254.361262>.
- C. R. Keeton. Computational Methods for Gravitational Lensing. *ArXiv Astrophysics e-prints*, February 2001.
- A. Korpela. *Modelling Complex Gravitational Microlensing Events*. PhD thesis, Victoria University of Wellington, 2007.
- S. Mao and B. Paczynski. Gravitational microlensing by double stars and planetary systems. *ApJ*, 374:L37–L40, June 1991. doi: 10.1086/186066.
- I. Newton, W., and J. Innys. *Opticks: or, A treatise of the reflections, refractions, inflections and colours of light. ...* Printed for W. and J. Innys, printers to the Royal Society, at the Prince’s-Arms in St. Paul’s Church-Yard., 1718. URL <http://books.google.co.nz/books?id=TwhbAAAAQAAJ>.
- B. Paczynski. Gravitational microlensing by the galactic halo. *ApJ*, 304:1–5, May 1986. doi: 10.1086/164140.
- S. Poindexter, C. Afonso, D. P. Bennett, J.-F. Glicenstein, A. Gould, M. K. Szymański, and A. Udalski. Systematic Analysis of 22 Microlensing Parallax Candidates. *ApJ*, 633:914–930, November 2005. doi: 10.1086/468182.
- S. H. Rhie. How Cumbersome is a Tenth Order Polynomial?: The Case of Gravitational Triple Lens Equation. *ArXiv Astrophysics e-prints*, February 2002.
- Ken F Riley, Michael P Hobson, and Stephen J Bence. *Mathematical Methods for Physics and Engineering; 3rd ed.* Cambridge Univ., Cambridge, 2006.
- M. C. Smith, S. Mao, and B. Paczyński. Acceleration and parallax effects in gravitational microlensing. , 339:925–936, March 2003. doi: 10.1046/j.1365-8711.2003.06183.x.
- T. Sumi. Moa-ii observation in 2009. Presented at the 14th workshop on Gravitational Microlensing, 2010.
- T. Sumi, D. P. Bennett, I. A. Bond, A. Udalski, V. Batista, M. Dominik, P. Fouqué, D. Kubas, A. Gould, B. Macintosh, K. Cook, S. Dong, L. Skuljan, A. Cas-san, F. Abe, C. S. Botzler, A. Fukui, K. Furusawa, J. B. Hearnshaw, Y. Itow, K. Kamiya, P. M. Kilmartin, A. Korpela, W. Lin, C. H. Ling, K. Masuda, Y. Matsubara, N. Miyake, Y. Muraki, M. Nagaya, T. Nagayama, K. Ohnishi, T. Okumura, Y. C. Perrott, N. Rattenbury, T. Saito, T. Sako, D. J. Sullivan,

- W. L. Sweatman, P. J. Tristram, P. C. M. Yock, The MOA Collaboration, J. P. Beaulieu, A. Cole, C. Coutures, M. F. Duran, J. Greenhill, F. Jablonski, U. Marboeuf, E. Martioli, E. Pedretti, O. Pejcha, P. Rojo, M. D. Albrow, S. Brilliant, M. Bode, D. M. Bramich, M. J. Burgdorf, J. A. R. Caldwell, H. Calitz, E. Corrales, S. Dieters, D. Dominis Prester, J. Donatowicz, K. Hill, M. Hoffman, K. Horne, U. G. Jørgensen, N. Kains, S. Kane, J. B. Marquette, R. Martin, P. Meintjes, J. Menzies, K. R. Pollard, K. C. Sahu, C. Snodgrass, I. Steele, R. Street, Y. Tsapras, J. Wambsganss, A. Williams, M. Zub, The PLANET Collaboration, M. K. Szymański, M. Kubiak, G. Pietrzyński, I. Soszyński, O. Szewczyk, Ł. Wyrzykowski, K. Ulaczyk, The OGLE Collaboration, W. Allen, G. W. Christie, D. L. DePoy, B. S. Gaudi, C. Han, J. Janczak, C.-U. Lee, J. McCormick, F. Mallia, B. Monard, T. Natusch, B.-G. Park, R. W. Pogge, R. Santallo, and The μ FUN Collaboration. A Cold Neptune-Mass Planet OGLE-2007-BLG-368Lb: Cold Neptunes Are Common. *ApJ*, 710:1641–1653, February 2010. doi: 10.1088/0004-637X/710/2/1641.
- P. Tisserand, L. Le Guillou, C. Afonso, J. N. Albert, J. Andersen, R. Ansari, É. Aubourg, P. Bareyre, J. P. Beaulieu, X. Charlot, C. Coutures, R. Ferlet, P. Fouqué, J. F. Glicenstein, B. Goldman, A. Gould, D. Graff, M. Gros, J. Haissinski, C. Hamadache, J. de Kat, T. Lasserre, É. Lesquoy, C. Loup, C. Magneville, J. B. Marquette, É. Maurice, A. Maury, A. Milsztajn, M. Moniez, N. Palanque-Delabrouille, O. Perdureau, Y. R. Rahal, J. Rich, M. Spiro, A. Vidal-Madjar, L. Vigroux, S. Zylberajch, and The EROS-2 Collaboration. Limits on the Macho content of the Galactic Halo from the EROS-2 Survey of the Magellanic Clouds. *A&A*, 469:387–404, July 2007. doi: 10.1051/0004-6361:20066017.
- A. Udalski, M. Jaroszyński, B. Paczyński, M. Kubiak, M. K. Szymański, I. Soszyński, G. Pietrzyński, K. Ulaczyk, O. Szewczyk, Ł. Wyrzykowski, G. W. Christie, D. L. DePoy, S. Dong, A. Gal-Yam, B. S. Gaudi, A. Gould, C. Han, S. Lépine, J. McCormick, B.-G. Park, R. W. Pogge, D. P. Bennett, I. A. Bond, Y. Muraki, P. J. Tristram, P. C. M. Yock, J.-P. Beaulieu, D. M. Bramich, S. W. Dieters, J. Greenhill, K. Hill, K. Horne, and D. Kubas. A Jovian-Mass Planet in Microlensing Event OGLE-2005-BLG-071. *ApJ*, 628:L109–L112, August 2005. doi: 10.1086/432795.
- H. J. Witt. Investigation of high amplification events in light curves of gravitationally lensed quasars. *A&A*, 236:311–322, September 1990.
- J. C. Yee, A. Udalski, T. Sumi, S. Dong, S. Kozłowski, J. C. Bird, A. Cole, D. Higgins, J. McCormick, L. A. G. Monard, D. Polishook, A. Shporer, O. Spector, M. K. Szymański, M. Kubiak, G. Pietrzyński, I. Soszyński, O. Szewczyk, K. Ulaczyk, Ł. Wyrzykowski, R. Poleski, The OGLE Collaboration, W. Allen, M. Bos, G. W.

Christie, D. L. DePoy, J. D. Eastman, B. S. Gaudi, A. Gould, C. Han, S. Kaspi, C.-U. Lee, F. Mallia, A. Maury, D. Maoz, T. Natusch, B.-G. Park, R. W. Pogge, R. Santallo, The μ FUN Collaboration, F. Abe, I. A. Bond, A. Fukui, K. Furusawa, J. B. Hearnshaw, S. Hosaka, Y. Itow, K. Kamiya, A. V. Korpela, P. M. Kilmartin, W. Lin, C. H. Ling, S. Makita, K. Masuda, Y. Matsubara, N. Miyake, Y. Muraki, M. Nagaya, K. Nishimoto, K. Ohnishi, Y. C. Perrott, N. J. Rattenbury, T. Sako, T. Saito, L. Skuljan, D. J. Sullivan, W. L. Sweatman, P. J. Tristram, P. C. M. Yock, The MOA Collaboration, M. D. Albrow, V. Batista, P. Fouqué, J.-P. Beaulieu, D. P. Bennett, A. Cassan, J. Comparat, C. Coutures, S. Dieters, J. Greenhill, K. Horne, N. Kains, D. Kubas, R. Martin, J. Menzies, J. Wambsganss, A. Williams, M. Zub, and The PLANET Collaboration. Extreme Magnification Microlensing Event OGLE-2008-BLG-279: Strong Limits on Planetary Companions to the Lens Star. *ApJ*, 703:2082–2090, October 2009. doi: 10.1088/0004-637X/703/2/2082.