FITTING BINARY LENS GRAVITATIONAL MICROLENSING EVENTS WITH EXAMPLE-BASED ALGORITHMS

BY

PIERRE LE ROUX VERMAAK

Thesis Presented for the Degree of

Doctor of Philosophy

in the Department of

ASTRONOMY

UNIVERSITY OF CAPE TOWN

15 August 2007

Abstract

Fitting binary lens models to gravitational microlensing events is currently a time-consuming and labour-intensive process.

Example-based algorithms more commonly used in the field of data mining were applied to simulated and observed light curves in order to facilitate the fitting of a simple, seven-parameter binary lens model with minimal human intervention. After examining some of the causes behind the poor performance of conventional fitting techniques, such as the ambiguity of binary lens light curves and premature convergence to local minima, attempts were made to overcome these difficulties by deriving features of light curves to be used instead of the curves themselves. Algorithms to select features were tested and applied to simulated data in order to find those most suited to regression. Regression algorithms using derived features were found to be less successful than those using pre-processed light curves.

A number of data mining algorithms were applied to simulated binary microlensing events in a series of experiments aimed at determining the best combination of algorithm, light curve pre-processing technique and training set construction, as measured by the correlation between known model parameters and fitting results. An REP tree induction algorithm, enhanced with the "Bagging" meta-data technique, proved most effective, achieving correlations in excess of 0.9 when trained using data sets biased towards light curves exhibiting high variance and separated into subsets by projected orbital radius of the lens system and the crossing angle of the source.

As proof of concept, light curves from two real microlensing events (EROS-2000-5 and 2003-OGLE-267) were successfully fitted using these techniques. The results indicate that example-based fitting holds promise as an aid to conventional fitting techniques. Future work is recommended to extend the range of fittable light curves, obtain error estimates from fits and fit more complicated models to the data.

Acknowledgments

This thesis was a joint venture between the South African Astronomical Observatory (SAAO) and the University of Cape Town (UCT) and was partially funded by a grant from the National Research Foundation of South Africa, for which I am very grateful. My co-supervisors Prof. Brian Warner from UCT and Dr. John Menzies from SAAO are owed a great debt for their extraordinary patience and flexibility. Thanks to Marguerite Armstrong from the UCT Astronomy Department for her ceaseless efforts in tracking down my paper work every year. Finally, I would like to thank all the contributors to the excellent open source WEKA data mining application as well as all those responsible for the splendid, free tools utilized during the completion of this thesis: gcc, valgrind, Knoppix, Kubuntu, tetex, vi, subversion, gnuplot, kghostview, kpdf, mozilla, amarok, debian, eclipse, f2c, xfig and the free and open source software community in general.

To Barbara, without whom I would just be wasting my time.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Introduction

Let's start by defining "Local Gravitational Microlensing" (LGM). It is a relatively new field of Astronomy, related to Gravitational Lensing (GL) but with a distinct literature and area of interest. Although the theory of Gravitational Lensing has been understood since at least 1936 [4], observations only became possible by the 1990s as the challenge posed to observers could only be met by the advent of modern computing power and digital instruments such as the CCD chip. Although pioneering observational groups overcame the original technical challenges ([5], [6], [7]), they were soon faced with the extraordinarily time-consuming task of interpreting Microlensing light curves. As of this writing, LGM observations are poised to enter a new era of abundant observations (see e.g. [8]) while the computational and theoretical interpretation of those observations is still a time-consuming, labour-intensive task.

The exponential increase in computing power since the 1980s [9] that allowed for analysis of data intensive observations, also saw the re-emergence of unconventional numerical techniques that are now known by umbrella terms such as "Evolutionary Computation" and "Data Mining" (which has been referred to as "statistics plus marketing"). Marketing or not, these techniques promised generality, ease of implementation and power and the urge to apply them to analysis of LGM analysis proved irresistible.

This work originated from both the real need in the LGM community to improve or speed up the interpretation of observational data as well as to explore the applicability of primarily data mining methods to the field.

1.2 Background

Before the aims of this project are laid out in detail, the next Section (1.2) describes the relevant theory of Gravitational Lensing and the current state of affairs in the field.

1.2.1 Gravitational Lensing

Gravity and Light

According to General Relativity [10], any massive body should distort the path taken by a ray of light. Although never observed in everyday life, the scale of mass and distance required for this effect to be observable are common in astronomy. A natural consequence of the "bending" of light by massive bodies was to predict that stars could affect light from background objects in this way. Einstein made this prediction in 1936 [4].

Einstein was not optimistic about this effect ever being observed, but the first observation of a similar gravitational lensing effect was made by Walsh, Carswell and Weymann [11] in 1979. In this case the lens was a foreground galaxy and the background source of light was a quasar. The observation of such lensing phenomena grew into the field of Gravitational Lensing during the 1980s and yielded a wealth of literature on the subject.

Photometry of lensed sources reveal short deviations in flux on a time scale of years (e.g. [12]), believed to be due to individual sub-solar-mass bodies in the lens galaxy aligning precisely with the background source and so fortuitously leading to additional amplification while the alignment lasts. This phenomenon was dubbed "Microlensing" because the angular radius of a single star's lensing influence was of the order of a micro arc second. See 2.1.1 for a definition of this size.

Galactic Microlensing as opposed to lensing

The form of Gravitational Lensing predicted by Einstein was to take place on a completely different distance and mass scale: both the background source of light and the lensing body would be stars or small systems of stars in our galaxy. It was thought that lensing on this scale would be impossible to detect, as an observer would have to monitor millions of background stars around the clock with great precision to detect just one such an event when a foreground and background star happen to line up precisely enough with the observer for a lensing effect to occur.

Astronomical technology advanced dramatically with the advent of accessible computer power and Paczynski [13] suggested that the time had come for detecting gravitational lensing in our own Galaxy (or in the Halo, at least), with stars acting as the sources of light to be lensed. The "local" galactic scale involved here (as opposed to the cosmic scale lensing discussed in 1.2.1 meant that actual images of the background star could typically not be resolved, only their combined flux and hence the magnification caused by the lens. Paczynski's prediction led to the formation of several observational groups dedicated to the detection of galactic lensing events (notably MACHO [5], OGLE [6] and EROS [7]). The survey groups were spectacularly successful and the first joint detection of a lensing event on the "Galactic", "local", scale was announced jointly by the MACHO and EROS collaborations in 1993 [14] and [15]. A new term was coined for this type of lensing: Gravitational Microlensing (GM). To make a distinction between this kind of lensing, where both the source and lens are in or near our own galaxy and are of roughly stellar mass, and the type of Microlensing that is seen when the constituents of a galaxy that is acting as a single lens cause perturbations, we will use the term Local Gravitational Microlensing (LGM) to describe the local galactic events.

Apart from confirming predictions that LGM could be observed, the survey experiments had some grandiose scientific goals for this brand new branch of astronomy and the new possibilities it presented. Arguably the most important goal was to determine the heretofore unknown constituents of dark matter. Survey group results presented strong evidence [16] that the majority of this "missing mass" was not to be found in the form of brown dwarfs or other Massive Compact Halo Objects (MACHOs). LGM has also yielded results in Galactic Structure, Stellar Atmospheres, Variable Star Surveys and of course exo-planet research (e.g. [17, 18, 19, 20]).

By 2000, more than 400 events had been detected, placing LGM firmly on the scientific map as an observable phenomenon.

Binaries and planets

Microlensing events where the lens is a binary star system were essentially noise to the dark matter experiments but these types of events hold the promise of yielding detections in an entirely different field of astronomy, namely the search for extra-solar planets. In 1991, soon after suggesting the feasibility of detecting LGM events, Mao & Paczynski [21] drew attention to the possibility of detecting the presence of an extra-solar planet during LGM events. They pointed out that the magnification caused by a binary lens can differ markedly from that of a single lens so that even a secondary of very small mass compared with the primary lens can be detected if the magnification is monitored closely over time. Detection of the small secondary lens in a binary is subject to various geometrical factors, including the projected orbital radius around its parent star, its mass ratio, the exact alignment of the source, lens and observer, etc. Nonetheless, Mao & Paczynski estimated that a planet roughly the size of Jupiter would stand a 5-10 per cent chance of being detected if its parent star were acting as a lens in an LGM scenario. Even nearby planets are extremely hard to detect by optical methods due to the overwhelming brightness of their parent stars, but LGM relies on gravitational detection only and promises detections at distances comparable to that of the Galactic Bulge, making it a potentially effective alternative to other planet detection techniques.

Figures 1 and 2 show some theoretical binary LGM light curves. Generally speaking, the smaller the planet, the smaller the disturbance to the normal single lens light curve, although the shape of the curve is highly dependent on geometry (i.e., projected distance between primary and secondary, angle of transition of the source, and the proximity of the transition path to either of the two lenses).

It is not known what proportion of lenses are single, binaries or even multiple systems. As shown in [22], a system with one star and several planetary bodies can be approximated fairly well by assuming that the effect of each planet on the light curve is independent of the others. Thus we can approximate a multi-planet system by considering each planet with the star as a separate binary system. The

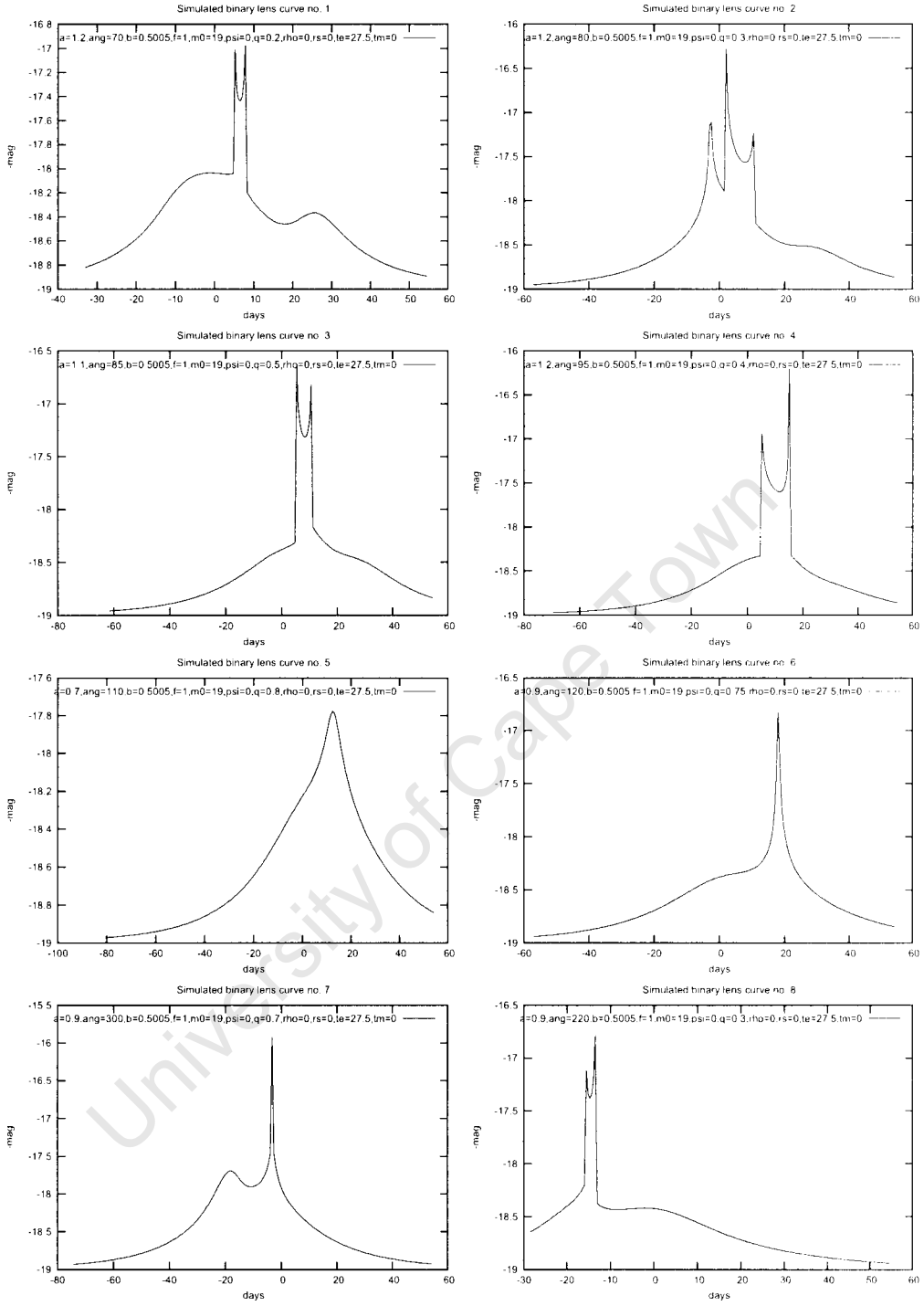

Figure 1: Examples of binary LGM light curves with large mass ratio lenses. These events all have mass ratios (q) larger than 0.2 and projected orbital separation (a) to place them within the lensing zone. Although the crossing angle (θ) was chosen to produce a dramatic curve, the impact parameter (b) is actually quite low ($b = 0.5005$ for all). The unit of b is Einstein angular radius, which will be defined and discussed later in the thesis.

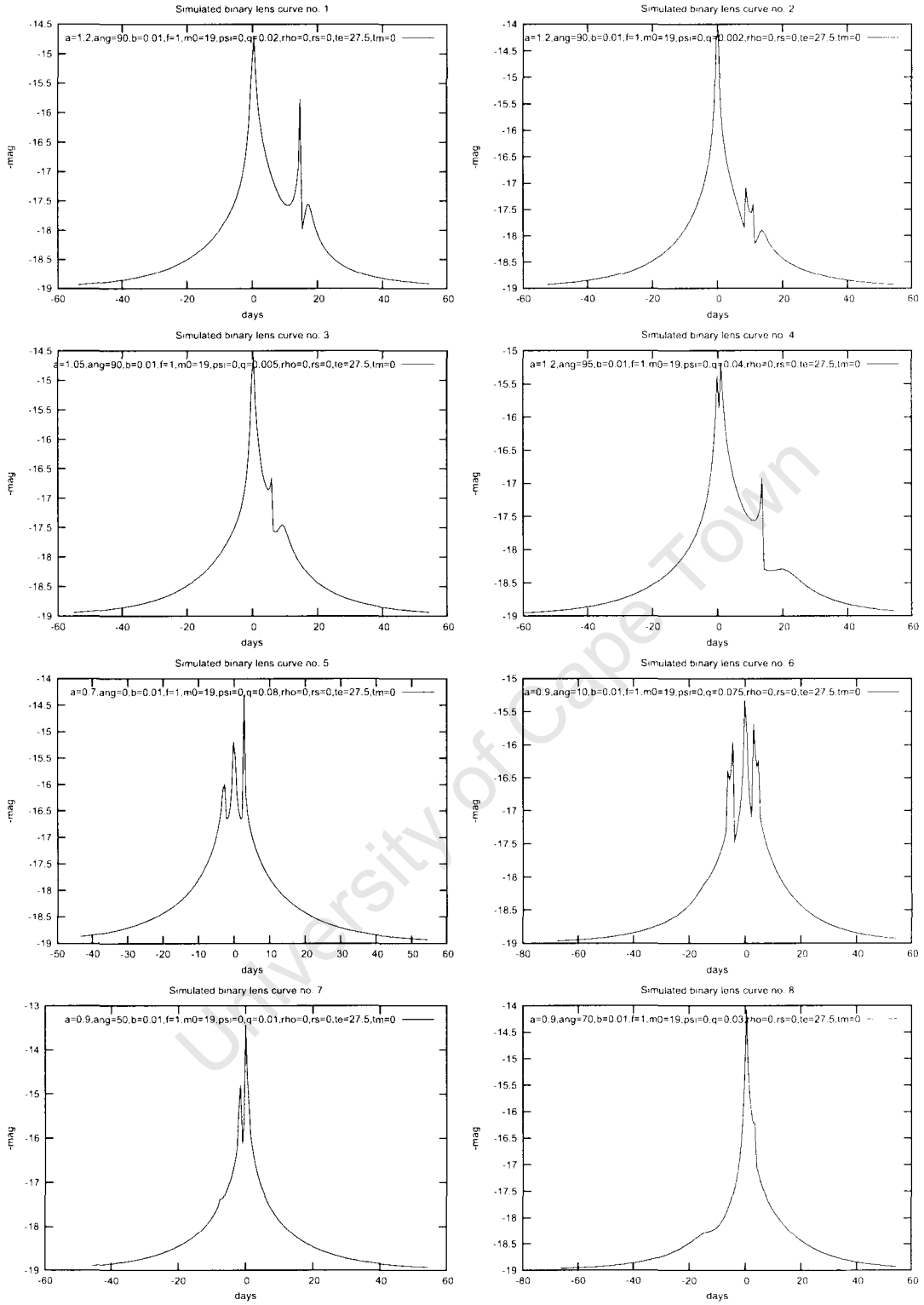

Figure 2: Examples of small mass ratio binary LGM light curves. These events all have $q \leq 0.08$ and were chosen with specific geometry to produce light curves with large distortions.

first practically uncontroversial detection of a planet by LGM was recently made by Bond et al (2004) [2]. Prior to this landmark the only candidates were fairly controversial, such as [23] which could be interpreted as either a planet orbiting a binary star or a higher-mass-ratio binary with rotation effects [24]. Some spectacular binary lens detections have yielded interesting results in a variety of fields such as [25] and [26]. Figures 3 and 4 show two impressive real binary light curves from the literature.

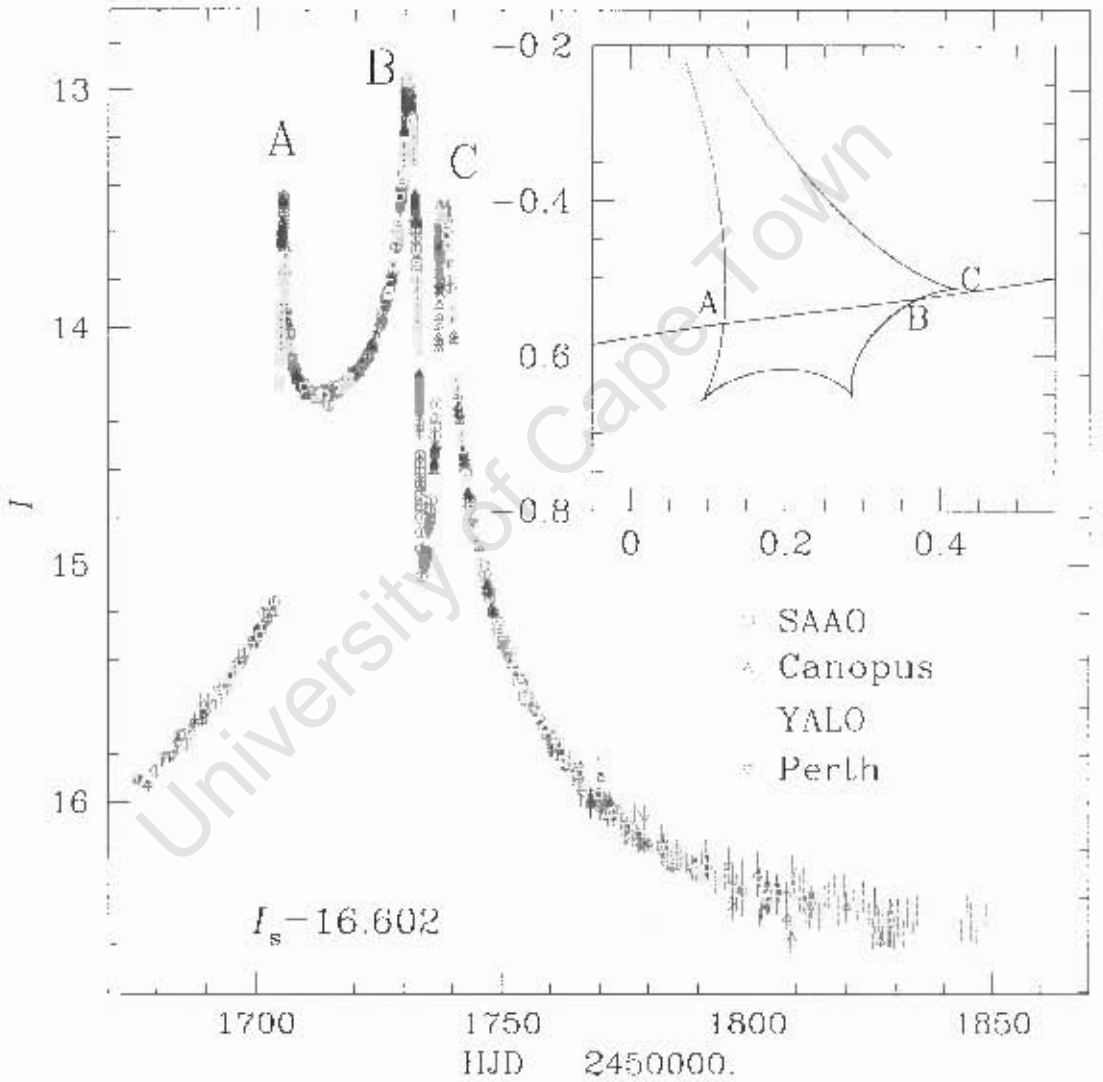

Figure 3: EROS-BLG-2000-5, one of the most spectacular binary lens events ever observed. Reproduced from [1].

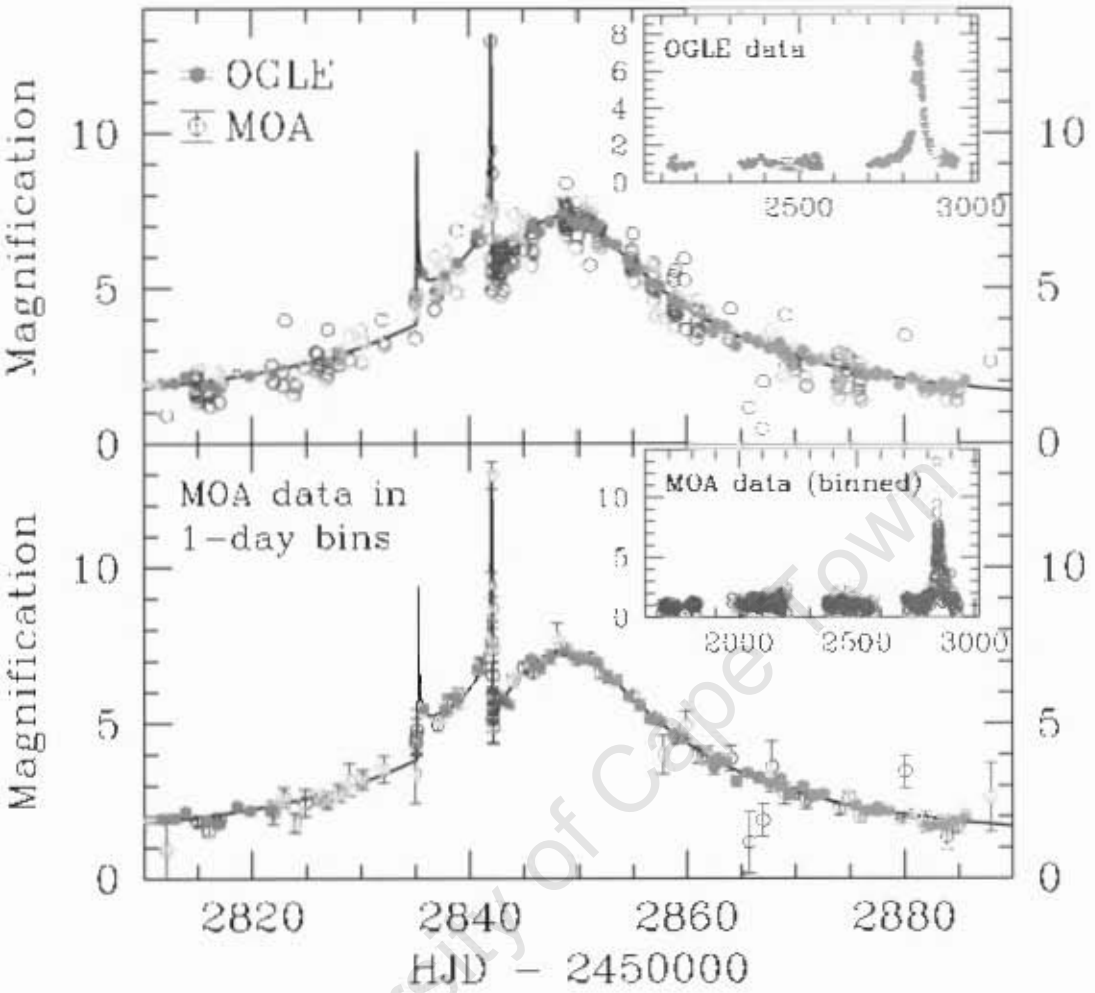

Figure 4: The almost certain planet detection binary lens event observed by OGLE/MOA (2004), reproduced from [2].

1.2.2 Current LGM (Local Galactic Microlensing) research

This Section is a brief review of past and current LGM research and an editorial view on some new developments in the field.

Past Local Galactic Microlensing observations

The initial Microlensing observations could be broadly categorized into "surveys" and "follow-up" (see e.g. [27] for a review of observations and results during this period). Survey groups generally observed millions of stars in dense fields such as the Galactic Bulge, nearby globular clusters or the LMC and SMC and even M31 using wide-field telescopes. The main purpose was detection of as many events as possible, not detailed (high-frequency, low-noise) observations. Typical survey group sampling rates were 1 or 2 points per event per night. Follow-up groups did not in general detect any new events, but observed events that were detected by the survey groups, at higher frequency and with lower noise, thanks to the greater amount of time spent on each event. They were totally dependent on the survey groups to provide the astronomical community with timely alerts on detection of an event in progress. Thanks to the cooperation of the survey groups, follow-up groups obtained some impressive results, although the uncontroversial detection of an extra-solar planet was not among them during the first round of experiments. Despite the negative result, PLANET was able to put some constraints on the existence of planets around lenses in their detection range [3].

Future LGM observations

The initial experiments above not only proved that LGM was a viable new observational field but also obtained important results. Several new experiments which can be described as "second phase" are planned or already under way (such as [28],) building on the success and lessons of the first phase and some of these are reviewed in [29]. Plans for the medium-term future of Microlensing frequently involve space-based missions (e.g. [30], [31], [32]).

Modelling and Interpretation

Past LGM experiments yielded a great many observed light curves and a good proportion of these differed from the "vanilla" single lens light curves defined by Paczynski [13]. PLANET, the follow-up collaboration of which the author was a

member between 1997 and 2000, observed more than 120 curves of which 11 per cent were clearly anomalous by eye [3]. The bottleneck to obtaining scientific results from LGM observations is due as much to the difficulty of interpreting the curves as to observing them. Although the simple single lens curves are relatively easy to model and to extract the relevant lens information from, interpretation of anomalous events requires a lot of effort. Difficulties with extracting information and attempts to overcome them form a major part of this thesis as well as the work of the observational groups. The difficulty arises because even the simple situation where a binary system is acting as the lens leads to a huge variety in theoretically possible light curves. In addition, although the mathematics of a binary lens scenario is simple enough to express in a few short equations, numerical computation is required to calculate the magnification caused by the lens. Faster methods of calculating amplification were developed by Asada in 2002 [33] but the solution remains numerical. This computation time rules out the use of simple methods that rely on approximations or the brute force of powerful computers and modellers are required to come up with interactive and ingenious procedures to find solutions to each specific curve. Section 3.2 describes in greater detail the difficulties in fitting LGM curves.

At present each observational group is applying their own set of methods, although successful attempts are obviously shared with the community. For example, PLANET successfully applied a staged method to at least three events ([25], [3], [1]) but it is only applicable to certain types of events and took of the order of months to apply to some events. Multiple solution sets exist for each observed light curve, necessitating data that are excellent both in quality and quantity in order to exclude all but one of these possible solutions. The problem is discussed further in Section 3.3.4 below.

Ambiguity is one of the hardest problems to solve when interpreting a light curve, and the resolution thereof was one of the driving factors in the establishment of follow-up observational groups. The problem is one of fundamental geometry

because very different lensing system geometries can produce very similar curves. A curve without sufficient data to distinguish between multiple possibilities is of less benefit. Up to 2002, the survey groups did not in general have enough data points in their light curves, necessitating a higher frequency of observation that was only attainable by follow-up observations. The MACHO-97-BLG-41 event was a good example of ambiguity of interpretation. The light curve had anomalous features ascribed to a rotating binary by PLANET [24] and a static triple system by the MPS collaboration [23]. Both groups had independent, high-quality data covering different parts of the light curve. Modellers with different data sets can verify or exclude each others' models, but no one can guarantee that they have the best solution, as parameter space is too large to search exhaustively. The methods introduced in this project do not solve these problems but attempt to provide an alternative to current LGM fitting techniques which could facilitate modelers.

Recent developments in LGM modelling

The modelling of LGM events, and anomalous ones in particular, has advanced since 2000 at which time the state of the art was probably the detailed analysis of EROS-BLG-2000-5 [1]. Advancement is crucial if modellers are to meet the requirements of ambitious detection missions such as the Microlensing Planet Finder [32], as there will be an enormous number of good candidates to model! Yet, most binary lens light curve fitting and exclusion analysis is still based on grid methods (e.g. [34], [35]). These methods step through the model parameter space generating model curves at each grid point to compare with an observed event. If the grid is fine enough, the results are reliable but the computation time to perform a near-exhaustive search is often crippling. Dominik summarizes and further develops the approach taken in [1] in [36], where the dimensionality of the binary light curve fitting problem is reduced by first performing a fit to specific regions of the curve. Asada [33] took a large step towards decreasing fitting time for binary lens events in

2002 by reformulating an equation at the heart of point-mass Microlensing methodology. This equation relates the position of the light source to the image positions for a given lens geometry and is a prerequisite for calculating amplification. Before that paper, the best formulation in use was a 5th-degree complex polynomial which translates into two coupled 5th-degree real polynomials. More about this formulation in 2.2.4. Asada [37] and Asada, Kasai and Kasai [38] then proceeded to use their new formulation to derive an analytical expression for the caustic curves of a binary lens system, in itself another great advance, immediately applicable to, for example, planet detection studies that use caustic-crossing as their detection criterion. A good example of single-lens finite source fitting is given in [39] for the observation of OGLE-2003-BLG-262, and Smith, Mao and Paczynski [40] point out yet another ambiguity caused by the modelling of parallax in single lens light curves. Finally, the characteristics of binary lens light curves in the vicinity of cusp caustics were definitively discussed by Gaudi and Peters in [41].

1.2.3 "Unconventional" Methods for Regression

The terms "regression" or "curve-fitting", which will be used interchangeably in the context of this project, refer to the process of extracting model parameters from observations (although the problem is defined clearly in 3.2). Methods used in this project include various forms of Data Mining algorithms, Artificial Neural Networks and Genetic Algorithms. So-called "conventional" methods as the term is used here refers to algorithms that minimize $\Delta\chi^2$ by performing a brute force grid search for the global minimum of the regression hyper-surface or by using local information such as the $\Delta\chi^2$-gradient. A discussion of conventional and unconventional methods can be found in Chapter 5.

1.3 Thesis

My thesis is:

- Unconventional fitting techniques such as those more often applied in the field of Data Mining or Artificial Intelligence research can be successfully employed in the analysis of Microlensing light curves.

- These techniques are alternative and complementary to χ^2-based LGM photometric light curve fitting.

- The use of these techniques provide a greater level of automation to the light curve fitting process than is attainable with conventional techniques. Currently, interpretation of observations by means of guided, conventional $\Delta\chi^2$-minimisation is a burden on human modelers.

- The use of these techniques speeds up the fitting process.

Most current fitting methods do not at first attempt to find an exact parameter set to model a given data set, nor attempt to fit the most complicated model. Instead, an iterative process is involved where certain parameters are determined first by finding an initial model (e.g. [35]). The primary objective of the methods explored in this thesis is to find these initial models much faster than what is currently the norm. If this stage could be speeded up, all the specific objectives set above could be met.

1.4 Overview

Chapter 2 discusses current LGM models, from the elementary four parameter single lens model to the simple seven parameter binary lens model and extensions to it. Section 2.2 discusses the implementation of binary lens calculations and their various complications. The challenges facing modelers who attempt to fit binary lens events is explained in greater detail in Chapter 3. These include the non-linearity of LGM binary lens events, model ambiguity, computation time and the small radius of convergence of a typical fit.

Chapter 4 introduces the concept of deriving and selecting light curve features as an aide to fitting. Various features are discussed, calculated and evaluated. Chapter

5 is a short review of "conventional" methods of $\Delta\chi^2$-minimisation and introduces a common ground for comparison between these and the more unconventional methods that follow in Chapter 6, which discusses and evaluates example-based regression. This chapter forms the backbone of the exploration into fitting methodologies, especially those based on data mining techniques. Evaluation is based on simulated data which is made more realistic by the introduction of simulated noise. Chapter 7 applies the best methods introduced in the previous Chapter to actual LGM observations and Section 8.1.5 discusses the future extension of this methodology to include extensions to the standard binary lens model.

2 Model

The LGM scenario can be easily described by a few simple equations, despite the extreme diversity of light curves that are theoretically observable based on a simple point-mass binary lens model. Fairly simple equations relate the relative position of the light source and the lens, with respect to an observer, to the image positions and their magnification, although only some can be solved analytically. The remaining equations have to solved by a variety of iterative, trial and error, numerical techniques. This Section describes the basic theory of LGM while 2.2 concentrates on the effective solution of the equations presented here.

2.1 Theory of LGM

2.1.1 Geometry

Lens and source plane

Figure 5 shows the basic LGM scenario. A bright source in the background is lensed by a massive object between it and the observer. Instead of observing the flux of the actual source, the observer measures the total flux from one or more lensed, distorted images of the source. Distorted images are not directly observable in the LGM regime (with current technology) due to their small angular separation, which is of the order of a milliarcsecond. Arguably the most important equation in LGM, Eq. 1, maps the position of the un-lensed source on the sky to the positions of its images (assuming a point-like source). Figure 6 shows some example images and the un-lensed position of the source on the sky. The shape, size and position of the images vary significantly as the source position changes.

The derivation of the equation that relates source to image position is based on simple geometrical arguments and can be found in e.g. [13], [42]. In the next Section we introduce a convenient unit of distance that further simplifies it.

In Local Gravitational Microlensing, the typical distance between observer and source is of the order of 30000 light years, or roughly the distance between us and the centre of our galaxy. The lens can be anywhere in this range with exactly half

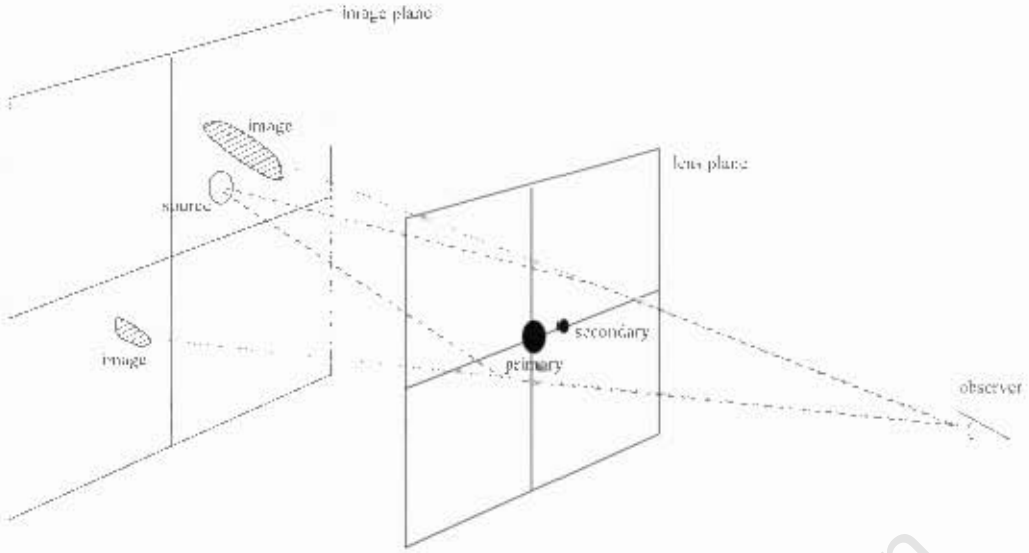

Figure 5: Gravitational Microlensing geometry. Illustration of a binary lens deflecting light from a background source. Multiple images are formed (2 for a single lens, 3 or 5 for a binary lens).

way being optimal.

Much more important than the exact distance to the lens is its alignment with the source and observer. From the observer's point of view, the lens needs to be aligned with the source so that they are within a few milliarcseconds of each other on the sky. The precision of alignment depends on the mass of the lens and distance to it and is of the order of milliarcseconds for LGM events. It is this stringent requirement on alignment that makes an LGM event a rare occurrence that requires the monitoring of millions of stars to detect just a few dozen events. This exacting geometrical requirement allows us to use some approximations that simplify the theory without degrading its accuracy. First, the lens can be called "thin". This approximation requires a lensing scenario where the size of the lens projected onto the observer-source axis is negligible as compared with the observer-source distance. This requirement is clearly met in LGM. The second approximation is to consider the lens as a point, or in the case of a binary or multiple lensing system, each body is approximated as a point.

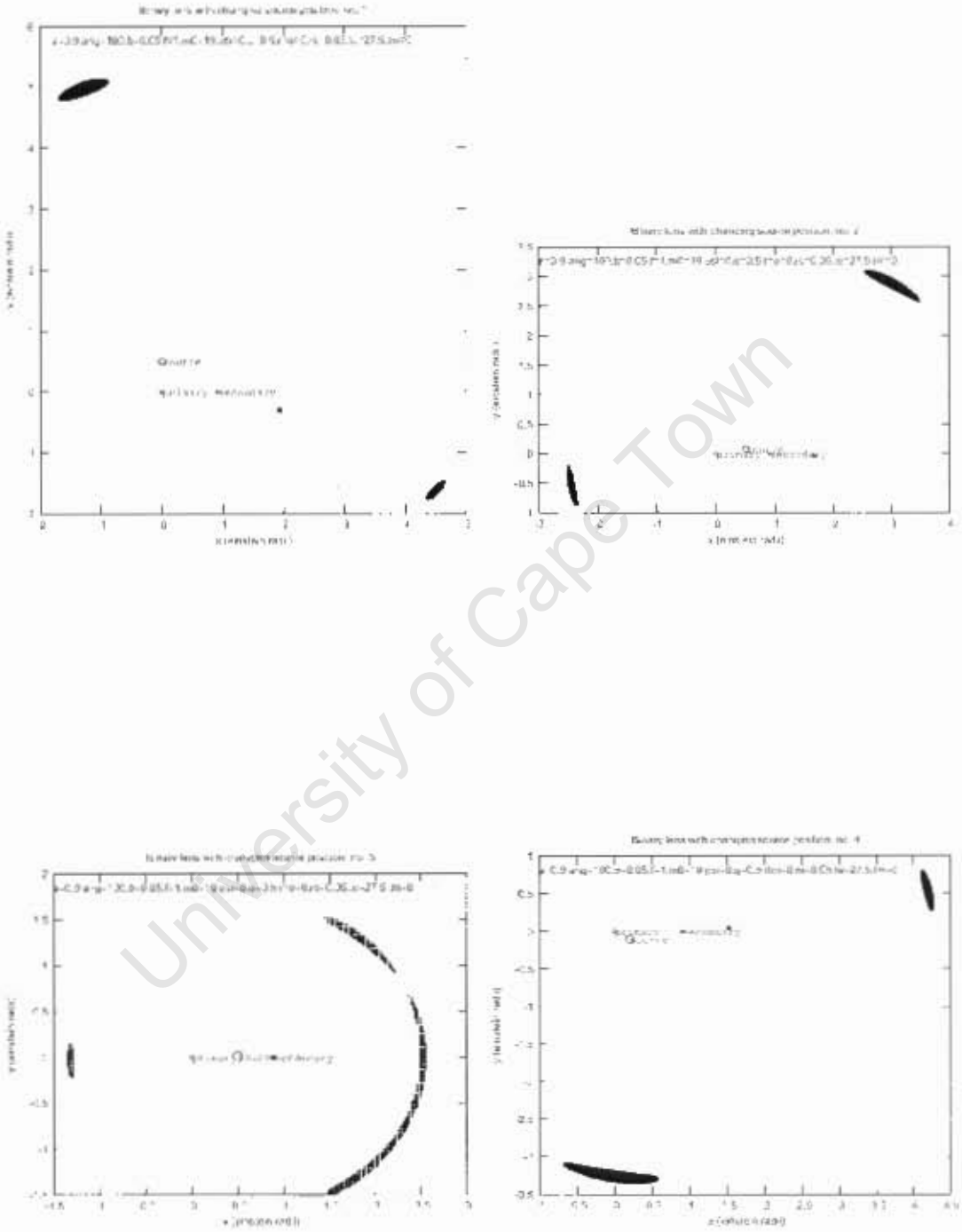

Figure 6: Images of the source during a Microlensing event. The lens geometry is the same for all four panels, but the source position is varied. The angular source radius is 0.05 θ_e, at the upper limit of angular source radii for LGM.

Lensing Equation and Angular Einstein radius

We now introduce a general lensing equation that describes the relationship between source position and image positions as projected onto a plane in the sky perpendicular to the observer's line of sight and located at the lens, valid for small deflection angles. The equation is at the heart of many Microlensing calculations as many observable quantities can be derived from it, including the total amplification of the source.

The point mass lensing equation was first derived by Paczynski [13] and Kayser et al. [42]. The version of the equation presented here uses complex coordinates, a methodology introduced by Bourassa and Kantowski [43] and applied to LGM by Witt [44]. The use of complex coordinates is a calculational convenience, introduced by Bourassa and Kantowski "because it is almost always easier to sum over complex functions than their components." The real and imaginary parts of the complex numbers map directly to real x- and y-coordinates, respectively, in the lens plane.

The relationship between the source's position and the position of its images in the LGM scenario is given by the dimensionless, normalized Equation 1

$$\zeta = z + \sum_{i}^{N} \frac{m_i}{\overline{z_i} - \overline{z}} \tag{1}$$

where m_i is the mass of lens i, ζ is the source position, the z_i are the lens positions and z is the image position(s), all unitless. $\overline{z}$ is the complex conjugate of z. The sum is over the total number of lenses and the total lens mass is normalized to 1.

The normalization of Equation 1 requires the introduction of the "angular Einstein Radius", an intuitive and convenient unit of angular radius applicable to LGM, given in Equation 2

$$\theta_E = \sqrt{\frac{4G}{c^2}\frac{M}{D_{rel}}} \tag{2}$$

where

$$\frac{1}{D_{rel}} = \frac{1}{D_l} - \frac{1}{D_s} \tag{3}$$

D_l is the distance to the lens, D_s the distance to the source, G is the Gravitational Constant and M is the mass of the lens.

Apart from simplifying the formulae associated with LGM, the angular Einstein radius is of fundamental importance to gravitational lensing and has a clear physical interpretation: the image of a point source that is exactly aligned with a point lens appears to the observer as a perfect ring around the lens with an angular radius of one angular Einstein radius. This is seen by putting $\zeta = 0$ in Eq. 1 for an $i = 1$ (single) lens. θ_E appears in various formulas and sets the angular distance scale for LGM events. For example, as a source nears the position of the lens on the sky its images move towards the Einstein ring. Images are normally found in the proximity of the angular Einstein Radius whenever non-negligible lensing is taking place. Extended (finite) sources cause extended images and these elongate along the ring when significant lensing is taking place. Fig. 6 shows this effect.

Amplification

The angular Einstein diameter of a typical LGM event with a mass of $0.3M_{sun}$ is slightly less than 1 milliarcsecond, making the angular scale of LGM effects too small to be resolved by current telescopes. Observers thus have to deal with the overall effect of lensing by measuring the combined flux from all images of the source. The source is amplified because the combined flux from the images is greater than the flux from an un-lensed source (again, see Fig. 6 for an illustration). One simple method of calculating the amplification when dealing with a finite source is in fact exactly this: one simply divides the total image flux by the source flux to retrieve the total amplification. Images of a point source are points themselves and do not have an area to add up but their amplification can be calculated efficiently by taking the determinant of the Jacobian of the source to image transformation in Eq. 1, given by

$$\det J_i = 1 - \frac{\delta\zeta}{\delta\overline{z_i}}\frac{\overline{\delta\zeta}}{\delta\overline{z_i}} \tag{4}$$

which is valid for each image i. The total amplification is the sum of Eq. 4 over all images,

$$A_{total} = \sum_{i}^{N_{images}} \frac{1}{\det J_i} \tag{5}$$

where N_{images} is the total number of images, in other words the total number of solutions to Eq. 1.

Caustics and critical curves

From Eq. 5 we see that $\det J_i = 0$ for any of the images implies infinite amplification of a given source. This unnatural prediction is a result of the assumption of a point source and is obviously not attained in nature. The geometry where $\det J_i = 0$ does indeed occur during lensing events and can lead to extremely high amplification, the exact height of which is determined by geometry and the size of the source. These $\det J_i = 0$ regions are therefore of utmost importance during a lensing event and often completely dictate the structure of the light curve. Witt [44] formulates these regions as a parametric curve in the image plane by Equation 6,

$$\frac{m_1}{(\overline{z_1} - \overline{z})^2} + \frac{m_2}{(\overline{z_2} - \overline{z})^2} = e^{i\phi} \tag{6}$$

with

$$\phi \in [0, 2\pi] \tag{7}$$

These regions are closed curves for any lensing system with more than one lens, and a single point at the position of the lens in the case of a single lens. The curves can be described as an implicit function of image position or source position. When described as a function of image position they are called "critical curves". The same curves translated to the source plane gives source positions where magnification is

infinite and are called "caustic curves". Translation between critical and caustic curves can be achieved via Eq. 1.

The caustic curve description is particularly useful when analyzing the structure of features in an LGM curve. Peaks correspond to caustic crossings or near approaches by the source to a caustic curve. Generally, one is more interested in caustic curves than critical curves as caustics are more closely related to the observed light curve. There is a wealth of literature available on caustics and LGM curve analysis based on caustic crossings. One of the most successful current methods of light curve analysis relies on determining the angles at which the source path intersects caustic curves as a starting point in determining the other LGM parameters [1].

2.1.2 Single lens

The simplest lensing geometry to analyze is that of a single lens. Single lens systems are the only ones where lens amplification can be expressed analytically and calculated by simply substituting the source position into the equation for amplification. Single lens events can be thought of as the "default" event in that an event is considered to be caused by a single lens (by the principle of Occam's razor) unless proved otherwise. The majority of LGM events are classified as single ([45], [3]).

Parameters

Figure 7 is similar to 5 but includes the parameters used to describe single and binary lens geometry in this document. Several other parameterizations exist but most are qualitatively similar. One point of note is that most other parameterizations differ from the one used in this thesis in their point of origin and units of angular radius. In this piece the point of origin is always taken to be the position of the primary. In most others, the origin is placed at the centre of mass of the binary. Similarly, the unit of angular distance used here is the angular Einstein radius of the primary, not the binary lens.

b is the impact parameter of the source path to the primary lens projected on

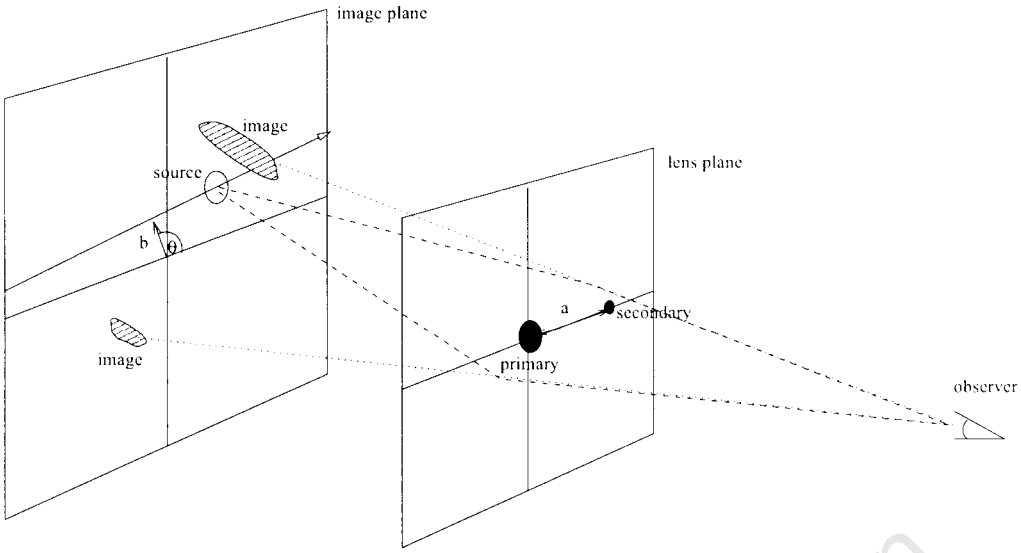

Figure 7: Parameters of the simple binary model. b and θ fully describe the linear path of the source. q and a describe the secondary lens. Not in the diagram, t_e, t_m and m_0, respectively the angular Einstein Radius crossing time, time of closest approach to the primary and un-lensed source magnitude, describe the translation and scaling of the light curve in a time-magnitude plane.

the sky, assuming a linear path. When the source path is not linear due to having parallax and binary lens rotation effects projected onto the source's movement, for example, b no longer represents the impact parameter, but rather the distance between source and primary lens at the time when a linear path would have had its closest approach. t_0 is the time corresponding to the source being at point b, that is the time when the source on a linear path is closest to the primary lens if the source path is linear. t_e has the unit of Days and is also known as the angular Einstein Radius crossing time. It sets the time scale for the LGM event and is typically between 10 and 20 days, although events with much shorter and larger time scales have been observed. m_0 is simply the un-lensed magnitude of the source.

Siting the origin at the position of the primary, unlike the standard parameter system which sites it at the centre of mass of the binary lens, facilitates event comparison except in the case of close binaries, where a better point-lens approximation is obtained by placing the origin at the centre of mass of the binary. Comparing

binary lenses to single lens systems for low mass ratio systems is easier if we site the origin at the position of the primary, as the origin remains in the same position. In the standard coordinate system the origin would move, which means the impact parameter b would have to change in order to align the central peak of the two curves in time.

The Single Lens Equation

Substituting $n = 1$ into Eq. 1 yields Eq. 8, the "single lens" equation. $n = 1$ is the only case where an analytical solution for the image positions can be obtained from the source position. The equation is second order in z and always yields two physical image positions for every source position.

$$\zeta = z - \frac{1}{\bar{z}} \tag{8}$$

To obtain the total amplification, we replace the two image positions in Eq. 5 yielding the simple single lens amplification equation,

$$A = \frac{u^2 + 2}{u\sqrt{u^2 + 4}} \tag{9}$$

The caustic curve in the single lens case consists of a single point at the lens position, while the corresponding critical curve is the Einstein ring. From the discussion in Section 2.1.3 we expect the image positions to approach the critical curve(s) as the source position approaches the caustics. Naturally this also holds true for the single lens case, so that the image positions approach the Einstein ring ever more closely as the source approaches the lens position. If the source covers the lens position completely, it implies that the images must cover the critical curves and form a continuous ring coinciding with the Einstein ring.

2.1.3 Binary lens

Binary lenses take centre stage in this thesis. The binary lens system is only a small physical step away from the single lens system in that the lensing system constitutes two massive objects in relative proximity (within roughly 10 Einstein radii of the primary body away from each other). If the two bodies are further separated, they can generally be considered as two independent single lenses. However, the introduction of the second body presents a step up in the complexity of amplitude calculations and especially modelling and fitting light curves. Binary lens formalism is of great importance to LGM theory as binary lens systems are used as an approximation for planetary systems. It is commonly assumed that a planetary system is well described by considering each of its planets in turn with the central star as an independent binary system. This can be shown to be a reasonable assumption in the majority of cases but is investigated in [22]. The great majority of literature on planet detection by LGM uses binary lens systems as the model for planetary systems. Binaries are also important due to the spectacular light curves observed during caustic-crossing LGM events found with higher mass ratio binaries. If the first caustic crossing is well observed, it is sometimes possible to predict subsequent crossings leading to high quality data and frequent sampling of the crossings. These highly detailed binary light curves have been used to do some novel stellar atmosphere work (e.g. [18]) as well as allowing more lens information to be extracted from the event than is possible with a single lens. More new original results have been produced thanks to this extra information, e.g. [46]. Simple binary geometry leads to an unbelievably large number of theoretically possible light curves and even small changes in the binary geometry can change a given light curve completely. This non-linear dependence of the light curve on the binary geometry is discussed in greater detail in Section 3.3.2.

Binary lens Parameters

In this thesis we will adopt the Einstein radius of the primary lens as the angular distance scale when dealing with multiple lens systems. As mentioned above this is a convention and is adopted partly for its ease in comparing binary systems to single lens systems. For the same reason we put the origin of our axes on the sky at the projected position of the primary lens. The secondary is placed on the positive x-axis. There is no loss of generality by fixing the secondary's y-position. This geometry is illustrated in Fig. 7.

With the above conventions, b remains the impact parameter of the source path to the primary lens, t_0 is the time corresponding to the source at B and m_0 is the un-lensed magnitude of the source star, and thus single lens parameters remain the same. Additional binary parameters are required. θ is the angle between the positive x-axis and the impact parameter vector $\vec{b}$. Note that $|\vec{b}| = b$. a is the angular distance between the primary lens and the position of the secondary lens as projected onto the sky. q is the mass ratio of the secondary to the primary lens.

Binary lens equation

Setting $m_1 = 1$ and $m_2 = q$ in Eq. 1 and using only two lenses, we have

$$\zeta = z - \frac{1}{\overline{z}} + \frac{q}{a - \overline{z}} \tag{10}$$

where the primary lens is at the origin and the secondary lens is on the positive x-axis at distance a. The equation is again normalized and unitless, but in the version presented here there is a slight departure from the standard normalization: the masses do not sum to 1. This convention is introduced so that the natural unit of angular distance becomes the angular Einstein radius of the primary lens mass, not the combined, binary lens mass. The author found this normalization to be more convenient, with the side effect that direct comparison to angles used in most other Microlensing works require conversion by a factor $\sqrt{1+q}$, derived by comparing normalizations.

The amplification can no longer be written as an explicit function of source position as was the case for single lenses. Instead the image positions z_i need to be calculated first from Eq. 10 and then replaced into 5. Unfortunately Eq. 10 cannot be solved analytically for the image position for a given source position. It yields either 3 or 5 solutions, corresponding to image positions. There are a variety of ways to solve this equation for the image positions to obtain amplification and also a variety of ways to calculate amplification without solving it. Methods are discussed in Section 2.2.

Caustics

The caustic curve for a single lens event consists of a single point at the origin, while the critical curve is a perfect circle centred at the origin with radius 1 θ_E. In stark contrast the caustics of binary lenses are complicated, jagged, closed curves, sometimes merging into a single curve and sometimes divided into several disjoint curves. They do not in general coincide with the lens positions. As shown above, Witt has derived an elegant parametric description for the critical curves of any binary lens geometry, Eq. 6. The binary lens equation 10 can then be used to map points on the critical curves to positions on the corresponding caustic curve.

Figure 8 shows some sample binary lens caustics and their corresponding critical curves. There is a variety of caustic curve morphologies (e.g. [47]) and light curves are parametric curves across such complicated caustic geometries, hence a large number of light curves types are allowed by the binary lens model. Some properties of binary caustic geometry are discussed in e.g. [48] and [49] and we will briefly discuss general and asymptotic behaviour of LGM caustic curves.

If the binary is widely separated, the critical curves approach perfect rings around each lens of angular radius equal to the θ_E of each lens i.e., identical to two individual single lenses. As a decreases, the critical curves are deformed until a point is reached where the critical curves merge into a single curve. If we let $a \Rightarrow 0$, the critical curve approaches a perfect circle again, centred at the projected centre of mass of the two

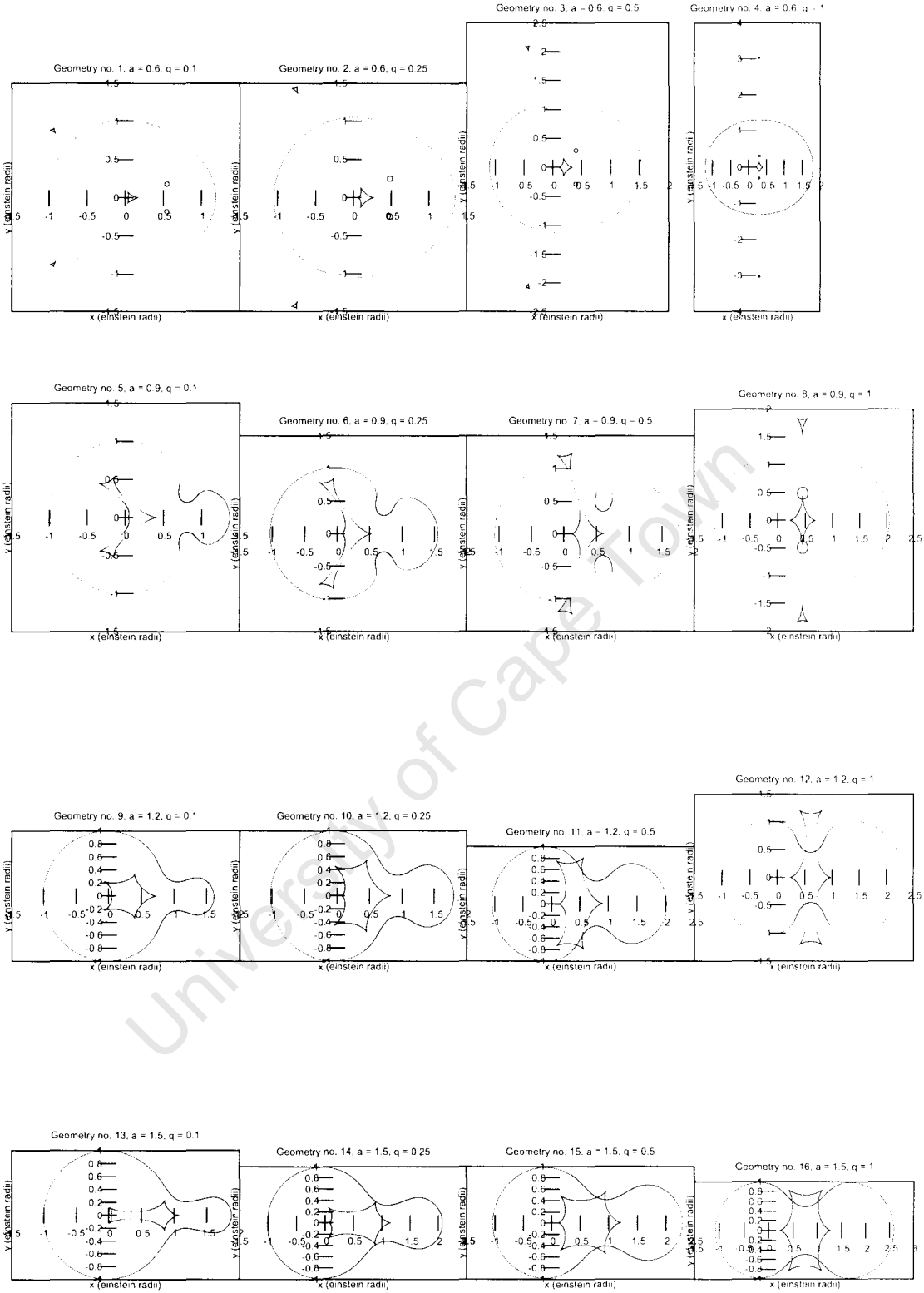

Figure 8: Binary caustic and critical curves for lens systems with various angular separations a and mass ratio q.

lenses and with angular radius equal to the θ_E of a single lens of the combined mass. Schneider and Weiss [50] (for the equal-mass case) and Erdl and Schneider [51] (for the general case) presented analytical derivations of the separation where the merging of critical curves occurs, as well as the corresponding critical curve behaviour.

There is a convention to separate binary events into "strong lensing" binary light curves and "weak lensing" binary events based on whether the source path actually intersects a caustic curve or not. The distinction is useful due to the different conventional fitting techniques that apply to these two types of light curves. Caustics are of central importance to binary light curve analysis and planet detection theory, as all the major features of a binary light curve are generally a direct result of the source's proximity to a caustic curve.

Planet detection

This Section aims to provide a brief review of the theory of detecting extra-solar planets (ESP's) by LGM. There is a possibility of detecting a low mass secondary lens in a binary lens system if the source path happens to cross a caustic curve, which would cause an observable peak in the light curve. Even planetary mass (mass ratios of $q < 0.01$) secondaries are theoretically detectable in this way (e.g. [21], [52], [53]). The mass ratio "cutoff" of $q < 0.01$ is somewhat traditional and in fact not quite correct. A more realistic level is actually around $q < 0.03$ as that would correspond to a ten Jupiter mass planet orbiting a three solar mass star. Calculating planet detection probability is beyond the scope of this thesis and is discussed only briefly, but as planetary LGM light curves are just binary light curves where the mass ratio is small, the techniques developed in this thesis to extract physical parameters from binary light curves should be relevant to planet detection theory. A planetary signature is expected to be a short time scale perturbation on an otherwise normal single lens light curve. The amplitude, shape and duration of the perturbation depends on the binary geometry and mass ratio as well as the way in which the

source path crosses a caustic curve. Planet detection utilizing LGM has seen some remarkable recent results and successes. The first uncontroversial detection of a planet by the OGLE and MOA collaborations occurred in 2003 [2]. A planet of a remarkably low 5.5 Earth masses was detected during event OGLE 2005-BLG-290Lb [54].

If a lensing system does contain a planet, the probability of detecting it is most sensitive to its projected orbital distance from its star. The most favourable such distance is between 0.618 θ_E and 1.618 θ_E from the primary lens. This distance band is called the "lensing zone", a term coined by Bennett and Rhie in [53]. Planets in the mass ratio range from unity down to $q = 10^{-3}$ stand a reasonable chance of being detected. The geometry of the lensing system determines the shape of the light curve and hence the odds for distinguishing a planetary light curve from a normal single lens light curve through the parameter set (b, θ, q, a). Detection probability is critically and non-linearly dependent on these parameters.

Many authors (e.g. [55], [52]) give detection probability as a function of projected orbital radius and mass ratio for a star with a single planet. This probability is generally accepted to be between 10 and 20 per cent for a Jupiter-mass lens orbiting a star like our sun ($q \approx 10^{-3}$). Note that the term "detection probability" is used loosely here, as it refers only to detecting an anomaly in an otherwise normal single lens light curve assuming that the system consists of a primary lens and secondary lens and is undergoing an observed LGM event. Most studies do not take into account the chances of detecting the event in the first place or the difficulty of extracting planet parameters from such a perturbation. Parameter extraction in the form of curve fitting proves to be extremely difficult and is fraught with ambiguity, especially for an incomplete data set. Most of this thesis is dedicated to the extraction of these parameters from binary light curves, although the mass ratio to be considered is in the range $0.1 < q < 1.0$, putting it outside of the planetary range and into the range of larger binary perturbations.

2.2 Binary lens Calculations

Binary LGM light curves are of particular interest, as most planetary systems can be approximated as binaries in the LGM context ([22] calculate a 1 to 15 per cent probability of multiple planets inside the lensing zone for a multiple system) and a lot of science information is potentially available from binary curves. The study and analysis of binary light curves requires an effective means to calculate them. This Section discusses some methods that are currently in use, makes some comparisons and motivates the use of the particular methods adopted for the rest of this thesis.

2.2.1 From lens position to source amplification

LGM photometry measures the total flux as a function of time through an aperture that contains the images of a background star that is being lensed by a foreground object. The aperture also contains other "background" sources of light that dilute the amplification of the source star to some degree. The total flux as a function of time forms a light curve.

Section 2.1 introduced the equations governing a simple Microlensing event and Figure 7 illustrated the seven parameters of the Simple Binary Lens Model (SBLM). We recap in Table 1.

Source, lens and observer are in motion relative to each other, making amplification a function of time. These dynamic effects are most easily dealt with if we consider a static lens and observer and project all movement onto the source position. By taking this approach we divide the calculation into a dynamic part that calculates the source position relative to the lens as a function of time, and a binary geometry part that calculates the amplification of the source as a function of the lens system geometry. Geometrical parameters in the SBLM are a and q, while b, θ, t_e and t_m describe the path of the source. The final parameter m_0 is the un-lensed magnitude of the source.

Table 1: The seven simple binary model parameters (SBLM)

Name	Symbol	Description
Angular Einstein radius crossing time	t_e	The time it takes for the source to cross an angular distance of one Einstein radius
Time of closest approach	t_m	The point in time at which the linear path of source is closest to the primary.
Un-lensed magnitude	m_0	The un-lensed magnitude of the source star
Angular impact parameter	b	The impact parameter between source path to the primary lens in terms of angular Einstein radius.
Crossing angle	θ or "ang" in some diagrams	The angle between the impact parameter and the positive x-axis
Mass ratio	q	The ratio of the secondary's mass to that of the primary
Orbital separation	a	The projected orbital separation between the primary and the secondary in Einstein radii

2.2.2 Approaches to calculating amplification

For any time t, we first obtain the source position as a function of θ, t, b, t_m and t_e. An extension to this simple model is to include a terrestrial observer's motion by projecting it onto the source position, in which case we need to include four new parallax parameters (ψ, ρ, λ and β) in the calculation of the source position relative to the lens. This extension is discussed in Section 2.3.3.

For a linear source movement model which neglects parallax effects, we define

$$t' = \frac{t - t_m}{t_e} \tag{11}$$

where t is simply the observation time. The position of the source is subject only to scaled linear motion

$$\zeta = t' \sin(\theta) + b\cos(\theta) + i\left(-t'\cos(\theta) + b\sin(\theta)\right) \tag{12}$$

where we are keeping for now with the description of source position in the sky

in the complex formulation.

Binary rotation of the lenses could also be included into this equation but are not considered in this thesis. In this simple linear model all movement is contained in Eq. 12 and the lens binary geometry remains static. Including lens binary rotation makes the secondary's position a function of time, leading to dynamic geometry.

We now have source position as a function of time by combining Eqs. 1 and 12. Unfortunately amplification cannot be explicitly expressed as a function of source position for systems with more than one lens. Mapping source position to amplification presents the most challenging part of the amplification calculation and several approaches exist.

The first approach to calculating amplification is to obtain image positions for a given source position and to calculate amplification from the image positions. This is a relatively time-consuming numerical calculation. An alternative approach is to calculate source position from image position which is a very efficient calculation as source position is explicitly stated in terms of image position and the model parameters in Equation 1. The amplification contributed by a specific image position is also explicitly stated. The drawback is that the source position that matches a given image position is not necessarily the source position that we are interested in. This leads to a situation where many thousands of image positions need to be calculated first, before we can map image positions to an arbitrary source position. This method is described in the following Section 2.2.3.

2.2.3 Solving for the source position from an image position

The next Section describes one technique of calculating amplification by mapping from a grid of image positions to their corresponding source positions. Equation 10 maps each image position z onto a single source position ζ. The key to this particular technique is to realise that the amplification of a source is directly related to the ratio of the densities of source points to their corresponding image points, e.g. ([12], [56],[57]). If we create a uniform grid of image positions and then map each to their

corresponding source positions through substitution in 10, the density of source positions gives the amplification at any point. Figure 9 shows an illustration of this process, called "Ray-shooting".

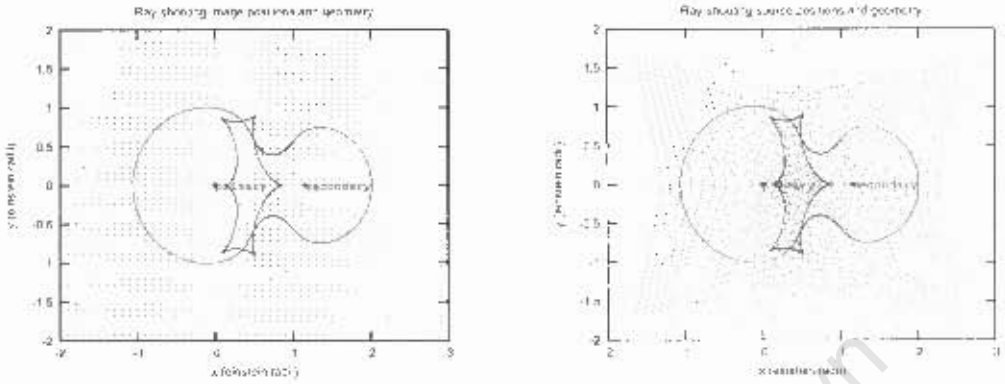

Figure 9: An illustration of ray-shooting. The figure on the left contains a flat distribution of lens system image positions (and the binary lens geometry for illustration). On the right, the image positions have been mapped to their corresponding source positions. The density of source positions is directly proportional to the amplification, as can be seen from the coincidence of caustic curves and ray-shoot maximum density.

Ray-shooting is widely used and offers some particular advantages and disadvantages over other methods, discussed in the next Section.

Ray-shooting

One of the primary advantages of ray-shooting is that it is an extremely simple procedure, requiring no more computational trickery than that of evaluating a simple function repeatedly, Eq. 1. There is an additional advantage over solving for image positions from a given source position. In the latter case, the equation to be solved for image positions grows prohibitively in complexity as additional lenses are added to the system. In contrast, ray-shooting can deal with a large number of lenses (e.g. 10000 in [58]). A term is simply added to the function to be evaluated for each lens in the system. Ray-shooting amplification may also be calculated to machine accuracy by increasing the density of image positions to be mapped to source positions. This accuracy is limited by storage space and data search constraints on the calculating

machine. These very requirements unfortunately greatly reduce the practical use of ray-shooting as a modelling technique in GM. A ray-shoot map can be constructed and reused for any source path through a lensing system, but a new map needs to be constructed each time the secondary lens position or mass ratio changes. These maps take a long time to calculate and consist of thousands of data points, depending on the precision required. In addition to the calculation time required, maps also require a large amount of storage space. A modeller attempting to vary the binary parameters a and q for fitting a model to data then faces the choice of either having a large amount of pre-calculated maps available, one for each (q, a) pair at some sampling frequency in q and a, or to calculate a new map each iteration if (q, a) is allowed to vary continuously. There is another disadvantage to the ray-shooting technique: to obtain the actual amplification of a source at a given position, the density of source points needs to be calculated. There are relatively efficient means of computing this density, such as convolving the source's brightness profile with that of the density map by means of a fast Fourier transform, but this is an additional computational burden. Simple ray-shooting was considered unsuitable for the fitting procedures in this project for the above reasons as well as the following considerations:

- Number of points in the map required for small source size R_s is very high if good precision is required.

- In this project I was primarily concerned with binaries, where the source to image method is effective.

- The number of (q, a) maps required to fit a general binary light curve was enormous, leading to huge storage space requirements. Note that others have lived with this restriction to produce good results, e.g. [3].

2.2.4 Solving for image positions from the source position

Obtaining the projected positions of the images of a source on the sky leads to a conceptually more direct route to overall amplification than the ray-shooting technique. Image positions are related to that of the source by Eq. 10, and Eq. 5 gives amplification as a function of image positions. The central difficulty is in mapping the source positions to image positions. Unfortunately, numerical methods have to be used as there is no general analytical solution to the lens equation for systems with two or more lenses. Most numerical methods for root-finding progress from an initial guess to a given precision by iteration, if all goes well. There are fairly efficient methods available in the literature with various strengths and weaknesses and this Section deals with selecting an appropriate method. The main criterion for success is accuracy: amplification has to be correct to within at least 0.001 mag in order to be used for fitting purposes, as the observational noise on LGM light curves is of the order of 0.01 mag. Accuracy is also required due to the sensitivity of fitted model parameters to the light curve. Robustness is another criterion.

The "best" algorithm is then simply the fastest one that meets the above criteria.

Complex polynomial with Laguerre's method and deflation

This method proved to be very effective in solving the binary lens equation. Following [44], Eq. 10 is rewritten as a fifth degree complex polynomial and then solved by any of a number of methods that apply specially to polynomials. The polynomial is obtained by multiplying the equation by its complex conjugate and gathering coefficients for powers of z, leading to

$$\sum_{i=0}^{5} z_i c_i = 0 \tag{13}$$

where

$$c_0 = -\zeta a^2, \tag{14}$$

$$c_1 = 2\zeta aq + \zeta a^3 - 2\zeta\overline{\zeta}a^2 - qa^2 + 2\zeta a. \quad (15)$$

$$c_2 = 4\zeta\overline{\zeta}a - \zeta\overline{\zeta}^2a^2 - \zeta - \zeta q^2 + aq - 2q\zeta + \overline{\zeta}a^2 - qa^2\overline{\zeta} + 2\zeta\overline{\zeta}qa + q^2a + \zeta\overline{\zeta}a^3 - 2\zeta a^2 - \zeta qa^2. \quad (16)$$

$$c_3 = \zeta a - 2a^2\zeta\overline{\zeta} + qa^2 - 2a\overline{\zeta} - 2q\zeta\overline{\zeta} - 2\zeta\overline{\zeta} + aq\zeta + \overline{\zeta}^2a^2 + 2a\zeta\overline{\zeta}^2 - \overline{\zeta}a^3, \quad (17)$$

$$c_4 = q\overline{\zeta} + \overline{\zeta} - \zeta\overline{\zeta}^2 - 2a\overline{\zeta}^2 + 2\overline{\zeta}a^2 + \zeta\overline{\zeta}a - aq. \quad (18)$$

$$c_5 = -a\overline{\zeta} + \overline{\zeta}^2. \quad (19)$$

Laguerre's method (e.g. [59], first applied to Eq. 13 in [60]) is useful as it does not require an accurate initial guess for the roots of the polynomial: it starts iterations from the origin and this turns out to be close enough for convergence. The particular version of the algorithm used in this thesis (from Press [59]) implements polynomial deflation so that each root that is found is factored out of the polynomial in order to simplify the task of finding the remaining roots. Critically, deflation also guarantees that the roots found will be unique, which avoids the enormous problem of converging repeatedly to the same root or failure to find multiple, closely-spaced roots. Laguerre can solve any polynomial, whether the coefficients are complex or not. The method is actually fairly simple and a concise description can be found in [59].

In practice the roots found after using deflation often need "polishing" by another method as a small error is introduced into the remaining polynomial when each root is factored out in turn. Laguerre's method itself can be used for this, but Newton's method works well in this case because it converges quadratically if a fairly accurate starting point is already known.

A complication of the polynomial method is that it always yields five solutions, whereas the original binary lens equation has either three or five, depending on geometry and source position. That means that the polynomial method yields two spurious solutions under certain circumstances. The simplest way to eliminate the

non-physical solutions is to test all polynomial roots in the original lens equation to see whether they are true solutions or merely solutions to the polynomial equation.

Newton's Method

Newton's method, a.k.a. Newton-Raphson (N-R), is highly effective at finding solutions to Eq. 10 when it has an initial guess in close proximity to the actual solution. This is often the case when a small change in the parameters has been made from the previous solution, such as a small increment of time added to t.

It has the advantage that it can be applied directly and therefore the problem of spurious solutions presented by turning the binary lens equation into a polynomial is avoided. Unfortunately, the method is of little use by itself when no accurate initial guesses for the roots are available, or when different roots are not widely separated. in which case Newton often converges to the same root from two different starting positions. All is not lost as [59] (again) describes a variant of Newton's method that converges to a root from an initial guess that can be much further away than the accurate starting points required by the original version. This makes N-R well-suited to calculating additional points on a light curve where image positions from the previous time point are available as initial guesses. If N-R fails, the previous points can be discarded and the image positions found by the reliable but slower Laguerre's method instead. Of course, the images belonging to the source at the first time point of the light curve still have to be calculated by a different method such as Laguerre.

Asada's Real Polynomial

The method of choice for solving the binary lens Eq. 10 up to about 2002 appears to have been the complex polynomial method described above. Asada made a breakthrough in binary lens modelling by rewriting the binary lens equation, i.e. the relation between source position and image positions as a function of lensing geometry, as a fifth-degree real polynomial with real coefficients [33].

In Asada's regime one solves for $\tan(\phi_i)$ as the roots to a fifth-degree real polynomial, where ϕ_i is the angle between the position of image i and the positive x-axis, with the coordinate origin on the position of the primary.

Asada defines the relationship between source and image angles $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ as

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \Big(\nu_1 \frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|^2} + \nu_2 \frac{\boldsymbol{\theta} - \boldsymbol{\ell}}{|\boldsymbol{\theta} - \boldsymbol{\ell}|^2}\Big). \tag{20}$$

He defines

$$\nu_1 = \frac{M_1}{M_1 + M_2}, \nu_2 = \frac{M_2}{M_1 + M_2}, \boldsymbol{\ell} = \frac{\boldsymbol{L}}{D_{\mathrm{L}}\theta_{\mathrm{E}}}. \tag{21}$$

ν_1 and ν_2 are the masses of the primary and secondary lenses, and ℓ is the binary separation vector.

Note that Asada uses the Einstein radius of the total mass of primary plus secondary as his unit of angular radius, whereas this thesis uses the mass of the primary only. Also, lens masses ν_1 and ν_2 are normalized to the total mass. In practice the unit difference requires a minor conversion from the units of this thesis to Asada's, pre- and post-calculation of amplification.

Eqs. 22 and 23 show Asada's polynomial for $\tan(\phi)$ and its coefficients:

$$\begin{aligned} &(a_5 \tan^5\phi + a_4 \tan^4\phi + a_3 \tan^3\phi + a_2 \tan^2\phi \\ &+a_1 \tan\phi + a_0) \tan\phi = 0, \end{aligned} \tag{22}$$

where, if we define the coordinates for the source, image and separation vectors as

$(\beta_x, \beta_y) = (\rho\cos\varphi, \rho\sin\varphi)$, $(\theta_x, \theta_y) = (r\cos\phi, r\sin\phi)$ and $(\ell_x, \ell_y) = (\ell, 0)$, respectively, with ρ, r and $\ell \geq 0$ we have

$$\begin{aligned} a_0 &= \nu\ell\rho^3 S^3, \\ a_1 &= \rho^2 S^2 + 2\ell\rho^3 C S^2 - \ell^2(2\rho^2 S^2 - \rho^4 S^2) \end{aligned} \tag{23}$$

$$
\begin{aligned}
& -2\ell^3\rho^3 CS^2 + \ell^4\rho^2 S^2 \\
& -\nu(5\ell\rho^3 CS^2 - 4\ell^2\rho^2 S^2). \qquad (24)
\end{aligned}
$$

$$
\begin{aligned}
a_2 \;=\; & -2\rho^2 CS - 4\ell\rho^3(C^2 S - S^3) \\
& +\ell^2(4\rho^2 CS - 2\rho^4 CS) + 4\ell^3\rho^3 C^2 S - 2\ell^4\rho^2 CS \\
& +\nu\Big[\ell(2\rho S + 8\rho^3 C^2 S - 2\rho^3 S^3) \\
& \quad - 10\ell^2\rho^2 CS + 2\ell^3\rho S\Big]. \qquad (25)
\end{aligned}
$$

$$
\begin{aligned}
a_3 \;=\; & \rho^2 + \ell(2\rho^3 C^3 - 10\rho^3 CS^2) \\
& +\ell^2(-2\rho^2 C^2 + 2\rho^2 S^2 + \rho^4) - 2\ell^3\rho^3 C + \ell^4\rho^2 \\
& +\nu\Big[\ell(-2\rho C - 4\rho^3 C^3 + 6\rho^3 CS^2) \\
& \quad + 6\ell^2\rho^2 C^2 - 2\ell^3\rho C\Big] \\
& +\nu^2\ell^2, \qquad (26)
\end{aligned}
$$

$$
\begin{aligned}
a_4 \;=\; & -2\rho^2 CS + 8\ell\rho^3 C^2 S - \ell^2(4\rho^2 CS + 2\rho^4 CS) \\
& +4\ell^3\rho^3 C^2 S - 2\ell^4\rho^2 CS \\
& +\nu\Big[\ell(2\rho S - 4\rho^3 C^2 S + \rho^3 S^3) \\
& \quad - 2\ell^2\rho^2 CS + 2\ell^3\rho S\Big]. \qquad (27)
\end{aligned}
$$

$$
\begin{aligned}
a_5 \;=\; & \rho^2 C^2 - 2\ell\rho^3 C^3 + \ell^2(2\rho^2 C^2 + \rho^4 C^2) \\
& -2\ell^3\rho^3 C^3 + \ell^4\rho^2 C^2 \\
& +\nu\Big[-\ell(2\rho C + \rho^3 CS^2) + 2\ell^2\rho^2 C^2 - 2\ell^3\rho C\Big] \\
& +\nu^2\ell^2. \qquad (28)
\end{aligned}
$$

C is simply $\cos(\phi)$, S is $\sin(\phi)$.

After obtaining all values of $\tan(\phi_i)$, each in turn is substituted into

$$
r\cos\phi = \frac{\tilde{R}_1(\tan\phi)}{\tilde{R}_2(\tan\phi)}, \qquad (29)
$$

where

$$\begin{aligned}\tilde{R}_1(\tan\phi) &= (\ell^2\rho C - \ell\rho^2 C^2 - \nu\ell + \rho C)\tan^2\phi \\ &\quad -\rho S(\ell^2 - 2\ell\rho C + 1)\tan\phi - \ell\rho^2 S^2. \qquad (30)\\ \tilde{R}_2(\tan\phi) &= \rho(C\tan\phi - S) \\ &\quad \times\left[\rho S + 2(\ell - \rho C)\tan\phi - \rho S\tan^2\phi\right]. \qquad (31)\end{aligned}$$

in order to obtain the image x and y position on the sky from

$$(x, y) = (r\cos(\phi), r\cos(\phi)\tan(\phi)) \qquad (32)$$

The Jenkins-Traub algorithm is used to solve this polynomial. The procedure is still not analytically tractable but Asada's formulation provides numerous advantages over the complex polynomial method:

- A real polynomial is much easier to solve than a complex one permitting the use of efficient algorithms such as Jenkins-Traub.

- Real roots are all genuine solutions of the lens equation. There are three real roots if the source is outside of a caustic region and five when inside, corresponding to the number of images present in each case. This is in contrast to the complex polynomial method which always provides five complex roots. Some of those roots correspond to physical images whereas two are spurious side-effects of the conversion of the binary lens equation into a complex polynomial. In practice all roots of the complex polynomial have to be substituted back into the binary lens equation to check whether they correspond to genuine images.

- Asada's formulation does not require complex arithmetic, easing the coding burden.

Table 2: Binary model parameter ranges for accuracy checks

Parameter	Minimum	Range
$a(\theta_E)$	0.6	1.6
$\theta(^\circ)$	0	360
$b(\theta_E)$	10^{-3}	1.0
q	10^{-5}	1.0

2.2.5 Quantitative Comparison

Several methods of obtaining amplification from source position were discussed above and this Section compares them based on the speed and accuracy of the method as implemented.

Internal consistency

The comparison data were obtained simply by calculating a set of 1000 random binary lens LGM light curves, utilising each method in turn. The set of event parameters was generated once and thereafter reused, so that all methods calculated light curves for the same set of random model parameters. Each light curve contained 400 data points. The first test was for "internal consistency" of each calculation method, i.e. whether the same calculation always generates the same answer. To measure consistency, each point on a light curve was calculated ten times. The standard deviation of the magnitude for each curve point was calculated and the maximum such standard deviation for the entire curve was saved. Event parameters were selected from a flat distribution over the ranges given in Table 2. The range chosen mostly coincides with the ranges used for events throughout this thesis but includes smaller q and smaller b in order to challenge the algorithm. Generally, the ranges presented the region of maximum interest where binary effects are most pronounced.

Point source calculations were considered, for which the most suitable methods were the complex polynomial- and Asada-type calculations.

Results In the ranges specified in Table 2, both methods consistently returned a maximum standard deviation over any given light curve of 5.26836×10^{-9}, a number which is assumed to be machine accuracy for the calculation performed. This was sufficiently low to allay fears of internal inconsistency in point lens calculations.

Inter-method consistency

Algorithms could be compared with one another in a similar way. The only two serious contenders for point source calculations were the complex polynomial and Asada methods. Light curves were calculated for the same set of model parameters, once by each algorithm. The absolute value of relative difference between these two methods was calculated for each point on a curve and the maximum difference over an entire curve was recorded. The distribution of this measurement is plotted in Figure 10.

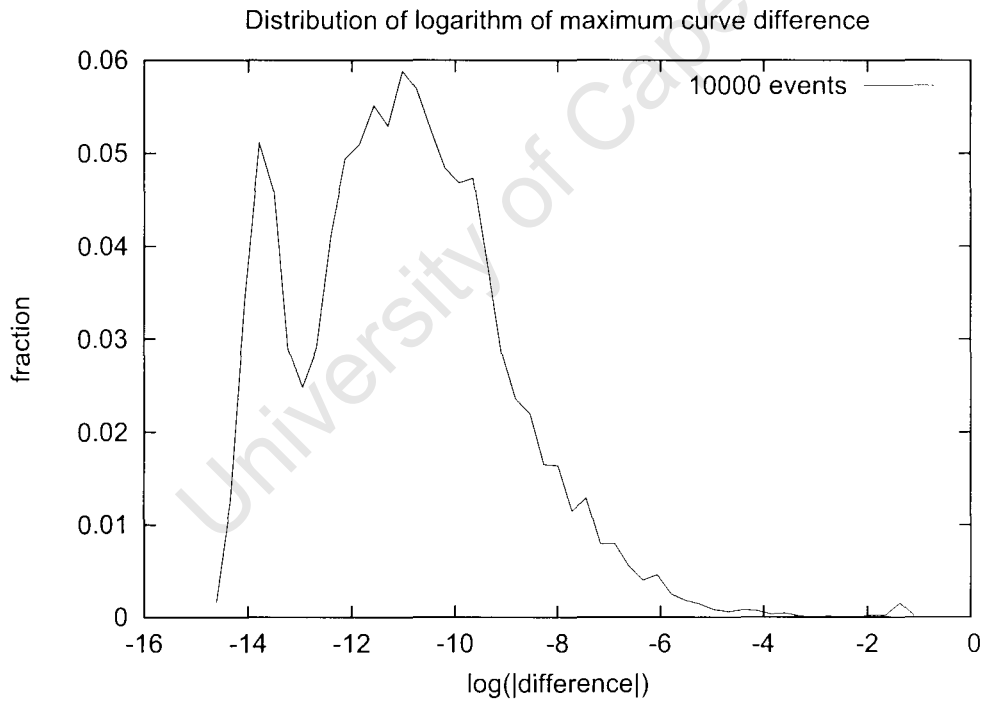

Figure 10: Distribution of absolute value of relative error between the complex polynomial and Asada's methods for point source binary lens magnitude calculations ($\frac{mag(b)-mag(a)}{mag(a)}$)

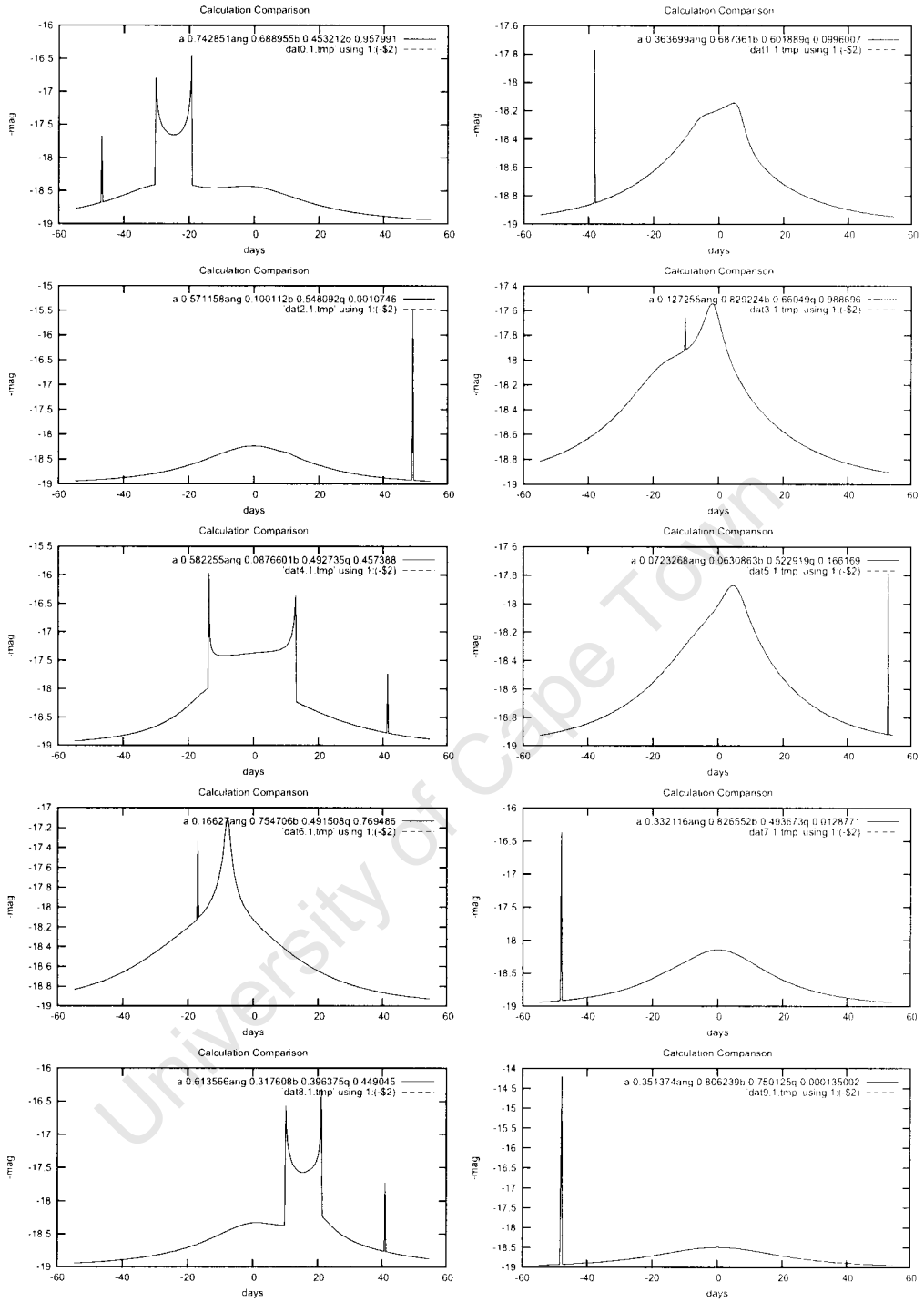

Figure 11: Ten sample plots of curves with identical parameters calculated by Asada's method and the complex polynomial method, plotted on the same axes. The sample consists of curves that differed at least one data point by more than one per cent. It appears from these plots that differences are only likely at extreme amplification during caustic crossings where the point source approximation will have failed in any case.

Figure 10 shows that a small number of light curves contain at least one point of difference of more than a per cent. These curves are re-plotted in Figure 11. In each and every sample case investigated, the difference occurred at the very peak of a caustic crossing, where the point source approximation will have failed in any case.

The above investigation instills a degree of confidence that single lens calculations are correct for either type of method used for binary lens calculations in this thesis. Of course, it may be that both methods are incorrect or that a coding error or incorrect parameter translation causes a consistent systematic error. This scenario can be investigated further by consideration of yet another reference calculation, preferably from an external source.

Speed

If accuracy is one aspect of the calculation method choice, speed is the other. Some care was taken to optimise both the complex and Asada algorithms and implementations. Figures 12 and 13 are two call-tree diagrams, produced by the excellent open-source tool VALGRIND [61], that show the execution time in profiler "counts" for both algorithms.

Call-tree diagrams simply plot the path taken through source code during the execution of a program. The timing or "count" at each node include the total count of all nodes beneath it as well as its own. It is thus a simple graphical display of how much time is spent in each part of a program and its subroutines.

To minimize dependency on hardware configuration, the numbers should be considered in a strictly relative sense. A simple comparison shows that this implementation of the Asada algorithm outperformed the implementation of the complex polynomial method by almost a factor of ten in this case. The Asada source amplification calls (for a number of curves and source positions) take about 150 million counts. Of these, about half are spent on peripheral calculations and half in the

Jenkins-Traub root finder which calculates $\tan(\varphi)$ in this implementation ("JenkinsTraub::rploy"). In the complex case, the total "CalAmp" count is 1400 million, 25 per cent of execution time is spent on finding the complex polynomial roots by Laguerre's algorithm ("SourcetoImageApprox" and children), and a further 75 per cent on polishing these roots ("NewtonImages"). Clearly there is an argument for using unpolished roots to save CPU time, but remember that the algorithms show similar accuracy with the settings as tested.

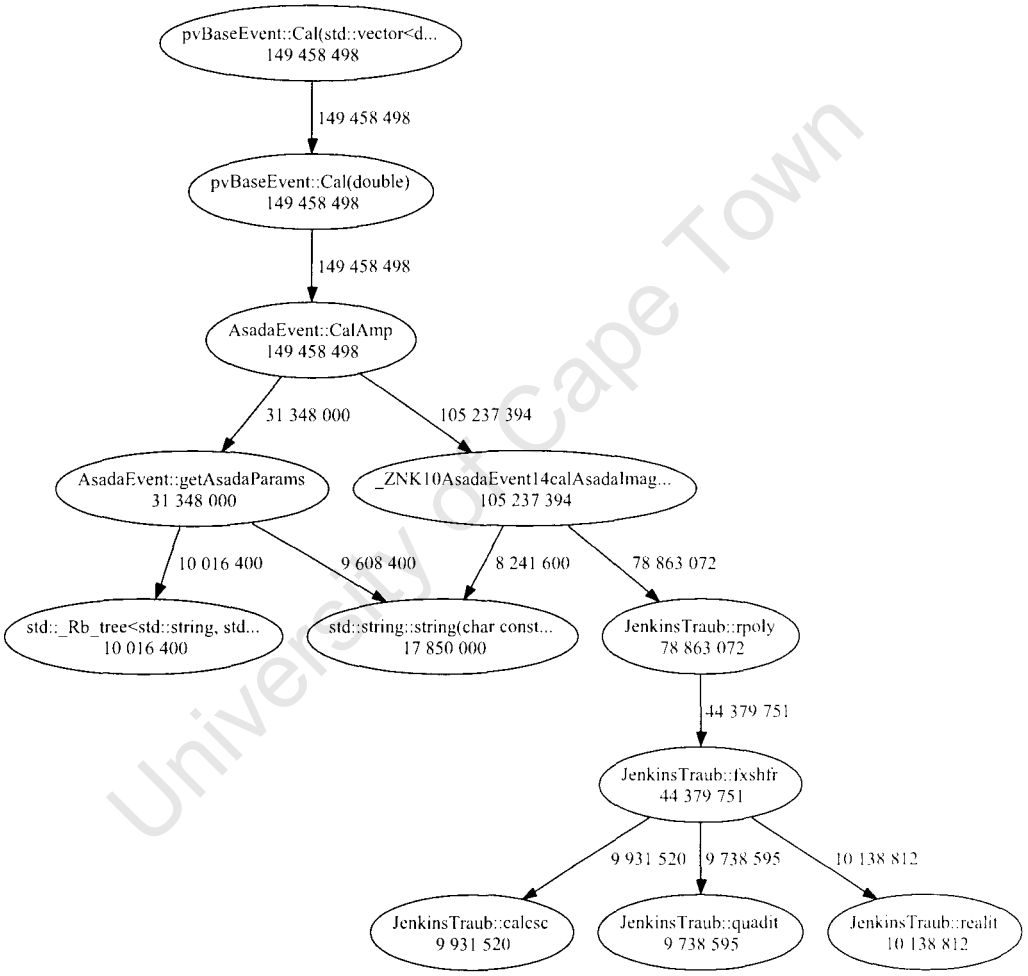

Figure 12: Call-tree with "counts" for the Asada-based amplification calculation algorithm. The call to calculate amplification for a given source position is "CalAmp", which is resposible for 150 million counts.

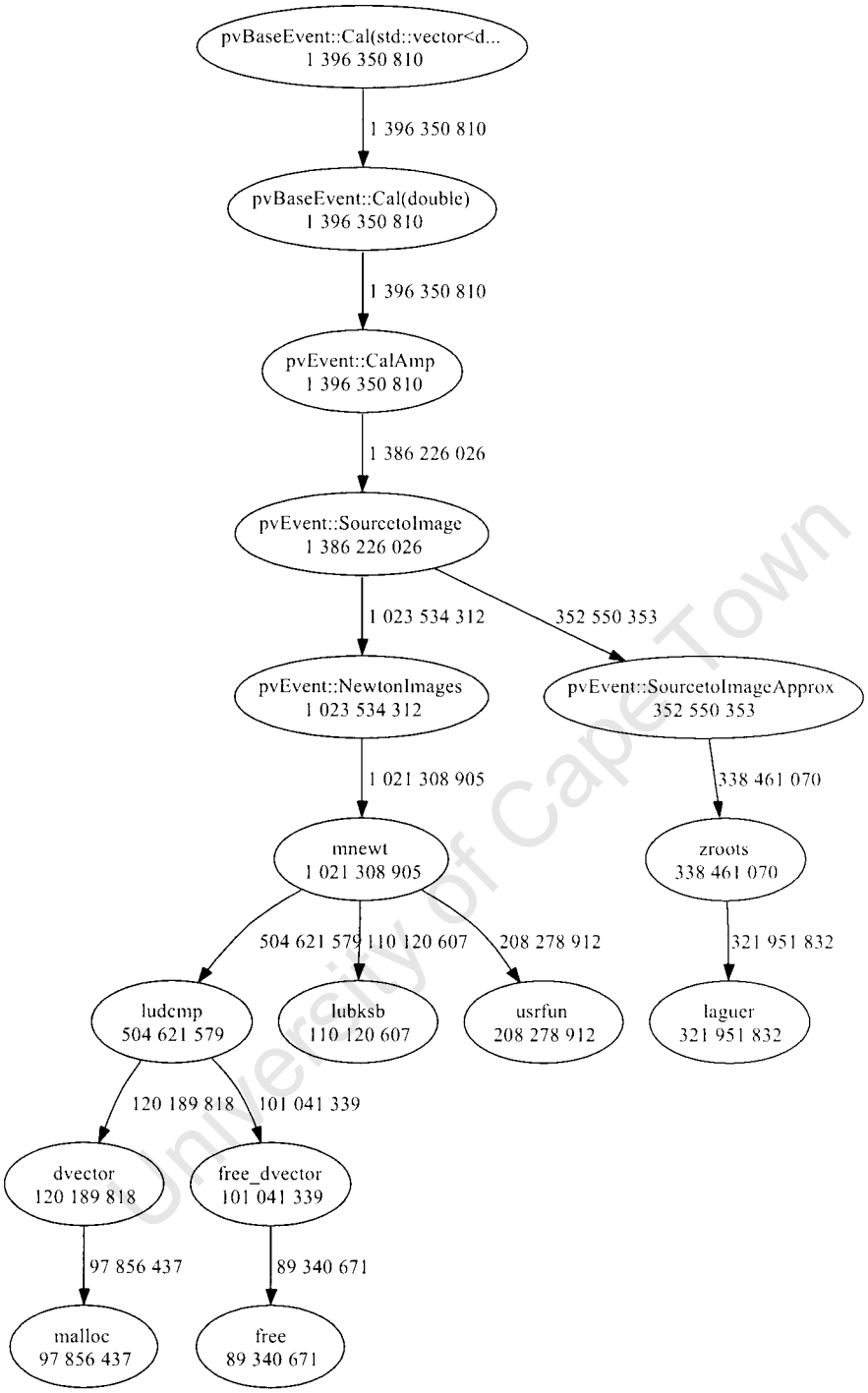

Figure 13: Call-tree for the complex-based amplification algorithm. The call to calculate amplification, "CalAmp" is responsible for 1400 million counts, an order of magnitude more than the Asada method for the equivalent calculation.

Final choice of algorithm

The Asada implementation was thus chosen for all subsequent point source calculations based on its comparable precision but faster execution time.

2.3 Extending the standard model

In Section 2.2 the foundations were laid for the binary lens, point-source, rectilinear-source-motion calculations, the "SBLM". The seven parameters in this model are the minimum required to build a binary gravitational lens model, but there is room for extending it to include additional effects. Indeed, there is little motivation to stop at the seven parameters we have examined so far except for a bias towards simpler calculations! The principle that drives the inclusion or otherwise of any imaginable effect should be Occam's razor. Hence in 8.1.5 the consequences of neglecting more complicated models is gauged.

Light curves that contain caustic crossings, or where the source approaches a caustic, require extensions to the standard model to take the source's finite size into account. This is clearly a shortcoming of the standard model which predicts infinite amplification for a point source.

Events with precise timing information or even just a particular geometry need to take non-rectilinear source motion into account as well, whether this projected movement is due to a rotating lens system or the Earth's movement. Some of these effects are easy to include into the model and require only a small increase in computation time above that required for the standard model, but some are not. The challenge posed by the effects to be discussed in this Chapter is that they are often negligible but occasionally of utmost importance, so that including them can lead to an entirely different light curve. Even when the extra effects are easy to calculate, they are generally not independent. Each additional parameter added therefore adds one dimension to the search space as far as fitting a model to the curve is concerned. The number of models to be tested in a grid-based search of the parameter space rises rapidly with the number of dimensions when their effects on

the model are dependent on each other.

The standard model consists of the minimum number of parameters required to encompass the basic geometry and dynamics of a binary lens system and source. In this Section, we plan to address the issues raised here by considering a number of "optional" extensions to the standard model. Extensions are assessed (in Section 8.1.5) and/or included in our fits based on their importance as an observable effect, which we will attempt to measure (in a fairly crude way) using a $\Delta\chi^2$-metric to compare extended light curves with standard light curves for a variety of generated events.

2.3.1 Resolved source effect

The finite source effect unfortunately requires a tedious, numerical calculation which takes many times as long as the point source equivalent. In this Section the basic theory is set out, as well as some methods of calculation.

Theory

This effect is conceptually simple: a real source is a disc of light (of varying intensity across the disc) with finite area. Different parts of the source are amplified by different factors and the total amplification is simply the integral of amplification over the entire source. Even though the amplification of the parts of the source on a caustic curve is infinite, the area of the source that is exactly on top of the one-dimensional caustic curve is an infinitesimal fraction of the total source luminosity.

The finite source scenario is illustrated in Figs. 14 and 15, in which three sources of different sizes cross a caustic structure along the same source path to give three different light curves. The basic effect of R_s on light curves is to smooth peaks out, making them broader and less pronounced with increasing source size, as one would expect through dilution of the caustic effect. A further 20 example R_s-affected light curves are shown in Figure 52

An optimistic modeller might think that the effect can be ignored except in regions of high amplification, i.e., peaks in the light curve due to caustic crossings.

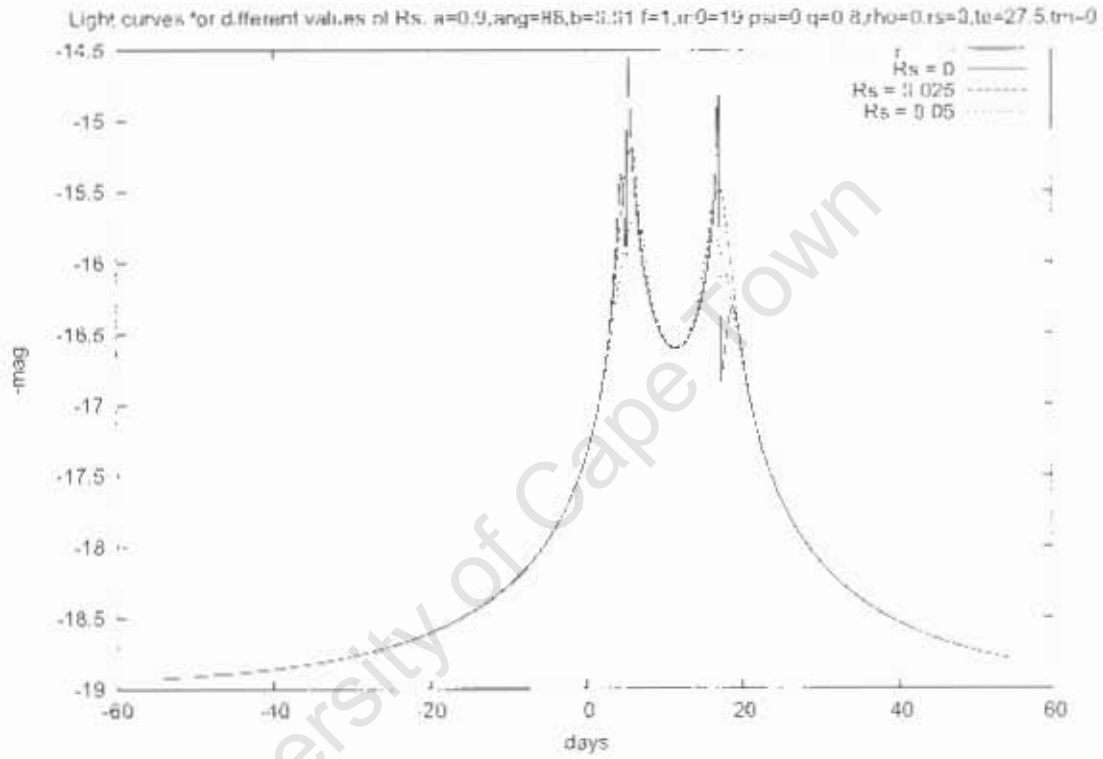

Figure 14: Three light curves, each based on the same standard model parameters, but with the addition of a resolved source (R_s) of varying angular radius in units of θ_E

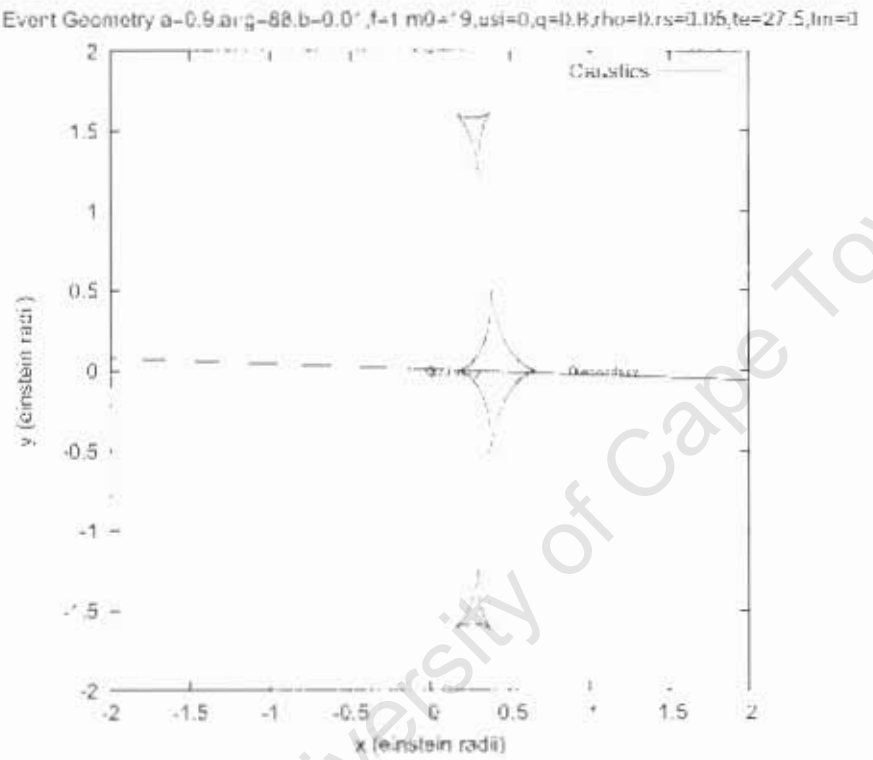

Figure 15: The binary geometry and caustic curves of the events generating the three light curves shown in Fig. 14

The need to include finite source effects is investigated in Section 8.1.5 where it is shown that the effect becomes significant from a $\Delta\chi^2$ point of view with resolved source sizes as small as $R_s = 0.005\ \theta_E$. While it is true that areas of peak amplification such as caustic crossings and approaches are the only regions affected by resolved source size, these regions can constitute a major part of the light curve. An example of this scenario is given in Figs. 16 and 17 where a source travels parallel to a caustic curve. A small source does not cross it and is only mildly amplified, while a resolved source covers the caustic and has its limb amplified by a large factor.

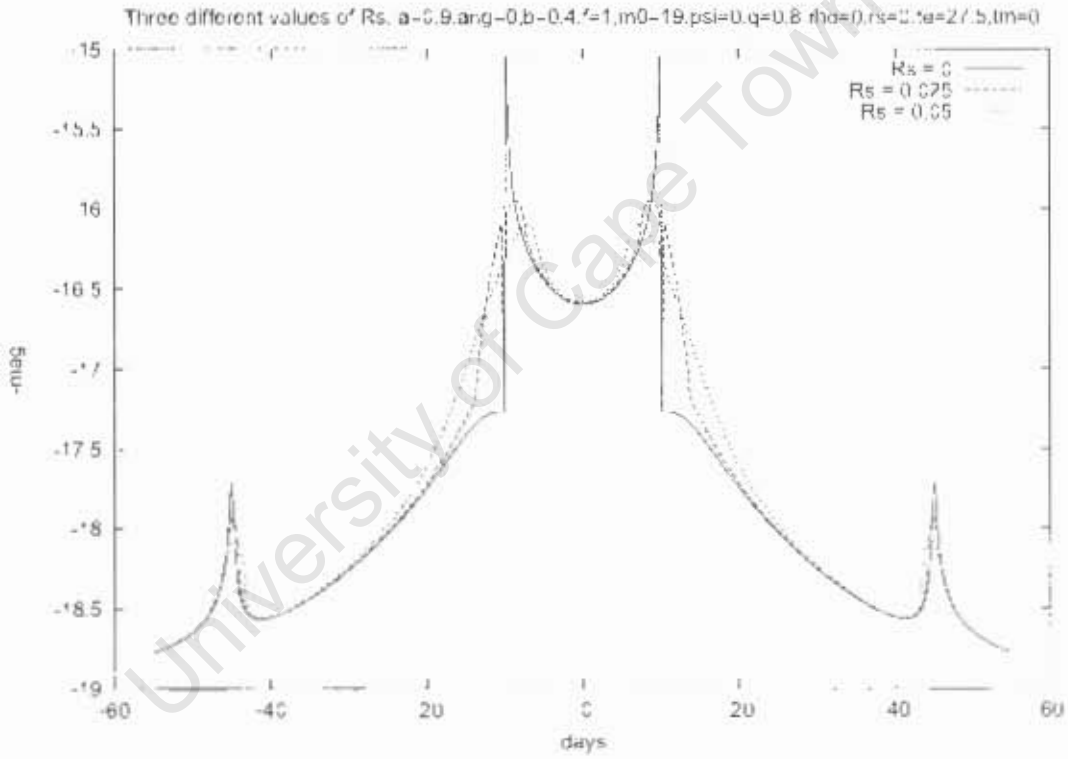

Figure 16: A binary event where the finite source affects a large region of the light curve.

The resolved source effect is also detrimental to the use of LGM events for the detection of extra-solar planets, e.g. [62].

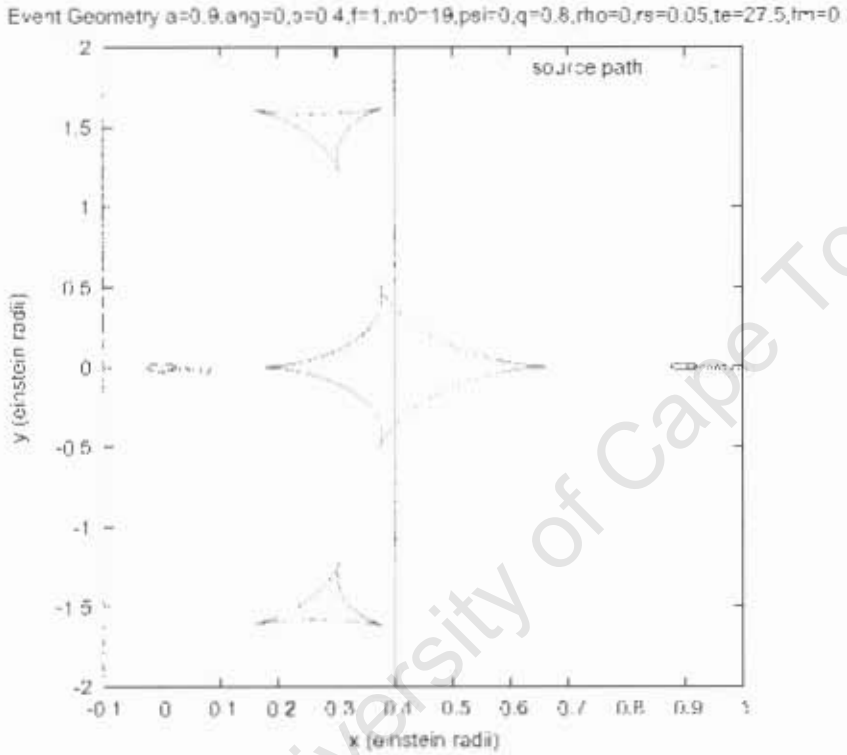

Figure 17: The binary geometry and caustic curves of the R_s-affected curve in Fig. 16

Occurrence

It is possible to model the expected distribution of finite source sizes after making various assumptions about the lens and source population, Galactic models and the like. The purpose of looking at the distribution of finite sources, and hence the distribution of R_s, in this thesis is to determine whether the parameter needs to be included in a realistic model. For this purpose it is arguably sufficient to look at the distribution of R_s as observed in actual events to date. One source of such data is the PLANET 5-year results which places constraints on the number of planetary companions in the observed stellar population. This study also provides the most reliable fits for R_s due to the high sampling frequency on PLANET light curves. The distribution of upper limits to R_s from [3] is shown in Figure 18.

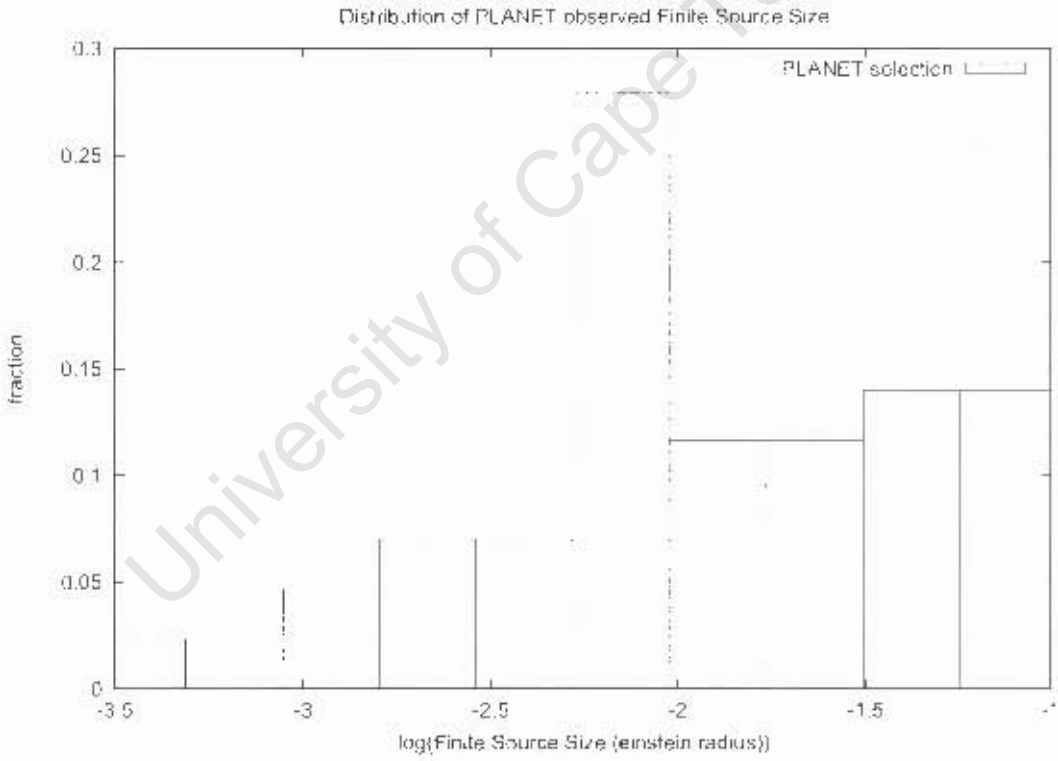

Figure 18: Distribution of upper limits to R_s from PLANET [3] for events meeting PLANET criteria.

PLANET points out that due to a selection effect the actual distribution of R_s

could be fairly inaccurate. Qualitatively the answer to the question on inclusion posed here remains the same, which is that effects of resolving the source need to be taken into account in the majority of cases when modelling realistic light curves. A large number of sources in the PLANET data have resolved angular source radius in the region of $0.01\theta_E$, placing them well to the right in the scatter plot shown in Fig. 53, which implies average values of $\frac{\Delta\chi^2}{d.o.f}$) well in excess of 1. Fig. 53 and those like it in Section 8.1.5 compare light curves created by the simple 7-parameter binary lens model to those created from an extended model. The comparison is made by way of the $\Delta\chi^2$-measure.

Calculation methods

Methods of calculating amplification of a finite source can be divided into two categories: those that integrate amplification over the source surface, which is the direct application of Eq. ??, and those that integrate over the images instead.

Source integration seems a more direct method but care must be taken in the handling of infinite amplification when the source profile intersects a caustic. This complicates numerical methods attempting direct source integration. For example, any finite summation method that relies on sampling the source amplification in the region of a caustic which does not take the caustic boundary into account, introduces an unbounded error into the integration due to the infinite amplification on the caustic curves.

An alternative is to integrate over the images themselves, as amplification can be obtained by dividing the flux of the images by the flux of the source, which is a simple numerical summation exercise. This method was adopted for this thesis, mainly due to its simplicity, and is described further below.

Image integration

This Section describes the method of choice adopted for finite source calculations. It is a version of image plane integration based on that of [53] but adapted to use a recursive flood-fill algorithm described below.

The idea is to start with the set of all images that correspond to the centroid of the source. In fact the images of any point contained within the finite source profile will do. These image points call each surrounding pixel (also in the image plane) recursively. When a pixel is called, it maps itself back to the source plane to check whether it maps onto the source by Eq. 1. If so, it adds itself to the integration, weighted by the brightness of the source at the mapped radius from the source centroid and calls all surrounding pixels to perform the same check. If not, the failed pixel does not call any other pixels.

Any source brightness profile can be accommodated in this way at minimal calculation cost. This is an extremely simple and quite efficient way to perform image plane integrations, as the expensive source-to-image calculation is performed only once to find pixels that are guaranteed to map back onto the source. Thereafter, image-to-source mappings are calculated which are computationally cheap.

The algorithm does suffer from some pitfalls. The size of an integration element in the image plane needs to be calculated before the calculation commences. If the adopted pixel size is too large, the calculation will be inaccurate, while a pixel size that is too small leads to a long calculation time. Unfortunately the required pixel size depends on the amplification, which is exactly what we are trying to calculate in the first place.

To avoid this chicken-and-egg scenario, various methods were tested to estimate amplification in order to find a reasonable pixel size. The simplest and probably the most reliable method was just to base the pixel size for the next point in a light curve on the size of the previous point's pixel size, adjusted to correct for the error in the previous point's size. Pixel size can also be based on a point source amplification which is available in any case as one source-to-image mapping is required to find the first image positions for the recursive integration. This is not ideal, as resolved source calculations are mostly required in regions of high amplification where the point source amplification tends to infinity.

In all cases, a final check can be made to ensure that the pixel size, and thus

the number of integration elements and the accuracy is within an acceptable range. Calculations with too few pixels are rejected and recalculated with a smaller pixel size. Calculations that are taking too long can be terminated and restarted with a larger pixel size.

Unfortunately, this method suffered an additional shortcoming that was not taken into account during calculations in this thesis. Dominik [63] showed that it is possible to miss the initial, seed pixel in a disconnected image altogether if the source overlaps a caustic region but its centroid falls outside the caustic region, and the source does not include a caustic cusp. In this case, the flood-fill algorithm presented here would miscalculate the total amplification. Fortunately, this error had no bearing on the results of this thesis as a point-source model was used throughout. The plots and distributions in the Future Work Section 8.1.5 should be marginally affected, but not the conclusions.

Alternative methods

[63] provides details of an efficient algorithm using Green's method to turn the integral over the image surface into a one-dimensional integral around the borders of the images.

Ray shooting calculations were discussed in Section 2.2.3 above. Ray shoot maps have the huge advantage that they can be used to produce light curves for resolved sources simply by convolving by the source profile. Convolution can of course be performed efficiently using the FFT algorithm. The drawback is the long calculation time required to produce the maps, as well as the high precision required for small source sizes: the smaller the source, the higher the density of points that are required to maintain a given precision.

Effects on binary fitting

The resolved source effect presents a challenge to LGM light curve modelling and fitting. It introduces a new variable into amplification calculations which cannot in general be fitted for independently and requires many times more calculation time

than the point source approximation requires. The most effective way to approach the problem, barring miraculously effective new calculation methods, is to try and separate the resolved source effects from the calculation of the other GM variables. For example, if the light curve is modeled in regions where the resolved source has little effect on the light curve, this calculation could be done using the point source approximation. For the majority of curves, a finite source affects only the caustic crossing regions of the light curve, corresponding to the high amplification peaks. There are notable exceptions, for example where a source approaches or crosses a caustic cusp or where a source's path runs parallel to a caustic curve as demonstrated in Figure 16. Given an observed light curve, it is not possible to exclude a resolved source a priori, and the validity of fitting a model to an LGM light curve by first approximating the event by a point source and then including the effect in a final fit is investigated in Section 8.1.5. Figure 53 in that Section shows the consequences to $\Delta\chi^2$ of neglecting the resolved source effect over the parameter ranges described in Table 4.

2.3.2 Blending

The standard scenario assumes that the background source of light that is lensed is the only source of light during a lensing event. This assumption is broken in all lensing events to some degree. Background, un-lensed light is always present in the crowded fields in which GM observations take place. This effect is known as "blending". A luminous lens can also contribute to blending and in fact some workers estimate that the majority of lenses towards both the Galactic Bulge and the Magellanic Clouds are stars in the Milky Way (e.g. [64], [65]).

Occurrence

It is further argued in [65] that binary events that exhibit a blending factor of $f > 0.17$ should be quite representative of blending for all LGM events. The distribution of the blending parameter f reported by MACHO in their 2000 results [45] and plotted in [65] shows that the majority of events have $f < 0.5$, which means

that moderate to serious blending can be assumed to be present in all realistic LGM events.

It is difficult to determine to what degree a given event is affected by blending unless a detailed fit is performed. Even if good data are available, it is hard to determine f for a single lens event due to a serious degeneracy between f and b, the impact parameter (e.g. [66]).

Theory

A blended event contains a fraction of un-lensed light, diluting the signal from the lensed light source. Figures 19 and 20 illustrate varying degrees of blending for a typical caustic-crossing event and the event's geometry, respectively.

If we assume that the total flux from a source of light in our aperture, A_{blend}, is due partly to Microlensing amplification A_{gm} of a source star with flux L_s and partly due to light from an un-lensed source L_x, the situation is as described in Eq. 33 which is valid for any fraction and source of un-lensed light, whether from a luminous source or a background star:

$$A_{blend} = \frac{A_{gm}L_s + L_x}{L_s + L_x}. \tag{33}$$

The blending parameter f is then defined as

$$f = \frac{L_s}{L_s + L_x} \tag{34}$$

and

$$A_{blend} = fA_{gm} + (1 - f). \tag{35}$$

Equation 35 shows that the blending effect is easy to calculate. In cases where the blending parameter f is the only parameter that changes from one light curve calculation to the next, it is not necessary to recalculate the standard model binary amplification. Instead the standard model amplification is merely modified by use of Eq. 35.

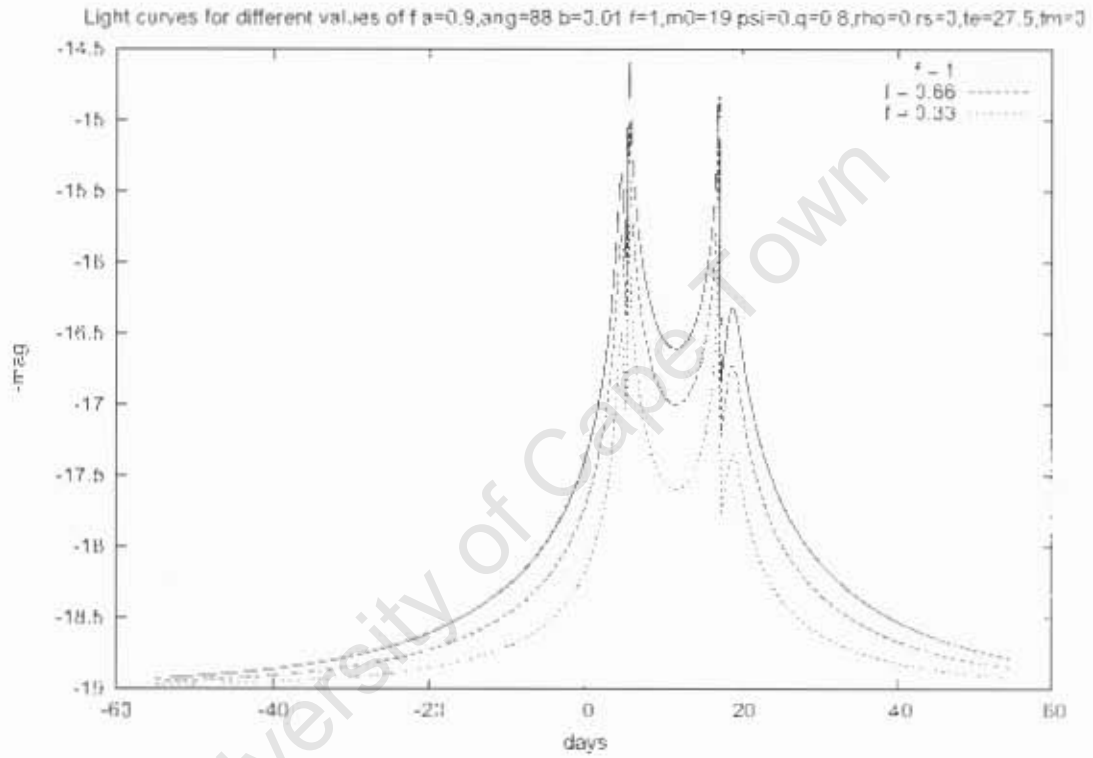

Figure 19: Three light curves, each based on the same standard model parameters, but with the addition of different amounts of blending (f)

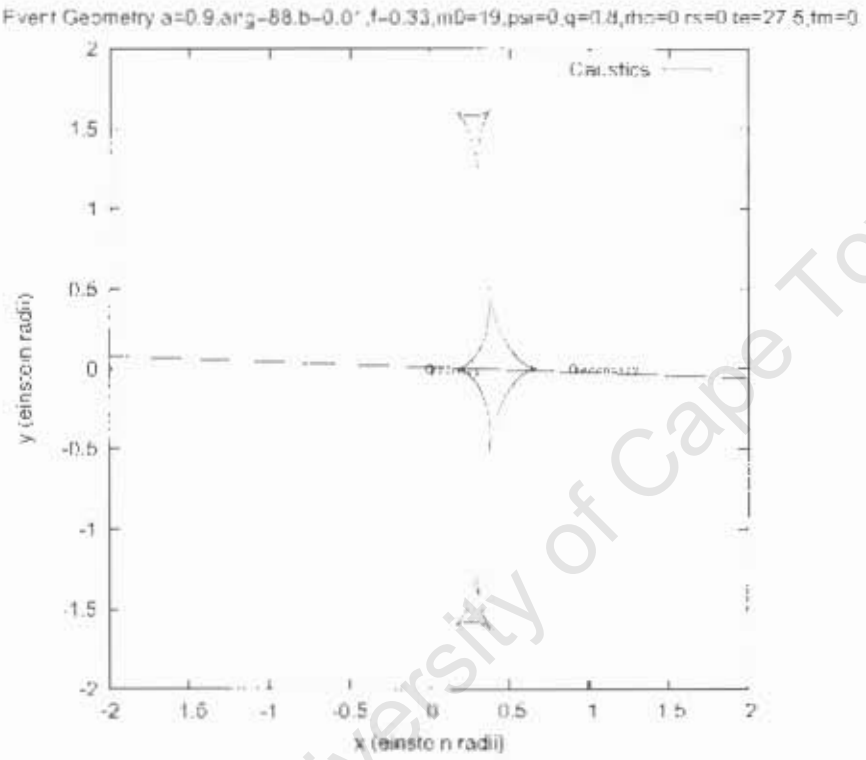

Figure 20: The binary geometry and caustic curves of the events generating the three light curves shown in Fig. 19

Effects on binary fitting

It is not possible in general to ignore blending while fitting for binary events, so blending should ideally be fitted simultaneously with other parameters as one cannot assume that it is possible to separate its effect. Unfortunately the blending effect introduces another variable into the LGM light curve model parameter search space. (Figure 53 much later in Section 8.1.5 illustrates the consequences of ignoring the resolved source effect during LGM binary lens fitting.)

2.3.3 Parallax

"Parallax" deals with the effect of the observer's movement on an LGM light curve. It can be a complicated extension to the model. The simple form of parallax was introduced by Gould [67] but has been considerably expanded upon. Although these models are beyond the scope of this thesis, which sets out to prove that unconventional methods can aid the fitting procedure, it must be noted that they add additional degeneracies to a Microlensing fit. There are also events which could not be explained without the introduction of higher-order parallax effects ([68].)

Nonetheless, the simple parallax which is referred to here as "annual parallax" illustrates the effect well and is discussed below. This extension incorporates the Earth's movement into the equation for the source position. The effect can be taken into account by modifying our linear equations of relative source motion given in Eq. 12.

The notion of parallax may also be extended to differences between light curves measured at different observatories, an effect which was modelled by PLANET in [1].

Theory

Calculations are simplified by projecting all relative movement in the observer-lens-source-system onto the source. This holds true when we wish to include the observer's movement, assumed to be the orbital motion of the Earth around the sun. When this movement is projected onto the source, the source position is a function

of the Earth's orbital motion, as described below. The equations below are stated without proof but follow the derivation by Dominik in [69].

If we define

$$\xi(t) = 2\pi \frac{t - t_p}{T} + 2\epsilon \sin\left(2\pi \frac{t - t_p}{T}\right) \tag{36}$$

and

$$\rho = a_{semi} \frac{1 - x}{r_E} \tag{37}$$

where

- t_p is the time of Earth's perihelion
- T is the Earth's orbital period
- ϵ is the Earth's orbital eccentricity
- x is the distance to the lens as a fraction of the distance to the source
- a_{semi} is the Earth's orbital semi-major axis,

and further use

- φ, the longitude in the ecliptic plane from the perihelion towards the Earth's motion
- χ, the latitude measured from the ecliptic plane to the ecliptic north pole.

we can define

$$x_1(t) = \rho\left(-\sin\chi\cos\phi\left(\cos\xi(t) - \epsilon\right) - \sin\chi\sin\varphi\sqrt{1-\epsilon^2}\sin\xi(t)\right) \tag{38}$$

$$x_2(t) = \rho\left(-\sin\phi\left(\cos\xi(t) - \epsilon\right) + \cos\varphi\sqrt{1-\epsilon^2}\sin\xi(t)\right). \tag{39}$$

If

$$\tau(t) = \frac{t - t_m}{t_e}, \tag{40}$$

we finally have

$$p(t) = \tau(t) + \cos\psi\,(x_1(t) - x_1(t_m)) + \sin\psi\,(x_2(t) - x_2(t_m)) \tag{41}$$

$$d(t) = b - \sin\psi\,(x_1(t) - x_1(t_m)) + \cos\psi\,(x_2(t) - x_2(t_m)) \tag{42}$$

where $p(t)$ is the movement of the source in a direction parallel to the unperturbed source path, and $d(t)$ is the source's movement in a direction orthogonal to the source path or in fact in the direction of the impact parameter vector $\vec{b}$. ψ is a rotation angle in the lens plane describing the relative orientation of $\vec{b}$ to the sun-Earth system.

Two of the parameters ψ and ρ described in this Section are event properties that have to be determined by fitting and I shall refer to them as "the parallax parameters". The remaining parameters like χ and ϕ can be directly deduced from the source's coordinates.

A further look at the above equations is in order. First, note that if ρ is zero, $x_1(t)$ and $x_2(t)$ are zero as well, and Eqs. 41 and 42 reduce to the simple rectilinear motion of the standard model, where

$$p(t) = \tau(t) = \frac{t - t_m}{t_e} \tag{43}$$

and

$$d(t) = b. \tag{44}$$

Secondly, the physical interpretation of ρ is as the Earth's semi-major axis projected onto the lens plane. This parameter is a measure of the "amount" of parallax in a light curve. In general if ρ increases, parallax effects increase as well.

This type of parallax begins to have a pronounced effect on LGM light curves for events with longer time scales: roughly $t_e > 60$ *days*, however the "Jerk-parallax" effect described in [68] affects events with $t_e < \frac{gr}{2s}$.

Three theoretical light curves with varying degrees of parallax and underlying geometry are shown in Figures 21 and 22. Twenty more parallax examples are shown in Figure 51.

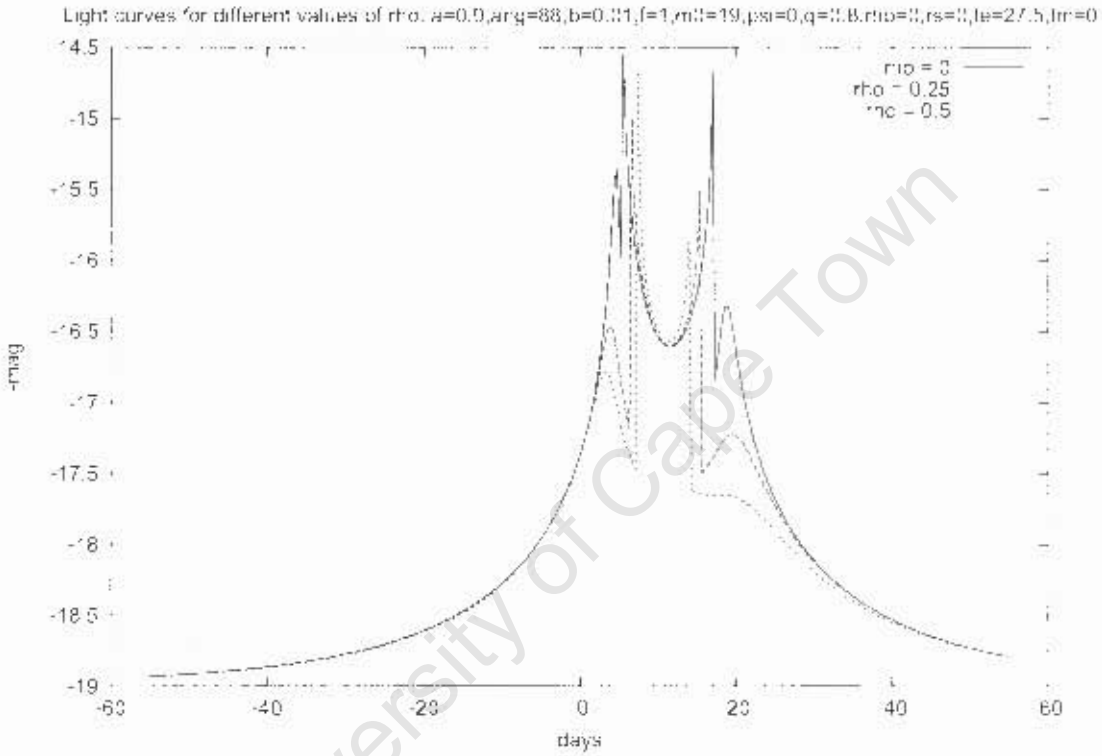

Figure 21: Three light curves, each based on the same standard model parameters, but with the addition of different amounts of parallax (ρ)

When all movement is projected onto the source, we can think of parallax as imposing a regular perturbation onto the otherwise rectilinear source path.

Effects on binary fitting

Parallax mostly affects single lens events of longer time scale so it may be possible to ignore this effect for short time scale single lens events. The scatter plot in Figure 53 and the anecdotal example in Figure 22 above show that the situation for

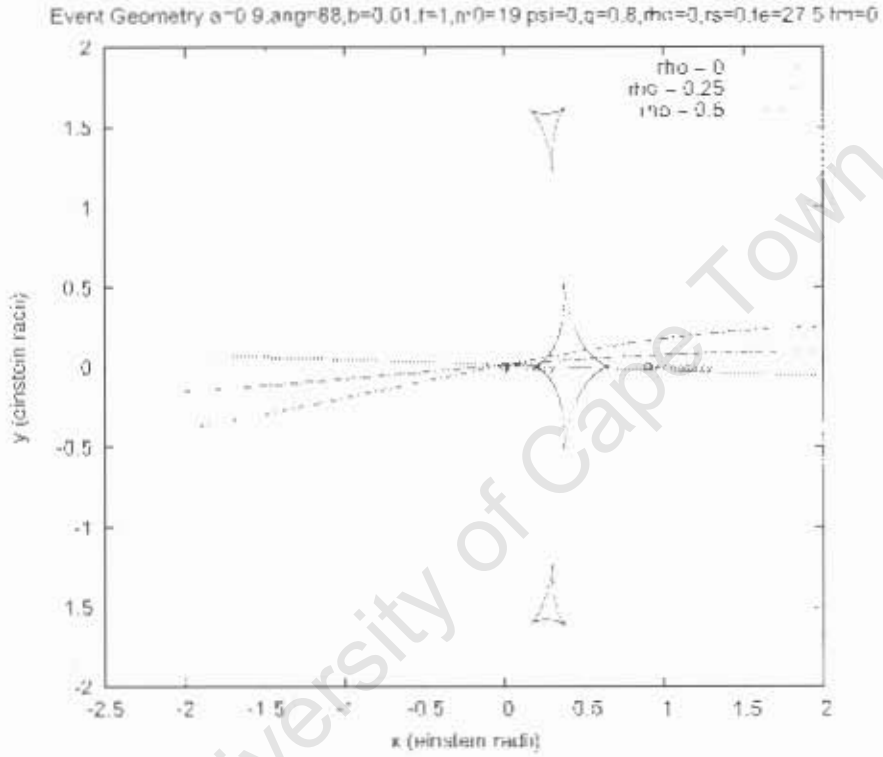

Figure 22: The binary geometry and caustic curves of the events generating the three light curves shown in Fig. 21

binary lens events is very different, where a non-zero value of ρ causes considerable perturbation to strong binary lens events (most of the events in our parameter range are strong or caustic crossing). Where parallax is modeled, it introduces at least two new variables to be fitted, once again multiplying by a large factor the amount of time required for a fit. As with the other extensions to the standard model, it would be ideal if a given light curve could be approached by first fitting the standard model and then refining the model by adding parallax parameters to the standard parameters.

3 LGM Light Curve Fitting Considerations

3.1 Introduction

We shall begin this Chapter by clearly defining the fitting problem in Section 3.2. In Section 3.3 it is attempted to explain why fitting LGM binary lens models to observations presents a particular challenge and why "conventional" fitting algorithms often fail. Chapter 4 deals with the topic of feature selection which attempts to pose the problem in a form that is more suitable for a particular purpose, whether that be to promote understanding or improve fitting efficiency.

Ideally, a modeller should attempt to fit the observations with the model that has the highest a priori probability of correctly describing the physical situation. If a model is insufficient to describe the observations, the model can be extended to include more complicated or unlikely scenarios, following the principle of Occam's razor. The desired output is thus the set of the fewest model parameters that best describe the observations. To qualify the term "best" may present a challenge to the modeller. It may well be that a model fits the observations extremely well by standard modelling criteria. On the other hand, the modeller may be aware of additional constraints from data not included in the fit, such as spectroscopic information that is not directly included in the light curve of an LGM event. A particular set of parameters may also present a physically impossible situation. Any additional information that the modeller has, such as a probability distribution for a given parameter, should be taken into account when modelling, but in this thesis we will focus exclusively on photometric data and modelling.

In the Sections to follow, we begin by fitting the standard model to artificial data generated using the same model. This model is mathematically simple and captures the essence and main difficulties inherent in fitting LGM observations, but it is not complete and various crucial extensions are left out at first. The consequences of ignoring extensions are investigated in Section 8.1.5 and real observations are finally fitted in Chapter 7.

3.2 Definition of the fitting problem

This Section aims to define the central thesis more clearly, in other words what is implied by "fitting a light curve".

3.2.1 Input

In this thesis we focus entirely on the photometric light curves of LGM events.

"Reduced" or processed light curves are simple x-y plots of the flux measured through an aperture that contains a background light source, also containing lesser contributions from other light sources, vs. time.

There is quite a lot of additional information available from the telescope, of which the most important is the photometric error estimate. The actual observational uncertainty or error is not trivial to obtain and is often in itself the result of considerable data reduction and calculation.

In the resulting x-y plot, source brightness, plotted on the y-axis, can be in magnitude or flux. The absolute value of the brightness is often unknown, primarily because LGM modellers are not really concerned with this absolute value as we deal with the ratio of lensed brightness to un-lensed brightness, or amplification. Light curves can also be plotted with amplification on the y-axis, although this may include a blending factor and is not necessarily equivalent to Microlensing amplification obtained from, e.g., Equation 9.

The time axis for a given event spans anything from an hour to several years.

Technically, survey groups require observations over several years to prove that an event is really due to LGM and not another form of brightness fluctuation. Observations of the actual event need to include time before and after amplification in order to establish and confirm the un-lensed baseline. The unit of time is most often days and typically given in Julian Date. Light curves to date consist of tens to a few hundred data points. From the photometric point of view and for the purpose of this thesis, that is the sum total of the input information available to the light curve modeller. Of course additional data may well be available that could influence

certain parameters in the fit. For example, the colour of the source known from spectroscopy may be used to constrain the blending parameter or the likelihood of a binary source. This information is often crucial and should be used whenever available, but we focus here on photometric data only and will assume here that no additional information is available to us.

3.2.2 Output

The desired output of any fitting procedure is the set of physical parameters describing the lensing event. The only way to relate the observed light curve to these physical parameters is by way of a model, such as those explored in Chapter 2.

3.2.3 Fitting as Mapping

In general fitting can be summarised as mapping from one vector space to another.

In the LGM scenario the two spaces in question are the light curve space consisting of time-brightness data points and the model parameter space which consists of (in this case continuous) variables entered into a light curve-producing model. Mapping from model parameters to light curve is a simple operation as each point on the light curve is uniquely determined by the parameters and model, even though the mapping is not mathematically explicit (see Section 2.2). During fitting we are unfortunately concerned with the inverse problem, that is mapping from light curve to model parameters which is neither uniquely determined nor explicit nor analytically tractable.

Many methods exist for tackling this type of problem. Conventional "fitting" is perhaps the simplest of these. Faced with the lack of a unique mapping from light curve to model parameters (which I shall call the "inverse" mapping), the fitter attempts to find a model parameter set matching a given light curve by mapping a number of parameter sets to their corresponding light curves (the "forward" mapping") until one of the generated light curves matches the observed light curve that

is being fitted.

Gradient-based algorithms attempt to use the improvement in a measure of proximity between generated light curves and the target light curve to guide the next choice of model parameters to map. Genetic Algorithms choose the next parameter set to map to its light curve based on manipulations of the "best" of a set of current parameter set candidates, as judged by some measure of proximity. Artificial Neural Networks attempt to approximate the unknown inverse mapping itself by using samples from the known "forward" model mapping.

A statistical fitting method currently enjoying success as applied to binary lens fitting is Markov Chain Monte Carlo (e.g. [70]). The method is particularly well suited to finding all local minima around an initial seed location, and can also be used to determine the covariance matrix at the minimum.

There is also a host of statistical techniques from the field of Data Mining which attempt to infer the inverse mapping from samples of the forward mapping, some of which are tested and applied in this thesis.

3.3 The Challenge

At first, the binary lens fitting problem does not seem particularly difficult to solve. The average light curve consists of tens or even hundreds of data points and we only need to extract a few parameters from this curve, e.g. 7 for the standard binary lens model (SBLM) and about twice that amount, depending on which extensions are included in the model, (i.e., finite source effects, lens system rotation, etc.) Problems that look a lot worse than the LGM fitting problems often pose no challenge for conventional fitting techniques. These may have many more parameters or far noisier data than are generally available for LGM light curves yet are easier and less labour-intensive to fit. What makes the LGM binary lens light curve fitting problem so difficult? This Section discusses the complicating factors.

3.3.1 Fitting considerations in general

This Section describes some aspects of fitting problems in general that combine to determine their difficulty.

Output Dimensionality

One of the most important considerations in a fitting problem is the dimensionality of the search space, or the number of model parameters in the case of LGM light curve fitting. The number of model parameter sets that would need to be tried randomly before the correct one is found scales exponentially with the output dimension. The standard model has 7 output parameters, t_e, t_m, m_0, b, θ, a and q. This is enough to present a serious challenge to "brute force" methods that simply check sets of model parameters to see whether they map reasonably well to a given input light curve.

To illustrate this (simplistically), if we decide that a frequency of 100 steps per parameter is sufficient to sample the output space, we have to compute 10^{14} light curves to check every model in our sample. Combined with the substantial computation time and even denser sampling required of LGM binary lens light curves, it is clear that brute force methods are impractical. As we shall see, the more successful fitting methods developed in this project rely on the reduction of the dimensionality of the search space.

Analytical formulation or the lack thereof

LGM benefits from the existence of a powerful and simple model that approximates the photometry of events in just a few equations (see Section 2). This mathematical formulation enables us to extract physical parameters from a given light curve by fitting. Unfortunately, the model equations cannot be solved analytically for amplification from source position but have to be numerically approximated.

Analytically soluble problems are generally much faster to fit than numerical ones thanks to the powerful mathematics that can be brought to bear on such problems. If we had an analytical formulation for the mapping from light curve to

model parameter space we would have solved the entire problem, as our formulae would provide an explicit solution in the form of the model parameter vector for any given light curve. No further fitting required. Unfortunately, not only do we lack an analytical formulation for the inverse mapping but in fact for the forward mapping as well. Light curves can be calculated by solving numerically for the image positions corresponding to a given source position (Eq. 1) and image positions can be substituted into Eq. 5 to obtain amplification, but the lack of an explicit analytical formulation drives us to numerical analysis which introduces a huge cost in calculation time as well as numerous uncertainties in precision and accuracy.

Data quality

LGM photometry data are generally of good quality, by which we mean that the signal to noise ratio of follow-up observations in particular is sufficiently high to expose the required amount of information. The signal quality required in order to produce accurate model parameters from light curve fitting is determined by the properties of the inverse mapping. LGM data quality is in fact required to be of rather high quality (low noise) and quantity (high sampling frequency) in order to avoid serious ambiguities in the model (see Section 3.3.1 below). Data of exceptional quality enable the modelling of subtle effects, for example those modelled in [1].

Non-linearity

The term "non-linear" is used somewhat loosely in this context to mean that the inverse mapping (and hence the forward mapping, too) is complicated: a small adjustment in model parameters may lead to a large change in the light curve generated by the model. In other words the amplification is a non-linear function of model parameters. The term "non-linear" is often used colloquially to describe badly behaved mappings, where a small change in input parameters can lead to an entirely different set of output parameters, or vice versa.

Fitting problems with this quality require high sampling frequency when performing any form of grid search of the output space. The problem is compounded

in cases of high dimensionality, as a large number (samples per parameter) will be raised to the power of another large number (dimensions) to determine the number of parameter sets required to perform a brute force search of output space. Non-linearity also brings about the problem of premature convergence. LGM non-linearity is described in some detail below (Section 3.3.2).

Premature convergence

In general a modeller cannot know whether a successful fit is in fact the best solution to the given problem. In the case of light curves an Levenberg-Marquardt algorithm (section 5.1) may determine that a certain set of model parameters produces a light curve that fits the model with a low χ^2 and that the model is at a local minimum in the χ^2-space. That is, any small change in χ^2 produces a model worse than the current best, but the local minimum is not necessarily the global minimum. In the case of Levenberg-Marquardt and other gradient-based algorithms it is merely the first minimum that the algorithm has come across, traveling from its starting position on the regression surface.

Premature convergence is common for highly non-linear mappings such as the SBLM. The more convoluted the mapping, the more local minima exist and the less information about the global solution is available to a fitting algorithm operating in the regression space.

It is important to note that χ^2 remains simply a measure of the goodness of fit given our imperfect observations and our error estimates. The χ^2 measure also assumes normally distributed observational errors and a linear dependence on the model parameters. In short, even if we did find the global χ^2-minimum, we would have no guarantee that we had the "true" solution to our fitting problem.

Ambiguity

An ambiguous input vector is defined here as an input vector that maps to more than one output vector, and thus a GM light curve is ambiguous if more than one model parameter set can produce it or a close match to it. The definition is used

loosely because it does not define how closely the two light curves need to resemble each other before they are said to be "equal", or how different model parameters need to be to qualify as "different". We will use the term to mean that two curves are similar enough so that they cannot be statistically distinguished from one another, assuming typical observational errors and using the $\Delta\chi^2$-measure. As a rule of thumb, models are taken to be different if they belong to different convergence wells in the comparison space.

Ambiguities are always a hindrance to fitting algorithms, but their negative effect can be minimized if they are known. Any one of an allowed set of solutions to a light curve is of course a local minimum in χ^2-space, so a mapping that contains many ambiguities also contains lots of local minima which presents the fitting challenge discussed above in Section 3.3.1. From this perspective models with ambiguities can be seen as an opportunity to find an entire set of good solutions if the relationship between valid model parameter sets are known. For example, if an LGM binary lens solution is found to have low mass ratio and projected orbital separation $a = 1.1\ \theta_E$, then the modeller knows that there is another valid model with $a = 0.9\ \theta_E$ that is also a good solution (see Section 3.3.4). Unfortunately ambiguity by definition makes it hard to decide which of the solutions is the correct one. Various degeneracies of LGM light curves are discussed in Section 3.3.4.

3.3.2 Non-linearity in LGM

Section 3.3.1 introduced the complications that arise from a non-linear mapping. This Section attempts to investigate, and where possible quantify, the severity of non-linear effects in the mapping of model parameters to light curves in LGM.

Small parameter adjustments

Figure 23 illustrates anecdotally the effect that a small model parameter adjustment can have on a binary lens LGM light curve. Figures 19, 21 and 14 are also good examples of this, although in these cases one extension parameter is being varied across its entire range.

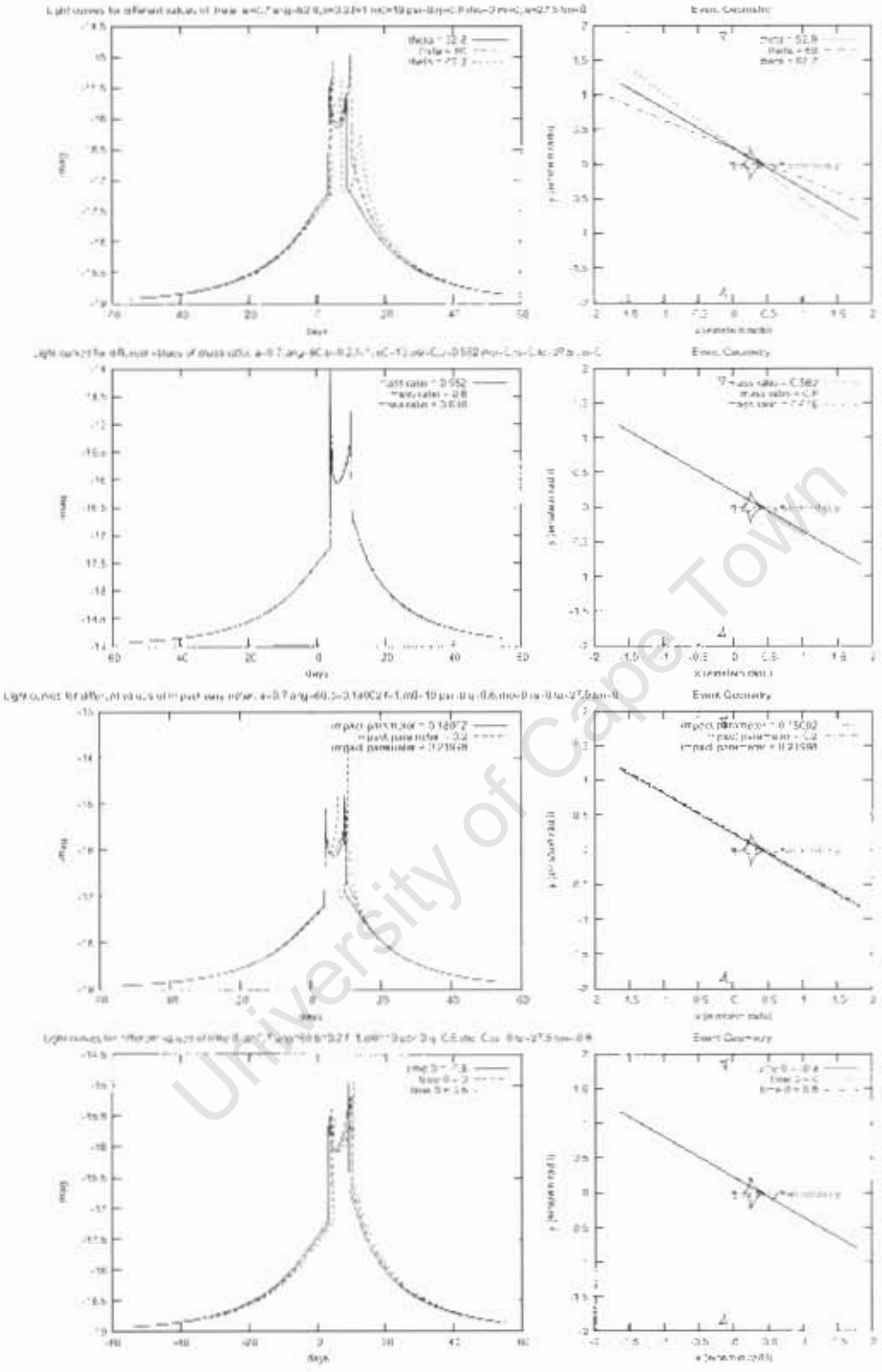

Figure 23: Standard model binary lens events that have their model parameters varied by only two percent can display very different light curves. The parameters being varied are θ, b, q and t_m respectively and geometry of the base event is shown in each case.

Large light curve changes caused by tiny changes in model parameters are the defining characteristic of the non-linear LGM mapping. A small change in the model parameters does not necessarily cause a correspondingly small change in the light curve. Here we were defining "change" as a visible difference by eye, but the more formal measure of $\Delta\chi^2$ between the original curve and the modified one also registers an enormous change for the small adjustment. Regression surfaces other than $\Delta\chi^2$ may not display this non-linear behaviour, i.e., a small change in model parameters may lead to a corresponding small change on this hypothetical regression score function.

3.3.3 The convergence well

Another consequence of the non-linear mapping from model parameters to amplification is the "narrowness" of the regression convergence well surrounding the global minimum of the regression hyper surface when attempting to extract model parameters from an LGM light curve during fitting.

An analogy is in order here. Imagine that the regression hyper surface is represented by a table top. The solution we are looking for is represented by a pit in the table and the aim of regression is to find this pit. The difficulty of this task arises from the fact that the table top is very bumpy, and every hollow represents a local minimum. We are also blindfolded and can only examine the height of the table top by touching one point on it at a time. Luckily we have some tools. Using a gradient descent algorithm to find the global minimum is comparable to placing a marble somewhere on the table and allowing it to roll around until it settles in a hollow, which helps a bit. The problem of premature convergence to a local minimum still fits this analogy: the marble will settle in the first hollow it finds, regardless of whether this is the deepest one on the table.

The convergence well in this scenario corresponds to the size of the unique, correct, deepest hollow that we are trying to find. In a simple problem, the entire table top would slope gently towards the target hollow (solution), and we would find

it easily by placing the marble anywhere on the table. Unfortunately the actual case in LGM fitting is a nightmare scenario. in which the table-top is contorted and the size of the average convergence well is (comparing the convergence volume in seven dimensions with the two-dimensional table top) just about $0.05^7 = 8 \times 10^{-10}$ of the surface of the table, or 1/50th of a millimetre across, much too small to see with the naked eye!

Convergence well calculation

The previous Section used a rough estimate for the size of the convergence well and in this Section it is attempted to justify this estimate by calculation. Admittedly, the "convergence well size" is a crude measure of the difficulty of a fitting problem, as the well diameter actually varies widely along the parameter axes of the regression hyper surface and also varies with the location of the global minimum in parameter space. Yet, it is a single number that provides a simple quantification of this aspect of the fitting problem. One way to measure the mean size of the convergence well is to use a fitting algorithm that simply runs downhill on the regression hyper surface from its start position and stops at the first local minimum it finds. If the solution found by the algorithm is correct, the initial starting position of the fit must have been inside the convergence well of the solution. By recording the success rate of fits to a large number of randomly generated light curves, the distribution of the $\Delta\chi^2$-convergence well size of binary lens fits to LGM light curves can be determined. Note that during this investigation we will keep variables such as the number of data points in the curve and the size of the observational errors fixed at reasonable values of 100 data points with an "error" of 0.01 mag.

Figure 24 shows the results of such a simulation. 100 curves were fitted using the Amoeba (Downhill Simplex) method (e.g. [59]). Each curve was randomly generated from the standard parameter ranges in Table 4. The starting position and the maximum model parameter starting error over all seven parameters were recorded. An example of how the "maximum error" for a given fit is calculated is

Table 3: Sample of "maximum error" calculation. Units are specified as a fraction of the range in each parameter.

Parameter	Actual	Start	Difference
a	0.19	0.22	0.03
θ	0.56	0.49	0.07
b	0.82	0.91	0.09
m_0	0.33	0.34	0.01
q	0.37	0.42	0.05
t_e	0.02	0.07	0.05
t_m	0.91	0.88	0.03
Maximum Error			0.09

shown in Table 3. A fit was considered to be successful if the $\frac{\Delta\chi^2}{d.o.f}$ of the solution model was less than 1.

The main result from this simulation is that the convergence well is small. The distribution of convergence well sizes peaks at about 2 per cent of the allowed parameter range. All parameters mostly need to be within a few percent of the correct value to have a reasonable chance of a successful fit with a simple gradient method! This is a severe constraint that makes LGM binary lens fitting so intractable to conventional fitting methods. Only 5 per cent of the events had a convergence well that stretched out further than 20 per cent of the allowed parameter range. Note that we do not mean 20 per cent of the parameter search volume, but instead that all parameters need to be within 20 per cent (absolute) of their correct values.

The convergence radius is likely to vary for different model parameters, and Figure 25 plots the success rate of Amoeba fits as a function of the start error for each parameter in turn. Note that these calculations were different from the ones that produced Figure 24 in that all parameters were set to their correct values except the parameter under investigation, which was perturbed. As a result the implied convergence well size by parameter represents the best case scenario, where all parameters but one were spot on before commencing the fit. All parameters were allowed to vary during a fit which explains the dismal convergence well size of an "easy" parameter such as m_0. If all the other parameters were fixed at their correct values during the fit, it would be hard to imagine that the routine would not be able

to recover the correct value for m_0. What happens in practice is that the downhill simplex method searches in all parameter directions and is attracted to and caught up in local minima, despite starting out perturbed in only a single parameter.

Although a's distribution peaks around 8 per cent, a fair number of events (20 per cent) can be fitted successfully even if a is initially perturbed by more than 40 per cent (absolute). θ seems particularly hard to fit. Its distribution peaks around 6 per cent and very few events (less than 5 per cent) that are perturbed by more than 40 per cent are successfully fit. b is a more forgiving parameter with a broad peak at about 12 per cent. In addition, 23 per cent of events will be fitted successfully even if b is initially perturbed by more than 40 per cent. m_0 is also relatively easy to fit. The peak is hard to discern on this sample but looks to be between 10 and 15 per cent, with 10 per cent of events fittable despite starting off more than 40 per cent (absolute) from their target. q is, perhaps surprisingly, a fairly easy fit and distributed much like b. t_e peaks around 10 per cent - harder than expected. Only 15 per cent of events can be successfully fitted if they start out perturbed by more than 40 per cent. Finally, t_m is fairly hard to fit again, peaking around perhaps 8 per cent. There are however a fairly large percentage of events (17 per cent) that can be fitted correctly even if they start at more than 40 per cent (absolute) perturbation.

The figure shows that even if all six of the other standard model parameters start out exactly correct, the parameter under investigation still needs to be known fairly accurately to be fitted successfully.

3.3.4 Ambiguity in LGM

Ambiguity and its associated challenges were discussed in general in Section 3.3.1. This Section deals with ambiguities that are specific to LGM light curves and the model used to describe them. Two ambiguous curves are shown in Figure 26. The caustic geometry and model parameters of the two curves differ completely, yet their $\frac{\Delta\chi^2}{d.o.f}$ is only 27. The two very different binary lens caustic geometries are plotted in Figure 27.

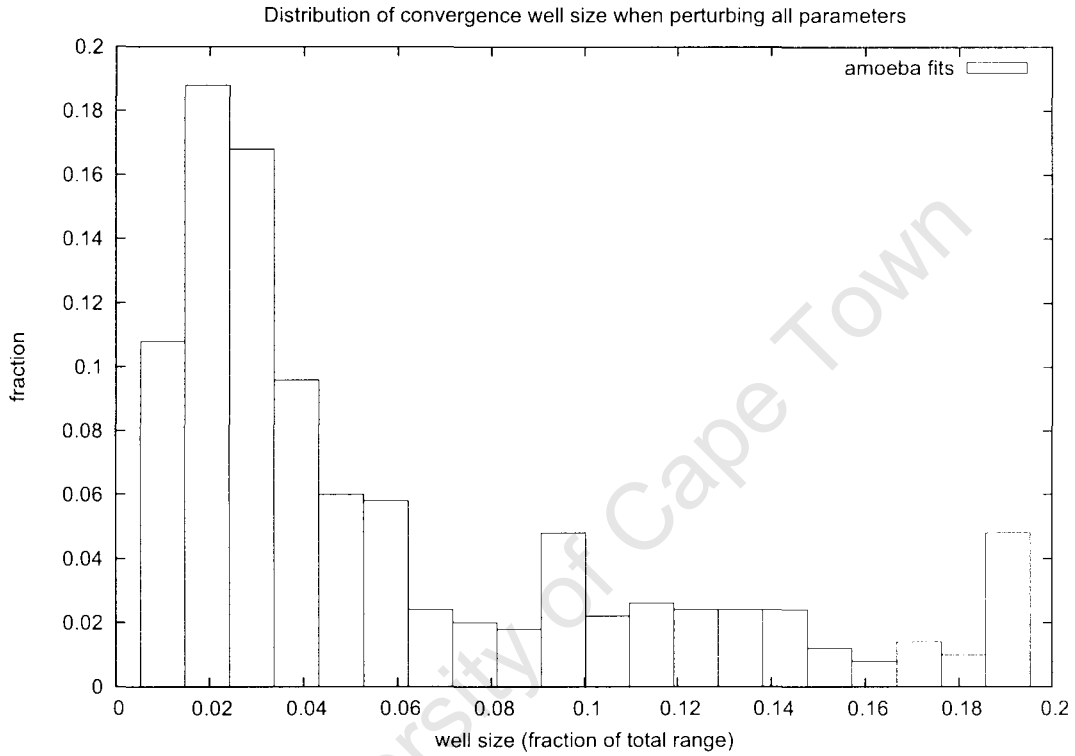

Figure 24: Distribution of convergence well size for multiple fits to 100 random events. The convergence well was defined as the absolute difference between the correct parameter and the random starting parameter for whichever parameter this was the largest. Note that the final bar at 0.2 in fact represents events that can be fitted at 20 per cent absolute perturbation or more.

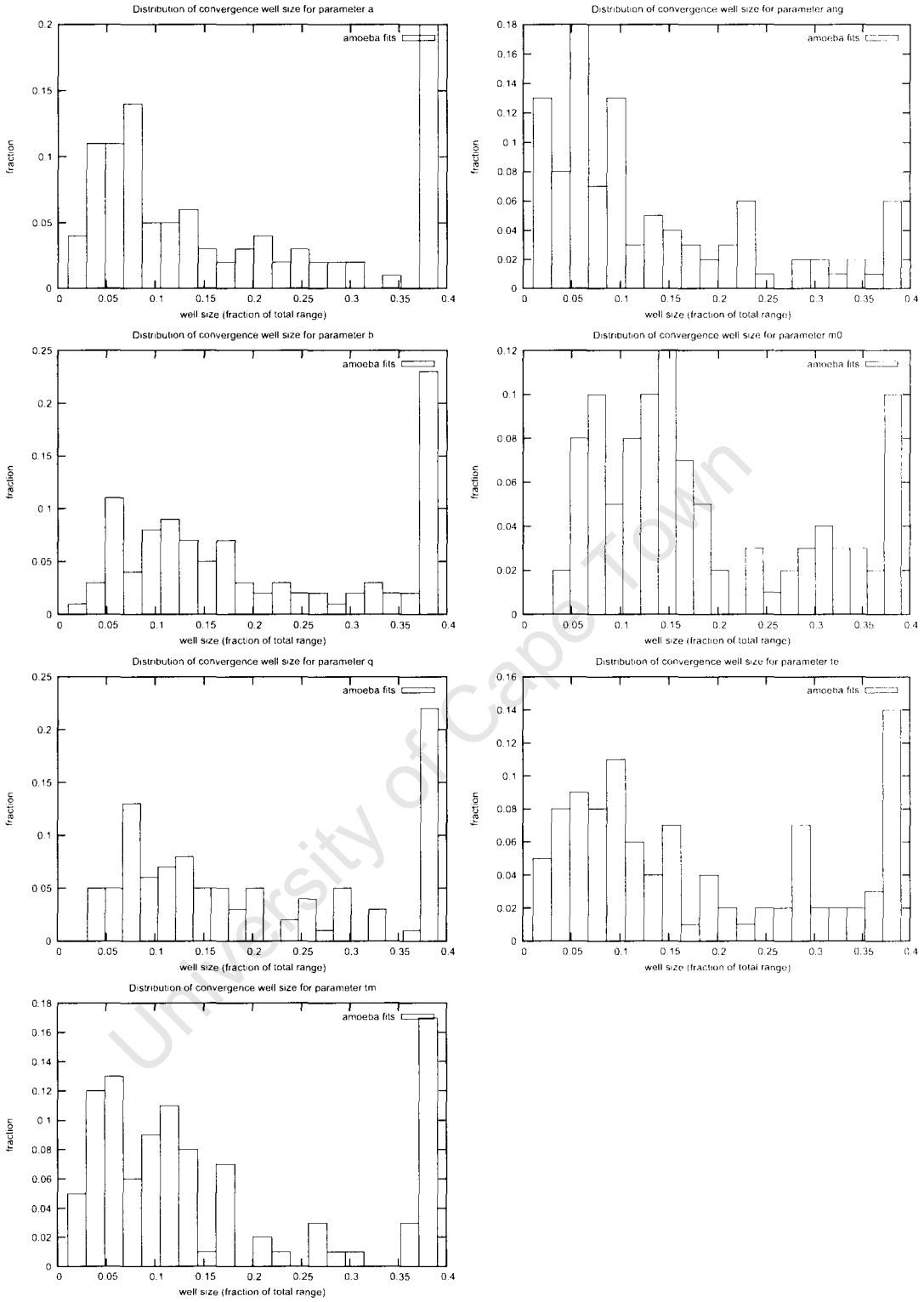

Figure 25: Distribution of convergence well size for multiple fits to 100 random events. This time one parameter was perturbed at a time. Note that the final bar at 0.4 in fact represents events that can be fitted at 40 per cent absolute perturbation or more.

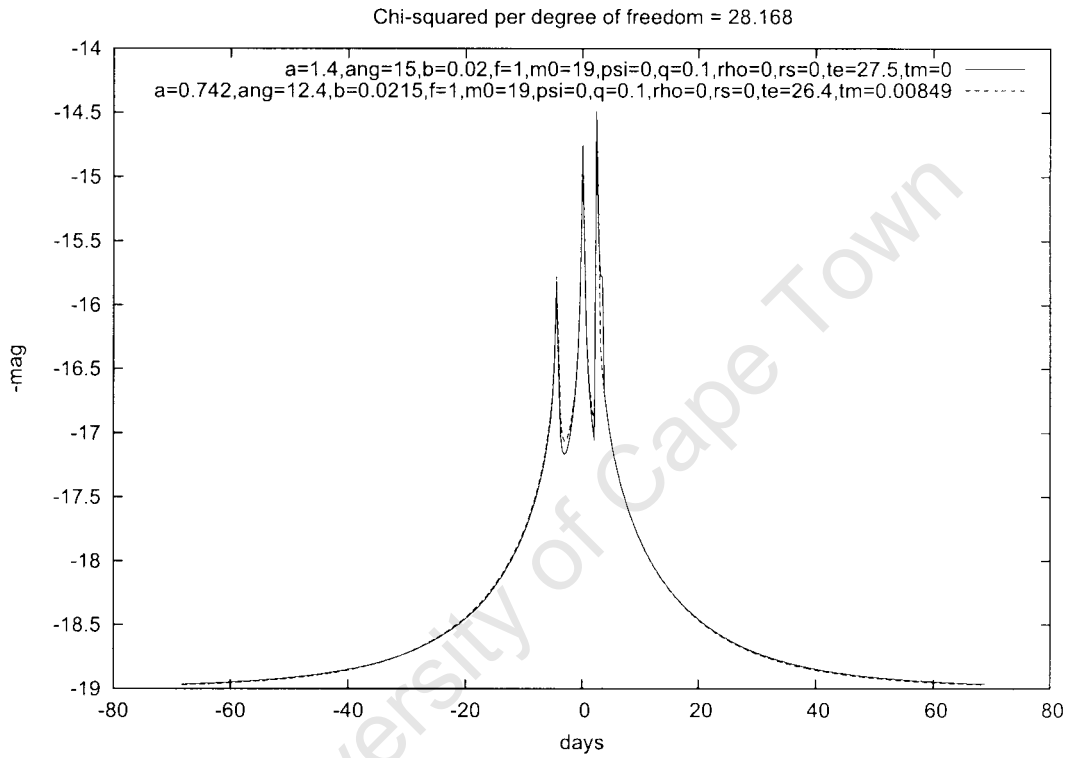

Figure 26: Two radically non-linear, but very similar-looking curves. Most binary model parameters are nearly equal, but the orbital separation a for these two events differ substantially.

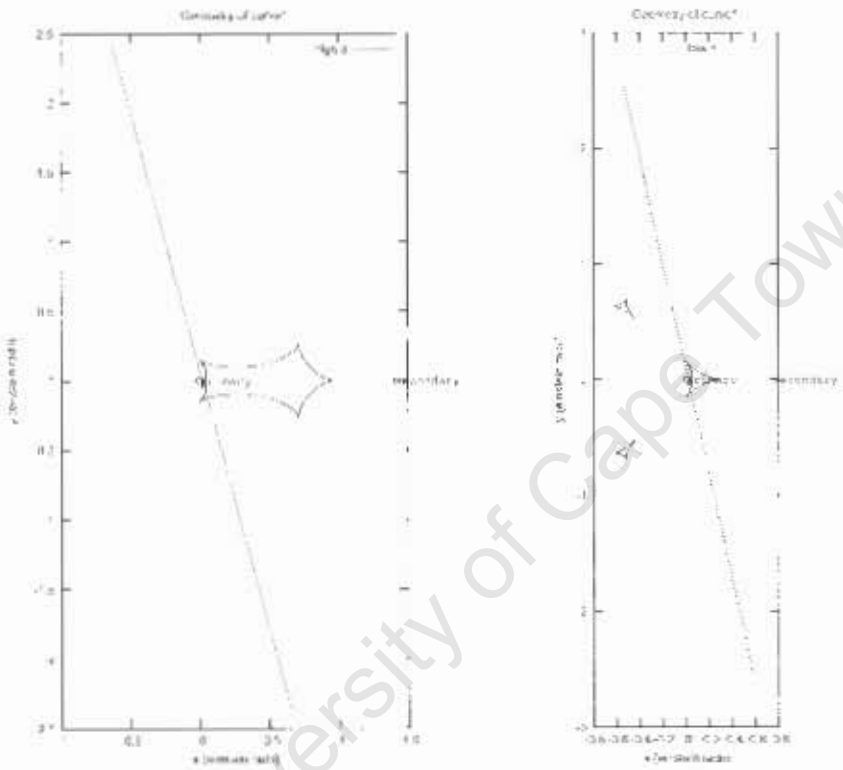

Figure 27: Caustic geometry for the two events plotted in Figure 26. The global caustic geometry is totally different but the source path shown produces similar curves.

The known binary lens LGM model ambiguities are discussed below.

Orbital separation

There is a well-known LGM binary lens ambiguity between light curves with orbital separations a and $1/a$ (e.g. [71]. This is particularly severe for small values of the mass ratio q and is demonstrated as our example ambiguity in Figures 26 and 27. This type of ambiguity has been observed and resulted in the inability to discriminate on photometric grounds alone between a wide binary and a close binary solution in the case of event MACHO-99-BLG-47 [72]. Fortunately in that case the physical implications of one of the two possible models excluded itself on the grounds of plausibility.

Blending

The "blending" effect was previously discussed in Section 2.3.2. Here we focus on the serious ambiguity it introduces to LGM light curves, especially those caused by single lenses. The ambiguity is well-described in [66].

In essence, a blended light source appears to be less amplified than it actually is because a portion of the total flux in the aperture originates from an un-lensed source. This additional un-lensed flux obscures the change in flux from the lensed source. Blending works against amplification, which is dependent on impact parameter b for single lens light curves. This leads one to suspect that an event with a small impact parameter and lots of blending can be mistaken for an event with a large impact parameter and little blending, as is indeed the case.

Wozniak and Paczynski [66] quantify the ambiguity by noting that in the limit of $b \gg 1\ \theta_E$,

$$A \approx 1 + \frac{2}{u^4} = 1 + \frac{2}{\left(b^2 + \left(\frac{t}{t_m}\right)^2\right)^2} \tag{45}$$

where u is the source-lens distance in this single-lens system.

If the quantities F_0 and $F(t)$ are defined as the un-lensed brightness of the

aperture and the brightness as a function of time respectively then

$$\frac{F(t)}{F_0} = (1-f)+f\left[1+\frac{2}{\left(b^2+\left(\frac{t}{t_m}\right)^2\right)^2}\right]$$

$$= 1+\frac{2f}{\left(b^2+\left(\frac{t}{t_m}\right)^2\right)^2} \quad (46)$$

$$(47)$$

by substitution. Note that in these equations f corresponds to the blending parameter defined in Section 2.3.2. It is easy to verify that substituting

$$f_1 = fC^4, b_1 = bC, t_{m1} = t_m C^{-1} \quad (48)$$

into Equation 48 yields exactly the same equation! Thus any (f_1, b_1, t_{m1}) set is also a valid solution if (f, b, t_m) is a valid solution. Wozniak and Paczynski also show that in the limit of $b \ll 1$,

$$\frac{F(t)}{F_0} = 1+\frac{f}{\left(b^2+\left(\frac{t}{t_m}\right)^2\right)^{\frac{1}{2}}}, (f \ll 1) \quad (49)$$

so that any set $(fC, bC, t_m C^{-1})$ is a solution if (f, b, t_m) is a solution.

These ambiguities raise grave concerns for the accuracy of conclusions drawn from photometric data of single lens light curves if blending is ignored. The ambiguity effect of blending on binary lens curves is less pronounced (e.g. [62]) but can be expected to affect any weak binary event in a manner similar to the one discussed here.

Parallax

The parallax extension to the 7-parameter model adds 2 parameters needed to describe non-linear relative motion of the lensing system. These (ρ and ψ) were discussed in 2.3.3 above and are used to model the effects of the Earth's orbit on an

LGM light curve. Smith, Mao and Paczynski [40] discussed a constant acceleration model which serves as a good approximation to the normal parallax effect. Using this model they discovered a degeneracy which affects weak parallax events, i.e., events with t_e close to the bottom of the range of where parallax can be observed. The degeneracy is in the two parallax parameters and t_e.

An extension to the parallax model made by Gould in [68] also introduced a new form of parallax degeneracy.

Weak parallax vs. planetary perturbations in long time scale, asymmetric events

In about 1 per cent of planetary events, planets can give rise to long time scale perturbations [3]. These are highly degenerate with parallax events of the type discussed in [67]. This degeneracy has also been observed, which led to the disqualification of event OGLE-1999-BLG-36 as a planetary event.

Close and wide binaries vs. planetary events at high amplification

Extreme separation ($a \ll 1\ \theta_E$ and $a \gg 1\ \theta_E$), medium mass ratio binaries generate caustic patterns close to the primary lens that are almost indistinguishable from those caused by medium separation, planetary companions ($q \ll 1\ \theta_E$). This degeneracy is discussed in, e.g., [71]. It has also been observed, if that is the correct term in so far as the degeneracy was an issue during analysis ([73]).

3.3.5 Sampling parameter space

Possibly the simplest (and most brutal) way of fitting a light curve is to decide on a sampling frequency in parameter space and to simply step through all sample models looking for one that generates a light curve that closely matches the data curve. Assuming the sampled curves adequately cover the allowed parameter ranges, the method has the advantage of being exhaustive. Of course, the problem with LGM is that a vast number of sample models are required. The author did attempt a naive library-based fitting solution without success, to be compared to Mao & Di

Stefano [74].

Storage restrictions

If the sampling requirements of this library are to be met, there is still the restriction of storing the calculated light curve for each grid point as they are needed for comparing with observed light curves when fitting with a library-based method. The number of data points per curve that need to be stored is a non-trivial issue, but if we assume that 100 points are good enough to distinguish between light curves to a level that produces a usefully small list of candidate models on comparison, we are faced with the prospect of storing 200 points of data of size "double" per curve. A small test shows that gzip compresses 100 light curves of 100 x- and y-points each into about 70kB. For a library method that needs to fit, say, all 7 standard model parameters and a sampling spacing of 5 per cent per parameter we are faced with

$$\left(\frac{100}{5}\right)^7 \times curves \times 0.7kB = 1.28 \times 10^9 \times 0.7kB = 896GB \tag{50}$$

of storage. Although this number is attainable on a personal computer one has to wonder whether this is the right approach.

3.3.6 Computation time

Computation time is a limiting factor when fitting binary lens LGM light curves due to the iterative, non-analytical nature of the amplification calculation. We have seen several different methods of computing amplification as a function of time described in Section 2.2.5. Regardless of the method used, amplification calculations are non-trivial and take such a long time that brute force is excluded as an LGM fitting operation.

4 Feature Selection

4.1 Introduction

The input to our fitting problem is an observer LGM light curve. In this Section we will simplify the actual problem by assuming that the observers have provided us with a reduced, cleaned light curve. This may have required the removal of outliers, the combination of data points from different observatories, rescaling of photometric errors and the like.

We can now feed this light curve directly into our regression algorithms or process it further. In fact, practitioners agree that regression performance often benefits from further processing of the inputs, a.k.a. "pre-processing". One very important form of pre-processing is feature selection or construction. This is the process of constructing new input variables from the existing ones through transformations, driven by domain-specific knowledge or by removing redundant variables. The latter is often referred to as subset selection but we will use the term "feature selection" to refer to any change made to the original input set, by either transforming, removing or adding variables and their derivatives.

New inputs can be any set of numbers derived from or combined with the original inputs. The aim is to find a transformation of input parameters that has more predictive power than the parameters themselves, as measured against an unseen, testing data set.

There are good reasons to reduce the number of inputs to fitting algorithms. The vast majority of classifiers and regression algorithms function progressively better as the number of inputs is reduced, provided that it is not reduced to the point where critical information that could be used in the identification of a given light curve is lost.

The somewhat counter-intuitive reduction in success rate with increasing input information is mostly due to the "curse of dimensionality" (i.e. [75]). There is a point at which adding input parameters does not justify the enlargement of the input

space that comes from increasing its dimension. In this way, feature selection is also related to data compression and, risking a philosophical statement, "understanding", in that the more compact the parameter set used for a complete description of the data is, the better we understand it.

The problem of finding the best input vector is that of maximising the amount of relevant information in the input vector while minimising the dimension of the input space. Using the correct input vector is absolutely crucial to successful approximation or fitting, but the question is how to find this input vector.

One can do so by trial and error, which consists of selecting different combinations of input vectors by intuition or at random, training a regression algorithm and checking performance: a prohibitively slow process with no guarantees offered on finding the best solution. Fortunately, the next Sections describe a selection of more realistic, formal methods that were investigated, applied and evaluated for LGM binary lens light curve fitting with the standard model.

Goal The goal of this Chapter is to find a representation of an LGM light curve that performs optimally at regression, presumably better than the light curve does by itself. I shall divide the process of deriving an optimal feature set for fitting LGM binary lens light curves into three dependent Sections:

1. Construction. New features are created or derived from the light curve.
2. Evaluation. New and original features need to be evaluated for their predictive utility.
3. Search. We require a feature-set searching methodology to derive the optimal set.

4.1.1 Know your data

Before we even start deriving new features, it is highly recommended to get to know the raw input data (i.e. [76]). We need to know what the typical light curve

Table 4: Standard Binary Lens Model (SBLM) ranges used throughout.

Parameter	Minimum	Maximum
$a(\theta_E)$	0.6	1.7
$\theta(^\circ)$	0	360
$b(\theta_E)$	0.001	1.0
$m_0(mag)$	18	21
q	0.1	1.0
$t_e(days)$	5	25
$t_m(days)$	-10	10

looks like, what the "typical" outlier looks like, what kind of degeneracies can be expected, etc. A knowledge of the data will improve our understanding of results to follow.

There are many ways of looking at our data. We shall start with a visual inspection of a number of randomly sampled light curves before moving on to statistical properties of the complete training sample. Figure 28 shows 50 light curves selected at random from the sample constructed from the parameter ranges shown in Table 4 using the standard model.

A few simple observations from the figure. First, almost all curves are indeed visibly perturbed from the typical single lens light curves. Most are asymmetrical. Secondly, a lot of curves have only a single peak and of these it would seem that many are fairly similar. This indicates large areas of degeneracy when attempting to recover model parameters for curves with a single peak.

Many curves seem to be crossing a largish caustic area, showing the distinctive caustic entry and exit spikes and an area of increased amplification while within the region. Many show peaks that are not due to actual caustic crossings and so must be a caustic cusp approach.

Statistical description of binary light curve data

In order to use a simulated light curve as an input vector, we take the step of creating it from samples at 100 regularly spaced points between a start and end point at $t_m - 2t_e \pm 0.5\ t_e$ and $t_m + 2t_e \pm 0.5\ t_e$. The uncertainty in the start and end

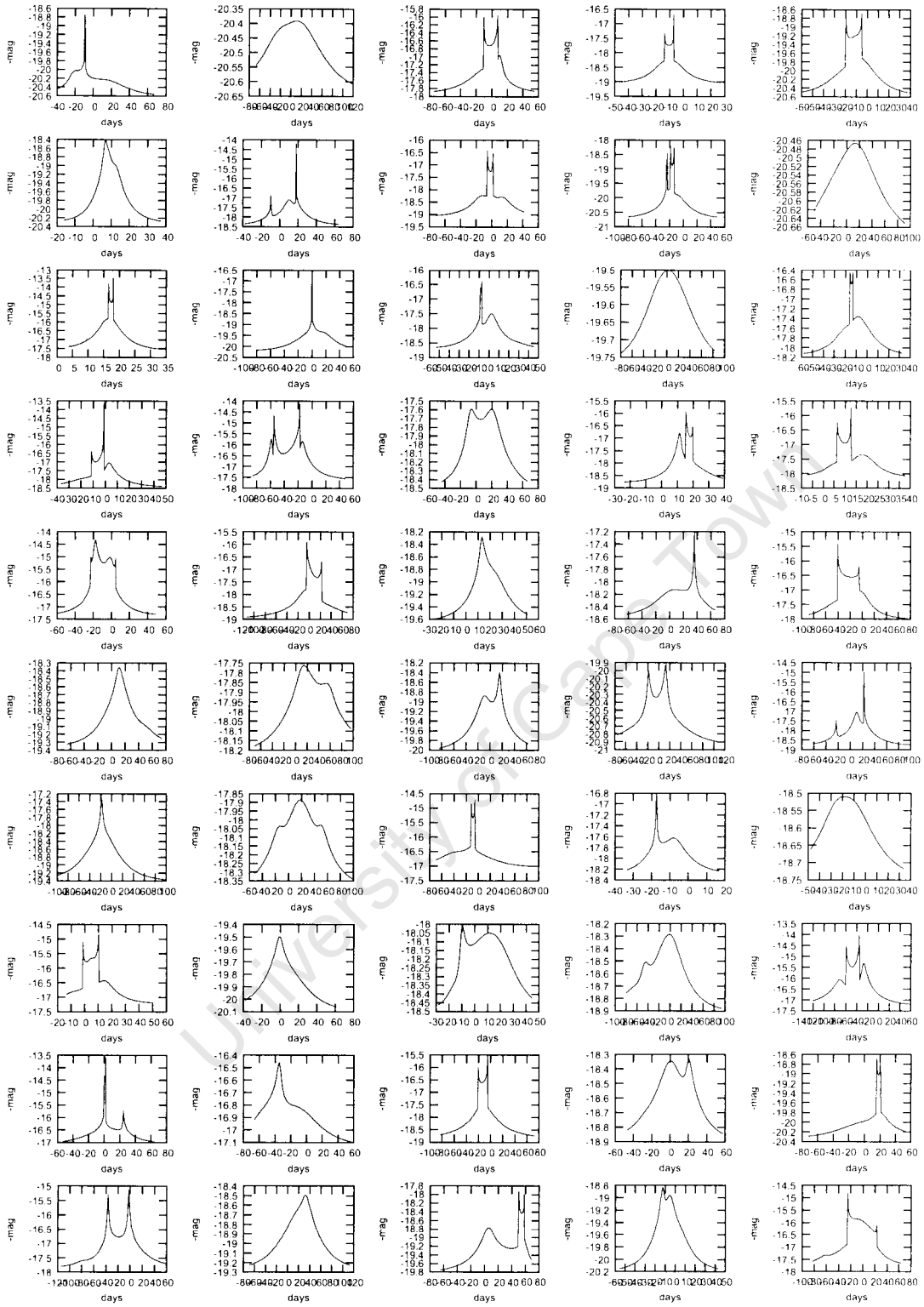

Figure 28: 50 randomly selected light curves from the parameter ranges in Table 4 using the standard model. This is our input space of choice for study of the standard model.

point reflects the fact that we do not know what t_e and t_m are before fitting. Our raw data thus consist of 100 evenly-sampled time points and 100 matching magnitude points. As we shall see below (in Section 4.2.1) we can discard all but two of the time points with no loss of input information, leaving us with 100 magnifications and the start and end times of the curve.

Figures 29 to 32 are distribution plots where the values on the x-axis are derived from the extremes of the range and the y-axis is a count. As the actual value of the count is not important (but the amplitude of the distribution is), the y-axis scale was omitted.

Figure 29 shows the distribution of start times for a sample of 10000 random events with parameters selected from Table 4. This distribution is a function of the ranges chosen for t_e and t_m, as well as an additional uncertainty of 0.5 t_e for each curve's starting point. There is not much else to learn from this distribution but it begins to place the light curves into context.

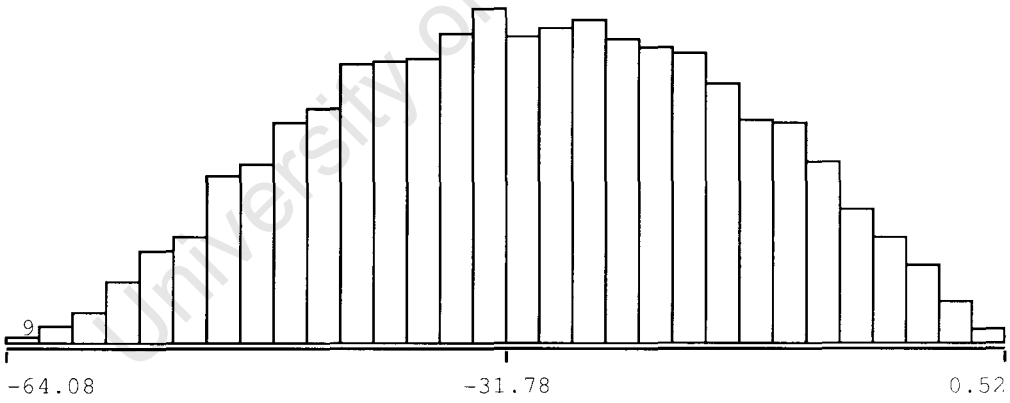

Figure 29: The distribution of curve start times (in days) for 10000 standard model events, randomly generated from the ranges in Table 4.

Figure 30 shows the distribution of start magnitude for all 10000 curves. This distribution is technically a function of all model parameters, as well as our choice of light curve start and end ranges, $t_m - 2t_e \pm 0.5t_e$ and $t_m + 2t_e \pm 0.5t_e$. Of course, the

magnitude this far from the peak of amplification is almost directly related to the flat distribution of start magnitudes chosen for the experiment. The distribution at points closer to the centre of the curves are more interesting.

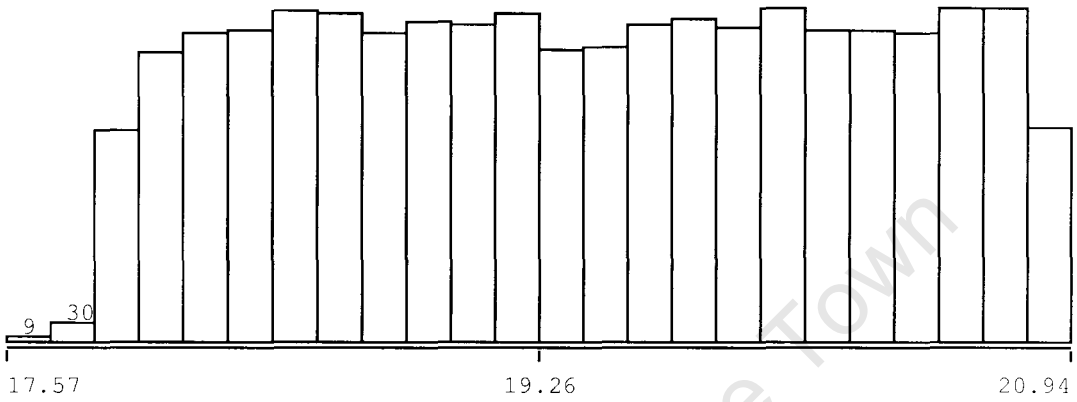

Figure 30: The distribution of curve start magnitudes (mag) for 10000 standard model events, randomly generated from the ranges in Table 4

Magnitude distributions at points deeper into the curves are plotted in Figures 31 and 32 which are at position 33 and 50 out of a hundred magnitude points, respectively. Position 50 corresponds on average to the centre of each curve. Both these distributions are complicated functions of the ranges chosen in Table 4. Both are highly asymmetrical as one would expect of a flat distribution in impact parameter b: single lens amplification scales approximately as the inverse of b at high amplification, and we know that a large number of light curve will resemble single lens curves.

The rather smooth, unimodal distributions discussed above suggest that there is no clear discriminator for a classifier or regression routine when looking at any given magnitude point in isolation. Relationships between points are likely to play the largest role when discriminating between models.

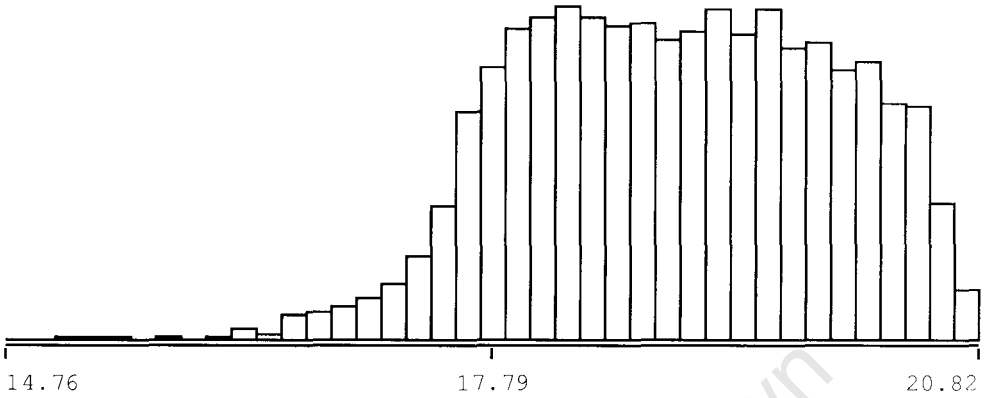

Figure 31: The distribution of curve magnitudes (mag) at point number 33 for 10000 standard model events, randomly generated from the ranges in Table 4

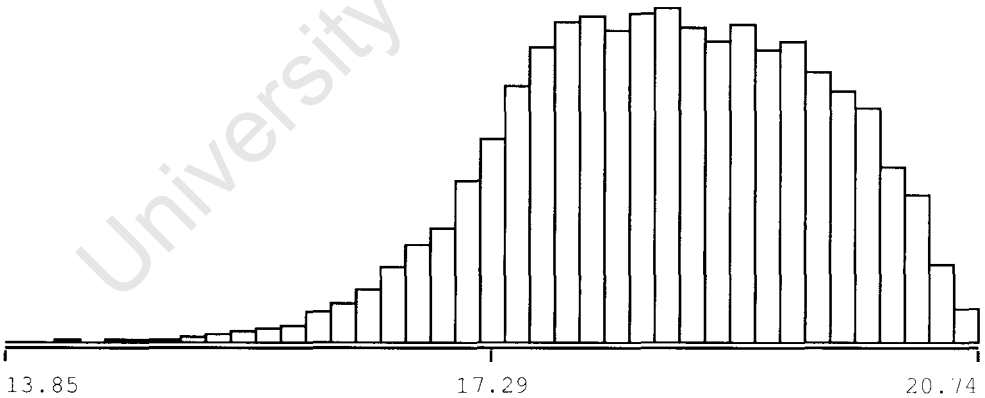

Figure 32: The distribution of curve start magnitudes (mag) at point number 50, which corresponds to the average peak position, for 10000 standard model events, randomly generated from the ranges in Table 4

4.2 Construction: Creating new features by transforming the light curve

In this Section we shall derive many features from an existing light curve, either by transformation or by use of domain knowledge or by hunch. The vast majority of these will then be mercilessly discarded in Section 4.3 when we evaluate their actual worth in fitting. Nonetheless the prediction is that at least some useful new features or transformations will be discovered which have the potential to facilitate the fitting process greatly.

4.2.1 Domain-specific construction

This Section deals with features that are specific to light curves. Later Sections will use generic feature construction methods in which the input vector is considered to be just a vector of values. Here we will attempt to incorporate properties of the standard model and its resulting light curves into features that should have high information content or predictive power. The number and position of peaks, for example. We know these numbers say something about model parameters. If we have a large number of peaks, we must have multiple caustic crossings which put constraints on the parameters a, q and b and θ. Peak number is a generic concept but we suspect it is significant due to our domain-specific knowledge.

The most obvious features of an LGM binary lens light curves are the peaks and troughs of the curve, but any bit of information can potentially serve as a feature. An infinite number of features may be extracted from a single curve, from the obvious to the more obscure, such as the slope at start and end-point, moments of the magnitude, etc. A good selection requires an intuitive approach, based on the modeller's knowledge of the problem. When a large number of plausible feature selections have been made, selection methods from Section 4.3 will be used to narrow the candidates down formally to the optimal set. Figure 28 shows example binary lens curves and the reader is invited to select a minimal set of features that carry maximum information regarding the possible lens geometry that was responsible for these curves.

The curve itself

A first choice of a feature set is not to choose one at all and to use the entire curve as input, thus retaining all information contained in the observations. This type of input was used successfully in [77] to fit the standard model to simulated light curves, although the sampled curve data were supplemented with their own extremum data.

This type of input is unsatisfactory for several reasons:

1. Even for this direct approach some pre-processing is required. A sampling frequency, start- and end-point need to be chosen to ensure that all curves can be turned into a uniform, evenly-spaced format for comparison by the fitting algorithm.

2. LGM binary light curves are highly non-linear and subsequently require a high sampling frequency to enable a successful fit by Artificial Neural Network. This leads to input vectors with very large dimensionality. In [77], input vectors had up to 117 entries. Large dimensionality brings about the "curse of dimensionality" whereby the success rate deteriorates with increasing input dimension despite the input of more information.

3. LGM light curves often contain gaps in coverage, for example due to bad weather. Some algorithms require full input vectors and this raises the issue of what to do where data are missing. The "Missing Data" problem (e.g. [78]) is a non-trivial drawback to the use of these algorithms. Other features, such as pre-fitting by simple models, may be less sensitive to gaps in coverage.

Nonetheless, raw light curve data proved effective in the experiments in later Sections. Perhaps that is because they do not lose information and modern algorithms have become more robust against the problem of missing values?

Extrema

LGM binary lens curves often have strong peaks and deep troughs. As discussed in Section 2.1.1, these are closely related to geometrical features of the binary lens system and the projected source path across the system, implying that features based on the curve extrema should be rich in geometrical information which is closely linked to model parameters. The (t,mag)-coordinates of extrema were used as additional input to the light curves themselves in Section 6. Higher-order extremum data, such as the second derivative magnitude at a peak/trough, can also be used. Unfortunately, extrema are often missed in practice due to bad weather or late alerts, leading to missing inputs and potentially crippling fitting problems.

Statistics and Moments

A single light curve can be described by various statistics and it is easy to imagine that some of these have a high information content. The mean, maximum and minimum of both time and magnitude values should prove useful. Higher order moments such as variance, skew and kurtosis could also be valuable as a rough indication of the asymmetry or number of caustic crossings in a curve. All of the above were calculated for both time and magnitude values.

Derivitaves

Derivatives of light curves may be useful as well. One could speculate that these would show a clear signature of peaks and troughs and may be used with the magnitude at start and end points of the curve, perhaps to provide information on the impact parameter, etc. Derivatives unfortunately cover a very wide range so in this thesis it was decided to use the tangent angle at each point as a more smoothly-varying replacement for the first derivative. The second derivative was used as calculated.

Several numerical schemes were considered for calculating derivatives. Local approximation with smooth functions such as splines was a viable option but in the end a simple difference method was used where the slope was simply the slope

of the linearly interpolated light curve. It was interesting to note the difficulty of approximating a potentially noisy light curve in its entirety with a simple smooth function. Figure 33 illustrates the problem. The light curve in the figure is not well approximated by Chebyshev polynomials, even using polynomials with 200 cofficients. This issue is discussed further in Section 4.2.2.

Statistics on the slope were also included.

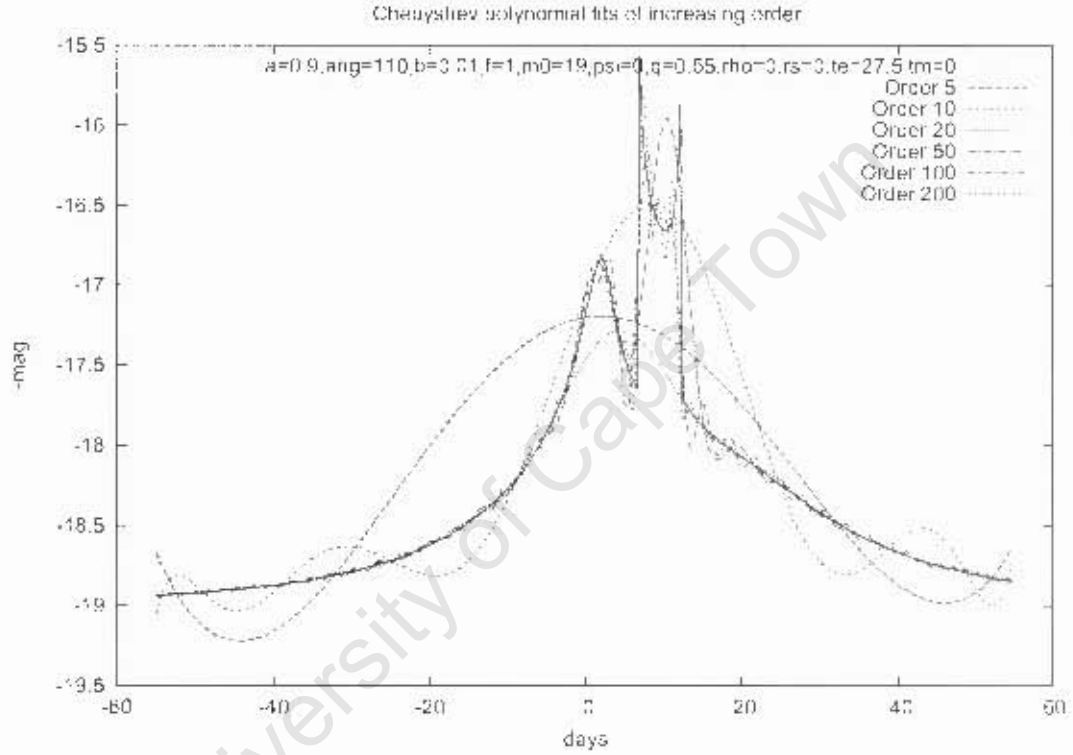

Figure 33: Chebyshev approximation of a non-linear but fairly typical binary lens light curve. The approximation is poor, even at 200 polynomial coefficients. Note in particular the dismal failure of the 20-coefficient polynomial which is still incapable of resolving the multiple peaks in this event.

Smoothing

Another candidate feature set consists of the moving average of N light curve points. Two such curves were added into the list of features, the first with an averaging length of 5 data points, the next with an averaging length of 20 points.

This simple form of smoothing has some potentially undesirable properties, such as the dilution of peaks and troughs. However, this is not necessarily a handicap as peak and trough information is already available in abundance in other features. Instead, the point of the smoothed curves would be to accentuate longer-term trends in the light curve.

Simple fits

An LGM light curve may be well defined by the parameters of a model that is simpler to calculate than the standard model. For example, fitting a 10th-degree polynomial to a light curve would yield 11 parameters and if the fit were good, these parameters would contain most of the useful information that was contained in the original curve. Fitting an alternative model to the data seems counter-intuitive, as the aim of the entire fitting exercise including feature extraction is to fit the LGM binary lens model to the light curve in the first place. The advantage is that simple models are much faster to calculate and may contain all the relevant information implicit in the data in a much more compact form, e.g. a few parameters - provided a suitable model can be found.

This is subject of course to the existence of a satisfactory simple model.

Single lens fits to binary curves The most effective way to reduce the difficulty of a fit is to reduce the number of parameters that need to be fitted. One way partially to achieve this is by first fitting a single-lens model to a light curve to obtain an estimate for t_e, t_m, m_0 and b, dramatically reducing the regression search space. The single lens model provides a very poor description of the majority of binary lens light curves. Figure 34 provides a quantification of the number of binary lens curves with parameters in the ranges specified in Table 4 that are successfully fitted by a single lens model. The criteria for success are $\frac{\Delta\chi^2}{d.o.f} < 2.0$ and $E_{max} < 0.1$, where E_{max} is the maximum error after fitting of any fit parameter. The combined success rate of less than 4 per cent exposes the limitations of the single lens fit.

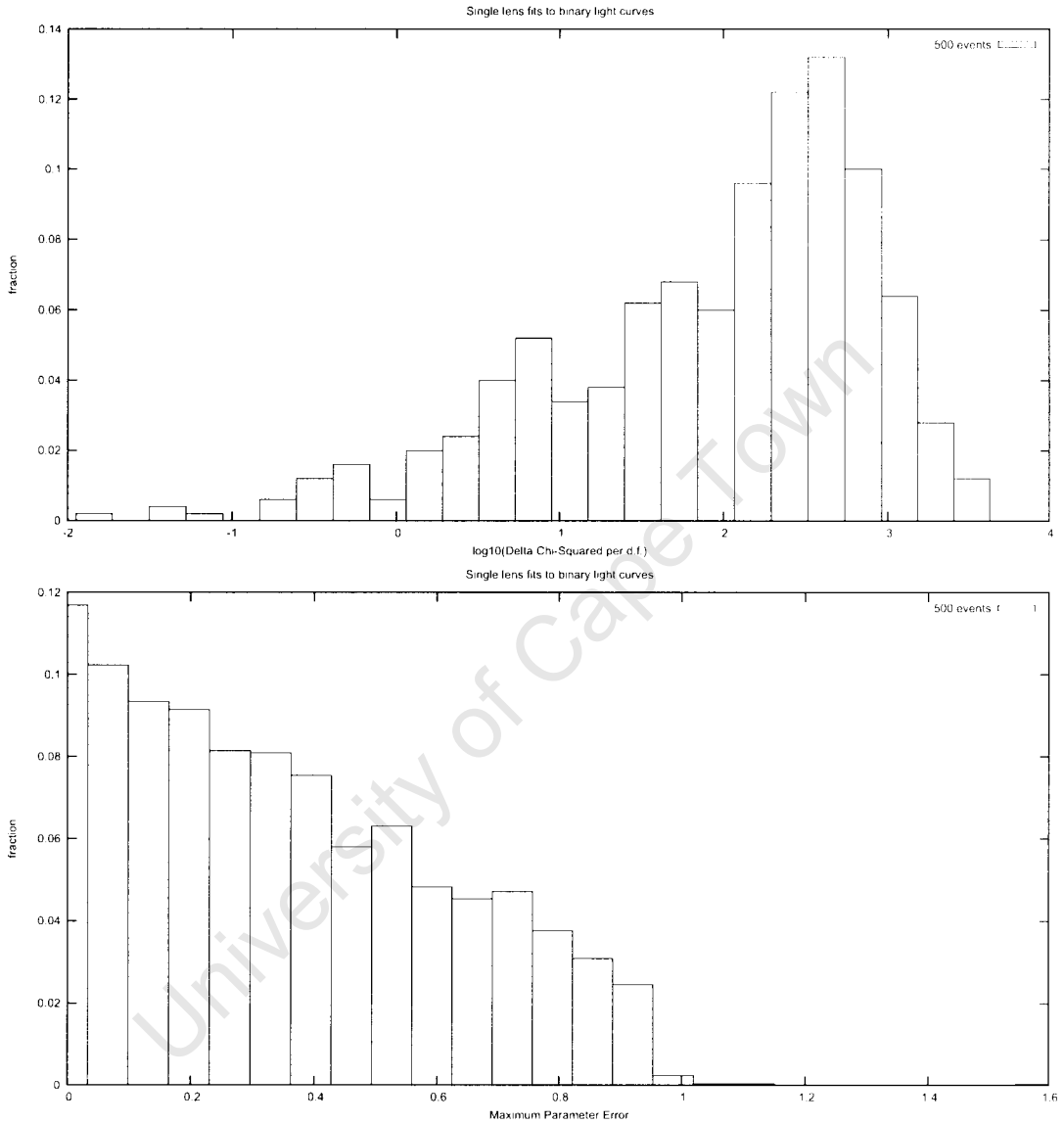

Figure 34: The distribution of $\frac{\Delta\chi^2}{d.o.f}$ for single lens fits to standard model binary events with parameters in the range specified in Table 4.

A visual illustration of single lens fits to binary events was considered useful and is shown in Figure 35. Most seem reasonable. There appears to be a small percentage that one feels the fitting routine could have done better at and some are clearly too ambiguous to fit as a single lens.

Single lens fits may be used in this way to reduce the parameter space by assuming that these parameters carry over to the binary lens problem in question, and then subsequently varying binary parameters. Alternatively, the single lens parameters may be varied in a subsequent binary fit that used the single lens values as a starting point. Finally, single lens parameters may be used as features in a binary lens regression but that was not attempted here due to the low combined success rate of just 4 per cent.

4.2.2 Generic Models

One can make the distinction between domain-specific models such as the single lens fit from Section 4.2.1, and "generic" models such as those obtained from spectral analysis or general linear regression, which will fit any observation given enough terms in the expansion. Both methods can be used during feature construction. If a light curve is fitted by, e.g., a Fourier expansion, the finite number of Fourier coefficients in an approximating expansion can be used as features for a regression algorithm. It is possible to achieve significant dimension reduction with this process, provided that the Fourier fit is good, which means that high approximation accuracy is achieved with a small number of coefficients. Unfortunately we shall see that most light curves are not well approximated by generic models.

Polynomials and Chebyshev Polynomials

All smooth functions can be approximated by polynomials. The aim in this Section was to replace an entire light curve with a polynomial approximation to reduce the input dimension. The simulated light curves that are to be fitted contain 100 points. If we could approximate the light curve well with a 5th-degree polynomial to use as a proxy we would reduce the input dimension from 100 parameters to just

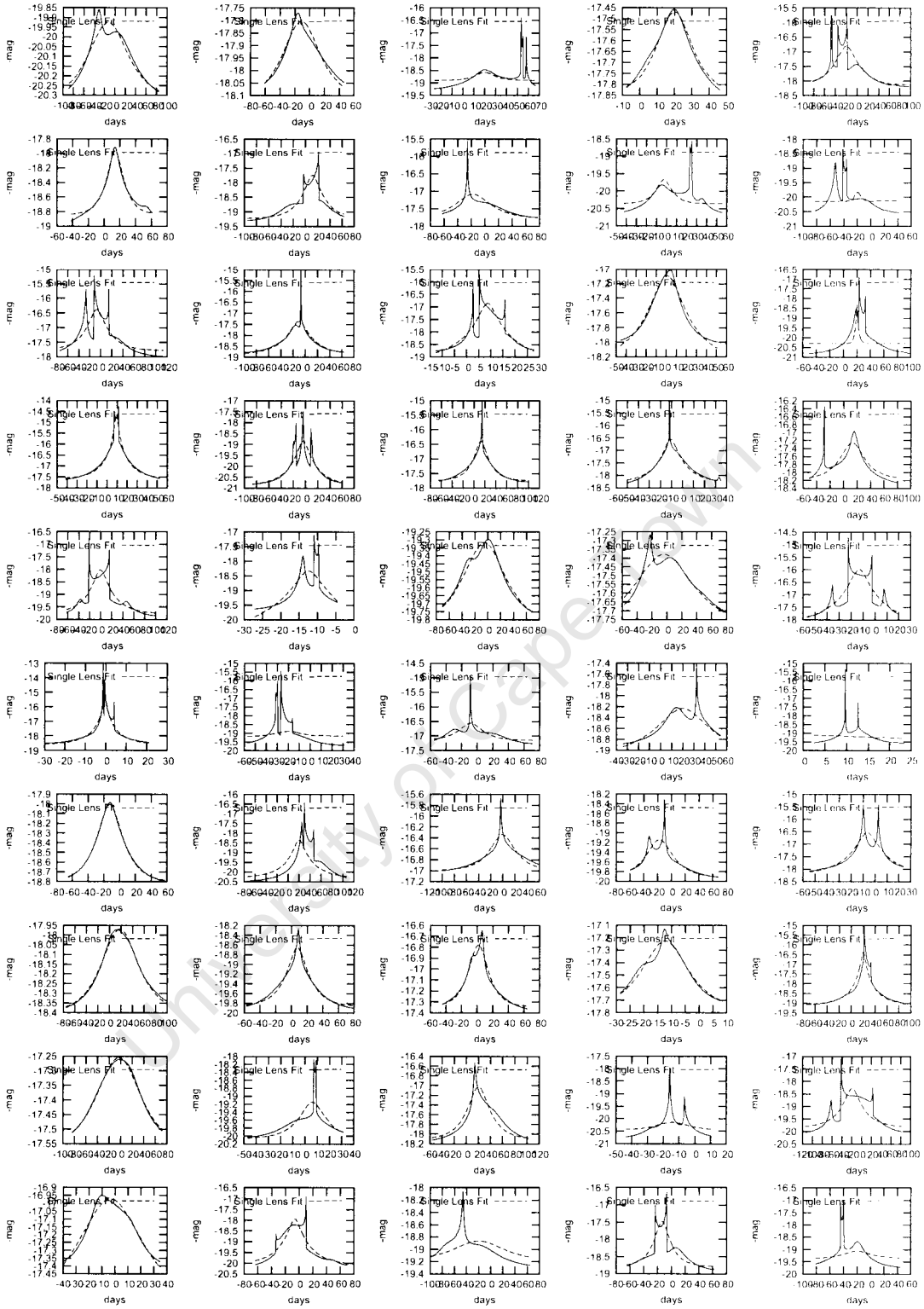

Figure 35: Single lens fits to 50 random binary events.

6.

Various polynomial approximation algorithms exist, e.g. [59]. Chebyshev's method was chosen because it is highly efficient and arbitrarily accurate, although the polynomials produced by the method are not the "Best Fit" or "MiniMax" polynomials. An example of a Chebyshev polynomial fit to a binary light curve is shown in Figure 33. The light curve in this figure is not well-approximated by the Chebyshev polynomial. A 5-coefficient polynomial is little more than an asymmetrical hump, whereas a 20-coefficient polynomial did not resolve the curve's multiple peaks. A 200-coefficient polynomial began to approximate the light curve as judged by eye, but at that stage the polynomial expansion had too many coefficients to achieve dimension reduction.

Figure 36 shows 20 more 20-coefficient Chebyshev polynomial approximations to randomly generated light curves. By eye, the figure indicates that smooth curves are approximated well. Unfortunately curves with discontinuities are very poorly approximated. Based on the passable performance of the Chebyshev polynomials, it was decided to add the 20 coefficients of a Chebyhev approximation to the feature set.

4.2.3 Linear Transformations

Linear transformation of a dataset is a fairly efficient operation and can lead to a transformed set with desirable properties as far as regression is concerned.

PCA

Perhaps the most famous of these is Principal Component Analysis [79], where an input set is transformed linearly to a new coordinate system where each axis is chosen in turn so-as to maximize the variance of the data along that axis. The direction of each new axis (component) of the transformation often yields insight into the data. Used in combination with the proportion of total variance of the data set along each new axis, we have a powerful tool for understanding our data as well as reducing its dimension. Understanding follows from analysis of the direction of

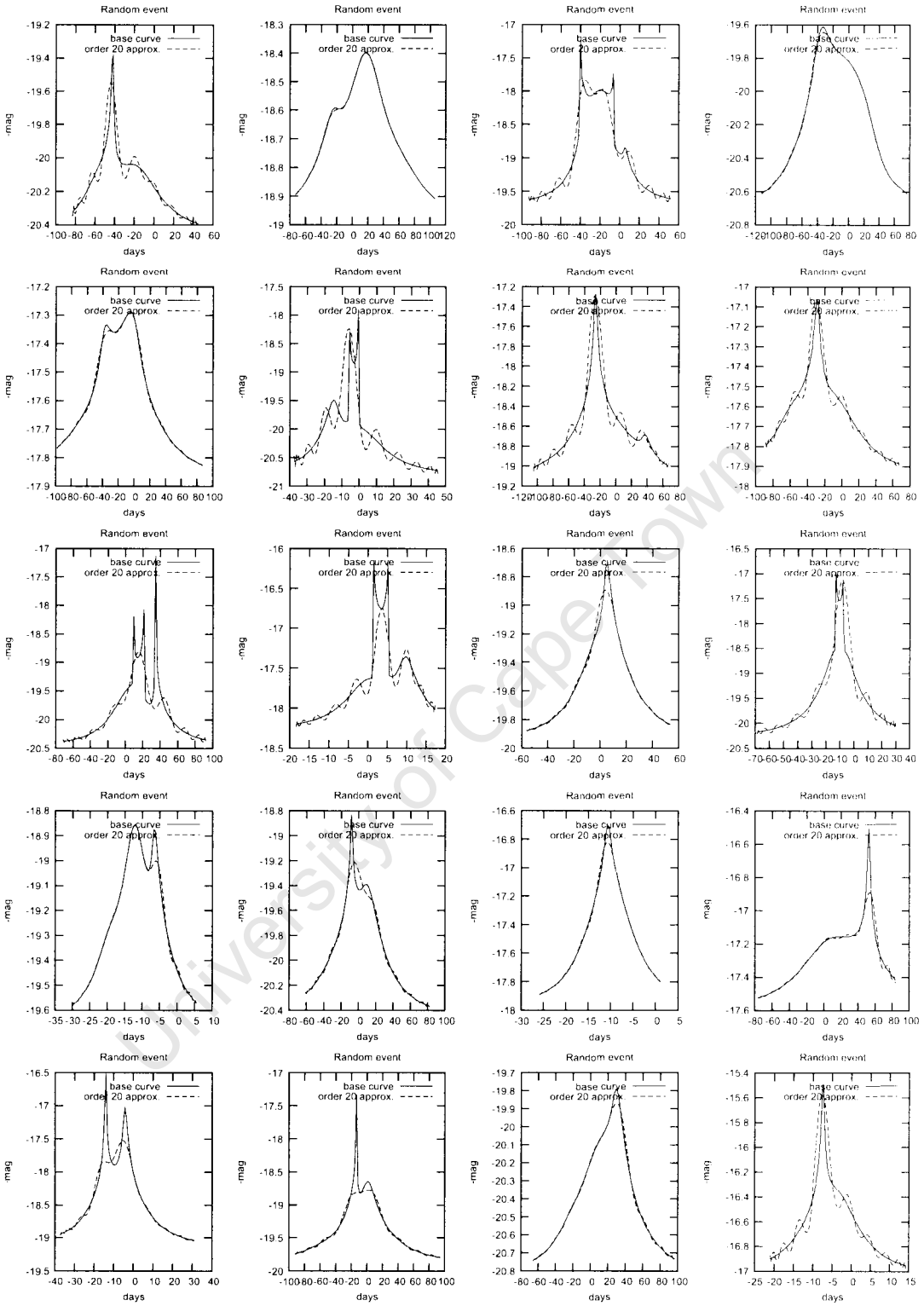

Figure 36: Chebyshev polynomial approximation of order 20 to 20 random binary lens fits.

new axes. Are they mostly along the axes of the original coordinate system? If so, the original data set is probably the most compact way of describing the data and we can skip the transformation. If the new axes point into directions in parameter space that are quite different to the original axes, the data are better described by a linear combination of the original input variables.

PCA was performed on our input data set of light curves with the WEKA software package [76]. The relative strength of the linear contribution of each light curve point by index for the first 4 principal components of our data set is shown in Figure 37. The first 10 principal components were added to our set of features for subsequent selection or rejection.

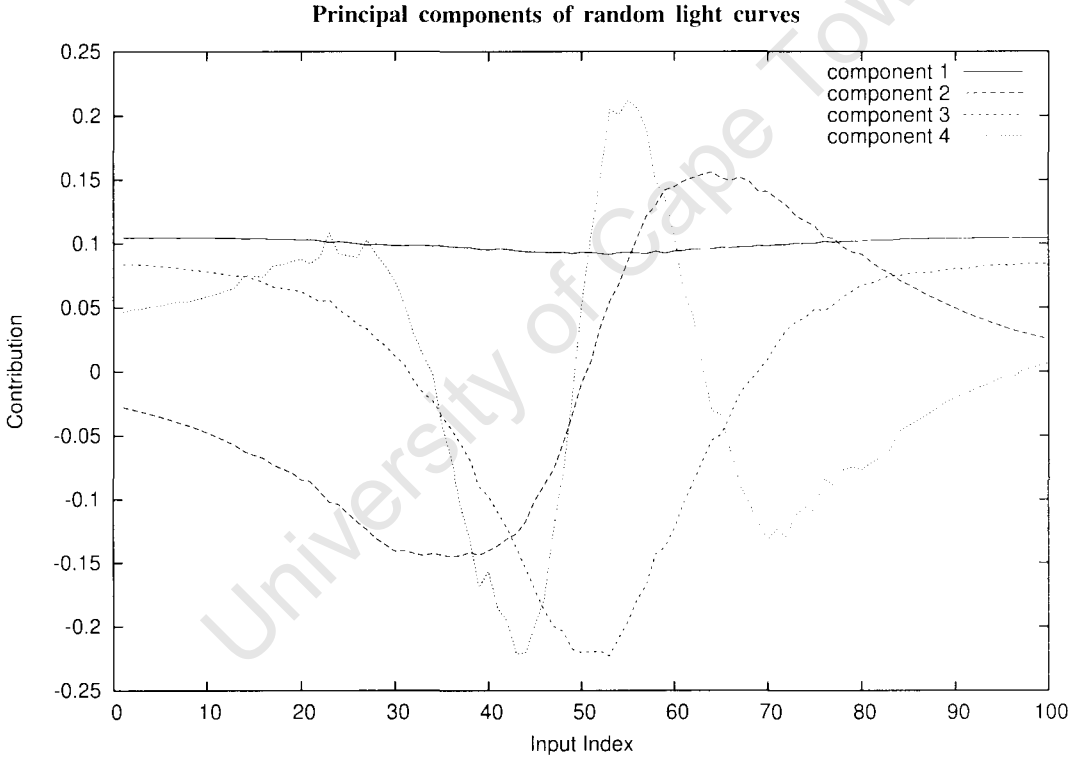

Figure 37: PCA results for binary light curves generated from Table 4. The first 4 components are shown, accounting for more than 96 per cent of the total variance.

One could interpret the figure in a straight-forward manner. The principal component, responsible for 88.4 per cent of the total variance, is almost the same as

Table 5: PCA of standard model data set, showing components covering more than 98 per cent of variance.

Component	Covered Variance	Cumulative
1	0.88413	0.88413
2	0.04257	0.9267
3	0.02819	0.95489
4	0.00601	0.9609
5	0.00574	0.96664
6	0.00331	0.96995
7	0.00291	0.97286
8	0.00225	0.97511
9	0.00204	0.97716
10	0.00155	0.9787
11	0.00146	0.98016

the mean of magnitudes for the light curve. The points closer towards the average peak position of the entire data set are slightly less important than the outlying points. This would appear to correspond to the variance introduced to all points by the un-lensed magnitude m_0 which simply parallel shifts the curve up or down. The second component appears to be symmetrical around the centre of the average of light curves in our sample. Curves that are asymmetrical with increased magnitude on the right will have a high value for the second component. Those skewed towards the left will have a negative value, and perfectly symmetrical curves will have a second component value close to zero.

The third component appears to be measuring the central magnitude against the magnitude in the wings, as the wings' contributions are positive whereas contributions from the centre are negative. We could speculate that the third component should be correlated with impact parameter b.

Additional components become harder to interpret. The fourth component appears to be another indication of asymmetry although more pronounced than the asymmetry measurement of the second component.

4.2.4 Genetic Programming

Feature construction by Genetic Programming is a fairly new technique (e.g. [80]). Genetic Programming is a versatile technique for evolving an analytical solution to any problem that can be written as a function, based on example data [81]. In its simplest form it is a straight-forward application of the Genetic Algorithm where individuals of the population are function trees that are evolved to approximate best the functional mapping implied by the example data. It should be noted that the simple Genetic Algorithm has been successfully applied to Microlensing problems in e.g. [82], and in fact also later in this thesis as a fine-tuning technique. Genetic Programming is a different technique, in that a function is evolved instead of a model parameter set for a pre-determined function.

This Section would be equally at home in the final regression Chapter, but Genetic Programming is applied here as a feature construction technique. The idea was to evolve an analytical, functional mapping between the input light curve and each standard model parameter in turn. The Genetic Programming algorithm was adapted from [83] to allow the use of mathematical functions. A population size of 2000 functions, to a maximum of 100 generations, was chosen. The fitness function was a standard $\Delta\chi^2$ over the training set which consisted of 10000 random events.

To avoid running out of memory and evolving down complicated cul-de-sacs, the maximum depth of any formula tree was set to 6. Apart from the Genetic Programming parameters, our input consisted of the set of input Microlensing variables and a set of operators that the algorithm chooses from to insert at branch nodes in the formula tree. The input to these runs was simply a single vector consisting of the magnitude values of a given light curve. Table 6 shows the list of operators used in the runs. These were selected ad-hoc based on past experience.

Figure 38 shows the result of a single Genetic Programming run for the four "difficult" parameters of the standard model. The plots are of the distribution of relative error for the data set of the predicted model parameter vs. the actual parameter value.

Table 6: Genetic Programming operators used for Feature Construction.

Operator	Description
Add	simple addition
Subtract	simple subtraction
Divide	protected division: check for zero denominator
Multiply	simple multiplication
Ln	protected logarithm: ln of absolute value, check for zero
Constant	a random constant within set range

The actual formulae generated by the run are not reproduced here as they ran over several pages and are not really human-readable despite being restricted to a maximum tree depth of 6. Analytical yes, elegant, no.

In this single run, Genetic Programming was largely unsuccessful at fitting the parameters a, θ, b and q. Not too much can be read into the results of just one such Genetic Programming run. For example, we don't know whether the algorithm failed to produce a formula for b because this was intrinsically difficult or just because we were unfortunate enough to choose a "bad" random seed. This type of uncertainty is a consequence of the stochastic nature of Genetic algorithms. The evolved formulae for the four model parameters were not added to our feature set because they failed to do much better than just choosing the centre of the model parameter range as a fit value.

Conclusions Genetic Programming is an exciting new technique with application in many fields. Unfortunately they were found wanting in the LGM binary lens model feature selection problem. The negative result was probably due to a combination of the genuine difficulty of the problem and the unsophisticated version of Genetic Programming applied. Sources like [81] suggest a variety of complicated techniques that may enjoy more success.

4.3 Feature Evaluation

The worth of a feature can be simply defined. The ultimate measure of its success is its predictive ability on an unseen test set. However, predictive success is a function of the set of features used, not any individual feature in isolation.

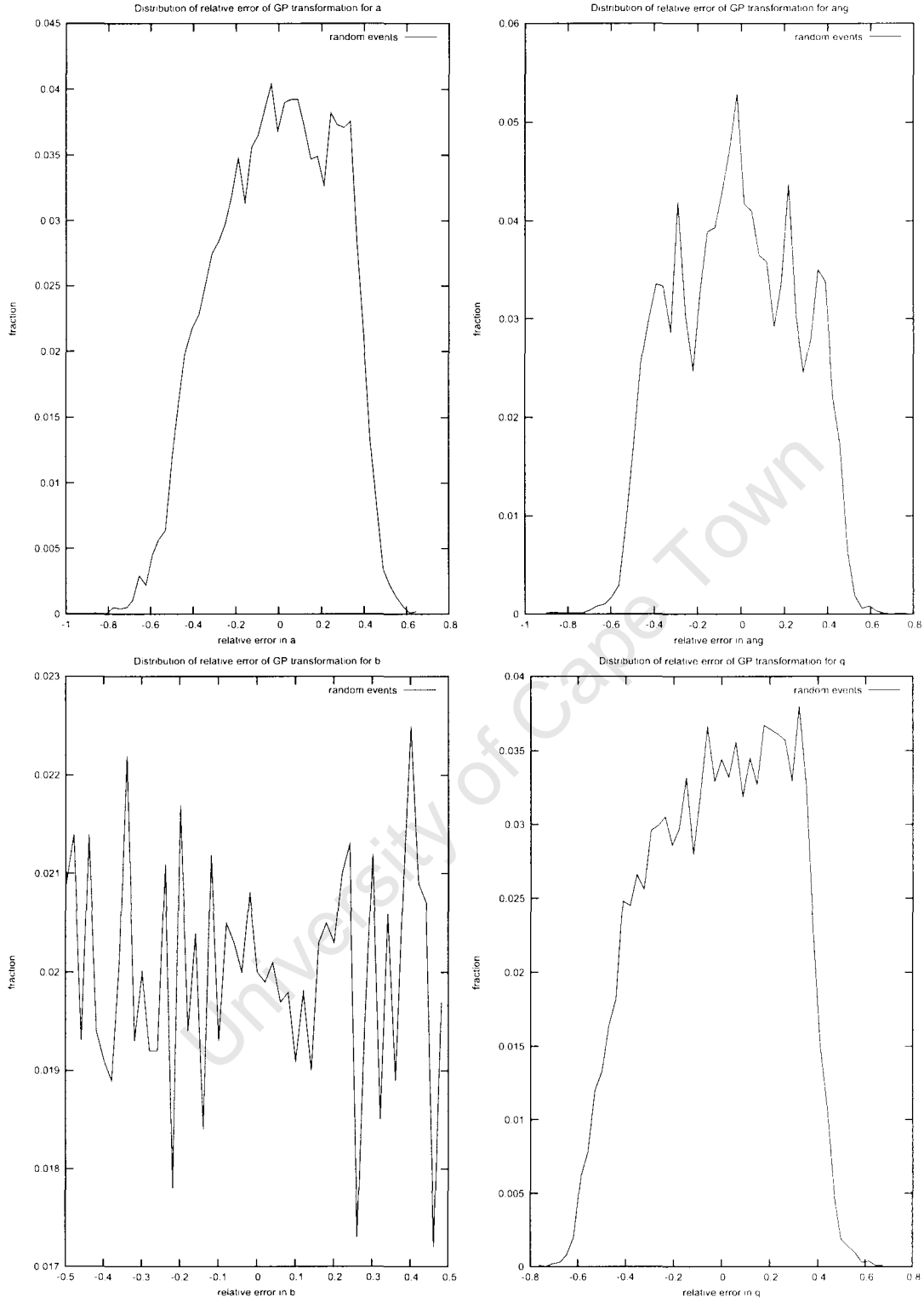

Figure 38: Relative error distribution for standard model parameters where Genetic Programming has provided a functional mapping as feature for subsequent regression. The results are discouraging as the error distribution encompass the entire range of the model parameters.

Many different feature selection algorithms exist. The most reliable evaluation method is exhaustively to train a classifier for each feature subset, something which is more than likely prohibitively time-consuming. Fortunately, alternative measures of feature sets' predictive power exist. These are discussed and applied with various search algorithms in the following Sections.

4.3.1 Filters and Wrappers

Feature selection algorithms can be divided into two categories: wrappers and filters. "Wrapper" methods train regression algorithms and use the classification accuracy of the trained regressor on an unseen test set as the measure of success. "Filter" methods use an alternative measure of predictive ability than the training of a regressor. The measure of predictive ability in the absence of actual evaluation on a test set is an approximation, but the relative efficiency of filter methods makes them vastly preferable to wrapper methods on real world problems with many potential features to select. Wrappers and filters may share the same feature selection search techniques, the distinction is solely in the way candidate feature subsets are evaluated.

4.3.2 Benchmark Computation time

Computation time and hence efficiency of the methods of evaluation and search were major factors in the selection of a feature set from the 600+ candidate features constructed from each light curve in Section 4.2. A few simple tests with small data sets and feature subsets indicated that some search and evaluation methods were much more efficient than others. There was little point in applying an inefficient method to the large feature set constructed in Section 4.2 so it was decided to perform a study to determine the relative running time and efficiency of various feature evaluation and search methods.

The timing study used a wrapper around the feature evaluation and search algorithms from the Weka data mining library [76] on a set of light curves consisting of different numbers of attributes in the form of evenly sampled magnitude points.

Table 7: Evaluation and Search algorithms in Speed Test.

Evaluation	Search
CfsSubsetEval	BestFirst
CfsSubsetEval	GeneticSearch
CfsSubsetEval	GreedyStepwise
ChiSquaredAttributeEval	Ranker
GainRatioAttributeEval	Ranker
InfoGainAttributeEval	Ranker
OneRAttributeEval	Ranker
ReliefFAttributeEval	Ranker
SymmetricalUncertAttributeEval	Ranker

Event parameters were selected from Table 4 as usual. The idea was not to find a subset of magnitude points to form a genuine feature set, but merely to test for computation time and consistency as a function of the number of both instances (light curves) and attributes. The full data set was reduced in turn to sets with 10, 20, 40 and 80 attributes and 50, 100, 200 and 400 instances and all combinations from these two ranges.

This left the choice of feature evaluation and search algorithms. Some feature selection algorithms were rejected outright on performance considerations. In particular, Weka's Support Vector Machine algorithm "SVMAttributeEval" was too slow on the test set so it would not be feasible to select features on our real data set of 600+ attributes and 5000 instances.

Some feature evaluation methods only work for nominal or discretized class variables. In other words, these are not really regression feature selectors but classification feature selectors. A large number of evaluators fell into this category so instead of rejecting them the variable we were attempting to recover was discretized into 10 categories on an equal-frequency basis.

The evaluation and search algorithm combinations chosen are shown in Table 7. The entries at the bottom the table all require the use of the Ranker algorithm for their search. CfsSubsetEval was given a choice of three search methods.

Table 8 shows timing results for a single run of the candidates in Table 7 for the smallest data set which contained only 50 instances and 10 attributes each.

Table 8: Ranked performance on small data set. 50 instances. 10 attributes. The time units are machine-dependent. Numbers should be interpreted by using their ratios.

Evaluation	Search	Time
CfsSubsetEval	GreedyStepwise	31
SymmetricalUncertAttributeEval	Ranker	83
InfoGainAttributeEval	Ranker	145
GainRatioAttributeEval	Ranker	150
ReliefFAttributeEval	Ranker	208
ChiSquaredAttributeEval	Ranker	315
CfsSubsetEval	GeneticSearch	345
CfsSubsetEval	BestFirst	402
OneRAttributeEval	Ranker	1583

These runs were fast in general but there was a lot of variation in running time. CfsSubsetEval with GreedyStepwise search was more than twice as fast as its nearest rival. SymmetricalUncertAttributeEval with Ranker, which was in turn twice as fast as its nearest rival InfoGainAttributeEval with Ranker. The Ranker search method appears to be effective for small data sets. OneRAttributeEval with Ranker is considerably slower than any other method and almost 50 times slower than the winner. Finally, the speed of CfSubsetEval depends strongly on the search algorithm it is used with; it ranked fastest using GreedyStepwise and second slowest using BestFirst.

Next are results for the largest data set under test, with 400 instances and 80 attributes. Comparison with Table 8 should provide us with a crude estimate as to how evaluation and search time scales with the number of attributes and instances. CfsSubsetEval with GreedyStepwise is still the fastest. CfsSubsetEval with BestFirst makes a dramatic move up the table from second slowest to second fastest. It is only about twice as slow on the large data set than it is on the small one, indicating that this method scales very well to large data sets. ReliefFAttributeEval with Ranker drops dramatically down the table to the last spot and a speed more than three times as slow as the nearest method, indicating very bad scaling for this method. The rest of the methods are all roughly 5-10 times as slow on the large data set as they are on the small one.

Table 9: Ranked performance on large data set. 400 instances, 80 attributes.

Evaluation	Search	Time
CfsSubsetEval	GreedyStepwise	201
CfsSubsetEval	BestFirst	912
SymmetricalUncertAttributeEval	Ranker	1170
InfoGainAttributeEval	Ranker	1192
GainRatioAttributeEval	Ranker	1303
ChiSquaredAttributeEval	Ranker	1408
CfsSubsetEval	GeneticSearch	2504
OneRAttributeEval	Ranker	6893
ReliefFAttributeEval	Ranker	22675

This very crude speed test has at least raised doubts on grounds of execution time about a single contender from our choice of evaluation and search algorithm, namely ReliefFAAttributeEval with Ranker. Theoretical performance numbers are available for most of these algorithms, but it was decided to test the Weka implementation that would be used in the final feature selection.

4.3.3 Benchmark Accuracy

Section 4.3.2 provided crude timings of the various candidate algorithms to be used for feature selection in Section 4.4. Another equally important consideration is the effectiveness of the candidate feature selection methods.

This Section attempts to benchmark the accuracy of the candidate methods against a test data set for which the attributes with most merit as predictors are known in advance. The candidate that fares well in this Section as well as the timing Section is chosen to perform feature selection on the full light curve data set.

One could expect the various evaluation and search algorithms to have different degrees of success based on the difficulty of the selection problem. Three data schemes of increasing difficulty were created as benchmarks. Each set consisted of 500 instances of 100 attributes each. 20 of the 100 attributes were relevant to the regression and the remaining 80 were simply random numbers between 0 and 1.0. Nominal evaluators split the target value into four categories on an equal-frequency basis.

In the three different data schemes, the targets were

1. simply the sum of 20 random attributes,

$$\sum_{i}^{20} a_i \tag{51}$$

2. the sum of the squares of 20 random, centred attributes,

$$\sum_{i}^{20} (a_i - 0.5)^2 \tag{52}$$

3. the sin of the sum of squares of 20 random, centred attributes

$$\sin\left(\sum_{i}^{20} (a_i - 0.5)^2\right). \tag{53}$$

Results are shown in Tables 10, 11 and 12. The tables show a startling variance in the relative accuracy of the methods employed. None of the benchmarks was an easy target. In all cases the target value was a function of all twenty relevant parameters. Case 1 was a simple sum and the correlation-based evaluator CfsSubsetEval fared extremely well with a true positive success rate of 85 per cent. In the next two cases we deliberately destroyed correlation between attributes and the target variable by using squares of centred values of a flat distribution and summing them. The tables were turned and CfsSubsetEval fared poorly at both problems. Although all methods struggled with case 3, ReliefFAttributeEval appeared to be a fairly consistent performer on all cases. OneRAttributeEval boasted impressive performance on the more difficult cases but fared worst on the easy case 1. Note that the success of an evaluator that simply chose 20 parameters out of 100 at random would have scored a respectable 20 per cent success rate. It appears that CfsSubsetEval fared so badly at uncorrelated but relevant attributes that it actively deselected them in case 2.

4.3.4 Conclusion from the Benchmarks

Results from the previous Section warned against naive application of a single evaluator and search algorithm combination. Attributes that are highly correlated with the target attribute will most likely be selected by CfsSubsetEval but relevant

Table 10: Case 1. Benchmark Accuracy

Evaluation	Search	True Positive	False Positive
ReliefFAttributeEval	Ranker	0.85	0.15
CfsSubsetEval	GreedyStepwise	0.85	0.0
CfsSubsetEval	BestFirst	0.85	0.0
SymmetricalUncertAttributeEval	Ranker	0.45	0.55
InfoGainAttributeEval	Ranker	0.45	0.55
GainRatioAttributeEval	Ranker	0.45	0.55
ChiSquaredAttributeEval	Ranker	0.45	0.55
CfsSubsetEval	GeneticSearch	0.4	0.6
OneRAttributeEval	Ranker	0.35	0.65

Table 11: Case 2. Benchmark Accuracy

Evaluation	Search	True Positive	False Positive
OneRAttributeEval	Ranker	0.55	0.45
ReliefFAttributeEval	Ranker	0.45	0.55
SymmetricalUncertAttributeEval	Ranker	0.3	0.7
InfoGainAttributeEval	Ranker	0.3	0.7
GainRatioAttributeEval	Ranker	0.3	0.7
ChiSquaredAttributeEval	Ranker	0.3	0.7
CfsSubsetEval	GeneticSearch	0.1	0.9
CfsSubsetEval	GreedyStepwise	0.05	0.95
CfsSubsetEval	BestFirst	0.05	0.95

Table 12: Case 3. Benchmark Accuracy

Evaluation	Search	True Positive	False Positive
OneRAttributeEval	Ranker	0.3	0.7
CfsSubsetEval	GreedyStepwise	0.3	0.7
CfsSubsetEval	BestFirst	0.3	0.7
ReliefFAttributeEval	Ranker	0.2	0.8
CfsSubsetEval	GeneticSearch	0.2	0.8
SymmetricalUncertAttributeEval	Ranker	0.15	0.85
InfoGainAttributeEval	Ranker	0.15	0.85
GainRatioAttributeEval	Ranker	0.15	0.85
ChiSquaredAttributeEval	Ranker	0.15	0.85

attributes that are not correlated in a linear sense will not be selected by this method. In fact no single attribute selection scheme attained consistently good results and several methods should be used on the final data set. Due to speed and high success rate for correlated attributes, CfsSubsetEval with GreedyStepwise will be used. ReliefFAttributeEval with Ranker will also be used due to its relative success with all cases above. Ironically it was the slowest method under test in the performance benchmarks. Finally, InfoGain with Ranker will also be used, due to the fact that it is fast and works on discretized data, which is an alternative direction to take in our feature selection plan.

All algorithms in the above tests have a number of parameters to configure their use. WEKA defaults were used for all runs, but these were not necessarily optimal and are problem-dependent. Nonetheless, experimentation with those parameters indicated that the settings were not critical in determining the success of any of the methods attempted above and that the conclusions hold.

4.4 Feature Selection Results

In this Section, a set of features constructed from generated, binary light curves is selected as input attributes for optimal regression. We narrowed the selection methodology down to just three algorithms in Section 4.3 and these three methods will be applied to the full feature set generated in Section 4.2.

We get a little ahead of ourselves because pre-processed light curves were used for feature selection. The pre-processing algorithm as well as motivation for its use are discussed in Section 6.1.3. Suffice to say at this stage that the raw data created by simply generating light curves from the parameter ranges in Table 4 required a pre-processing stage in order to tremendously enhance the accuracy of feature selection and regression. The process translates and scales light curves to a common base magnitude and start time. Although accuracy is increased for the SBLM parameters a, θ, b and q, it is no longer possible to fit for t_c, t_m and m_0 after pre-processing. These are fitted for in a subsequent fine-tuning stage.

Table 13: Summary of all potential features for regression.

Name	Description	Number of features in set
x_i	x-axis of curve; time, if not processed	100
y_i	y-axis; processed magnitude	100
$turnx_i$, $turny_i$	(x, y) of first seven light curve extrema	14
$xmax$	Maximum x-value in curve	1
$xmin$	Minimum x-value in curve	1
$xadev$	Absolute deviation of all x-values	1
$xsdev$	Standard deviation of all x-values	1
$xskew$	Skew of all x-values	1
$xcurt$	Kurtosis of all x-values	1
$ymax$	Maximum y-value in curve	1
$ymin$	Minimum y-value in curve	1
$yadev$	Absolute deviation of all y-values	1
$ysdev$	Standard deviation of all y-values	1
$yskew$	Skew of all y-values	1
$ycurt$	Kurtosis of all y-values	1
$slope_i$	First derivative of y to x	99
$slopeturnx_i$, $slopeturny_i$	(x, y') of first seven extrema	14
$slopemax$	Maximum y'-value in curve	1
$slopemin$	Minimum y'-value in curve	1
$slopeadev$	Absolute deviation of all y'-values	1
$slopesdev$	Standard deviation of all y'-values	1
$slopeskew$	Skew of all y'-values	1
$slopecurt$	Kurtosis of all y'-values	1
$slopslope_i$	Second derivative of y to x	98
$5smooth_i$	5-point smoothed y	95
$20smooth_i$	20-point smoothed y	80
$chebyfit_i$	First 20 Chebyshev-polynomial coefficients of fit to y	20
pca_i	First 10 Principal Components of y	10

We summarize the full list of features available (all derived from 100 time-magnitude points) in Table 13.

For maximum accuracy, a separate feature set was derived for each binary model parameter. Differentiating feature sets in this way may provide some insight into model behaviour.

4.4.1 CfsSubsetEval with GreedyStepwise

The CfsSubsetEval with GreedyStepwise run completed within two hours on a 2.8GHz PC running 10-fold cross-validation. The features selected for each of the

Table 14: CfsSubsetEval with GreedyStepwise feature selection results for all fittable Parameters

a	θ	b	q
y0	y81	turnx7	turnx4
y57	turnx1	slopesdev	tangent31
turny0	turny1	chebyfit6	tangent36
xcurt	turnx3	chebyfit19	tangent43
ysdev	turnx5		tangent44
yvar	tangent1		tangent46
tangent0	tangent44		tangent54
tangent22	tangent46		tangent55
tangent27	tangent49		tangent56
tangent29	tangent50		tangent57
tangent33	tangent53		tangent66
tangent66	tangent55		tangent67
tangent71	tangent97		slopeturnx1
tangent78	5smooth16		slopeturnx2
tangent82	5smooth79		slopeturny2
tangent88	20smooth6		slopeturnx3
tangent96	chebyfit1		slopeturnx4
tangent98	pca2		slopeturny4
slopeturny0			slopeturnx5
slopeslope9			slopeturny5
slopeslope36			slopeadev
slopeslope61			slopesdev
slopeslope90			slopevar
chebyfit6			slopecurt
pca1			slopeslope43
			slopeslope45
			slopeslope46
			slopeslope47
			slopeslope48
			slopeslope49
			slopeslope50
			slopeslope51
			slopeslope52
			pca23

four standard model parameters which are fittable after pre-processing are shown in Table 14.

Table 14 contains a varying number of features per model parameter. The CSF.GS method selects only those features that are correlated with the target values, but not highly correlated with values that have already been selected. Does the selection make intuitive sense?

Features that had the highest predictive power for projected orbital separation (a), according to this algorithm, are three brightness values at the start and close

to the centre of the light curve. The brightness of the first extremum was also important. A hand-waving argument indicates that the first extremum will indeed carry some information on the value of a; from Section 2.1.3 we know that the value of a is crucial in determining the size and position of spikes in the light curve due to its importance (with q) in generating the magnification caustic pattern of a given event. In the extreme case where a is outside of the "lensing zone" (roughly $0.618\ \theta_E < a < 1.618\ \theta_E$) binary effects would be hard to observe, which implies that the only extremum in the curve will be at the single lens amplification peak. The parameters *ysdev* and *yvar* also provide an indication of the extent of binary features in the light curve and have been selected. After these, the algorithm selected the value of tangents at points all along the light curves, as well as second derivative information across the curve. The selection of the kurtosis of time values (*xcurt*) will have to prove its worth as this parameter seems intuitively meaningless after pre-processing: light curves have been scaled and translated to a common time starting point and scale. Finally, a middle-order Chebyshev polynomial coefficient and the second PCA coefficient are also selected.

The crossing angle θ should also be connected to the best indicator of structure in the light curve, due its role in determining which path the source takes through a caustic pattern. After one sample of the brightness curve near its end, the following parameters chosen made sense: the relative time-position of most of the turning points and one brightness as well. In fact, somewhat curiously, the positions of light curve troughs were chosen, not the peaks in between. This kind of feature provides high-level information on the structure of the curve. After those came a selection of tangents at the very edges and near the centre of the curve. The symmetry in selection was pleasing and helped the algorithm to pass another sanity check. Our light curve data set is symmetrical around the average peak position and this should be reflected in the feature selection at all times. A Chebyshev Polynomial coefficient and a PCA coefficient of low order were also selected.

Selected features for impact parameter b were strange. The most predictive

feature according to CfsSubsetEval with GreedyStepwise selection was the time-position of the eighth turning point and that did not make intuitive sense. After that the standard deviation of tangents (*slopedev*) was selected, a medium-order Chebyshev coefficient and a very high order Chebyshev coefficient.

Features selected for mass ratio q seemed fairly sensible. They included some tangent points, a large amount of turning points in the first derivative, most of the first derivative statistics and a large number of second derivatives at the centre of the curve.

Although most of the selections did make sense, there were some that were almost certainly incorrect, for example the kurtosis of x-values. These examples make it clear that feature selection algorithms could not be completely trusted in all cases.

4.4.2 ReliefFAttributeEval with Ranker

The ReliefFAttributeEval with Ranker run would have taken days to complete on an 2.8GHz PC in stark contrast to the CfsSubsetEval with GreedyStepwise selection run. This was expected based on the results of the speed tests performed in Section 4.3.2. To speed up the process, the algorithm was instructed to use only 1000 examples at a time during its nearest neighbour comparison phase, and no cross-validation was performed. Results are shown in Table 15.

The algorithm selected a much more uniform set of features across the four model parameters we are investigating. Tangents played a large part in all selections. The top features in all cases were light curve tangents from the centre of the curve, followed by second derivatives close to the centre, extremum height information and the occasional y-value itself.

Although the selection seems feasible, it is clearly very different from that made with the CfsSubsetEval with GreedyStepwise selection algorithm, even though both algorithms have the same goal and should ideally have given similar results.

Table 15: ReliefF with GreedyStepwise Feature Selection Results for All Parameters

a	θ	b	q
tangent50	tangent49	tangent52	tangent52
tangent51	tangent50	tangent53	tangent39
tangent52	tangent51	tangent44	tangent56
tangent48	tangent48	slopevar	tangent41
tangent49	tangent52	tangent55	tangent61
tangent53	tangent47	tangent54	tangent38
tangent47	tangent53	slopesdev	tangent44
tangent65	tangent45	tangent47	tangent43
slopeslope49	tangent46	tangent46	tangent62
tangent46	tangent40	tangent40	tangent45
tangent60	tangent54	tangent56	tangent40
turny2	turny2	tangent45	tangent54
turny4	tangent44	tangent48	tangent42
tangent66	turny4	tangent43	tangent59
slopeslope51	slopeslope34	slopeadev	tangent46
tangent54	slopeslope35	tangent51	tangent57
tangent64	slopeslope59	turny0	tangent55
slopeslope67	tangent43	turny2	tangent60
tangent45	tangent41	tangent58	tangent47
tangent97	turny1	tangent57	slopeslope67
tangent44	turny0	tangent41	tangent53
tangent67	tangent55	tangent23	tangent63
turny1	tangent36	tangent49	tangent58
tangent59	tangent42	tangent24	tangent35
tangent96	tangent59	turny4	tangent36
tangent95	tangent37	tangent93	tangent37
tangent37	tangent39	tangent50	tangent48
ymax	tangent56	ymax	tangent34
tangent93	turny3	tangent92	tangent64
tangent94	tangent60	tangent42	tangent51
tangent43	tangent27	turny1	slopeslope37
5smooth50	tangent57	tangent62	slopeslope31
5smooth51	turny6	tangent87	tangent65
y52	tangent35	tangent21	slopeslope39
tangent61	tangent38	tangent86	tangent67
5smooth49	ymax	tangent94	turny4
y49	ysdev	yadev	tangent33
5smooth46	yadev	turny3	tangent68
5smooth45	tangent29	tangent39	tangent66
5smooth48	tangent61	tangent22	slopeslope33
ysdev	yvar	tangent90	slopeslope41
5smooth47	y39	tangent91	tangent50
yadev	5smooth45	tangent77	slopeslope57
slopeslope53	y40	tangent95	slopeslope59
y51	5smooth51	5smooth46	y38
turnx1	slopeslope33	ysdev	turny0
y47	y47	5smooth45	y36
tangent41	slopeturny7	tangent61	y35
tangent58	5smooth50	tangent20	y37
y53	tangent62	tangent96	turny2

4.4.3 InfoGain with Discretization and Ranker

A third and final algorithm was used to select features and to compare to the two algorithms chosen in the benchmarking Section. It was chosen primarily because of wide usage in the literature but also because it works on discretized data, a new consideration in our feature selection regime. The algorithm was fast in these calculations, completing in minutes on an 2.8GHz PC.

Data were discretized by simply dividing the available ranges into ten bins each. More complicated discretization schemes exist, for example supervised algorithms that attempt to discretize data in conjunction with a measure of classification performance. We chose a simple scheme here to avoid confusing the issue. 10-fold cross-validation was applied.

Results are shown in Table 16.

Once again a set of features were selected that seemed feasible. This time there were some similarities with the selection made by the CfsSubsetEval with GreedyStepwise selection algorithm. The PCA parameter $pca1$ is selected for a in both cases, as are some extremum information and values of brightness close to peak. The selection for θ has more in common with the ReliefFAttributeEval algorithm's selection but its selection for q with heavy emphasis on the smoothed curve is something new.

4.5 Conclusions and Comparison of Feature Selection Algorithms

Some conclusions can be drawn from the feature selection algorithms run above in Section 4.4.

1. Constructed features play the dominant role. Raw magnitude values are not selected at all by CfsSubsetEval. Instead, the slope curve and turning points in the light curve and slope curve play a major role. Even more esoteric features like PCA components and Chebyshev Polynomial coefficients are frequently selected. These constructed, selected features will be compared with more mundane benchmarks in the Sections to follow.

Table 16: Features selected by InfoGain with Ranker on discretized data.

a	θ	b	q
pca1	ysdev	slopevar	20smooth41
y49	yadev	slopesdev	20smooth40
5smooth48	ymax	slopeadev	20smooth42
5smooth49	yvar	tangent19	20smooth43
5smooth47	pca0	tangent20	20smooth39
y51	yave	tangent21	20smooth44
turny1	pca24	tangent77	20smooth38
turny3	chebyfit0	tangent22	5smooth49
y52	20smooth30	tangent79	5smooth48
y48	20smooth29	tangent78	5smooth47
y50	20smooth28	tangent80	20smooth45
5smooth50	20smooth31	tangent18	5smooth46
5smooth46	20smooth27	tangent23	5smooth50
y53	20smooth32	yadev	y49
y47	tangent49	tangent76	20smooth37
y54	tangent48	tangent81	pca24
5smooth45	tangent47	turny0	20smooth46
5smooth51	20smooth26	tangent46	5smooth51
20smooth41	20smooth33	tangent45	y52
5smooth44	tangent50	tangent53	y51
y46	tangent51	tangent75	y48
5smooth52	20smooth50	tangent47	5smooth45
20smooth42	20smooth51	tangent24	y53
20smooth39	20smooth34	tangent55	20smooth36
20smooth40	20smooth25	tangent17	y50
20smooth43	tangent46	5smooth26	pca0
y45	20smooth49	tangent52	20smooth47
y55	20smooth48	tangent82	5smooth52
20smooth38	tangent52	tangent74	y47
5smooth53	20smooth35	5smooth27	y54
5smooth43	20smooth47	5smooth25	yave
y56	x35	tangent42	20smooth35
y57	x67	20smooth14	5smooth53
5smooth54	x73	20smooth16	5smooth44
20smooth44	x55	tangent44	20smooth48
20smooth37	x83	tangent56	y55
y58	xave	tangent48	20smooth34
y59	x34	5smooth24	5smooth43
y44	x84	5smooth69	5smooth54
5smooth42	x54	20smooth67	y56
chebyfit4	x9	tangent54	20smooth49
20smooth45	x46	tangent43	y46
5smooth55	x66	tangent58	y45
y43	x49	20smooth66	pca1
20smooth36	x60	tangent25	20smooth33
5smooth56	x48	5smooth68	chebyfit0
5smooth41	x61	y72	5smooth42
turny4	x47	y71	20smooth50
y60	x7	tangent51	y57
5smooth57	x8	20smooth13	5smooth55

2. Different algorithms did not necessarily select similar feature sets. CfsSubsetEval and ReliefFAttributeEval broadly agreed on the selected feature sets but there were differences, especially in the selection for the mass ratio q.

3. Selection of obviously incorrect parameters like the kurtosis of x-values (meaningless after pre-processing) did not instill confidence in feature selection algorithms. Nonetheless, intuitively sensible parameters were chosen in most cases.

These feature selections were carried over to the following Chapter for fitting accuracy comparisons.

5 Fitting the Standard Model

This thesis is directed at easing the difficulty that conventional fitting techniques have with fitting model light curves to observations. Hundreds of regression and curve-fitting techniques are in use in every field of science and industry but each problem has specific properties which make some methods more applicable than others. The "conventional" techniques of modellers mostly start out with an initial guess somewhere in search space and attempt to approach the global minimum in the regression surface, most often χ^2 as a function of model parameters when fitting a curve, by iteration. The most effective of these (such as Levenberg-Marquardt) use gradient information to make each iteration progress optimally towards the solution, with various refinements to avoid pitfalls such as overshooting the solution [59]. These methods mostly fail dismally with the SBLM and light curve data, due to a number of problems discussed in 3.3.

The "unconventional" (in the context of regression in Astronomy) techniques discussed here attempted to overcome the challenges posed in the LGM scenario by arriving at a solution through an entirely different process that does not involve normal regression iteration, or else changing the parameter space and regression function so that a conventional technique may succeed. Powell's Method, Library Methods, Levenberg-Marquard's Method, Amoeba (Downhill Simplex) and others including gradient methods were considered "conventional" while Genetic Algorithms, statistical data mining (such as those in the WEKA software package) and neural networks were considered "unconventional" in this role.

5.1 Conventional Optimisation / Regression

A general discussion on the topic can be found in e.g. [59], but this Section presents a summary in the context of LGM binary lens light curve fitting.

One could divide conventional minimisation algorithms into gradient- and non-gradient methods. Non-gradient methods are often simple, and rely on evaluation of the function that is to be minimised, not its gradient. Gradient methods also

require the evaluation of the function's gradient by analytical or numerical means, often leading to faster convergence than non-gradient methods.

Conventional methods are generally based on finding the minimum of a regression hyper-surface that is a measure of the "closeness" of a test solution to the observed light curve. $\Delta\chi^2$ is a commonly used as this measure of goodness of fit.

A modern approach that is being successfully employed in binary lens fitting (e.g. [70]) is the stochastic Markov Chain Monte Carlo method, again using $\Delta\chi^2$ as a measure of goodness of fit.

The data mining and Artificial Neural Network-based techniques work on a different principle. There is no measure of goodness-of-fit and no iterative search in model parameter space. The mapping between model parameters and input light curve, or derived features of it, is directly approximated by a complicated function that is derived from example data.

There are several advantages to using conventional methods:

1. They work very well on problems with simple regression surfaces that have only a few minima and are relatively smooth.

2. They are very fast when the $\Delta\chi^2$-function is easy to calculate. Gradient methods such as Levenberg-Marquardt are currently the industry standard, primarily due to their speed.

3. They are well-understood. Their theory is generally simple and their behaviour is predictable.

4. Algorithms often provide error information in the form of covariance matrices or uncertainties in the solution, provided noise is Gaussian and other such mild assumptions.

5. Any local solution can be run to machine accuracy, provided one has the time.

Yet, the disadvantages of conventional minimisation techniques render them almost useless when applied in isolation to some problems, in particular fitting LGM binary lens light curves. The disadvantages that bring this disaster about include:

1. Most conventional algorithms will only find the nearest minimum, not the global minimum. To converge on the global minimum they require a starting point that is close to this minimum to begin with. How "close" depends on the geometry and smoothness of the regression surface.

2. Gradient methods require a smooth regression surface. They also require a preferably inexpensive method to calculate gradients for the function that is being minimised.

3. Some methods are not robust, i.e. they occasionally fail to converge at all. A well-known example is the Newton-Raphson algorithm.

These methods are to be put to use fitting LGM binary light curves. This Section will illustrate the shortcomings of conventional methods by attempting to fit simple LGM binary lens light curve models to curves generated by the same model.

5.1.1 χ^2-minimisation by conventional algorithms

Some pessimism is in order here, as LGM binary light curve regression surfaces do not fulfill any of the requirements for successful application of conventional regression methods.

1. The regression surface is utterly non-linear and densely populated with local minima.

2. Gradients of the regression surface are expensive to calculate.

3. We have already determined that the size of the regression well in a typical problem is so small that all parameters are required to be within 2 per cent of the correct solution if downhill methods are to succeed (see Section 3.3.3).

The next few Sections illustrate some typical problems when using conventional methods to fit simple LGM models to data.

Amoeba Algorithm and Local Minima

Section 3.3 mentions the existence of numerous local minima as a major challenge to conventional methods when fitting binary lens models to data. The Amoeba method discussed in 3.3.3 is well-suited to illustrating this problem. The success rate of Amoeba fits was plotted as a function of the starting error in 3.3.3. The amoeba method's success or failure is determined almost exclusively by the size of the convergence well, as is the success rate for all downhill methods that are incapable of escaping a well.

Fit history

Figure 39 displays plots of the course of Amoeba fits to light curves. $\frac{\Delta\chi^2}{d.o.f}$ is plotted as a function of time for a sample of 6 randomly chosen light curves. These plots provide some insight into an algorithm's convergence properties. In this case Figure 39 relates a sad tale of premature convergence as $\frac{\Delta\chi^2}{d.o.f}$ often drops sharply indicating fast convergence. but remains high which means that the algorithm is converging to a local minimum.

Conclusions

A few observations from the experiments performed above.

1. Amoeba does not work by itself. Simple downhill algorithms such as Amoeba are mostly useless for LGM binary fitting unless a highly accurate initial guess is available. From Section 3.3.3 this guess would have to be accurate to better than about 2 per cent of the allowed parameter range in all parameters.

2. Neither would other "greedy" methods. The Amoeba method was used here as a representative of all methods that simply rush towards the nearest local minimum. Figure 39 indicates that the regression space is littered with local

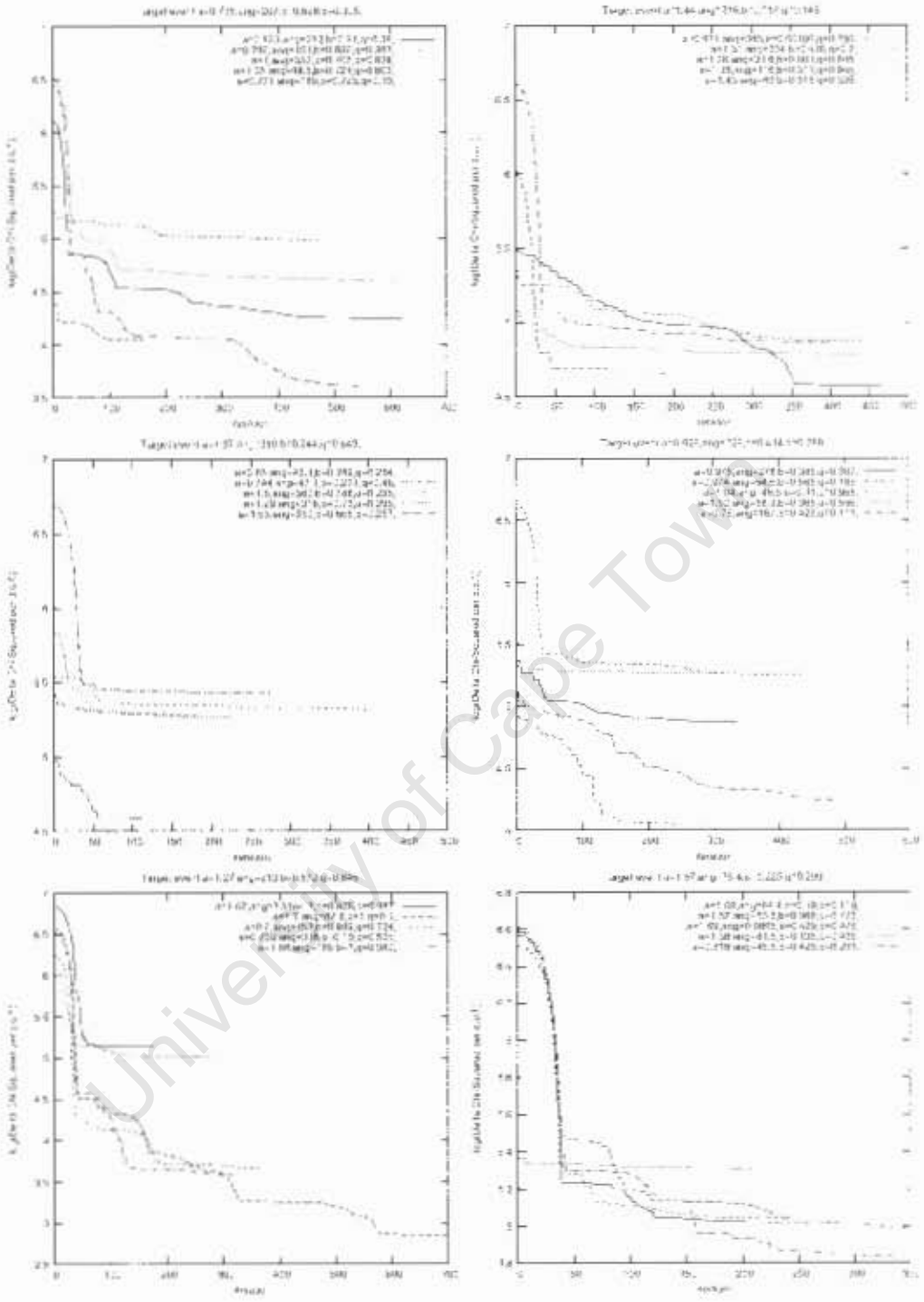

Figure 39: Fits to six random SBLM light curves. Each panel contains the convergence history of five fits to an event from different starting points. The logarithmic y-scale for $\frac{\Delta\chi^2}{d.o.f}$ indicates that no fit succeeded.

minima and these minima would trap any algorithm that was incapable of extricating itself from a local convergence well.

3. Amoeba converges rapidly. Apart from the bleak results offered in Figure 39, it at least shows that the Amoeba method is capable of converging very rapidly to a local minimum and this ability makes it useful as a fine-tuning algorithm.

5.2 Common procedures

5.2.1 Parameter space and ranges

The range of parameters shown in Table 4 was chosen from which to produce simulated curves for fitting.

These ranges were specifically chosen to include those parts of the binary lens parameter space that produce light curves that are discernible from single lens light curves. Had this not been the case, there would have been no photometric basis for fitting a binary lens model in the first place. The ranges chosen are in fact smaller than the full range where perturbations occur. For example, only values of a between 0.6 θ_E and 1.7 θ_E are considered, although values of a from as low as 0.2 θ_E to even greater than 5 θ_E can cause perturbations to the single lens curve with reduced likelihood (e.g. [62]). The choice made was a compromise between generality and precision, as the fitting success rate declines with the broadening of the allowed parameter space. The range of a that is being investigated coincides with the lensing zone (see [21] for a discussion on this "zone") and so includes most binary events that are likely to be detected.

5.2.2 Light curves for fitting

Fitting methods were developed and tested on simulated light curves, which differed from real light curves in the following respects:

1. Completeness. Artificial light curves had no gaps in coverage, as is frequently a problem with real data.

2. Artificial curves were initially noise-free. Although no noise was added to light curves, an uncertainty of 1 per cent was assumed throughout.

3. Artificial curves consisted of exactly 100 evenly distributed magnification points, starting at a random time t_{start}, where

$$-3 < \frac{t_{start} - t_m}{t_e} < -2 \tag{54}$$

and ending at a random time t_{end} in the range

$$2 < \frac{t_{end} - t_m}{t_e} < 3 \tag{55}$$

Real curves frequently consist of at least as many data, but they are not precisely evenly distributed.

6 Example-based Regression

Introduction In this Chapter, example-based algorithms and the features selected in Section 4.4 were used to perform regression on LGM binary lens light curves. Clean, evenly sampled, simulated light curves built from the 7-parameter SBLM were used in preceding Sections, but in this Chapter simulated noise and temporal gaps are added. Although we are dealing with a regression problem, the term "classifier" is used interchangeably with the term "regression algorithm" throughout. This reflects the data mining origin of many of the algorithms tested in this Section.

The goals of this Chapter were

1. to find suitable algorithms for LGM binary light curve regression,
2. to test the features selected in Section 4.1 with real classifiers and against benchmarks,
3. to take the first step towards creating a regression scheme for analysis of real LGM events.

Training data consisted of 10000 sample light curves generated randomly from the ranges given in Table 4. In fact this training set was split into 80 per cent training data and 20 per cent "unseen" testing data to test classifier generalization. All features selected in Section 4.4 were calculated for the data set. Although a variety of libraries were used (TORCH [84], JAGA [83], JOONE [85]) and some algorithms were constructed from scratch, the final calculations were performed using the excellent WEKA package [76]. WEKA was used through both its GUI and its Java API.

Some of the algorithms used in this Section have a large number of settings, introducing a new set of variables into the search for suitable classifiers. Where this was the case, the defaults suggested by the authors of WEKA were used. Where a large number of settings were available, some experiments were performed to determine suitable settings, but these tests were not particularly rigorous. What

these experiments did show was that regression accuracy was much less dependent on any algorithm's parameters than on the quality of data and features passed into the algorithm.

The remainder of this Chapter documents several experiments that were performed in order to fit the standard model to a simulated, ideal light curve. Each experiment attempted to answer a specific question in the search for a general regression technique for LGM binary lens light curves.

6.1 Benchmarks and Pre-processing

6.1.1 Benchmarks

Results from regression experiments in this Section could not be interpreted in a vacuum, hence a benchmark comparison was invented to be used throughout as proofs. Another sanity check which was performed occasionally was to run regression algorithms on data sets with randomized target parameters, which of course should always provide null results and form a sensible baseline to compare the algorithms to.

The first benchmark algorithm was a simple Powell fitter adapted from [59]. The Powell algorithm does not require gradient information. The second benchmark was a classifier invented for the purpose, called the "Cached Fitter". The Cached Fitter consisted of the following algorithm

1. Create a set of 10000 noiseless light curves randomly from a fixed parameter range and store them.

2. Pre-process these light curves to facilitate comparison.

3. To fit a new light curve, simply pre-process it and calculate $\frac{\Delta\chi^2}{d.o.f}$ between each stored curve and the processed input curve.

4. Stored model parameters corresponding to the lowest value are returned as the best fit.

Table 17: Correlation between predicted and actual variables for a noise-free training set for parameters a, θ, b and q. CF refers to Cached Fitter and PF to a Powell Fitter.

Data Set	CF, preprocessed	CF, scrambled	PF, preprocessed	PF, scrambled
a	0.08	0.08	-0.10	-0.07
θ	0.01	0.00	0.01	-0.07
b	0.34	0.05	0.12	0.06
q	-0.04	0.05	0.03	-0.02

6.1.2 A look at the benchmarks

This Section tested the Cached Fitter and Powell benchmark against pre-processed data sets and a data set with randomized target parameter sets.

Data

A data set of 10000 noise-free light curves was generated from the ranges in Table 4. This data set was pre-processed and fit with the CachedFitter and PowellFitter. The target parameters were then scrambled and re-fit.

Model

The simple binary lens model.

Results

Table 17 shows the correlation of the fitted variable to the actual for both types of fitters using the three different data sets.

The benchmarks fared very poorly, illustrating the difficulty of fitting binary light curves with naive algorithms. The only sign of a statistically significant fit was the Cached Fitter's ability to produce correlation of 0.34 for the impact parameter b. The scrambled data sets provided null results as expected and also indicated the error bound on these correlation results was at least as large as 0.07.

Conclusions

This simple experiment provided us with a null result as well as a benchmark to be used for comparison to algorithms explored further on.

6.1.3 Pre-processing of light curves

Real Microlensing events are not regularly sampled and have variable start- and end-times. Unfortunately many of the features to be used rely on the fact that our idealized light curves contain exactly 100 evenly-spaced points with no gaps. Real light curves also have their time axis denoted in Julian date and the curves need to be centred on some feature. A form of pre-processing where light curves are translated in time to a point of reference and interpolated at regular intervals is therefore required in order to extract sensible features. For example, the x- and y-positions of extrema in the light curves are meaningless in a noisy light curve unless some form of smoothing is applied.

Training curves and the curves to be fitted were subjected to the same pre-processing algorithm. A related question was whether training curves should be noisy or not and so both noisy and clean training curves were tested.

"Peak" pre-processing The following simple algorithm was used for centering and found to be fairly robust to anomalous light curve shapes. Some aspects are discussed in more detail below.

1. Invert the light curve around the time axis, in other words the brightest point in the curve now has the highest value, and the faintest point in the curve is at 0.

2. Find the time corresponding to the brightest point.

3. Translate the light curve in time to this point and discard points fainter than 20 per cent of the peak brightness as measured from the faintest point in the curve.

4. Scale the time axis so that the curve starts at -0.5 units and ends at 0.5 units.

There are a few apparent disadvantages to these forms of pre-processing. Firstly, one cannot expect to regain the translation and scaling parameters like t_e, t_m and

m_0 from a processed curve as this information has been purposefully removed. In pre-processed curves we shall therefore concentrate on the more difficult parameters a, θ, b and q which can still be fitted for. Secondly, pre-processing is not perfect; for example, large gaps in the light curve around the cut-off points could lead to a curve that starts or ends prematurely and is not centred correctly.

On the other hand it was hoped that pre-processing would present the feature-selection and training algorithms with more potent information than a raw light curve would.

6.1.4 Pre-processing noisy curves

The above centering routine was required in all cases. Noisy data required additional processing as discussed in this Section.

Smoothing

Many feature extraction operations require a smooth curve but unfortunately smoothing is not a straight-forward operation. This is mainly due to the conflict between smoothing too much, which passes over and hides features and smoothing too little, which leaves spurious extrema and phantom features in the curve. Getting the balance right proved to be challenging.

Several classes of smoothing algorithm were considered.

Splines Fitted B-splines were used to smooth noisy light curve data. In this methodology the user chooses the order of splines to be used that connect a set of evenly-spaced nodes as well as the size of a smoothing penalty. The B-spline coefficients are then calculated to optimize the combined fitting and smoothing penalties. A detailed description of B-splines can be found in e.g. [86].

In practice, setting the order did not have a large effect on the resulting smoothed curve and third order splines were used throughout. Finding the correct smoothing parameter was considerably more difficult because no single value suited all curves. An iterative solution was ultimately implemented that started with a low smoothing

penalty. If the resulting, smooth curve had more than 16 extrema, the smoothing penalty was increased and this process was repeated until the number of extrema fell below 16. This process was found to be reliable.

A note on other smoothing techniques A number of alternative smoothing techniques were considered. Perhaps the simplest was smoothing by an averaging window passed over observations. This method was unsuitable due to the fact that smoothing diminishes localized features such as small peaks which are significant in binary lensing.

A promising variation on windowed smoothing is the Savitzky-Golay smoothing algorithm which also sums data points in a moving window but with specific weights designed to preserve all features, including the size of extrema [59]. Unfortunately the method is only applicable to evenly-spaced data which ruled it out for use with noisy curves.

Unconventional smoothing techniques like fitting by a neural network would have provided a smooth approximation to even highly non-linear curves but were not attempted on suspicion that a parameterized representation of the network would not be unique.

That left fitted splines as the most obvious choice, although there was surprisingly little publicly available source code for reliable fitted b-spline algorithms. The public FORTRAN implementation from [87] was converted to C++ to perform the task.

More on Truncation

An algorithm that aims to centre a noisy light curve in time as well as shift it to a baseline magnitude also requires a decision to be made on where to truncate the light curve. This also turned out to be a non-trivial exercise. The requirements for a truncation methodology are at least the following:

1. The method should never over-truncate, in other words discard parts of the

curve with high information content.

2. The method should be consistent in its truncation across all possible light curves.

3. The method needs to be robust to noise, including gaps in observations.

The aim of truncation was to take a light curve that contained an arbitrary number of observations of an un-lensed (no measurable magnification) source and a smaller number of observations of lensing in progress and to truncate the curve before and after the lensed portion of the light curve, leaving just enough of the curve on either side to include a return to the un-lensed baseline to within observational error.

As noted above, a simple method was finally adopted. It discarded all points that were fainter than 20 per cent of the distance between the brightest and faintest points in the curve. Below are some alternative methods that were considered but not used.

Slope-based truncation A first attempt at truncation used the instantaneous slope of the curve to attempt to find the points where magnification of the source becomes obvious. The method was rejected due to several fatal flaws.

The first was the large uncertainty in the first derivative of a noisy curve, leading to false positives. This could be partially compensated for by a smoothing algorithm but it was hard to smooth optimally. Too much smoothing led to an algorithm that would miss a short-lived magnification distinct from the main peak. Too little smoothing and the number of false positives remained large. An equally serious issue was caused by gaps in the data, leading to very misleading numerical derivatives. These in turn could be compensated for by interpolation, but this introduced unacceptable uncertainty into the process.

Fit-based truncation Another obvious technique was to fit a model to a noisy light curve and to truncate the light curve based either on the model parameters or by using a smooth model for slope-based methods. The paradox is that one cannot fit the correct binary lens LGM model to the curve because that is too hard and is after all the aim of the truncation exercise to begin with. A simplified model has to be fitted and no problem-specific simplified model was deemed sufficient. Approximations of the Chang-Refsdal type [48] were rejected because they assume small mass ratios. Single lens fits are also inadequate for the majority of binary events. Finally, generic "model" fits like Chebyshev polynomials were found wanting in Section 4.2.1.

Standard deviation-based truncation This method was based on data points moving outside of a standard deviation band of points in the un-lensed portions of the light curve. The procedure measured the standard deviation of the ten left-most points in the data set. It then started at the left-most data point and moved the point of truncation to the right until it found three consecutive data points more than three σ away from the mean of the ten left-most points. The procedure was repeated in mirror-image on the right-hand side of the light curve. The difference between the left- and right truncation points made up a width and finally a region of the curve with three times the initial width and centred at the mid-point of the initial selection was retained for further processing.

This method was also found wanting and abandoned, mainly because in curves with very noisy wings it led to over-truncation (discarded too much of the curve).

Interpolation

Several of the implied light curve features discussed in Section 4.2 implicitly require an interpolation scheme when applied to real light curves. Even the simple feature consisting of the light curve itself requires equally spaced points which cannot be achieved without interpolating the irregularly-spaced real data points.

Three interpolation schemes were attempted for this purpose: fitted splines, cubic splines and simple linear interpolation. Fitted splines have the advantage that they are robust to noise but they do not preserve detailed features in the curve. Cubic splines pass through all points in the curve but are very sensitive to noise and introduce false features into the curve between data points. Linear interpolation passes through all points and is completely robust to noise as it is bounded by successive data points but unfortunately is not very accurate.

Experimentation led to the final adoption of linear interpolation for feature selection where required.

6.2 Classifier Comparisons (without Feature Selection)

6.2.1 Introduction

In this Section the main category of regression algorithm is introduced. It was mentioned in Section 6 that many algorithms and approaches were attempted but eventually the bulk of the work was performed with the excellent WEKA "data mining" package [76].

14 WEKA classifiers were tested, spanning a variety of classifier categories. The LeastMedSq, LinearRegression, SimpleLinearRegression and PaceRegression algorithms were used to perform simple regression. Neural networks are represented by the MultiLayerPerceptron and RBFNetwork classes. SMOReg is a support vector-based regressor. IB1, IBk, KStar and LWL are nearest-neighbour-based lazy classifiers and DecisionStump, M5P and REP are tree-inducing classifiers.

6.2.2 Fitting raw light curves as a benchmark

In this experiment pre-processed light curves themselves were fit with the classifiers listed in Table 18.

Goal

To set a benchmark for fitting and feature selection by fitting the raw light curves themselves. This benchmark was an acid test for the feature selection process in

Table 18: Key to raw light curve benchmark classification.

Key	Classifier
1	functions.LeastMedSq
2	functions.LinearRegression
3	functions.PaceRegression
4	functions.SimpleLinearRegression
5	lazy.IBk
6	trees.DecisionStump
7	trees.M5P
8	trees.REPTree
9	functions.MultiLayerPerceptron
10	functions.RBFNetwork

Chapter 4. If the selected feature sets fared no better than the raw curve then the validity of the process would be doubtable.

Data

An input instance consisted of the first and last time values in the curve and all 100 evenly-spaced magnitude points from the same light curves as used in previous Sections. The light curves were noiseless and pre-processed.

Model

The standard 7-parameter LGM binary lens model (SBLM).

Classifiers

A subset of WEKA classifiers were used as a representative set of algorithms in e.g. Section 6.3.1. In particular, the SMOreg, KStar and LWL classifiers were left out for performance reasons due to the large size of the input vectors.

Results

Tables 19 show results for the classifiers used on these unprocessed light curves, with the key to classifiers given in Table 18. These results are compared to those of a data set of selected features in Section 6.3.1.

The first thing to notice about these results are that they all beat the two benchmark algorithms in Section 6.1.1 with ease, despite being used with almost identical data. This bodes well for the use of example-based regression. Having

Table 19: Pre-processed light curves without feature selection used with simple classifiers. Correlation between target and predicted variables on an unseen test set.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	0.36	0.35	0.36	0.25	0.71	0.20	0.35	0.60	0.38	0.27
θ	0.31	0.32	0.32	0.18	0.68	0.23	0.46	0.56	0.46	0.01
b	0.23	0.23	0.24	0.17	0.64	0.26	0.37	0.57	0.24	0.12
q	0.36	0.36	0.37	0.30	0.55	0.29	0.36	0.46	0.19	0.25

Table 20: Training time (seconds) of default classifiers on light curves with no feature selection.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	285.47	44.76	13.40	0.54	0.22	2.34	402.96	8.78	2985	17.77
θ	273.34	44.22	13.23	0.52	0.06	2.52	346.96	8.98	2954	17.50
b	284.36	37.13	13.29	0.49	0.06	2.50	332.94	7.63	2938	17.46
q	308.31	46.29	13.36	0.49	0.05	2.45	355.69	8.26	2736	17.43

said that, none of the algorithms produces a correlation value better than 0.71, which is not good enough for practical fitting purposes. To put the best values in perspective, the best result (for the value of a with an IBk nearest neighbour algorithm) corresponds to a mean absolute error of 0.15 θ_E in our range of 0.6 θ_E < a < 1.7 θ_E.

One algorithm's performance completely outstrips the rest: the simple IBK nearest-neighbour algorithm beats all other algorithms for all four model parameters.

Timing

Table 20 shows the training times of all classifiers on the light curve data sets. The training times are not prohibitive except perhaps for the neural network (MultiLayerPerceptron). The LeastMedSq and M5P Tree algorithms were also fairly slow and bad value for computing time, given their mediocre correlation results.

Conclusions

This Section determined that example-based regression was already more effective than benchmarks even when used on light curves themselves. Only minor pre-processing was applied to the input data.

6.3 Classifier Comparisons (with Feature Selection)

In this Section we compare the fitting performance of several classifier algorithms from the WEKA library on simulated SBLM events. Although this experiment is easily performed thanks to WEKA, it needs to be noted how hard it is in general to place such a variety of algorithms on a common platform. Before switching to WEKA, this thesis used algorithms from a variety of sources, including third party libraries (as mentioned in Section 6), home-grown code and adaptations from, for example, [59].

6.3.1 General Classifier And Feature Set Comparison

Goal

To compare the performance of feature sets selected by CfsSubsetEval, ReliefFAttributeEval and InfoGain algorithms using 10 different, real-valued, regression algorithms from WEKA. Both feature sets and algorithms were evaluated.

Data

Data consisted of features selected by the three algorithms in Section 4.4. Light curves were pre-processed and each model parameter had its own feature set and trained its own specialist classifier. Original light curves consisted of 100 evenly-sampled, noise-free points from curves generated using the parameter ranges in Table 4.

Model

7-parameter SBLM (Standard Binary Lens Model).

Classifiers

The following 10 WEKA algorithms, which span a number of general techniques, were used. The first three were simple statistical regression techniques, the next two were representative neural networks: the typical back-propagation multi-layer perceptron and a radial basis function network. Simple linear regression algorithms

Table 21: Key to CfsSubsetEval feature set classifiers

Key	Classifier
1	functions.LeastMedSq
2	functions.LinearRegression
3	functions.PaceRegression
4	functions.MultilayerPerceptron
5	functions.RBFNetwork
6	functions.SimpleLinearRegression
7	lazy.IBk
8	trees.DecisionStump
9	trees.M5P
10	trees.REPTree

Table 22: Performance of 10 example-based regression algorithms on the CfsSubsetEval-selected feature set. Results are in the form of the correlation between known and fitted variables.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	0.50	0.51	0.51	0.55	0.34	0.31	0.52	0.34	0.54	0.56
θ	0.60	0.61	0.61	0.63	0.48	0.41	0.64	0.42	0.65	0.62
b	0.51	0.52	0.52	0.54	0.48	0.51	0.36	0.46	0.57	0.55
q	0.51	0.51	0.50	0.27	0.31	0.31	0.42	0.34	0.54	0.51

were represented primarily for benchmarking purposes, as was the very simple decision stump single-node tree. IBk is a nearest-neighbour algorithm and finally two advanced decision-tree inducing algorithms, M5P and REPTree were used. Full descriptions of all the algorithms are available in [76]. They are listed in Table 21.

Results and Discussion

Table 22 shows the results of the CfsSubsetEval-based run with all ten algorithms, with the legend given in Table 21. Similarly Tables 23 and 24 show the same type of results for ReliefFAttributeEval- and InfoGain-selected feature sets.

Overall, the experiments were partially successful. Most classifiers fit the model

Table 23: Performance of 10 example-based regression algorithms on the ReliefFAttributeEval-selected feature set. Results are in the form of the correlation between known and fitted variables.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	0.45	0.45	0.45	0.53	0.00	0.23	0.56	0.21	0.64	0.58
θ	0.59	0.63	0.63	0.59	0.44	0.44	0.56	0.45	0.66	0.63
b	0.60	0.60	0.60	0.59	-0.02	0.52	0.53	0.45	0.67	0.60
q	0.44	0.44	0.44	0.44	0.30	0.19	0.45	0.26	0.44	0.51

Table 24: Performance of 10 example-based regression algorithms on the InfoGain-selected feature set. Results are in the form of the correlation between known and fitted variables.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	0.40	0.40	0.41	0.45	0.21	0.31	0.66	0.31	0.55	0.58
θ	0.42	0.55	0.55	0.54	0.03	0.41	0.59	0.41	0.62	0.61
b	0.57	0.57	0.57	0.50	0.03	0.52	0.53	0.45	0.62	0.66
q	0.30	0.30	0.30	0.33	0.25	0.30	0.48	0.30	0.30	0.44

Table 25: The highest correlation between known and fitted variables from three selection methods and ten classifiers.

Parameter	Selection Method	Classifier	Correlation
a	InfoGain	IBk (nearest neighbour)	0.66
θ	ReliefFAttributeEval	M5P (tree)	0.66
b	ReliefFAttributeEval	M5P (tree)	0.67
q	CfsSubsetEval	M5P (tree)	0.54

parameters to an accuracy of between 0.4 and 0.5 as measured by the correlation between the actual and fitted variables. These were not good enough to be called a successful fit but this experiment allowed the drawing of some conclusions about example-based regression and determined the choice of parameters and algorithms to be used in subsequent experiments.

Notable failures included the radial basis function neural network classifier (number 5 in Table 22). It exhibited erratic behaviour and often failed to fit a parameter at all, with a correlation of close to zero. Simple linear regression (6) and the decision stump algorithms (8) fared poorly as well. This was to be expected as these algorithms are very simple and we have already established that the LGM binary lens problem is complex.

All three feature sets performed similarly and there were best results in certain parameters for all three selections. The M5P tree-inducing algorithm fared very well, achieving the best results for three out of four parameters. Table 25 summarizes the best results across all selection methods and classifiers.

It should be noted that the majority of the remaining classifiers did not fare much worse than the champions in Table 25. If we were to choose a subset of the classifiers used here, one could argue to retain at least three: the IBk nearest neighbour search,

the M5P tree inducer and the REP tree inducer. The REP is particularly useful because it is so fast and almost matches M5P in performance.

Table 25 should be compared to the benchmarks in Section 6.2.2 where no feature selection was used at all. This leads to quite a shock: with no feature selection at all, the IBk nearest neighbour classifier in fact slightly outperforms all but one of the feature selections with correlations of 0.71 for a, 0.68 for θ, 0.64 for b and 0.55 for q. Of course the feature-selected data sets had far fewer inputs to the algorithms than the 100 brightness points present in the pre-processed light curves. Still, the overall goal was not compression of inputs but regression performance.

Timing

"Training time" is of secondary importance in a straight-forward case of LGM regression because the classifiers can generally be trained once and then reused. Nonetheless, training and testing time could be important if classifiers are to be trained dynamically; for example, if used iteratively where each iteration is performed on a subset of the preceding parameter space. Training and testing times were recorded and are briefly discussed here.

While there is not a large dispersion in the fitting accuracy of the various algorithms in this experiment, their training and testing times differ enormously. Tables 26 and 27 show the time taken to train each classifier and evaluate the unseen test set from the CfsSubsetEval-selected data using the trained classifier. Simple regressions and trees were the fastest to train. The slowest classifiers were the least median squared, multi-layer perceptron and M5P-tree algorithms. These took minutes to train instead of the seconds or even fractions of a second required by the fastest methods.

Testing or evaluation time was similar and fast for all algorithms except the nearest-neighbour IBk algorithm. This makes perfect sense as this type of algorithm calculates nearest neighbours for every evaluation, although in return it does not require training.

Table 26: Training time (seconds) for 10 classifiers using the CfsSubsetEval feature sets.

DataSet	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	102.50	1.28	1.07	199.47	8.12	0.13	0.02	0.60	171.54	1.94
θ	80.77	0.34	0.45	109.75	2.19	0.07	0.02	0.38	159.37	1.22
b	45.56	0.13	0.04	16.56	0.87	0.01	0.01	0.06	148.37	0.33
q	121.87	2.73	1.69	331.38	5.05	0.15	0.02	0.74	172.19	2.66

Table 27: Testing time for 10 classifiers using the ReliefFAttributeEval feature sets (seconds).

DataSet	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
a	4.70	4.26	4.30	4.79	3.97	3.91	72.46	4.07	6.66	4.51
θ	4.58	4.36	4.38	5.89	4.19	3.98	54.81	4.22	5.69	5.17
b	4.41	4.11	4.17	4.40	4.06	4.12	18.02	4.11	4.24	4.30
q	4.02	3.98	4.07	4.72	3.83	4.06	94.97	4.03	4.00	4.19

Summary of results and Conclusions

This Section produced useful results as well as a number of questions that will need to be addressed. To summarize the results:

- All but one of the classifier algorithms used in this experiment performed fairly well. Radial basis function neural networks (or the WEKA implementation of these) were too erratic to use.

- IBk nearest neighbours and the M5P tree inducing algorithm fared best on feature selections. although IBk is a fairly slow algorithm to apply. It does not require training. M5P is slow to train.

- The REP classifier had good overall performance and is very fast to train.

- Data sets containing selected features outperformed the corresponding benchmark data sets that used light curves themselves for all but one classifier (IBk).

- The IBk classifier applied to light curves themselves outperformed the data sets containing selected features.

The following conclusions can be made:

1. Feature selection does not necessarily improve performance of classifiers over those using simple, pre-processed data.

2. The classifiers that take the longest to train are not necessarily the best.

3. The IBk, M5P and REP algorithms would be best to continue with.

4. None of the feature selection methods were overall winners. ReliefFAttributeEval-based selection was good overall but is much slower than the other two methods (CfsSubsetEval and InfoGain).

We are also left with some new questions:

1. These results apply to noise-free data. Everything may yet change when noise is added in the next Section.

2. The best results were obtained for the IBk classifier using no feature selection at all. Perhaps even better performance could be achieved by selecting a subset of points from the curve?

3. There were good results from all three types of feature selection. Could the best features be combined?

6.4 Noisy Data

Real data contain various sources of noise and an attempt was made to work with noisy data sets, even if these data sets were completely simulated. The goal of introducing noise into our simulations was to determine if example-based regression techniques could fit real Microlensing events.

6.4.1 Motivation for simulating noise

These experiments deals with example-based regression and we shall see that many thousands of examples will be required in order to achieve acceptable results. Thousands of fully analyzed, real LGM events are required as examples for training

if we are to extend our methods to fitting unknown events from the telescopes. Unfortunately the total number of Microlensing events observed to date are of the order of a few thousand and a much smaller number of these are binary events. Even the relatively small number of candidate binary lens events have not always been completely analyzed.

The simulation needs to include realistic observational effects, including noise and temporal gaps in coverage. A model for these effects was required to progress further.

6.4.2 Modelling noise

At first, a model for noise was developed based on OGLE observations and analysis of three binary events [88]. The noise characteristics of these events were chosen because they were samples from the OGLE-III survey [89] and fairly typical of modern Microlensing surveys. Plots of the three binary events are shown in Figure 40.

Microlensing data show frequent gaps due to bad weather, daylight and other outages as well as noise dependent on crowding in the field, data reduction method, magnitude of the source and various other observational factors. The sources and distribution of noise are often complicated and hard to model, leading to an approach where a simple curve was fitted to sample data from these three OGLE events, instead of attempting to model noise from first principles.

6.4.3 A Fitted Noise Model

It was found that the photometric noise from the three OGLE events OGLE-2003-BLG-170, OGLE-2003-BLG-267 and OGLE-2003-BLG-291 was well-approximated by an exponential function shown in Equation 56

$$A_0 e^{mag - m_0} + e_0 \tag{56}$$

where A_0, m_0 and e_0 are parameters fit to photometric error as a function of

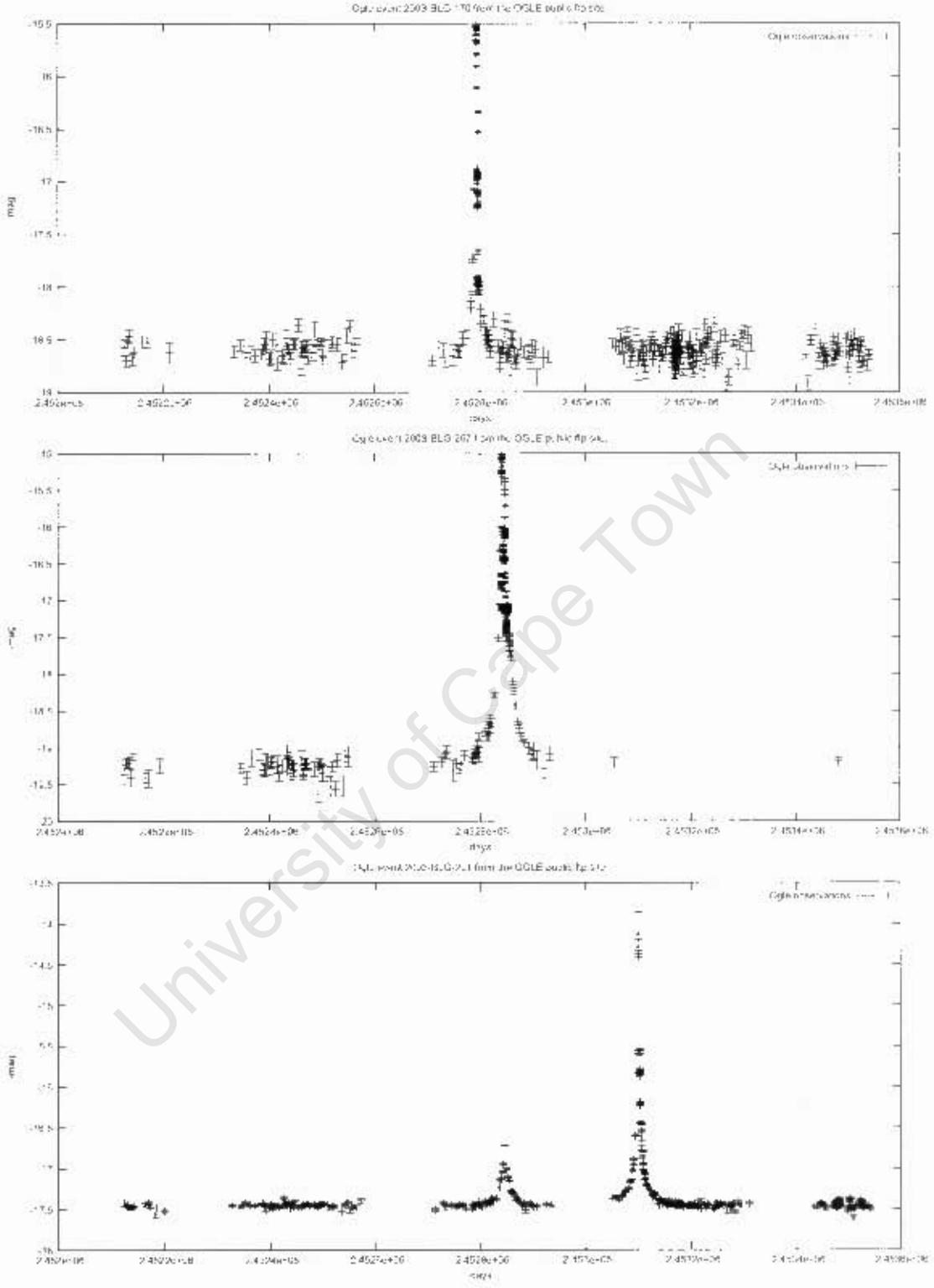

Figure 40: OGLE binary LGM events.

lensed brightness for a data set containing all observations from the three OGLE events above. An assumption on the "noise on noise" was required in order to perform this fit and this was that the uncertainty in photometric error was equal to the reported photometric error itself plus 0.05 mag. Other, simplifying assumptions were that photometric error was a function solely of lens magnitude and that the same noise model could be applied to all OGLE-observed events.

The fit is shown in Figure 41. The noise model appears to be pessimistic at the bright end of the scale, most likely due to the assumptions made on the deviation of reported photometric error. This was considered acceptable as erring on the conservative side.

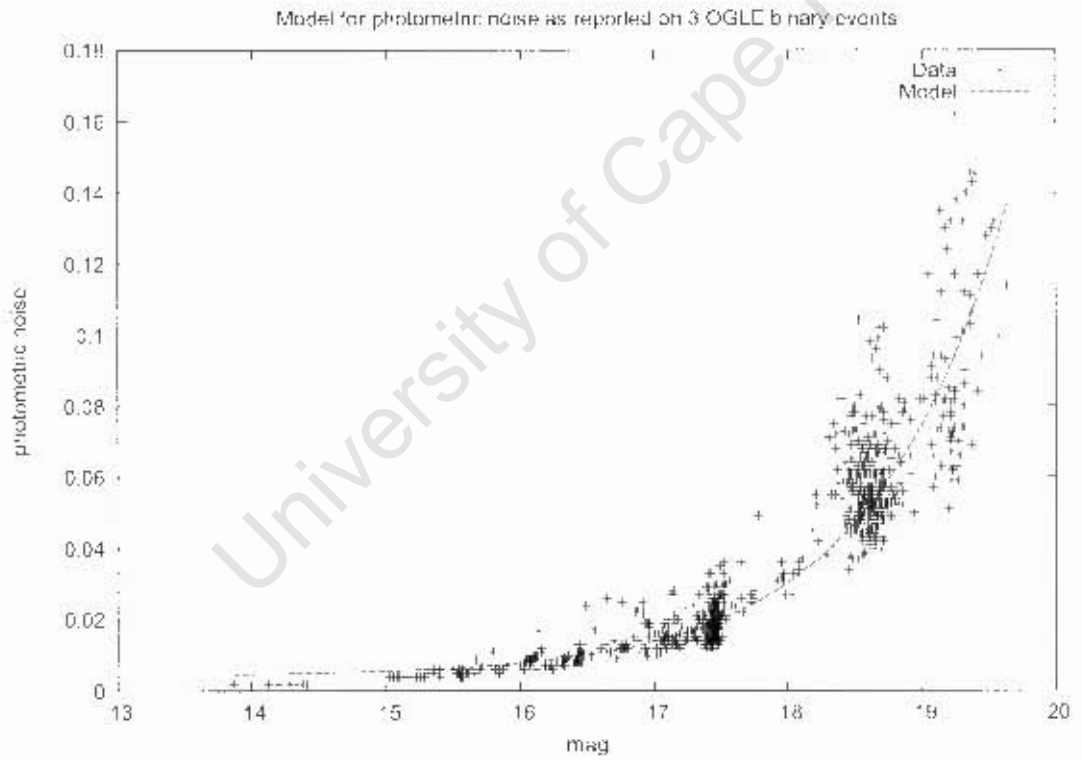

Figure 41: An exponential fit to a function of photometric error vs. brightness for three OGLE events.

Temporal Gaps

A model of OGLE observations also needs to address temporal gaps in the observations as these are a major feature of all photometric Microlensing observations and could have a major influence on the fit. Another simple approach was adopted under the following assumptions:

1. the time between discrete observations was dependent only on the absolute magnitude of the lens event and

2. the three OGLE binary events 2003-OGLE-BLG-170, 2003-BLG-267 and 2003-BLG-291 are representative of LGM observations of binary lens events.

Using this approach, observation time points were generated by roulette-wheel sampling of the distribution of OGLE time gaps at a given lens brightness (magnitude). If, for example, OGLE performs additional observations while the source is strongly lensed, this will be reflected in the sampled time gaps because the distribution of OGLE time gaps at bright magnitude will peak at a day or less. Similarly the distribution of OGLE time gaps peaked at several days for simple baseline observations during the surveying stage.

This approach allowed for the indirect inclusion of observational effects such as bad weather, daylight, observational outages and the like. An apparent weakness was that gaps at a given brightness were determined stochastically, so that periodic effects like regular breaks due to daylight or monthly deterioration due to the full moon did not occur periodically, but stochastically instead. This could have affected the value of processed features for light curve analysis if they were based on frequency transforms like Fourier or wavelet analysis. As the features considered in Chapter 4 were not in this category, this was not seen as a major weakness.

Another aspect of simulating time gaps in this way was that these gaps were based on absolute time, not t_e-scaled time which implied that events with a longer duration (higher t_e) contained more observations on average than events with low

t_e. To prevent inclusion of events that were practically impossible to analyze, events with less than fifty observations were discarded, leading to a t_e-bias in the generated data set. The bias was considered acceptable for the purposes of this experiment.

6.4.4 A simulated data set

Fifteen example, simulated light curves are reproduced in Figure 42 as a sanity check and to get a general feeling of the nature of the simulated data. These should be compared to the three genuine OGLE events shown in Figure 40.

6.5 The effects of Noise on fitting

6.5.1 Experiments with the OGLE noise model

In this Section the noise model from Section 6.4.1 was used to generate a simulated data set for testing the extension of example-based fitting techniques to noisy data. When dealing with noisy data one may wonder whether it is best to train the algorithms with noise-free or noisy curves to obtain the best performance on noisy test curves. Two experiments were performed in this Section for comparison although the results are also relevant in isolation.

Experiment A - OGLE noise model for both training and testing curves

A noisy data set of 10000 light curves was created, of which 8000 were used for training and 2000 for testing. Pre-processing was applied to the training and testing curves in the manner described in Section 6.1.3 above and feature selection was undertaken anew for the noisy data set. In these experiments we discarded all curves with less than 50 points, whereas the previous noise-free experiments were performed on curves that contained as few as 20 points. In order to allow for direct comparison, the noise-free experiments from Section 6.3.1 were re-run here with CfsSubsetEval feature selection.

In Section 6.3.1 we managed to eliminate a few of the classifiers as either too slow to train or simply ineffective. Only six classifiers were used in these experiments and they are given in Table 28. The first three are simple algorithms included for

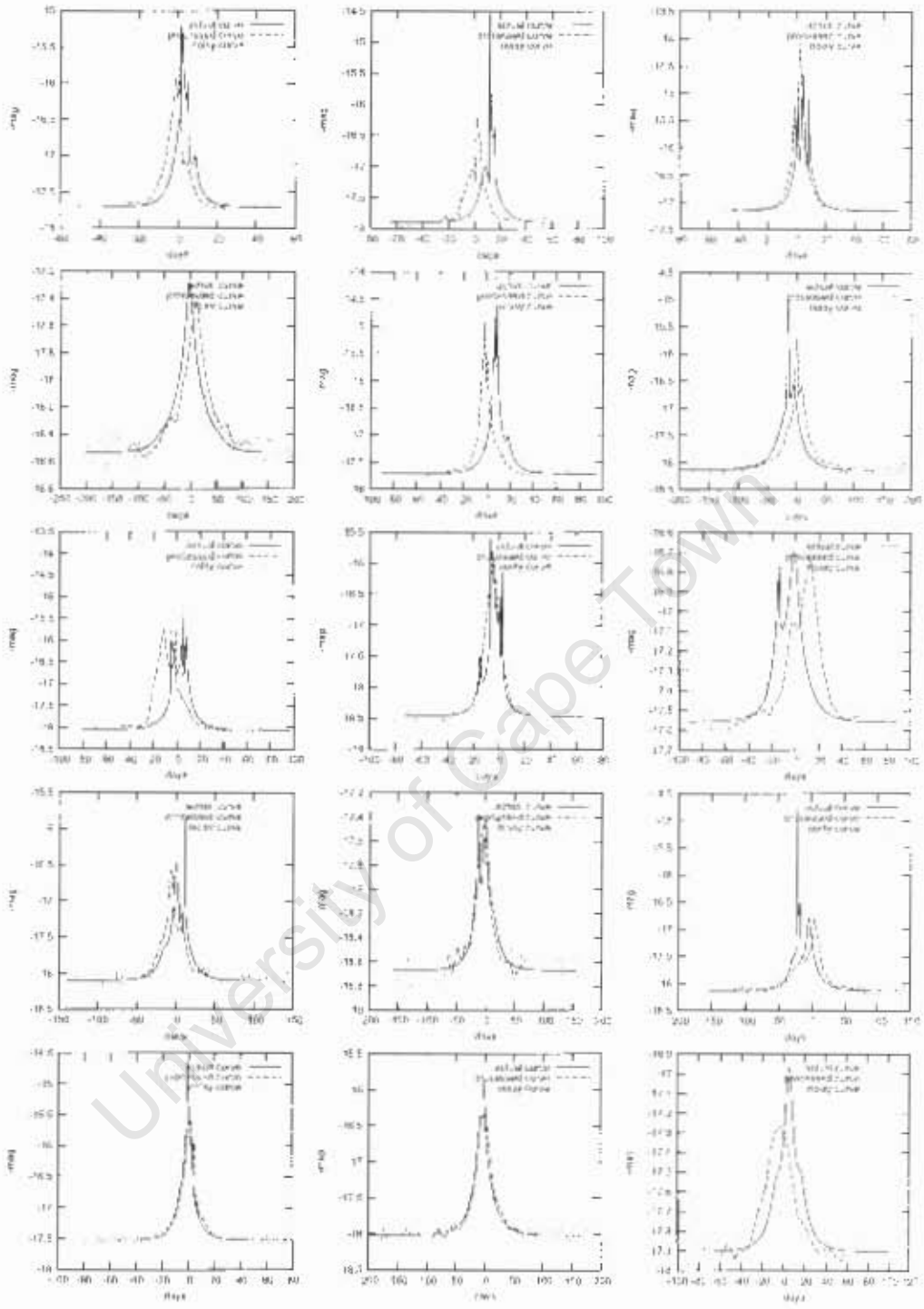

Figure 42: 15 randomly selected light curves from the new sample of noisy light curves. Solid lines indicate the actual noise-free and highly-sampled light curve. Dots indicate the data points in the final noisy curve, and the dashed line indicates a smoothed, processed version of the noisy curve. Note that the processed curve has been re-centred making direct comparison awkward in this image.

Table 28: Key to CfsSubsetEval feature set classifiers after eliminations.

Key	Classifier
1	functions.LeastMedSq
2	functions.LinearRegression
3	functions.PaceRegression
4	lazy.IBk
5	trees.M5P
6	trees.REPTree

Table 29: Correlation between predicted and actual variables for noise-free light curves containing at least 50 points.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)
a	0.47	0.56	0.56	0.66	0.61	0.63
θ	0.58	0.60	0.60	0.60	0.70	0.61
b	0.53	0.53	0.53	0.47	0.60	0.57
q	0.50	0.50	0.50	0.51	0.54	0.51

benchmarking while the last three were the best performers from Section 6.3.1.

Results

Table 29 shows results for noise-free data with curves with less than 50 points discarded. Comparison to Table 22 shows that the new data set performed slightly better than the previous experiment where only curves with 20 points or less were discarded. This was not unexpected; perhaps one surprising aspect of the result was that a data set that contained a large number of curves with fewer than 50 points fared almost as well as the current set.

Table 30 shows the results as correlation between predicted and actual values of parameters a, θ, b and q respectively for the noisy data set. All six classifiers fared poorer on noisy data. The reduction in correlation was roughly 0.1 to 0.2 except for the IBk classifier which showed a large decline for all model parameters. The noisy fit for mass ratio q only manages a correlation of 0.13 as opposed to the noise-free correlation of 0.51.

Experiment B - OGLE noise model and perfect training curves

This experiment was similar to the preceding one but training data were created free of noise, whereas the testing data were created with noise. The goal was to

Table 30: Correlation between predicted and actual variables on an OGLE noise model training set for parameters a, θ, b and q. Both training and testing curves are noisy.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)
a	0.44	0.44	0.44	0.30	0.49	0.42
θ	0.39	0.41	0.41	0.21	0.44	0.39
b	0.36	0.36	0.36	0.21	0.39	0.34
q	0.29	0.29	0.29	0.13	0.34	0.30

Table 31: Correlation between predicted and actual variables for OGLE noise model test curves, with classifiers trained on noise-free training sets. Only IBk nearest neighbours and M5P classifiers were used.

Data Set	IBk	M5P
a	0.17	0.06
θ	0.24	0.22
b	0.18	0.36
q	0.08	0.12

determine whether better results could be obtained for test curves that contained noise if the training curves were noise-free.

Results

Results are shown in Table 31. They are disastrous and show poor correlation between target and fitted binary lens parameters.

Conclusion

This experiment showed quite clearly that training curves should be noisy if they are to be used to fit noisy light curves.

6.5.2 Quantifying Noise Effects

The fitting experiment in Section 6.5.1 demonstrated a collapse in the fitting ability of the previously reliable IBk and even M5P algorithms when faced with a realistic noise model based on OGLE observations.

This Section sought to establish the point of failure as more and more noise was introduced into the simulated data set. At which stage do example-based classifiers become useless?

6.5.3 A parameterized noise model

A simpler noise model that enabled a parameterized, gradual introduction of different kinds of noise was developed for this experiment. The model takes only two parameters. The first is simply the standard deviation of Gaussian noise around the LGM binary lens model's magnitude, σ. The other kind of "noise" deals with irregular sampling of light curves. It was assumed that light curves are sampled at regular intervals twice per day. However a parameter p_{gap} was introduced to govern gaps in this regular sequence. On generation of each point of the sequence a gap of one full day would be introduced with probability p_{gap}.

The two parameters p_{gap} and σ were used to control noise in artificial light curves.

6.5.4 Regression as a function of noise

This Section contains two studies into the behaviour of example-driven fitting algorithms as a function of increasing noise. Noise is introduced via Gaussian perturbation and the likelihood of gaps in the data.

Experiment with Gaussian noise

The aim was to determine the effect of Gaussian noise on fitting accuracy. Features were re-selected for each noise level and each parameter, ensuring optimal fitting at each stage.

Table 32 shows the correlation between the four binary lens model parameters and their predictions, as fitted by four different algorithms. The Gaussian noise parameter is set to 0 per cent, 1 per cent, 2 per cent, 5 per cent and 10 per cent respectively.

This experiment allowed us to make a few interesting observations.

Table 32 shows that fitting accuracy does decline with increasing noise as expected but with an interesting catch; there is some evidence that introducing noise with standard deviation of 1 per cent in fact improved fitting as compared to the noiseless case. One hand-waving argument for this improvement is that the addition of a small amount of noise enabled both the feature selection and training algorithms

Table 32: Correlation between predicted and actual model parameters for a Gaussian noise model as standard deviation is increased from zero per cent to ten per cent.

parameter	σ (per cent)	LinearRegression	IBk	M5P	REPTree
a	0	0.55	0.67	0.42	0.64
a	1	0.54	0.62	0.54	0.62
a	2	0.50	0.59	0.54	0.58
a	5	0.54	0.51	0.59	0.56
a	10	0.40	0.34	0.50	0.43
θ	0	0.63	0.63	0.72	0.65
θ	1	0.61	0.53	0.63	0.62
θ	2	0.60	0.51	0.66	0.62
θ	5	0.56	0.46	0.61	0.57
θ	10	0.29	0.19	0.34	0.29
b	0	0.48	0.46	0.57	0.54
b	1	0.50	0.41	0.58	0.56
b	2	0.48	0.37	0.55	0.53
b	5	0.40	0.25	0.43	0.41
b	10	0.23	0.17	0.35	0.28
q	0	0.48	0.48	0.51	0.54
q	1	0.53	0.49	0.60	0.58
q	2	0.48	0.47	0.55	0.54
q	5	0.42	0.31	0.45	0.42
q	10	0.17	0.07	0.26	0.20

to avoid over training, leading to better performance on an unseen test set. Selected features were briefly examined in order to justify the selection half of this argument. Selected features are not shown here (although see Table 34 for a similar data set with 2 per cent Gaussian noise and 10 per cent chance of missing a day's worth of data) but a comparison of features selected for orbital separation a, of noiseless and slightly noisy data (1 per cent) shows a very similar set. PCA parameters were discarded in favour of Chebyshev polynomial coefficients. Similarly for parameter b, the y-position of the 4th turning point of the tangent curve was discarded but other parameters were very similar. q, which shows perhaps the most improvement from 0 per cent to 1 per cent noise was even harder to interpret. It maintained a similar set but added PCA parameters and selected a few more features of each type. Examining the feature selections were inconclusive.

Certain classifiers were more robust to Gaussian noise than others. In this experiment the M5P regression tree shows the smallest decline in efficiency with noise while the nearest neighbour routine IBk shows the largest. This is a significant

result because IBk was the most efficient algorithm in a noiseless environment and indicates that one's choice of algorithm would be dependent on the quality of your data.

The behaviour of fitting accuracy as a function of noise also differs considerably for different model parameters. For example, a remains fairly well-fitted even at 10 per cent noise while it becomes very hard to fit q at this level. This result indicates that the mass ratio q has a more subtle effect on the curve than does projected orbital separation a, rendering it vulnerable to Gaussian noise.

Finally we note that some algorithms show a considerable variance in performance even when applied to very similar data. M5P seems to fall into this category. Table 32 shows the results of single classifier runs and the fact that M5P achieves correlation of only 0.42 for the classification of a for a noiseless data set while achieving 0.59 in the presence of 5 per cent Gaussian noise indicates considerable variance in an individual run's efficiency. The other algorithms behave less erratically. Variance in performance is unlikely to be due to the set of selected features, as these are selected using 10-fold cross-validation and hence quite robust.

Experiment with gaps

This experiment examines the effect of gaps in simulated light curves caused by artificial "observational outages".

Table 33 shows the correlation of the four binary lens model parameters with their predictions as fitted by an M5P algorithm. The probability of missing a day's worth of observations (p_{gap}) is increased from 0 per cent to 40 per cent. One consideration in this experiment is the minimum number of data in a simulated light curve. In the previous experiments this was set to 50. In this experiment the introduction of gaps produces curves with fewer data points so the threshold for accepting a simulated light curve was decreased to just 20 points.

Another point to consider is that although the initial raw light curve will contain these gaps, curves are pre-processed in the way described in Section 6.1.3, in other

Table 33: Correlation between predicted and actual variables for a noise model as the probability of missing a day's worth of observations (p_{gap}) is increased from 0 per cent to 40 per cent. Four classifiers were used.

parameter	p_{gap}	Linear Regression	IBk	M5P	REP
a	10	0.52	0.51	0.54	0.60
a	20	0.51	0.35	0.60	0.58
a	40	0.55	0.35	0.63	0.62
θ	10	0.59	0.52	0.63	0.57
θ	20	0.56	0.39	0.61	0.56
θ	40	0.51	0.30	0.54	0.47
b	10	0.49	0.36	0.54	0.53
b	20	0.47	0.33	0.51	0.47
b	40	0.43	0.27	0.46	0.43
q	10	0.48	0.34	0.52	0.47
q	20	0.46	0.30	0.49	0.46
q	40	0.32	0.24	0.38	0.34

words curves will first be smoothed whereafter the gaps will actually be filled by interpolation. True information loss cannot be recovered in this way but the pre-processing routine may lead to some robustness to missing data.

First, a comparison across classifiers. As in the previous experiment with Gaussian noise, the IBk nearest-neighbour classifier which was so successful when applied to noise-free data showed most sensitivity to temporal gaps. It is the worst performer at p_{gap} of 40 per cent in all four model parameters. The rest of the results told a story of general decline with the increase of temporal gaps as expected, although the regressions of certain model parameters were more robust than others. In particular, regression of a and θ seemed quite robust and this made sense as these are parameters that tend to affect the entire light curve. q was most seriously affected by temporal gaps which could have been the result of curves where important peaks were missing from the data, making the measurement of q impossible. It is easy to see this would be the case for low mass ratios (small q) where the secondary lens would have a fairly local effect on an otherwise single-lens light curve.

Finally there is one feature of the results that seem hard to explain, which is the robustness and in fact improvement of fitting performance with increasing p_{gap} for the projected orbital separation a. Perhaps the argument about noise preventing over-training raised in the discussion of the previous experiment's results is also

relevant here.

Noise Experiments: Conclusions

We have examined both Gaussian noise added to brightness observations and the effect of temporal gaps. The results are largely consistent and indicate that:

1. Naturally, noise does have a detrimental effect on example-based fitting.
2. There are a number of regression algorithms that are quite robust to noise.
3. For these algorithms the addition of noise should not prove a major obstacle.
4. A small amount of noise may actually lead to improved regression performance.

Experiments from here on included noise for additional realism. The simple noise model was used with Gaussian noise at 2 per cent and p_{gap} of 10 per cent.

6.6 Experiments to Improve the Fit

6.6.1 Classifiers adjusted for higher accuracy

In the previous experiments all classifiers were used with their default settings. There was no reason to doubt these WEKA default settings, but classification problems are unique and require fine-tuning on a problem-by-problem basis.

Goal

The goal was to see if classifiers with more powerful settings resulted in a better fit and whether the additional training time that the settings required were worth it.

Data

A simulated, noisy data set of 10000 events was used, based on the noise model in Section 6.5.4. The standard deviation was set to 2 per cent and the probability of skipping a day's observation set at 10 per cent. CfsSubsetEval feature selection was used for each parameter, and this selection is shown in Table 34.

Table 34: CfsSubsetEval feature selection for a "slightly noisy" data set: Gaussian noise with standard deviation of 2 per cent and a probability of missing a day's observations set at 10 per cent.

a	*0*	*b*	*q*
y0	turnx2	slopesdev	turnx4
y1	turnx3	slopeslope21	yskew
y43	turnx5	slopeslope35	tangent35
y45	turnx7	slopeslope63	tangent39
y56	xvar	chebyfit6	tangent40
y57	tangent20		tangent41
y99	tangent45		tangent43
turny0	tangent49		tangent44
turny1	tangent52		tangent45
tangent19	tangent79		tangent46
tangent20	chebyfit1		tangent52
tangent23	pca2		tangent53
tangent24			tangent54
tangent25			tangent55
tangent27			tangent56
tangent29			tangent57
tangent31			tangent58
tangent32			tangent59
tangent33			tangent60
tangent34			tangent66
tangent35			tangent68
tangent36			tangent69
tangent62			slopeturny2
tangent65			slopeturnx3
tangent66			slopeturny3
tangent67			slopeturnx4
tangent69			slopeturny4
tangent71			slopeturny5
tangent72			slopeturny6
tangent74			slopeadev
tangent76			slopesdev
tangent78			slopevar
tangent80			slopecurt
tangent81			slopeslope15
tangent82			slopeslope45
slopeturny0			slopeslope46
slopeadev			slopeslope47
slopecurt			slopeslope48
slopeslope8			slopeslope49
slopeslope15			slopeslope50
slopeslope17			slopeslope51
slopeslope37			slopeslope52
slopeslope40			pca27
slopeslope58			
slopeslope61			
slopeslope84			
slopeslope88			
5smooth43			
5smooth54			
5smooth55			
20smooth40			
20smooth41			
20smooth43			
chebyfit2			
pca1			

Table 35: Key to CfsSubsetEval feature set classifiers with more powerful settings.

Key	Classifier	Change
1	functions.LeastMedSq	Sample size for linear regression improved to 100 from 4.
2	functions.LinearRegression	No additional attribute selection performed.
3	functions.PaceRegression	Estimator changed from Bayes to ordinary least squares.
4	lazy.IBk	4 distance-weighted nearest neighbours considered instead of 1. Cross-validation enebled.
5	trees.M5P	Regression tree enabled.
6	trees.M5P	Minimum number of instances reduced to 2 from 4.
7	trees.REPTree	Pruning switched off.

Model

The 7-parameter SBLM.

Classifiers

The list of more powerful classifiers is given in Table 35.

Results

Table 36 shows regression results for the modified regression algorithms and their unmodified (default settings) benchmarks. It is apparent that with the possible exception of the IBk algorithm, adjusting the default settings to attempt to boost fitting power simply did not work. Even for IBk, the one algorithm where a considerable improvement was made, the more powerful setting only brought fitting accuracy in line with what the other algorithms were already achieving.

Conclusions

This experiment leads to the following conclusions:

1. Improving the classifiers themselves by increasing their complexity has little effect on regression success.

2. This in turn indicates that the ability to fit light curves is bound by the

Table 36: Correlation between target and fitted model parameters for high-power settings on standard algorithms vs. their default settings.

Data Set	1	2	3	4	5	6	7
a	0.50	0.53	0.53	0.59	0.65	0.65	0.61
a with default settings	0.50	0.53	0.53	0.47	0.66	0.66	0.61
θ	0.54	0.55	0.55	0.63	0.63	0.63	0.57
θ with default settings	0.54	0.55	0.55	0.53	0.62	0.62	0.63
b	0.51	0.51	0.51	0.50	0.57	0.57	0.45
b with default settings	0.51	0.51	0.51	0.38	0.59	0.59	0.56
q	0.46	0.46	0.47	0.45	0.55	0.55	0.45
q with default settings	0.46	0.46	0.47	0.30	0.51	0.51	0.51

complexities inherent in the problem, not the complexity of any given classifier.

3. Most of the algorithms used in this and the Sections above are already performing well, given the data set.

4. To improve accuracy dramatically one would have to adjust the input data in some way. For example, the model parameter ranges in Table 4 could be subdivided to minimize degeneracy, or additional features with more predictive power would have to be used.

6.6.2 Biasing for High Variance

The previous Sections used a data set consisting of 10000 events that were randomly selected from the standard model parameter ranges given in Table 4. A number of these curves are quite similar to single lens curves, making a binary model inappropriate. A good example is the case of small mass ratio q with small binary separation a and large impact parameter b: the small secondary lens causes a minor perturbation at large impact parameter that will not yield any information on the crossing angle θ, for example. The parameter ranges in Table 4 were chosen so that all of the sample events were in the "lensing zone", or zone of large perturbation, but large perturbations were not guaranteed.

In this experiment 70000 new, noisy events were generated in the same way and from the same range as for the previous experiments, but all events without a large variance in the processed brightness ($yvar < 0.4$) were discarded. In other words

the data were purposefully biased toward disturbed events. Processed brightness refers to the final y-value of a light curve after pre-processing as discussed in Section 6.1.3. The point of the high-variance bias was to preserve events that an observer would classify by eye as "binary lens" due to its large departure from a single-lens light curve.

Goal

To determine if a data set skewed towards the inclusion of strongly perturbed light curves results in more effective regression. If this turned out to be the case, events that showed large perturbations could be fitted with a data set consisting of highly perturbed events, perhaps achieving better results.

Data

The original data set consisted of 70000 events. After selecting events based on $(yvar > 0.4)$, 8204 events remained. This number was close enough to the 10000 events used in previous Sections so that the size of the data set should not have influenced results.

The data set was subjected to feature selection yielding the set of features for each parameter as given in Table 37. It was interesting to compare this table to Table 34 in Section 6.6.1: features selected in the absence of high-variance bias. The features selected for the high-variance data were almost completely contained in the unbiased data set, with the exception of specific, similar features such as adjacent y-points. In other words, similar but fewer features were selected for the high-variance set. What conclusions could be drawn from this selection? Perhaps that features in the high-variance set were more clearly defined so that fewer derived features were required to reach the same conclusions about a model parameter fit? It was hard to tell without further tests beyond the scope of this experiment.

In addition to the CsfSubsetEval-selected feature set, the pre-processed curves themselves were also tested using the same high-variance bias for purposes of proof.

Table 37: CfsSubsetEval with GreedyStepwise Feature Selection Results for a data set biased towards large variance.

a	θ	b	q
turnx0	y63	yadev	ymax
turny1	turnx1	slopeturny1	yskew
turny2	turnx3	slopeturny2	ycurt
turny3	turny3	slopesdev	tangent42
ysdev	tangent47	slopevar	tangent45
yvar	tangent49	slopeslope43	tangent46
tangent16	tangent50	slopeslope45	tangent53
tangent19	tangent53	slopeslope57	tangent54
tangent21	slopeturnx3	chebyfit8	tangent56
tangent26	slopeturny6	chebyfit9	slopecurt
tangent28	slopeave		slopeslope49
tangent31	slopeskew		slopeslope56
tangent46	5smooth33		
tangent53	20smooth2		
tangent70	chebyfit1		
tangent72			
tangent75			
tangent76			
tangent81			
tangent98			
slopeturny5			
slopeadev			
slopesdev			
slopevar			
slopecurt			
slopeslope31			
slopeslope65			
slopeslope89			
slopeslope96			
chebyfit6			
chebyfit14			
pca1			
pca4			

Table 38: Key to classifiers used on the variance-biased data set.

Key	Classifier
1	functions.LinearRegression
2	lazy.IBk
3	trees.M5P
4	trees.REPTree

Model

The standard 7-parameter LGM binary lens model.

Classifiers

Four quite different classifiers with proven track records from previous Sections were used.

Results

Results are shown in Table 39 with the key to classifiers in Table 38. Results for an unbiased data set as well as a data set consisting of pre-processed curves without feature selection are also shown.

The effect of biasing for high variance brought about a large improvement in fitting performance. The most dramatic improvement was a correlation increase of 0.2, achieved for the IBk nearest-neighbour classifier, but performance was increased for all classifiers and model parameters (except parameter q with the REP tree, but this was considered to be within margin of error). Comparison to the curve-only data set (no feature selection) shows that classifier performance on a set of selected features is on a par with the benchmark, non-feature-selected curves for all model parameters. In particular, the M5P tree performs best on a feature set, as opposed to light curves.

This experiment proves that biasing the training data to include only high-variance events leads to substantially better regression performance. We have not yet considered the other implications of this bias. Firstly, if instances of our algorithms are trained only on high-variance events, these classifiers would obviously not be capable of fitting low-variance events which constitutes a loss of generality. It may

Table 39: Correlation between target and regression variables for an unseen high-variance test set. Results for the same classifiers using a biased, curves-only data set (no feature selection) and the unbiased data set from Section 6.6.1 are included for comparison.

Data Set	(1)	(2)	(3)	(4)
a	0.67	0.67	0.76	0.69
a without bias	0.53	0.47	0.66	0.61
a curves only	0.45	0.79	0.55	0.70
θ	0.68	0.67	0.74	0.71
θ without bias	0.55	0.53	0.62	0.63
θ curves only	0.41	0.75	0.65	0.63
b	0.59	0.52	0.62	0.57
b without bias	0.51	0.38	0.59	0.56
b curves only	0.31	0.63	0.39	0.58
q	0.51	0.43	0.54	0.49
q without bias	0.46	0.30	0.51	0.51
q curves only	0.09	0.58	0.36	0.49

be argued that this is not a serious loss as we can imply from the improvements in this experiment that low-variance events were not well fit to begin with. Observers can easily discriminate high- and low-variance events before fitting, which means that events could be funneled down two different paths for regression: this better-performing high-variance method and an alternative method specialized to low-variance events. Secondly, binary events that are detected are likely to have higher variance due to the obvious selection effect. That means that these methods should still be applicable to a large proportion of events that are easily distinguished as binaries a-priori.

Another question that comes to mind regarding bias is what effect high-variance bias has on the model parameter distributions that we are now training with and hence which parameter ranges can be successfully fit using these methods. Figures 43 to 46 show the distributions of simple model parameters a, q, b and θ in the biased data set. The distributions of b and q are fairly flat, which means that high-variance bias is not imposing a restriction on the ranges of these parameters that can be successfully fit (within the ranges used in these experiments, in any case). Model parameters a and θ are affected by the bias. Events with $0.6\ \theta_E < a < 1\ \theta_E$ have higher variance than those with $1\ \theta_E < a < 1.7\ \theta_E$, hence this parameter range will

dominate training. In θ the effect is the most dramatic. Roughly speaking events with $-90° < \theta < 90°$ have high variance whereas the rest do not. The explanation appears to be simple: events in the favoured range are those where the source path intersects the typically large binary caustic structure to the right of the primary lens (for these model parameters) as shown in Figure 8. The range of θ with low variance reflects the large number of events that miss caustic structure altogether due to an unfavourable crossing angle on the "wrong side" of the projected binary lens.

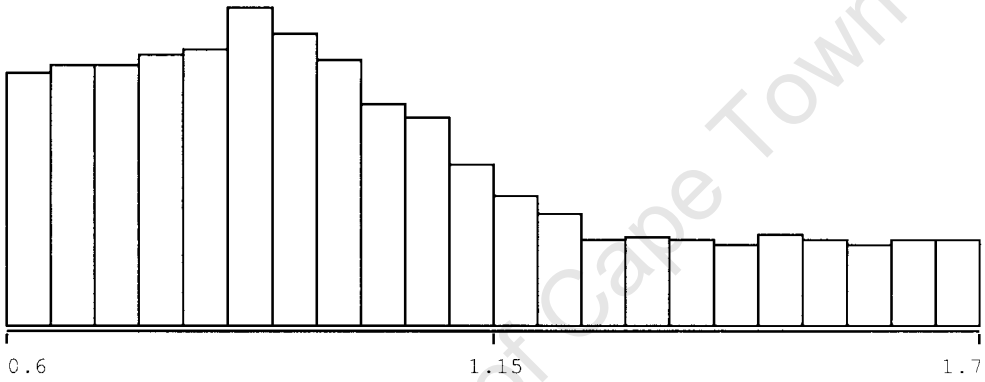

Figure 43: Distribution of a (θ_E) for a variance-biased data set.

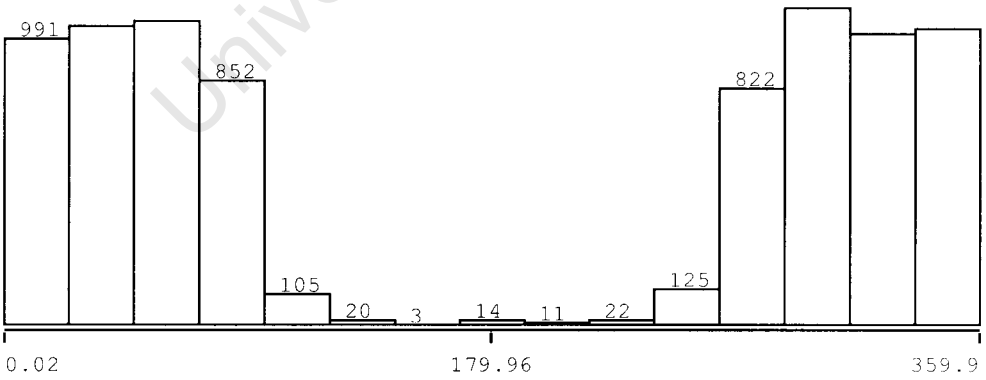

Figure 44: Distribution of θ (°) for a variance-biased data set.

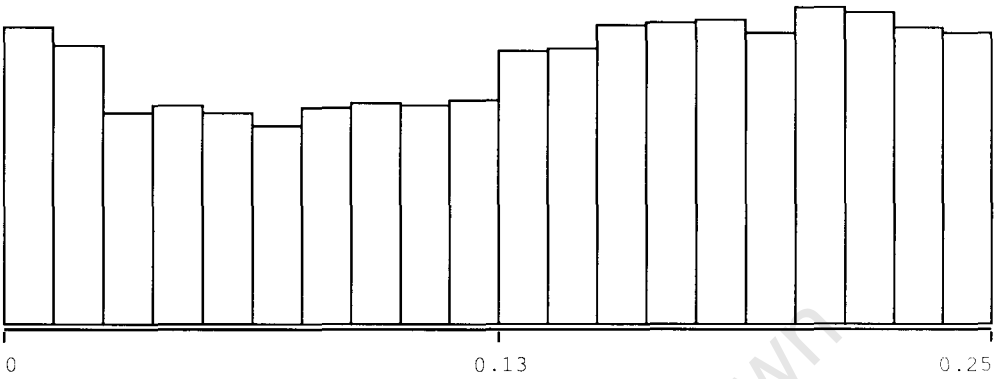

Figure 45: Distribution of b (θ_E) for a variance-biased data set.

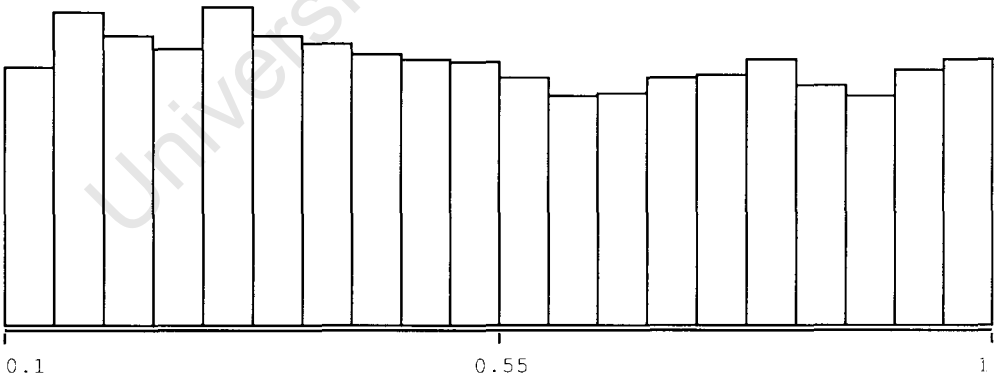

Figure 46: Distribution of q (*ratio*) for a variance-biased data set.

Timing

Timings of both test and training runs were similar to those of the biased data set.

Conclusions

1. Biasing the training and testing events for high variance leads to much improved fitting performance.

2. The price to pay for better performance is that a smaller subset of events can be fitted.

6.6.3 Discrete classifiers

Previous experiments in this Chapter determined continuous model parameters from continuous observations. In this experiment both the features and their target values were discretized, allowing access to a number of classification algorithms that were not available to continuous data.

Goal

To determine the effects of discretization of the input and target data and to evaluate classification performance of a number of discrete classification algorithms.

Data

The base data consisted of the same set of noisy events (10000 of them) used in the experiment in Section 6.6.1, but with each attribute and target value discretized into 10 bins. The CsfSubsetEval-selected features selected in Section 6.6.1 were used for this experiment and once again there was a comparison to a benchmark of pre-processed light curves with no feature selection. Each y-point in the benchmark curves was also discretized into 10 bins.

Model

The standard 7-parameter SBLM.

Table 40: Key to discrete classifiers used on the standard noisy data set.

Key	Classifier
1	trees.Id3
2	trees.J48
3	trees.RandomForest
4	trees.RandomTree
5	trees.REPTree
6	bayes.BayesNet
7	rules.OneR
8	trees.DecisionStump

Classifiers

Most of the classifiers used in this experiment (Table 40) work only on discrete data sets and could therefore not be used directly on the input data. For a detailed discussion on these classifiers, see [76].

Results

Results are shown in Table 41 in the form of the percentage of correctly classified parameters for each data set, classifier and model parameter. Unfortunately one could not derive a sensible correlation measure for discrete classification for direct comparison with previous experiments, but some of these results look impressive. Each model parameter was split into ten bins. and the RandomForest manages to correctly classify θ from its curve in two thirds of cases. The other model parameters fair less well, with the best result obtained for a being 35 per cent from its curve using Random forest while b attains 24 per cent from RandomForest and q reaches almost 22 per cent, again using the pre-processed light curve and the RandomForest algorithm.

Clearly the best discrete classifier in our experiment is the RandomForest routine, but a few performed well including J48, the REP and even the BayesNet. In all cases the classifiers fared better on light curves than on feature-selected data sets.

It was helpful in this case to view the confusion matrix for classification as well. Table 42 shows that of classifying model parameter a with the RandomForest classifier. For this classification the success rate was around 30 per cent. but the

Table 41: The percentage of correctly classified model parameters for each data set and each model parameter using discrete classifiers. Classifier type keys are shown in Table 40.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
discrete a	21.00	27.35	30.15	20.05	24.45	26.10	17.95	16.90
discrete θ	35.95	43.70	47.15	40.50	40.55	46.10	27.30	21.50
discrete b	15.40	18.25	16.85	16.00	17.60	18.05	18.05	12.50
discrete q	12.60	15.45	18.50	12.75	16.60	18.05	15.65	14.95
discrete a, curves only	25.75	31.50	35.35	25.85	26.90	20.15	17.70	16.15
discrete θ, curves only	52.55	57.45	66.85	51.40	52.05	44.60	29.05	22.95
discrete b, curves only	16.00	19.30	24.20	19.05	17.00	12.05	14.60	12.65
discrete q, curves only	16.50	17.70	21.80	17.80	16.90	14.00	14.90	14.95

Table 42: The confusion matrix for the RandomForest classification of model parameter a from selected features. Category names are in units of θ_E.

a	b	c	d	e	f	g	h	i	j	classified as
783	150	30	15	13	10	14	13	23	23	a=(-inf-0.710095]
266	428	159	37	15	23	22	21	28	27	b=(0.710095-0.820084]
74	201	298	130	61	23	26	33	26	25	c=(0.820084-0.930072]
35	64	143	263	151	75	48	31	32	23	d=(0.930072-1.04006]
24	34	54	173	250	163	80	49	52	36	e=(1.04006-1.150049]
34	32	26	70	174	215	173	131	70	58	f=(1.150049-1.260037]
26	36	25	44	82	163	235	179	146	82	g=(1.260037-1.370025]
39	41	34	39	61	115	174	200	207	152	h=(1.370025-1.480013]
74	31	37	28	42	61	135	203	259	252	i=(1.480013-1.590002]
81	54	40	20	35	58	78	115	256	301	j=(1.590002-inf)

confusion matrix shows that the situation is in fact better than indicated by this single number; the matrix shows quite clearly that when a parameter was misclassified, it was mostly classified in an adjacent bin. This can be seen from the strength of the diagonal and adjacent diagonals in the matrix.

Timing

All the discrete classifiers in this experiment trained and tested very rapidly. Training times are given in seconds in Table 43. Testing time was sub-second for all classifiers except the nearest-neighbour-based IB1 algorithm due to its many Euclidean distance calculations.

Conclusions

1. Discrete classification performs well.

Table 43: Training time (seconds) for all the set of discrete classifiers used in this Section. Several had sub-second training times.

Data Set	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
discrete a	9.33	2.09	37.96	5.09	3.31	0.51	0.17	0.32
discrete θ	1.06	0.49	11.45	1.19	0.76	0.06	0.02	0.02
discrete b	0.22	0.20	3.79	0.35	0.33	0.03	0.01	0.01
discrete q	15.08	7.49	160.74	22.26	13.30	1.27	0.64	0.61
discrete a, curves only	11.30	5.56	158.77	18.04	11.70	1.49	0.61	0.61
discrete θ, curves only	16.17	8.05	177.16	22.91	14.01	1.29	0.62	0.62
discrete b, curves only	15.54	7.78	167.75	23.23	13.76	1.48	0.62	0.60
discrete q, curves only	3.28	1.67	29.19	3.85	2.35	0.25	0.12	0.12

2. When parameters are incorrectly classified, they are mostly placed in adjacent categories; a reassuring sign that the algorithms are performing sensibly.

3. The algorithms tested faired better on the light-curve data itself than on selected, derived features.

The success of discrete classification as evidenced in this experiment opens up several possibilities for fitting schemes. For example, events can be classified into various bins or parameter subspaces. The choice of subspace could be based on a division that minimizes known degeneracies in the model. The next experiment examines the effect that reducing the model parameter range may have on regression accuracy.

6.6.4 Dividing the Input space

So far we have considered the model parameter range from Table 4 as our range of interest as it encompasses most binary lens perturbations to the single lens curve. In this experiment the parameter space was subdivided in an attempt to boost efficiency. Of course, if applied in this way to real-world events, a fit would have to be performed per subspace. Although each subspace might take time to train, interrogation time for the methods examined so far is always very quick so this does not represent a serious criticism of the approach.

Some care needs to be taken when conducting this experiment so as not to inadvertently increase the density of samples in our range, which would make for an

unfair comparison to results from experiments like those in Section 6.6.2. For this reason, the experiment is split into two parts. The first part is run with a sample density similar to that of Section 6.6.2 and the second part is run with a much higher sample density. This allowed for the separation of effects due to sample density vs. those of reducing light curve degeneracy due to the choice of parameter subspace.

Goal

To determine the effect of dividing the input space into subspaces that reduce the theoretical degeneracy of light curves. The effect of sample density is also investigated.

Data

The data range from Table 4 used in previous experiments was reduced. Projected orbital separation a was reduced to $0.6\ \theta_E < a < 1.0\ \theta_E$ from $0.6\ \theta_E < a < 1.7\ \theta_E$. The crossing angle θ was reduced to just $0^\circ < \theta < 90^\circ$. These particular choices were made based on the known degeneracies discussed in Section 3.3.4. Only 1250 events were generated in this range in the first phase so that the density of samples would be comparable to previous experiments. A subsequent experiment was run with much higher sample density.

Model

The standard 7-parameter SBLM.

Classifiers

Four classifiers with a proven track record were used in this experiment. given in Table 38.

Results

The results in Table 44 show that the data set consisting of events from a reduced parameter range outperforms the full range despite the constant sample density. The feat is made more impressive by the fact that the correlations are higher than those

Table 44: Correlation between target and fitted model parameters using four classifiers and just 1250 events.

Data Set	M5P	LinearRegression	IBk	REP
a	0.75	0.70	0.52	0.68
θ	0.70	0.73	0.68	0.71
b	0.64	0.56	0.34	0.62
q	0.46	0.47	0.32	0.45

Table 45: Correlation between target and fitted model parameters using four classifiers and a large set of 30000 events, still operating on a reduced parameter range. Both selected features and pre-processed light curves by themselves were tested.

Data Set	M5P	LinearRegression	IBk	REP
a	0.86	0.72	0.66	0.84
a curves	0.90	0.39	0.92	0.89
θ	0.81	0.72	0.67	0.79
θ curves	0.84	0.56	0.88	0.83
b	0.75	0.59	0.50	0.71
b curves	0.76	0.35	0.80	0.76
q	0.64	0.52	0.49	0.64
q curves	0.59	0.42	0.72	0.67

in Table 39, even though the parameter ranges are smaller (for parameters a and θ) which means that the absolute error has decreased considerably.

Density of events

The same experiment was repeated to investigate the effect of density of sample events on fitting accuracy. Model parameter ranges were left the same but this time 30000 events were generated, split into a training set of 24000 and a test set of 6000 events. Keeping with the comparisons made in previous Sections, the data sets containing selected features were also compared to a run containing only pre-processed curves. Results for both data regimes are shown in Table 45: the correlation between target and fitted model parameters using the same four classifiers as in the previous Section.

This experiment yields the best results so far in this thesis. The M5P tree classifier reaches a correlation of 0.86 between fitted and target variables on a feature-based data set; the best of any regression scheme based on selected features so far. However, some of the curve-based regressions fair even better. The IBk algorithm

applied to processed curves yielded 0.92 for model parameter a and 0.88 for parameter b. IBk also yields the highest ever correlation for b. at 0.80 using curves and q at 0.72 also using curves. M5P also fares very well with comparable performance using either curves or feature sets as input. It is important to note that our subspace in the model parameter range was a reduction in a and θ only, yet remarkably better results are achieved for parameters b and q as well.

Conclusions

1. Reducing the parameter space of our experiment in a and θ leads to a dramatic improvement in fitting performance. The performance improvement was not due to increased sample density. The most likely explanation for the improvement would be the removal of known model degeneracies when the parameter range is reduced specifically in this way.

2. On the other hand, increasing sample density also led to large improvements in performance, with the best results achieved so far out of any experiment.

6.6.5 Meta-classifiers

There is another avenue of exploration to be followed in order to boost the passable results obtained so far. So-called "meta-classifiers" use a combination of regression algorithms or re-train an existing classifier based on its residuals with respect to a given problem (e.g. [76]).

Goal

To determine if meta-classification is a viable regression strategy for LGM binary events, by testing a number of representative meta-classifiers.

Data

The noisy, feature-selected dataset considered in Section 6.6.1 is used here and the results from that Section are a suitable benchmark to measure this experiment against.

Model

The standard 7-parameter LGM binary lens model (SBLM).

Classifiers

Two scenarios were considered: where the underlying classifiers were three of those known to perform well on their own (IBk, M5P and REP classifiers) and a second scenario where the underlying classifiers were simple (e.g. DecisionStump, LinearRegression, ZeroR).

Results

Results for the strong underlying algorithms are shown in Table 46. AdaptiveRegression and Bagging show a negligible improvement over the performance of the M5P algorithm working by itself. The "Vote" algorithm using M5P, IBk and the REP tree fares slightly better than M5P does by itself for parameters a, θ and q, but in fact slightly worse for b.

Things got more interesting when the results for weak underlying classifiers were considered, shown in Table 47. The outstanding result from this part of the experiment was the performance of (3), Bagging with a REP tree underlying. It reached a correlation of 0.74 for a and above 0.61 for the remaining parameters. The Bagging classifiers in Table 46 do not fare this well, peaking at 0.68. In the case of the M5P algorithm, it looked like the meta-classifiers had very little effect on the performance of the underlying. Yet for the REP tree, the Bagging algorithm enhanced the accuracy of the underlying substantially, as can be seen from the underlying's benchmark performance (7). Similarly to the M5P underlying, Bagging with LinearRegression appears to perform no better than LinearRegression by itself. Our first observation, then, is that the performance of a meta-classifier can be critically dependent on the type of underlying classifier in the realm of the LGM binary lensing problem. The same ambivalence was seen for the AdaptiveRegression meta-classifier, which enhanced the performance of the underlying DecisionStump algorithm, but not that of an M5P tree.

Table 46: Correlation between target and fitted model parameters using three meta-classifiers. The meta-classifiers used the M5P tree as the underlying algorithm, except for the Vote algorithm that used IBk, M5P and REP tree.

Data Set	M5P tree	AdaptiveRegression	Bagging	Vote
a	0.66	0.66	0.66	0.68
θ	0.62	0.63	0.66	0.68
b	0.59	0.59	0.60	0.56
q	0.51	0.51	0.53	0.53

Table 47: Correlation between target and fitted model parameters using three meta-classifiers and comparisons, with key in Table 48. The data sets used are exactly the same as for those shown in Table 46.

Dataset	(1)	(2)	(3)	(4)	(5)	(6)	(7)
a	0.53	0.50	0.74	0.51	0.53	0.31	0.61
θ	0.55	0.54	0.68	0.54	0.55	0.44	0.63
b	0.51	0.52	0.61	0.52	0.51	0.47	0.56
q	0.46	0.46	0.64	0.47	0.46	0.31	0.51

Table 48: Key to meta classifiers used on the standard noisy data set.

Key	Classifier
1	LinearRegression
2	AdaptiveRegression (Decisionstump underlying)
3	Bagging (REP underlying)
4	Vote (LinearRegression, ZeroR and DecisionStump underlying)
5	Bagging (LinearRegression underlying)
6	DecisionStump
7	REP

Table 49: Correlation between target and fitted model parameters using three meta-classifiers and comparisons on a data set consisting of processed curves without feature selection. Classifier keys are the same (Table 48).

Dataset	(1)	(2)	(3)	(4)	(5)	(6)	(7)
a	0.32	0.40	0.74	0.30	0.32	0.16	0.61
θ	0.29	0.41	0.69	0.30	0.29	0.19	0.48
b	0.22	0.40	0.75	0.24	0.22	0.20	0.63
q	0.27	0.36	0.64	0.31	0.27	0.30	0.49

Timing

The training time attributed to meta-classifiers used in this experiment rather obviously depended on that of the underlying. The use of an algorithm that is not that slow to train but benefits from meta-classifiers, such as REP, appears to be a workable solution.

Conclusions

In summary

1. Meta-classifiers can boost the performance of some classifiers
2. REP and the DecisionStump appear to benefit in this test whereas the M5P algorithm does not.

7 Fitting Real data

In this Chapter the techniques of example-based fitting that had been experimented with in previous Chapters are finally applied to real data. The goal of this Section was still to prove the concept of applying example-based fitting to LGM events, not to re-analyze the published data thoroughly.

Only two real events were fitted using this methodology and hence this Chapter suffers from a selection effect. The two chosen events both showed strong binary features and were fairly complete in their coverage of the light curve, making them similar to the events the classifiers were trained with. A number of events were considered for which no fit was attempted. MACHO-98-BLG-35 was a controversial, possible planet detection but the planetary perturbation would have been way too small to pass the high variance requirement. It could have been attempted with a training set specially prepared to detect planetary signals, but this was beyond the scope of the thesis. OGLE-2003-BLG-170 had too few data points during the actual event to attempt a fit. OGLE-2003-BLG-291 appeared to be an obvious case of binary source instead of binary lens Microlensing. OGLE-1999-BLG-23, MACHO-97-BLG-41 and other candidates with publicly available data were not attempted simply because of time constraints.

Although the small number of real fits casts some doubt on the validity of this Section, there is no reason to suspect that more events could not have been successfully fitted. The method worked as designed when applied to OGLE-2003-BLG-267 and EROS-2000-BLG-5, which merely required extension of the training parameter range and limitation of training events along simple lines such as the minimum number of peaks in a given training curve.

7.1 Preparation

7.1.1 Classifiers and Example Data

A final choice of regression algorithm needed to be made. The results of our experiments implicated three classifiers, which were compared once more using specially prepared data sets. As in some previous experiments, two data sets per model parameter were constructed; the first was a feature-selected set using the CsfSubsetEval and Greedy-Stepwise selection algorithm. The second data set consisted of just the pre-processed y-values from our light curves, as discussed in Section 6.1.3.

Lessons from previous experiments were applied. 50000 events were used in each data set: the largest set to date. Constructing as large a data set as was practical was motivated by Section 6.6.4. Events were biased toward high variance by excluding all events with a pre-processed variance in y-values smaller than 0.4, motivated by Section 6.6.2. The parameter space was divided into four by splitting the ranges of two parameters, a and θ into $0.6\ \theta_E < a < 1.0\ \theta_E$, $1.0\ \theta_E < a < 1.7\ \theta_E$ and $0° < \theta < 120°$, $240° < \theta < 360°$, as motivated by Section 6.6.4. Note that no events were generated with $120° < \theta < 240°$. That's because the distribution of θ for events with high variance included a negligible number of events with θ in this range. The upper limit of impact parameter b was actually extended from $0.25\ \theta_E$ as used in all previous experiments, to $1.0\ \theta_E$. It was thought that this extension would allow for a larger number of events to be eligible for fitting without reducing the algorithms' accuracy too much, as events were already biased towards high variance which would exclude damaging training events which did not show much binary deviation.

The classifiers used are given in Table 50. The choice of these three was based on their strong performance in the experiments. IBk was improved from its standard configuration (as suggested by Section 6.6.1) to take four nearest neighbours into account, and scale their contributions by the inverse of their Euclidean distance to the input vector. The M5P tree algorithm was left at its default settings based on its previous success in this configuration and the REP tree was combined with the "Bagging" meta-algorithm as this proved highly successful in experiment 6.6.5.

Table 50: Choice of three regression algorithms used on the final training data set.

Key	Classifier
1	IBk-nearest-neighbour classifier with 4 nearest neighbours and d^{-1} distance scaling.
2	M5P tree with default settings.
3	REP tree with the Bagging meta-algorithm.

Selected Features

The data sets used in this Section were optimized for effective regression using results from all previous experiments and selected features are briefly revisited here. There were four data ranges to consider for each of the four model parameters to be fitted, corresponding to the divisions in θ and a. Selections using the CsfSubsetEval and Greedy-Stepwise algorithms are presented in Tables 51 to 54.

The feature selection results were largely as expected. There are strong similarities between features selected for model parameter ranges which result in events that are flipped around the y-axis of the light curve. For example the events from Table 51 are mirror images in the y-axis (statistically speaking) of those in Table 53. The same applies to Tables 52 and 54. If feature selection were perfect, one would have expected these matching tables to contain identical results, but they do not. Overall the selected features make sense by way of a hand-waving argument, but the real test of their merit is a quantitative comparison of training results as undertaken below.

Training Results and final algorithm selection

Correlation results for each combination of all four parameter range subdivisions, three classifiers, two feature sets (pre-processed curves and selected features) and four model parameters are given in Tables 55 to 58.

The training results of our final classifiers are strong, achieving correlations between target and fitted variables above 0.74 on unseen test sets for some combination of data set and classifier, for each model parameter in all four ranges. Correlations as high as 0.94 are achieved for all parameters but q, which reaches a maximum of

Table 51: CsfSubsetEval with Greedy-Step-Wise feature selection for a set of 50000 events with high-variance bias and slight noise using model parameter ranges $0.6\ \theta_E < a < 1.0\ \theta_E$, $0° < \theta < 120°$, $0.001\ \theta_E < b < 1.0\ \theta_E$, $0.1 < q < 1.0$.

a	θ	b	q
turnx1	y98	tangent10	y14
turnx6	turnx6	tangent56	y99
turnx7	ysdev	tangent93	turnx0
ymax	tangent53	slopeturny5	turny1
yskew	slopeturny5	slopeturny6	turnx7
tangent34	slopeturny6	slopevar	ymax
tangent43	slopeadev	slopeslope35	ysdev
tangent62	slopesdev	slopeslope60	yvar
tangent66	slopevar	slopeslope64	yskew
tangent86	slopeslope32	slopeslope66	tangent45
slopeadev	slopeslope35	slopeslope68	tangent53
slopesdev	slopeslope48		slopeskew
slopevar	slopeslope62		5smooth11
slopecurt	slopeslope64		chebyfit8
slopeslope24	slopeslope66		chebyfit10
slopeslope29	slopeslope68		chebyfit12
slopeslope38	5smooth1		pca4
slopeslope70	chebyfit3		
slopeslope73	chebyfit5		
slopeslope76	chebyfit9		
slopeslope82			
chebyfit10			
chebyfit12			
chebyfit13			
chebyfit14			

Table 52: CsfSubsetEval with Greedy-Step-Wise feature selection for a set of 50000 events with high-variance bias and slight noise using model parameter ranges $1.0\ \theta_E < a < 1.7\ \theta_E$, $0° < \theta < 120°$, $0.001\ \theta_E < b < 1.0\ \theta_E$, $0.1 < q < 1.0$.

a	θ	b	q
y4	ymax	y1	ymin
y5	ymin	turnx0	tangent56
y99	yadev	turnx3	slopeadev
turny0	ysdev	yadev	slopesdev
turny3	tangent60	tangent51	slopevar
yadev	slopeturny5	slopeturny6	slopecurt
ycurt	slopeturnx6	slopeadev	5smooth1
slopeturnx5	slopeturny6	slopesdev	20smooth48
slopeturnx6	slopeskew	slopevar	chebyfit10
slopeturnx7	slopecurt	slopeslope62	
slopeave	slopeslope43	slopeslope65	
slopeslope24	slopeslope46		
slopeslope27	slopeslope49		
5smooth81	slopeslope53		
chebyfit6	slopeslope56		
chebyfit10	slopeslope60		
chebyfit12	slopeslope63		
pca8	slopeslope79		
pca11	5smooth3		
pca12	chebyfit5		
	chebyfit6		
	chebyfit8		
	chebyfit10		
	chebyfit16		

Table 53: CsfSubsetEval with Greedy-Step-Wise feature selection for a set of 50000 events with high-variance bias and slight noise using model parameter ranges $0.6\ \theta_E < a < 1.0\ \theta_E$, $240° < \theta < 360°$, $0.001\ \theta_E < b < 1.0\ \theta_E$, $0.1 < q < 1.0$.

a	θ	b	q
turnx0	y1	turnx3	turnx1
turny1	turny4	ymin	xcurt
turnx7	ymin	tangent42	yskew
ymin	ysdev	tangent73	tangent45
yskew	tangent61	slopeturny7	slopeturnx3
tangent3	slopeturny2	slopevar	slopeave
tangent26	slopesdev	pca6	slopeskew
tangent31	slopeslope30		5smooth28
tangent35	slopeslope34		5smooth83
tangent61	slopeslope36		chebyfit5
tangent67	slopeslope51		chebyfit8
slopeadev	slopeslope67		
slopesdev	chebyfit3		
slopeslope14	chebyfit5		
slopeslope20			
slopeslope40			
slopeslope42			
chebyfit10			
pca26			

Table 54: CsfSubsetEval with Greedy-Step-Wise feature selection for a set of 50000 events with high-variance bias and slight noise using model parameter ranges 1.0 $\theta_E < a < 1.7\ \theta_E$, 240° $< \theta <$ 360°, 0.001 $\theta_E < b < 1.0\ \theta_E$, 0.1 $< q <$ 1.0.

a	θ	b	q
y0	y95	y48	turny5
y95	turny4	turny4	ymin
turnx0	ymax	turnx7	tangent43
turny2	ymin	ymin	slopeadev
turny3	xcurt	tangent48	slopesdev
yadev	yadev	tangent76	slopevar
ycurt	tangent39	tangent86	slopecurt
tangent92	slopeturny3	slopeadev	5smooth95
slopeave	slopeturny6	slopeslope34	20smooth33
slopeslope69	slopeturnx7		chebyfit10
slopeslope72	slopeskew		
20smooth9	slopecurt		
chebyfit6	slopeslope35		
chebyfit10	slopeslope38		
chebyfit12	slopeslope42		
chebyfit14	slopeslope49		
chebyfit16	slopeslope52		
pca8	slopeslope55		
pca13	chebyfit5		
	chebyfit6		
	chebyfit8		
	chebyfit16		

Table 55: Correlation between target and fitted model parameters for the range 0.6 $\theta_E < a <$ 1.0 θ_E,0° $< \theta <$ 120°,0.001 $\theta_E < b <$ 1.0 θ_E.0.1 $< q <$ 1.0. listed by algorithm type.

Dataset	IBk (4 neighbours)	M5P (default)	REP with bagging
a	0.78	0.89	0.91
$a(curveonly)$	**0.94**	0.87	**0.94**
θ	0.48	0.69	0.76
$\theta(curveonly)$	**0.84**	0.73	**0.84**
b	0.46	0.63	0.67
$b(curveonly)$	0.83	0.76	**0.85**
q	0.62	0.60	0.65
$q(curveonly)$	0.70	0.53	**0.74**

Table 56: Correlation between target and fitted model parameters for the range 0.6 $\theta_E < a < 1.0\ \theta_E$,240° $< \theta <$ 360°,0.001 $\theta_E < b < 1.0\ \theta_E$,0.1 $< q <$ 1.0. listed by algorithm type.

Dataset	IBk (4 neighbours)	M5P (default)	REP with bagging
a	0.80	0.88	0.90
$a(curveonly)$	**0.92**	0.84	**0.92**
θ	0.58	0.67	0.72
$\theta(curveonly)$	0.84	0.74	**0.86**
b	0.64	0.66	0.69
$b(curveonly)$	0.84	0.77	**0.88**
q	0.56	0.54	0.61
$q(curveonly)$	0.67	0.59	**0.75**

Table 57: Correlation between target and fitted model parameters for the range 1.0 $\theta_E < a < 1.7\ \theta_E$,0° $< \theta <$ 120°,0.001 $\theta_E < b < 1.0\ \theta_E$,0.1 $< q <$ 1.0. listed by algorithm type.

Dataset	IBk (4 neighbours)	M5P (default)	REP with bagging
a	0.64	0.64	0.73
$a(curveonly)$	0.82	0.70	**0.84**
θ	0.66	0.83	0.86
$\theta(curveonly)$	**0.92**	0.87	**0.92**
b	0.74	0.77	0.80
$b(curveonly)$	0.91	0.85	**0.92**
q	0.61	0.62	0.65
$q(curveonly)$	**0.78**	0.56	**0.78**

Table 58: Correlation between target and fitted model parameters for the range 1.0 $\theta_E < a < 1.7\ \theta_E$,240° $< \theta <$ 360°,0.001 $\theta_E < b < 1.0\ \theta_E$,0.1 $< q <$ 1.0. listed by algorithm type.

Dataset	IBk (4 neighbours)	M5P (default)	REP with bagging
a	0.70	0.68	0.76
$a(curveonly)$	0.85	0.78	**0.86**
θ	0.69	0.84	0.87
$\theta(curveonly)$	**0.94**	0.89	0.93
b	0.80	0.80	0.83
$b(curveonly)$	**0.94**	0.86	0.93
q	0.64	0.63	0.67
$q(curveonly)$	0.80	0.69	**0.81**

0.81.

REP with Bagging applied to pre-processed curves (no feature selection) appears to do particularly well, although simple IBk with 4 nearest neighbours does outperform it slightly on occasion. Classifiers trained on extracted features appear to lag behind in all cases.

Conclusions

With the final experiment completed, we made our choice of algorithm, example data and pre-processing technique.

Data:

1. Parameter $0.6\ \theta_E < a < 1.7\ \theta_E$, split into two sets at $1.0\ \theta_E$
2. All of parameter θ except $120° < \theta < 240°$, split into a set from $0°$ to $120°$ and another from $240°$ to $360°$.
3. Parameter $0.001\ \theta_E < b < 1.0\ \theta_E$
4. Parameter $0.1 < q < 1.0$
5. 50000 training events per data set
6. Pre-processed curves only. Based on raw performance only, feature-selected sets were not used.

Classifiers:

1. REP with Bagging was chosen to be used throughout. Although IBk also performed well, it had a slower classification time which would also count against it in bulk fitting operations.

7.1.2 Fine-tuning

The regression experiments in Chapter 6 achieved correlations of up to 0.91 for model parameters, but even at this accuracy the fitted model parameters would

not build a curve that was a good match to the target curve on a $\Delta\chi^2$-basis. The example-based fit at no time attempts to minimize $\Delta\chi^2$, even though it is attempting to discover model parameter values by other means.

Still, the minimum value of $\Delta\chi^2$ corresponds to the maximum likelihood model parameter set (e.g., [59]). Therefore a further, fine-tuning operation is required to complete the fit, which performs a more traditional $\Delta\chi^2$-based fit using the result of the example-based regression as a starting point.

Unfortunately, traditional fits to Microlensing events fare poorly, as shown in Section 5.1. The starting point of a traditional, gradient-based algorithm has to be within the convergence well of the event in order to find the true minimum of the χ^2-surface. Hence the problem: example-based fits are required to be accurate to less than the size of the convergence radius, which we have seen is a distribution that peaks at a mere few per cent.

Fine-tuning Techniques

Two fine-tuning techniques were employed for the final fits. The first was a publicly available [90] implementation of the industry standard Levenberg-Marquardt gradient-based method while the second was a bespoke implementation of a Genetic Algorithm. The Levenberg-Marquardt algorithm was extremely fast (O(seconds)) but was prone to premature convergence, while the GA was very slow (O(hours)) but much more robust. The model parameters from the result of example-based fitting was used as the starting point for both algorithms, although they were incorporated in different ways.

An important omission should be mentioned here. The Markov Chain Monte Carlo method has recently been successfully applied to binary lens problems to fine-tune seed solutions [70]. This method appears adept at finding all local minima in χ^2-space around the location of a seed solution and also provides a covariance matrix for error estimation at the minimum. The method was not investigated for fine-tuning in this thesis because it unfortunately escaped the author's attention

Table 59: Table of Genetic Algorithm parameters used for fine-tuning example-based fit results.

Parameter	Value	Description
Population Size	1000	The number of light curves calculated for each iteration.
Mutation Rate	0.005	The probability that a bit in an individual's genotype will invert.
Cross-over Probability	0.8	The probability that a fitness-selected individual will cross over its phenotype with a partner.
Selection Slope	100	The evolutionary pressure - fittest individuals are 100 times more likely to reproduce.
Elitism	5	The fittest 5 individuals automatically survive the iteration.
Bits per parameter	16	The number of bits for each model parameter in each individual.

until a very late stage.

For the Levenberg-Marquardt algorithm, the starting point was simply set equal to the example-based fit result. Levenberg-Marquardt is an unbounded algorithm. In the case of GA, which is bounded, the bounds were centered around the regression solution. The GA was also "seeded" with the regression solution, meaning that this parameter set was re-inserted into the fitting population at each iteration. GA fitting parameters are given in Table 59.

7.2 Fits to Observed Events

More than a thousand LGM events had been detected at the time of writing. It was decided to apply the methodology developed here to two binary events to complete the proof of concept. As we shall see the fits were both successful, but very few conclusions could be drawn from such a small number of fits. There was also a selection effect as two events were chosen that were in good agreement with the training data used in the experiments in previous Sections.

7.2.1 EROS BLG-2000-5

The EROS collaboration [7] issued an alert on 5 May 2000 that EROS BLG-2000-5 was a probable Microlensing event. The follow-up group MPS [91] issued a subsequent alert on 8 June 2000 that the event was rapidly brightening (in a manner inconsistent with single lens Microlensing). The PLANET collaboration acted on these alerts and proceeded to gather data for one of the most detailed light curves ever observed for an LGM event, with more than 1300 data points from four observatories by the time of publication of their modeling paper [1].

This event was chosen due to the availability of a complete single-band light curve from a single site. (I-band data from the South African Astronomical Observatory) which eliminated complications caused by the requirement to align observations from separate sites. It was also a fairly easy target as the light curve's coverage of the event was fairly complete, including the first caustic crossing and the density of observations exceeded that of the simulated events used for training. The event exhibited strong binary lensing features in the form of a caustic entry and exit, as well as a caustic cusp approach.

Data

Only I-band DOPHOT-photometry data from a single site (SAAO) were used. The same minor cleaning process as used by PLANET in [1] was applied, whereby data points taken during seeing of more than 2.1 arcseconds and measurements with uncertainty of worse than 3 per cent were discarded a-priori. That left a total of 428 observations, down from 515 in I-band taken at SAAO.

First Attempt

The very first attempt at fitting, using the classifiers trained in Section 7.1.1, failed to find any reasonable solutions, with a minimum $\frac{\Delta\chi^2}{d.o.f}$ across the four quarters of our fit of 4802 after fine-tuning with Levenberg-Marquardt. This failure was fortunately easily explained by the fact that the actual orbital separation of the event as fit by PLANET is $a = 2.55\ \theta_E$, where the Einstein radius is in our units,

that is using the mass of the primary, only. That value was larger than the top of the range for a used to train the classifiers, at $a = 1.7\ \theta_E$ and thus impossible to fit with our classifiers. The training range needed extension.

Enlarged Parameter Space

The range of orbital separation a was subsequently extended to include the regions $0.2\ \theta_E < a < 0.6\ \theta_E$ and $1.7\ \theta_E < a < 6.0\ \theta_E$. As these two new ranges were split into two ranges each by θ, a full fit now required example-based regression and fine-tuning of eight regions in total, after which the best solution was selected on a lowest $\frac{\chi^2}{d.o.f}$-basis.

The first fit was unsuccessful, as was evident from the huge values of $\frac{\chi^2}{d.o.f}$ (above ten thousand) in all parameter ranges after fine-tuning. Models were clearly incorrect by eye, and in fact this kind of inspection revealed the tell-tale signs of premature convergence to a poor local minimum, despite the parameter starting points provided by the example-based fits. Fine-tuning by GA and Levenberg-Marquardt both failed, but why?

Unfortunately a few minutes of visual inspection reveals the answer that the SBLM cannot possibly be a good model to fit to the light curve of EROS's BLG-2000-5. The event is a very clear case of "strong lensing", i.e., a caustic is entered and exited at least once. The SBLM predicts infinite amplification and a discontinuous step on caustic entry and exit. Obviously this light curve does not display infinite amplification, but it is influenced by the effects of resolving the source in other ways as well. The light curve's caustic crossing peaks are rounded and there is no sign of discontinuity. In fact, the duration of the caustic exit is around two and a half days from peak to trough, instead of a discrete step.

It is precisely these features that ruin the fine-tuning part of the fit, as large discrepancies from the SBLM are measured around all regions of high amplification or steep slope, even when each SBLM parameter is correct; the SBLM does not include resolved source effects.

The correct approach would have been to extend the model to include resolved source effects throughout, as discussed in Sections 2.3 and 8.1.5. However, the scope of this thesis was the SBLM and an interim solution to determine whether the fit could succeed, without widening the scope too much, was simply to remove the parts of the light curve that were clearly in conflict with an assumption of zero source resolution. These parts are the tops of all peaks as well as the caustic exit region, although the light curve in its entirety is still sent to the example-based portion of the fit, which requires peak information and should be much more robust to noise.

The trimmed light-curve was successfully fit by the fine-tuning operation, at least to $\frac{\chi^2}{d.o.f} = 17$ which sounds fairly high if taken at face value. Practitioners who analyzed this event in great detail [1] concluded that the uncertainties reported for photometry were too small. No such correction had been applied here, which increased the value of χ^2 above what would normally be expected of a good fit. In addition, this result was obtained using genetic fine-tuning only, which is known for extremely slow convergence close to the solution and may not have quite arrived at the global minimum.

The SBLM parameters obtained from our standard fitting procedure are given in Table 60 (in the units used in this thesis), and the light curves are plotted in Figure 47. Parameters derived by the PLANET collaboration in [1] are also shown in the Table, translated into the SBLM. The mass ratio q does not need translation. The lens equation in the PLANET paper is stated in terms of the (angular) Einstein radius of the total binary mass, whereas the SBLM in this thesis uses the Einstein radius of the primary. Translation requires scaling of PLANET parameter d_{tc} by a factor of $\sqrt{1+q}$ while the Einstein radius crossing time is scaled by the inverse of this factor. PLANET utilized a parameterization which makes some of their fitted parameters hard to translate into the SBLM. In particular, the usual distance of closest approach to the primary or centre of mass of the binary was replaced by the distance of closest approach to the caustic cusp, which is itself dependent on

Table 60: SBLM parameters in thesis units for event EROS BLG-2000-5 using the example-based fitting procedure with genetic fine-tuning.

Parameter	Example-Fitted Value	Fine-tune Value	PLANET [1]
$a(\theta_E)$	2.269	2.31	d_{t_c}=1.928 implies a=2.549
$\theta(^\circ)$	296	345	α'=74.18 implies θ=344.18
$b(\theta_E)$	0.161	0.321	-
$m_0(mag)$	N/A	17.64	-
q	0.806	0.743	0.7485
$t_e(days)$	N/A	64.52	t'_e=99.8 implies t_e=75.5
$t_m(days)$	N/A	18.04	-

the event geometry. The fitted, un-lensed source magnitude is not presented in the PLANET paper - only the recalibrated, un-lensed magnitude.

Another issue is that PLANET's much more detailed modelling process included parallax and extended source effects, which the SBLM did not. Apart from these translation and comparison issues, the fitted results presented here show good agreement with the PLANET results.

Discussion

EROS BLG-2000-5 was successfully fitted using the methodology in this thesis up to the limitations of the SBLM. The example-based portion of the fit came to within one per cent of the fine-tuned value for a, while the worst-fitted parameter was b, at a difference of 16 per cent of the total range under consideration. These values were close enough to a deep local minimum (in all likelihood the global minimum) to allow a fine-tuning algorithm to succeed, although not a gradient-based method like Levenberg-Marquardt, indicating that the example-based result was unlikely to be inside the covergence well.

Achieving success with this fit was important. The same event proved difficult and time-consuming for practitioners using standard fitting methodologies. It is fair to say that an example-based fit would have been a useful aid.

7.2.2 OGLE-2003-BLG-267

OGLE-2003-BLG-267 was discovered in 2003 by the OGLE-III Early Warning System [92]. The event is another example of strong lensing, apparently featuring

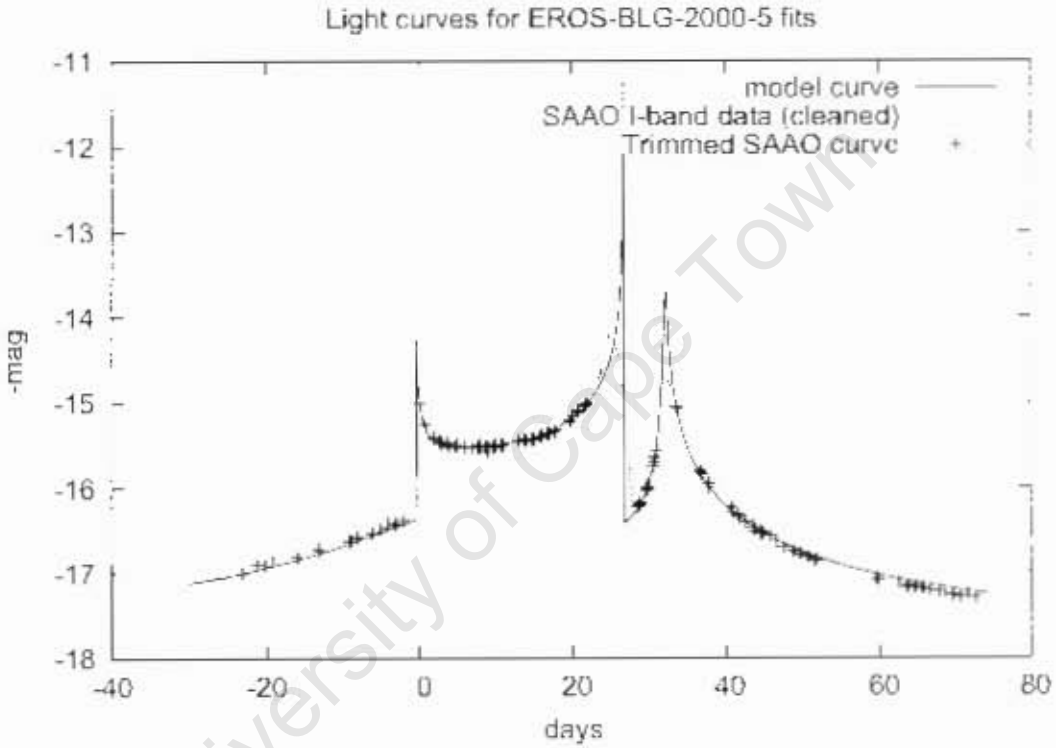

Figure 47: Original light curve (with minor cleaning) as published by PLANET ([1]), points used for fine-tuning and the SBLM light curve as fitted (after fine-tuning).

a caustic crossing and two cusp approaches (in the binary lens interpretation). The caustic crossings are covered by OGLE observations and the event was modeled by OGLE in [88].

Data

Data are publicly available and consist of 259 I-band points reduced by difference photometry [88]. At first, data were not cleaned or changed in any way but for subsequent fits the error estimates were adjusted by simply adding 1 per cent to all errors. The effect was to reduce the large dependency of χ^2-based fine-tuning on a small number of points with very small error estimates.

First Attempt

The classifiers trained in Section 7.1.1 failed to fit the event. $\Delta\chi^2$-values were in the hundreds of thousands after fine-tuning. Visual inspection revealed that the fits to some regions of parameter space produced a curve fairly similar to the data, but missing the cusp approached on either side.

Multi-peak Criterion and Error Adjustment

It was decided to introduce some information specific to the light curve into the fitting procedure. The classifiers were re-trained with a data set of events that had 3 peaks in brightness or more. This criterion was applied both to the classifier data sets as well as the Genetic fine-tuning algorithm, which was achieved simply by assigning a low fitness value to individuals in the GA population that did not meet the criterion. The "high variance" restriction introduced in Section 6.6.2 was dropped for this training set, as a minimum peak number criterion was considered strict enough to eliminate the large proportion of irrelevant training events that did not resemble the target event.

Unfortunately fits using this criterion were still unsuccessful until the OGLE error estimates were adjusted. The lesson from the SBLM fit to EROS BLG-2000-5 was that the SBLM model is not a perfect description of real LGM events and

therefore allowances have to be made. The data were adjusted by adding a constant value of one per cent to the OGLE error estimates, as well as truncating the peaks and caustic exit data points for the fine-tuning portion of the fit.

This immediately led to the successful fitting of two degenerate models to the light curve, with $\frac{\chi^2}{d.o.f} = 19$ and 24, respectively. The models were quite close in all parameters but θ, which differed by roughly 180 degrees. The cause of degeneracy is simple in this case: one solution is a near time-reversed version of the other. As the peaks were truncated for the fine-tuning portion of the fit, both models were viable. The χ^2-value of the complete curve, without peak truncation (and without additional fitting) favoured the first solution (solution A in Table 61) at $\frac{\chi^2}{d.o.f} = 3171$ vs 12255, respectively. Subsequent fine-tuning fits with extended models and all available data should not have any difficulty in choosing the best of the two solutions, as Figure 48 shows a large difference over data points not included in the SBLM fine-tune fit. The fit was not pursued further at this stage. Both solutions and their model light curves are shown in Table 61 and Figure 48.

Fitted parameters from two close binary models (numbers 1 and 5) in a study by Jaroszyński et al [88] are also shown in Table 61. Their model parameters required scaling by the factor $\sqrt{1+q}$, as was the case with PLANET for EROS BLG-2000-5, before they could be directly compared. The binary parameters a and q from the paper are shown in the Table for two different models, the first being very similar to the SBLM while the second includes binary rotation and parallax effects. The fine-tuned Solution B shows reasonable agreement with models 1 and 5 from [88]. It would have been optimistic to expect perfect agreement, given the number of points at peak excluded in the SBLM fit.

Table 61: SBLM parameters for both viable models in thesis units for event OGLE-2003-BLG-267 using the example-based fitting procedure with genetic fine-tuning.

Parameter	Example-Fitted Solution A	Fine-tune Solution A	Example-Fitted Solution B	Fine-tune Solution B	[88] simple model	[88] with parallax and rotation
$a(\theta_E)$	0.561	0.686	0.485	0.757	d_9=0.353 implies a=0.456	d_6=0.551 implies a -0.739
$\theta(^o)$	46.3	91.3	309	275	-	-
$b(\theta_E)$	0.138	0.0211	0.097	0.074	-	-
$m_0(mag)$	N/A	19.25	N/A	19.15	-	
q	0.519	0.426	0.536	0.699	0.668	0.797
$t_e(days)$	N/A	40.7	N/A	31.7	-	-
$t_m(days)$	N/A	-7.2	N/A	13.2	-	-

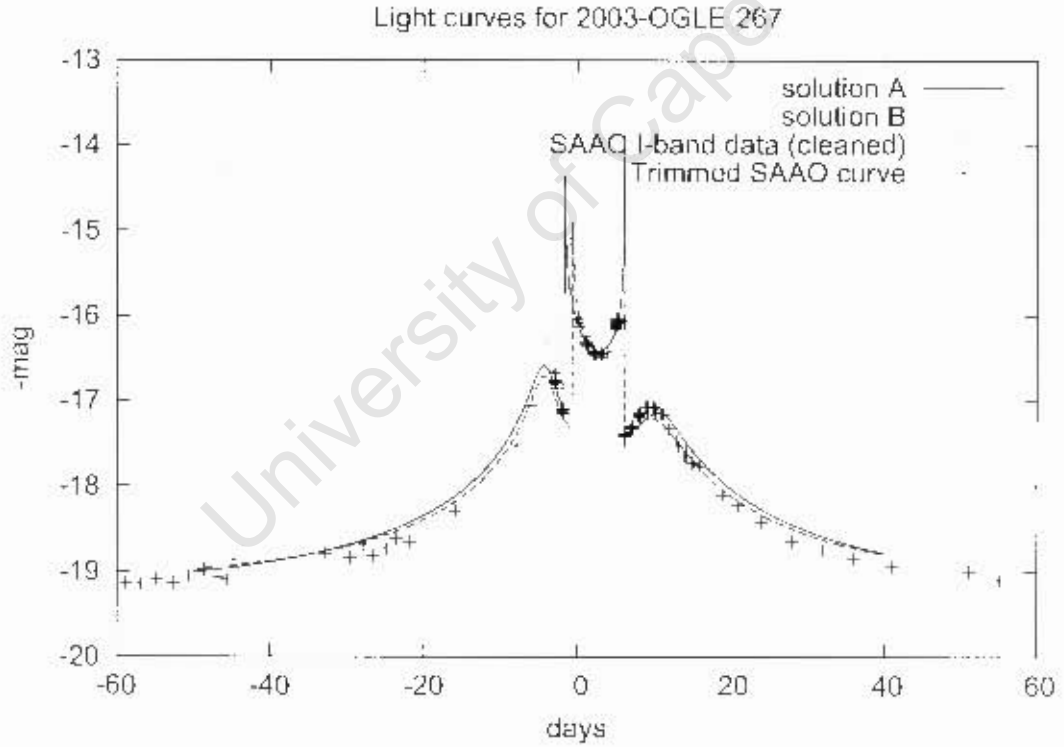

Figure 48: Original light curve, points used for fine-tuning and two SBLM light curves corresponding to Solutions A and B (after GA fine-tuning).

8 Discussion, Conclusions and Future Work

This is the final Chapter, devoted to a discussion of the thesis, conclusions that can be drawn and suggestions for future work.

Overview of Thesis and Results The majority of effort for this thesis went into developing reliable software for the simulation of LGM events using the standard binary lens model and testing the regression capability of various "unconventional", mainly example-based algorithms, as applied to the SBLM and simulated light curves. The model and this implementation of it was discussed in detail in Chapter 2. A highlight was the introduction of Asada's [33] method of lens image calculation, which led to a factor of six improvement in calculation speed. Chapter 3 showed, by way of small studies such as the convergence well size calculation in Section 3.3.3, why SBLM-fitting is particularly difficult.

Chapter 4 discussed the derivation and use of "features" from binary lens light curves that could be used for regression. There were several results from this Chapter. It was found that series expansions are poor approximations of binary lens light curves, as are fits to binary lens events using single lens models. Genetic Programming, an extremely general and powerful technique, also failed at providing a mapping between light curves and model parameters that would have been accurate enough to either fit the parameters by itself or serve as an aid to fitting. Genetic Programming is a wide area of research, however, and although this implementation failed it would be premature to abandon the method altogether in future work. Feature selection algorithms were tested against simple benchmarks for speed and accuracy, and proven to work - some better than others.

Feature selection on simulated SBLM events proceeded with three competing algorithms in Section 4.4. Selection algorithms favoured derived features instead of light curve points themselves. Overall, the three competing algorithms selected similar, but not identical, features. That inconsistency and the selection of some features which clearly bore no relevance to the parameter mapping at all showed

that the selection methodology was far from perfect.

Chapter 5 set the scene for the data mining-based fits that followed. It demonstrated the problem of premature convergence that gradient-based methods have with the SBLM.

In Chapter 6, experiments were performed that varied the composition of data sets, feature selection and regression algorithms, to determine the most effective combination of these to fit the SBLM to simulated light curves. A benchmark was set by two conventional algorithms, namely Powell's method and a simple comparison library. Thereafter, the pre-processing methodologies for noiseless as well as noisy light curves were developed. Pre-processing was found to be of cardinal importance in SBLM fits. The algorithm developed here included smoothing by fitted B-splines, truncation of the light curve outside the region of interest and interpolation to fill in gaps in observations. It proved to be fairly reliable in subsequent experiments and greatly facilitated fitting accuracy, probably due to the reduction in model dimensionality that it brought about as parameters m_0, t_e and t_m were eliminated from the fit.

The first experiment compared the fitting accuracy of ten example-based algorithms, using pre-processed light curves; the second performed the same experiment, using selected features instead of the light curves themselves. The experiments were useful because the two neural network algorithms were discarded on the spot due to inconsistency and performance issues. Three algorithms performed very well in these experiments and subsequent ones: the REP tree inducer, the M5P tree inducer and the simple, nearest-neighbour, IBk algorithm. These experiments also showed that pre-processed light curves out-performed the data sets consisting of algorithmically-selected, derived features.

Section 6.4 introduced artificial noise into the experiments by way of two different noise simulation models. The first was based on the noise and gap distributions of actual observations while the second used a simple model to add Gaussian noise and randomised gaps to simulated data. The effect of noise on fitting accuracy

was subsequently tested and found to be damaging to accuracy, especially for some classifiers, such as the IBk nearest neighbour classifier. It was also established in a small experiment that noisy light curves are fitted more accurately using an algorithm trained on noisy light curves than on idealised light curves. Further experiments with noise showed that the mass ratio, q, is much harder to fit with increasing Gaussian noise than the other parameters.

A subset of successful classifiers were run with parameters set away from their defaults in an attempt to improve their accuracy. This did not yield worthwhile results, indicating that the accuracy of example-based SBLM fits is constrained more by difficulties in the problem and how it was posed than shortcomings of the algorithms themselves. Building on this result, an experiment was performed biasing the training events towards high brightness variance. This procedure led to a large improvement in accuracy, presumably because a large number of events which were too similar to the single lens model and hence not uniquely fittable by any method, were excluded.

Discrete classifiers were put to the test in Section 6.6.3 and found to be effective, although a direct comparison with their regression counterparts was not possible.

Up to that point a parameter range covering the "lensing zone" had been used, but Section 6.6.4 experimented with dividing the example data sets along parameter boundaries corresponding to known degeneracies or time-symmetries, namely at $a = 1.0\ \theta_E$ and $\theta = 180°$. Two large improvements in fitting accuracy were achieved in this way: the first by avoiding ambiguity in the input data, the second by increasing the density of samples in the training set. This experiment brought regression correlation above 0.9 for some model parameters for the first time.

The final experiment (Section 6.6.5) applied meta-algorithms of various kinds to light curves. For example with "voting", the idea is that algorithms of different types should complement each other and that taking the mean prediction would result in higher accuracy than is obtainable with a single algorithm. In practice results were not improved and could turn out worse than the best of the algorithms

did by themselves. Results were mixed; most meta-algorithms slightly improved on the accuracy achievable by a single classifier. Success was also dependent on which meta-algorithm was applied to which single algorithm. A strong combination was discovered when the Bagging meta-algorithm was used with the REP tree-inducing algorithm - the result was quick to train and of comparable accuracy to the (much slower) IBk, nearest neighbour algorithm, which was also more susceptible to noise. Bagging with an REP-tree combination was chosen for example-based fitting of the SBLM model to light curves of real Microlensing events in Chapter 7.

Fits to observed events started with the training of classifiers combining all the results from the preceding experiments. IBk, M5P and REP tree (with Bagging) algorithms were trained using large data sets of 50000 events with simulated noise, biased towards high variance and trained on a model parameter space divided into four parts. Test set correlations were high: between 0.74 and 0.94 for all parameters. A final comparison was made between data sets consisting of selected features and simple pre-processed light curves. The REP algorithm with Bagging and direct use of curves outperformed the other two algorithms and the alternative event description once again, and this combination was applied to the observed events. Fine-tuning methodology was also developed in Section 7.1.2, giving us the option of a very fast Levenberg-Marquardt algorithm that frequently converged prematurely or a slow but robust Genetic Algorithm to start from the result of an example-based fit and complete it based on a χ^2-metric.

The first event to be attempted was EROS-BLG-2000-5, a densely-sampled, strong binary lensing candidate. The classifiers trained earlier failed to find a successful fit because the event's projected orbital separation, a, was outside the parameter range used for training the classifiers. This was easily corrected by training four additional classifiers on extended ranges, which yielded a better example-based fit, but still did not converge to a good solution during Genetic fine-tuning. Success was achieved by realizing that the limits of the SBLM had been reached. The event showed non-negligible deviations from the SBLM and these were sabotaging the fit.

The removal of small regions of the light curve at the peaks of amplification and the caustic exit led to a successful fit.

The second event was OGLE-2003-BLG-267: another caustic-crossing event but with much sparser sampling and more noise than EROS-BLG-2000-5. The standard example-based fit also failed for this event and led to a slight change in approach whereby classifiers were retrained on a data set that only contained events with more than three amplification peaks. Error estimates were also adjusted upwards and the small regions at peak and caustic exit were removed. The training restriction of high variance was dropped, however, broadening the scope of fittable events (with three or more peaks). The fit was successful after applying these changes.

The Verdict Successful fits to real (as opposed to simulated) events were achieved with example-based methods, although only two were attempted. Perhaps the first question to answer was whether the effort of developing an example-based methodology was justified at all. At the inception of this thesis, fitting binary lens models to Microlensing events was a painful, manual undertaking. It has in fact remained so, despite the enormous advance in computational power available to practitioners over the past few years. The difficulty of this fitting problem meant that the thesis objective has remained relevant - any aid to the fitting process should still be welcomed by the community, and example-based methods present an intriguing alternative approach. Unlike most regression algorithms, they are not based on a χ^2-metric at all. Of course it is fairly obvious from fits to real observations that the example-based method is not fully automated at this stage, either. More work is required before this goal can be achieved, but there is every hope that it could be. Based on the work performed here, the verdict is that example-based regression methods can be successfully applied to LGM binary lens fits and have a place in the future of the analysis of these events.

8.1 Future Work

This short section deals with a number of topics that merit further investigation or fall outside of the scope of this thesis.

8.1.1 Fits to More Observed Events

An obvious shortcoming was that the methodology was only applied to two real events due to time constraints. Future work should attempt to fit an unbiased sample of all binary lens events observed to date.

8.1.2 Error Estimates

This thesis presented many fitting results without error estimates - an unfortunate consequence of example-based fitting. Although the various algorithms' accuracy for the training and testing data sets were always measured, the algorithms in this thesis do not provide an indication of the uncertainty for any individual fit without further work. These numbers are not analytically available as they would have been for some of the more conventional algorithms such as Levenberg-Marquardt, for which the final Jacobian matrix provides an error estimate under assumptions of normally distributed noise, etc.

In practice it is likely that all example-based fits would be followed by a fine-tuning operation using conventional algorithms, most of which provide the user with error estimates. As last resort, and actually likely to be the most accurate estimate of uncertainty anyway, χ^2 should be calculated as a function of perturbations from the solution, in each model parameter. Such a calculation would yield an error estimate for that particular solution. It is impossible to know with certainty whether any given solution is the global solution, regardless of fitting methodology.

8.1.3 Model Ambiguities

Ambiguity in the model causes great difficulty for example-based methods because it leads to conflicting training data. Ambiguities were dealt with in a rather

casual way in this thesis by dividing the input parameter space along the boundaries of known ambiguities. This trick will not work when the model is extended to include additional effects like parallax and blending because the ambiguities get more complicated and numerous. A new approach would be required, perhaps keeping to the data mining theme and utilizing the rapidly growing field of clustering, or pre-classifying events automatically and training a specialist algorithm in each category.

Perhaps another type of "model ambiguity" that needs to be provided for in Future work is the handling of events that are simply impossible to fit. In the SBLM there are in fact large areas of the parameter space where a value for, say, q (mass ratio) simply cannot be extracted because no caustics were crossed and the binary event looks exactly like a single lens event. The data sets used in this thesis attempted to deal with this issue by creating training sets biased towards high variance or only including events with a minimum number of peaks. These methods were effective but quite crude.

8.1.4 Feature Selection

Some time was dedicated to feature selection techniques and these were fairly successful. Feature sets performed almost as well as pre-processed light curves despite containing a fraction of the amount of data in the original curve. There is hope that more work in this area could boost the efficiency of training data based on features. The technique could benefit greatly from an algorithmic approach to constructing features in the first place. The features used in this thesis were a somewhat random selection.

8.1.5 Fits Using Extended Models

The 7-parameter SBLM formed the basis of much of this thesis's LGM light curve fitting strategy because it does a good job of describing most light curves, but it is by no means complete. Extensions to the SBLM are described in Section 2.3 and this Section explores the consequences to fitting of including, and in fact

excluding, known extensions to the SBLM.

Additional parameters present serious challenges to fitting algorithms. The first is simply that the regression search space grows exponentially with the number of model parameters that have to be fitted. That essentially disqualifies methods that are linearly dependent on the parameter search space size, such as naive, library-based methods. Secondly, some extensions are hard to calculate in the numerical computational sense, most notably the resolved source effect which takes more than 100 times as long to calculate an amplification point in this implementation as the point lens model we have been using till now. Thirdly, new degeneracies are introduced with new parameters. The problems associated with degeneracies are severe and are discussed in Section 3.3.4. Finally, "scale" parameters t_e, t_m and m_0 are no longer separable from the rest of the lens parameters as was the case in the SBLM. The parallax effect is dependent on absolute time, or at least is now dependent on the time of year and the time scale of an event, t_e.

Perhaps the most effective way to deal with additional parameters would be to break the fitting process into parts such that some parameters are resolved before others, thus breaking the exponential dependency of fitting space volume on parameter number. This technique was employed throughout this thesis as well as in successful advanced fits in the literature (e.g., [1]).

In the case of extending the SBLM, one could start by fitting the SBLM, resolve its parameters and then apply extensions. Unfortunately Microlensing model parameters are not generally independent and we have not investigated whether this approach would be valid - that is, whether the global minimum of the regression surface using an extended model would necessarily be close to the SBLM minimum. Therefore the next Section is a short investigation on whether extensions to the SBLM can be treated as perturbations or would have to be included at all stages of the fit.

Table 62: Ranges of model extension parameters R_s, f and ρ.

Parameter	Minimum	Maximum
R_s	0	0.05 θ_E
f	0	1.0
ρ	0	1.0 θ_E

Light curve changes due to Extensions

The impact on our fitting procedure of extending the simple 7-parameter SBLM can be investigated by measuring the effect the new parameters have on an SBLM light curve. In other words, we generate a set of parameters for the standard model and compare its corresponding light curve to the light curve that is generated by an extended model that has standard parameters equal to that of the standard model. One way of making this assessment is to look at the $\frac{\Delta\chi^2}{d.o.f}$ between extended curves and their SBLM counterparts. Fig. 49 shows the distribution of $\frac{\Delta\chi^2}{d.o.f}$ for 1000 extended model light curves as compared with the SBLM curve with the same 7 standard parameters as was used for the extended curve. The extension parameters were randomly selected from the ranges given in Table 62. The extension parameters R_s, f and ρ were considered; resolved source size (in Einstein radii), blending parameter and the parallax scale parameter, respectively. The secondary parallax parameter ψ was not considered, as ψ is irrelevant unless ρ is non-zero.

Fig. 49 is quite alarming. The distributions for ρ and f peak at a $\frac{\Delta\chi^2}{d.o.f}$ value close to 1000 and are almost entirely above 1, which shows that any inclusion of blending or parallax radically alters the simple model curves. These parameters need to be taken into account if there is any chance of them occurring in genuine LGM events.

R_s has a bi-modal distribution and in fact most values of R_s will not cause a significant change in the corresponding simple model light curve. Figs. 50, 51 and 52 are useful for illustrating the point made by Fig. 49 and discussed in 2.3. The parallax parameter ρ drastically alters light-curve shape, causing large deviations to the simple model light curve. In some cases parallax-affected curves look like completely different events.

We can extract more detail by plotting $\log(\frac{\Delta\chi^2}{d.o.f})$ as a function of the additional

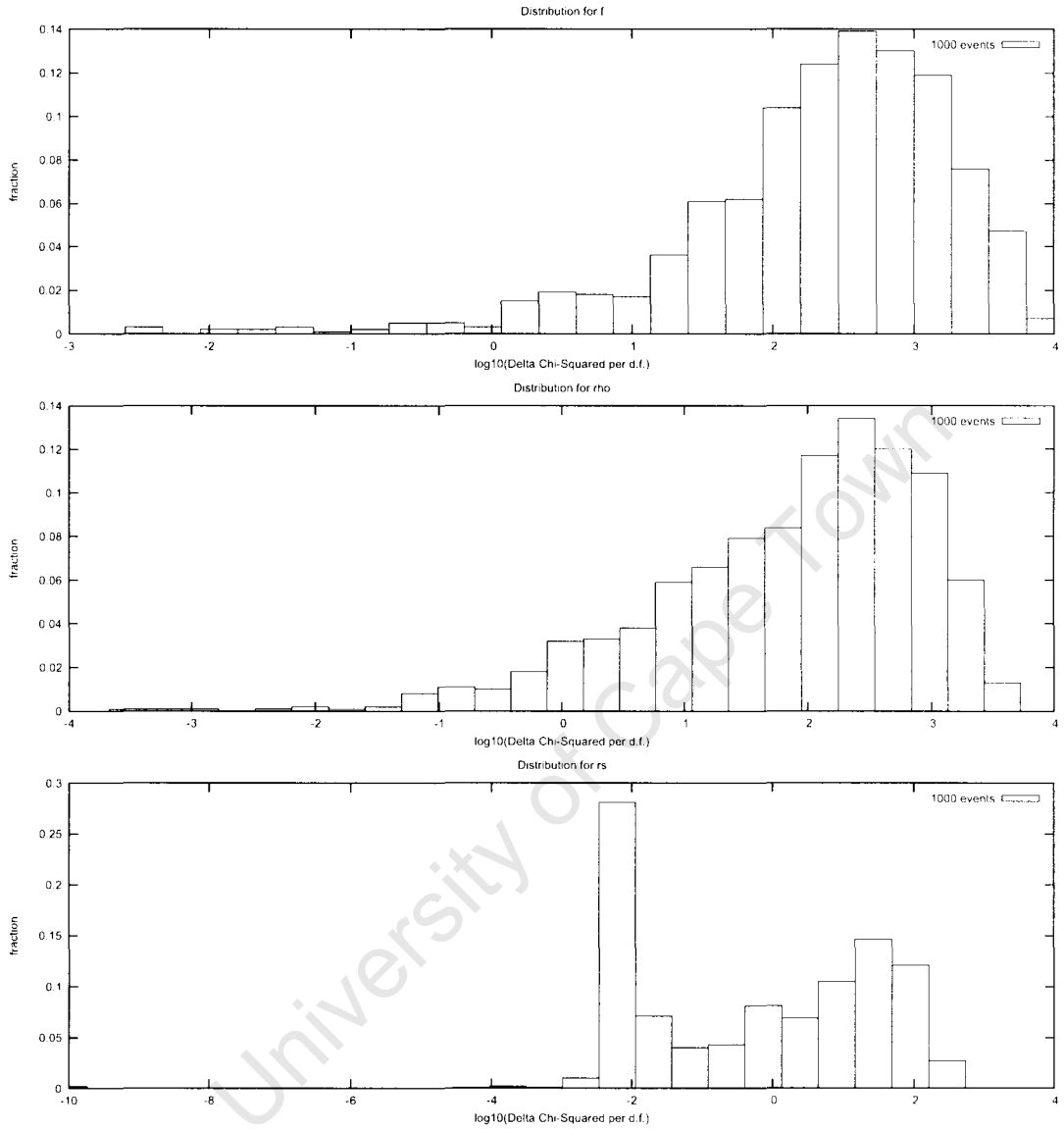

Figure 49: The distribution of $\log(\frac{\Delta\chi^2}{d.o.f})$ for the extension parameters R_s, f and ρ. Note that a completely flat distribution over the allowed range of each extended parameter was used in constructing these $\log(\frac{\Delta\chi^2}{d.o.f})$-distributions. In other words, these give some indication of the sensitivity of light curves to model extensions on a basic level but do not reflect the actual occurrence of the underlying physical effects in observed light curves.

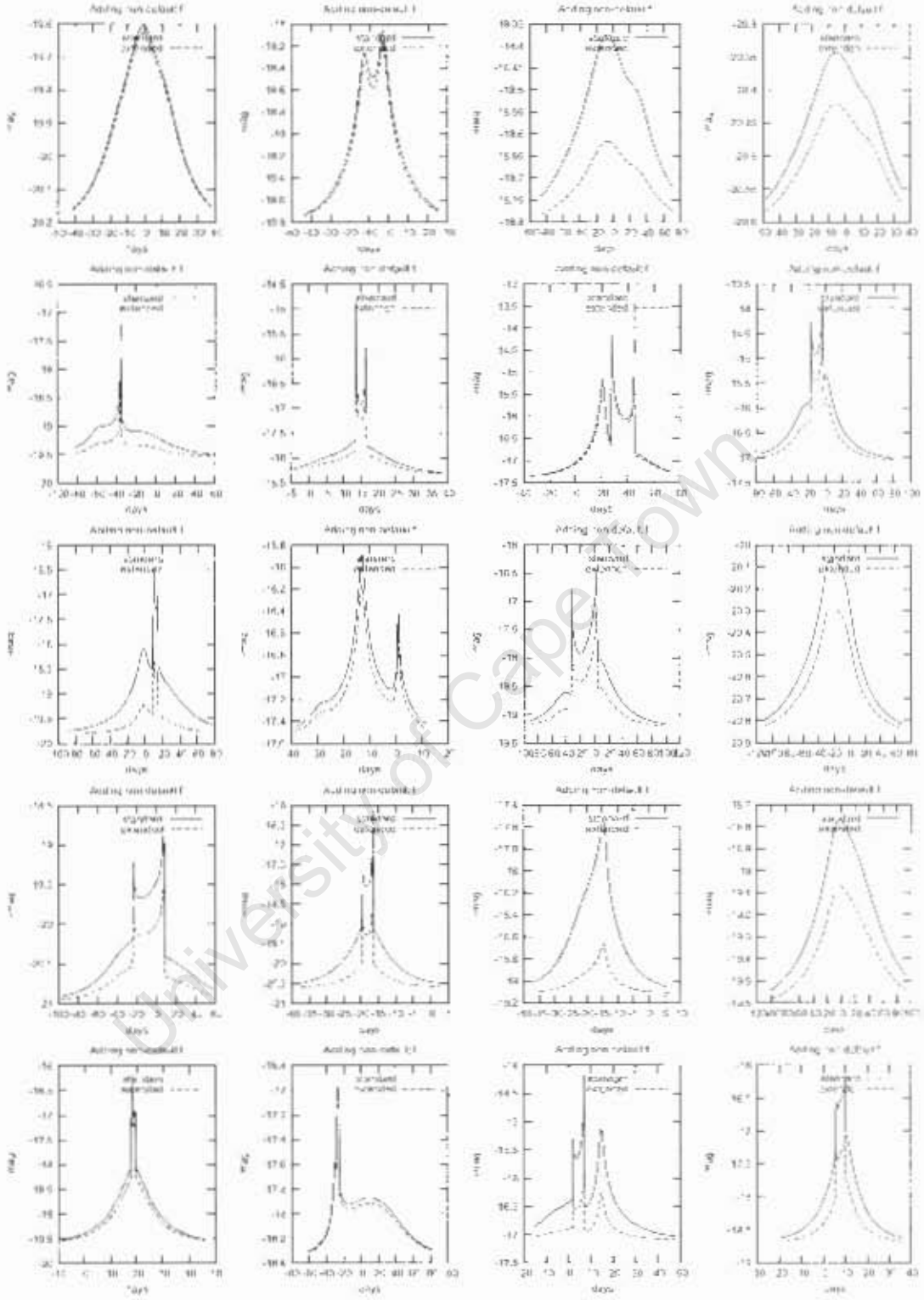

Figure 50: Examples of randomly generated standard models and the light curve corresponding to the same standard parameters but with a value of f different to the SBLM value of 1.0.

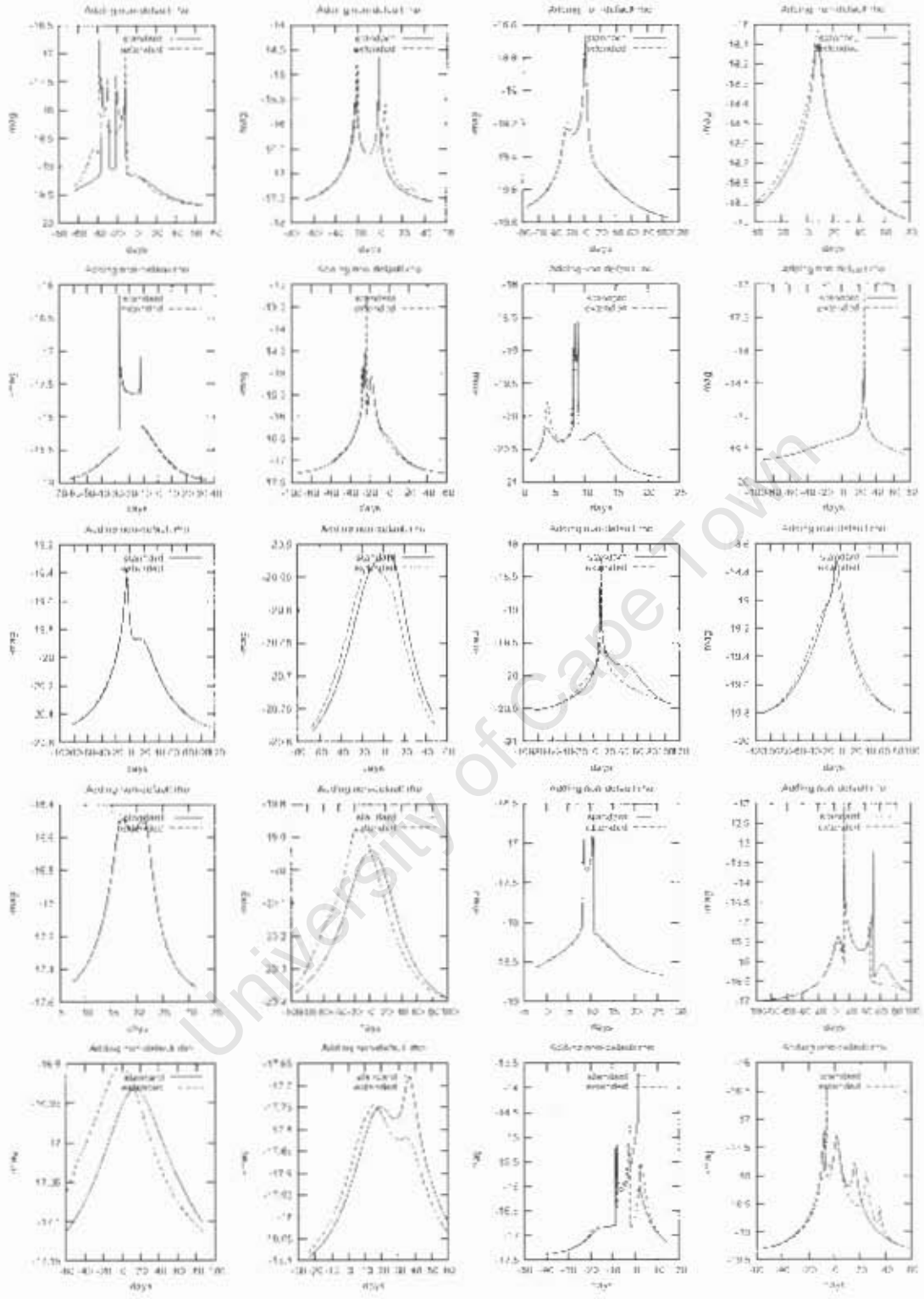

Figure 51: Examples of randomly generated standard models and the light curve corresponding to the same standard parameters but with a value of ρ different to the SBLM value of 0.0.

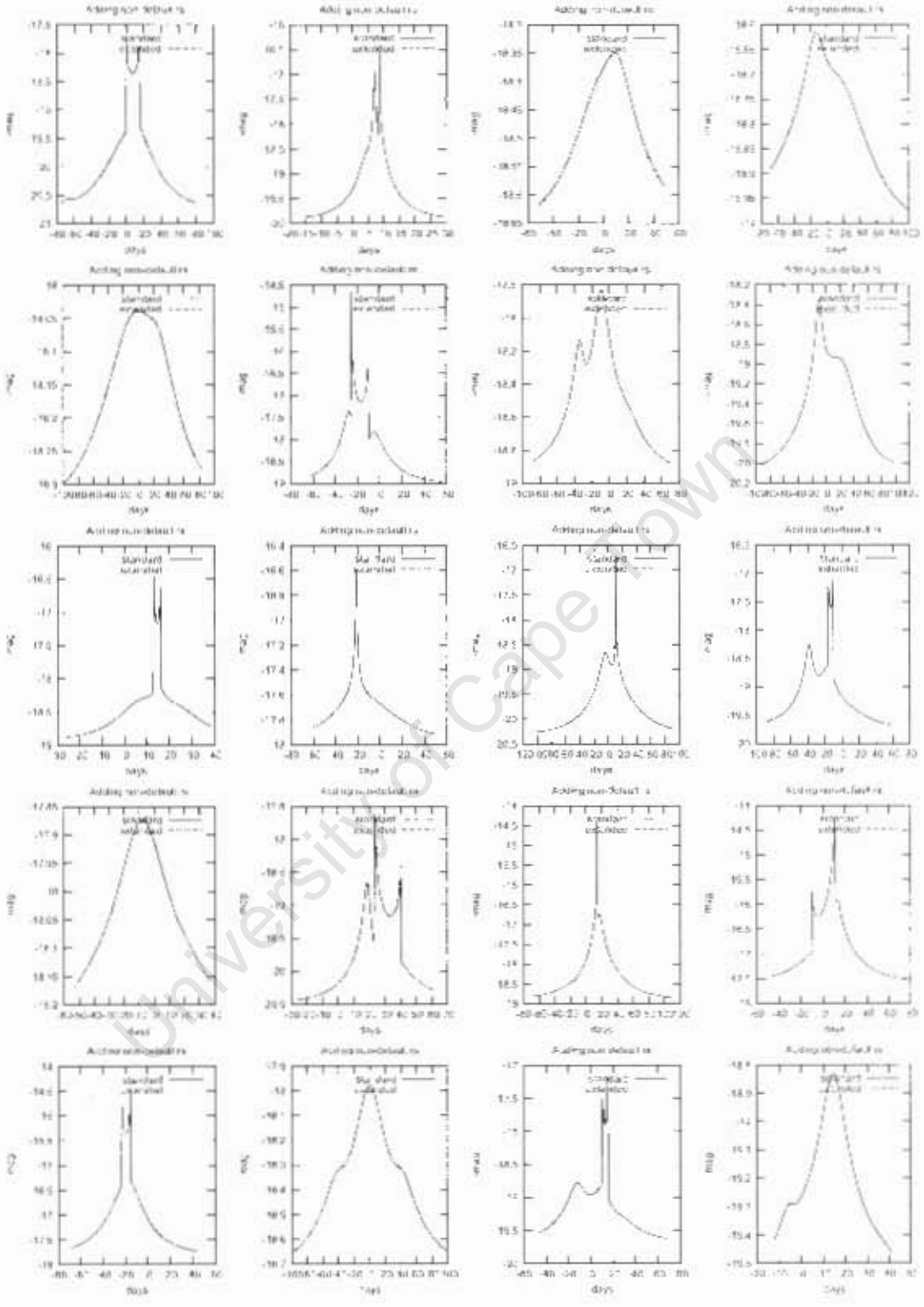

Figure 52: Examples of randomly generated standard models and the light curve corresponding to the same standard parameters but with a value of R_s different to the SBLM value of 0.0.

model parameter separately, as shown in Fig. 53. The number of points above $\log(\frac{\Delta\chi^2}{d.o.f}) = 0$ agrees with the results of the distribution plots in Fig. 19. These plots provide additional information in the form of the extension parameter magnitude that is required to cause significant perturbations. As noted in Section 2.3.1, the algorithm used to calculate extended source amplification in this thesis suffered from an inaccuracy in some situations where the source overlaps a caustic (but not a cusp). This inaccuracy would almost certainly not materially affect the extended source results presented here.

In the case of f (blending), only curves that are practically untouched by blending have a $\frac{\Delta\chi^2}{d.o.f}$ of less than 1. This end of the plot corresponds to 0 on the x-axis, denoting zero difference between the standard model value for f of 1 and the extended setting for f ranging from 1 at this end to 0 at the right of the plot. Just about any value of f smaller than 0.9 will cause significant deviation from the standard model.

The situation is equally serious for ρ. As the standard model value of ρ is zero, the x-axis shows the value of ρ directly. Any non-zero value of ρ at all is likely to cause major deviations from the standard model light curve.

Finally, the scatter plot shows that R_s may indeed have little or no effect on the curve for a large range of values. The standard model default value for R_s is again zero, the x-axis translates directly into the value of R_s. However, scatter is large enough to exceed $\frac{\Delta\chi^2}{d.o.f} = 1$ for at least some events at all values of R_s from about $R_s = 0.005$, indicating that not even this extension parameter can reasonably be left out of binary light curve calculations.

Conclusions

The results of this small study need to be taken with a pinch of salt as the actual distribution of standard and extension parameter values was not used and the results would be highly dependent on these. However, the results are alarming enough to warrant the conclusion that these well-known extensions cannot be excluded from

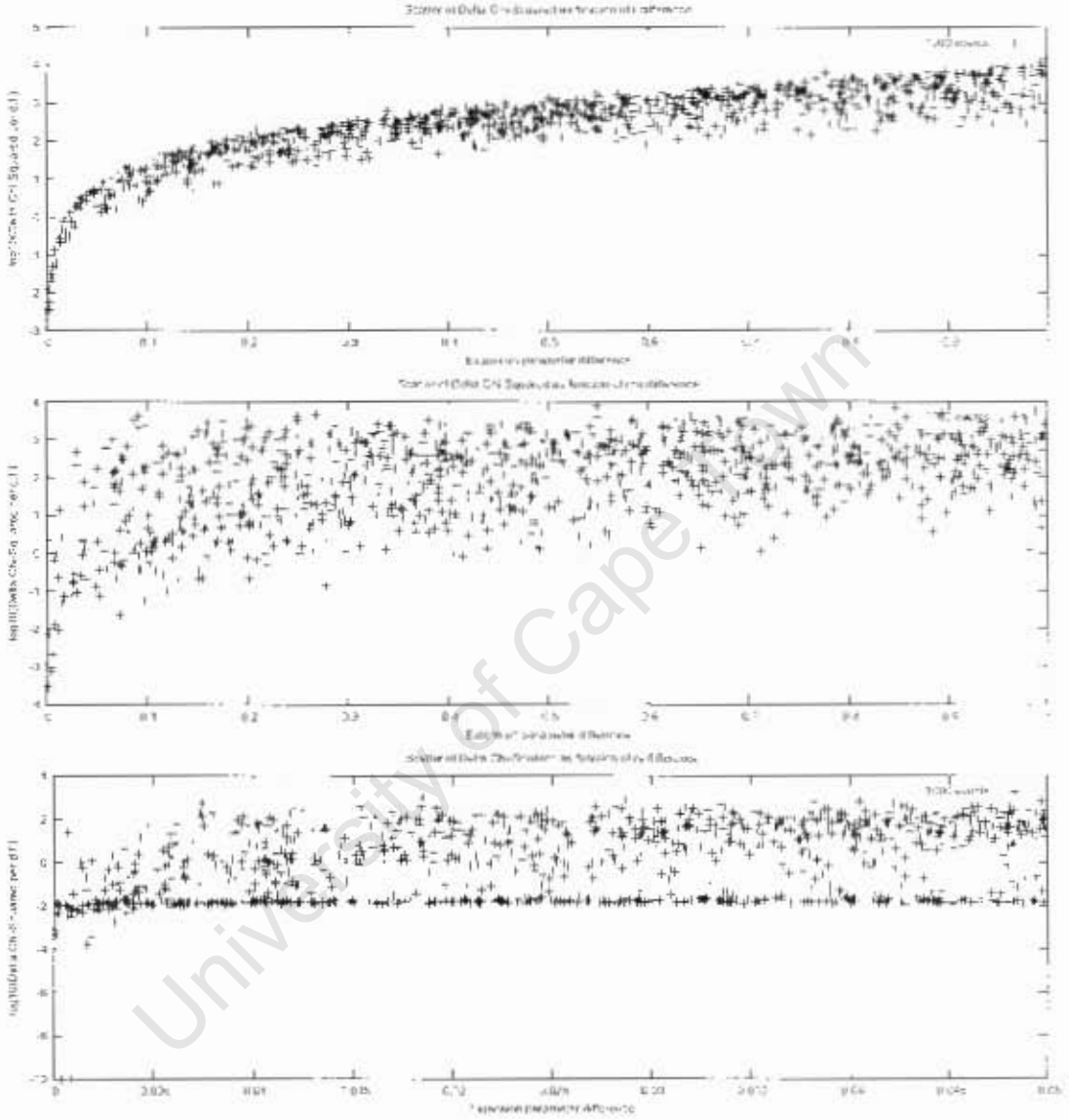

Figure 53: A scatter plot of $\log(\frac{\Delta\chi^2}{d\omega_i f})$ for the extension parameters R_s, f and ρ as a function of R_s, f and ρ, respectively.

fits to real events because they have more than just perturbatory effects in general. It is also unlikely that the SBLM global minimum will always be close to the global minimum of the extended model.

Does that mean that all SBLM fits are invalid? Practitioner experience indicates otherwise. This topic requires a more detailed analysis, in particular with the use of realistic model parameter distributions.

List of References

[1] J. H. An, M. D. Albrow, J.-P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, M. Dominik, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams, "First Microlens Mass Measurement: PLANET Photometry of EROS BLG-2000-5," ApJ, vol. 572, pp. 521–539, June 2002.

[2] 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, vol. 606, pp. L155–L158, May 2004.

[3] B. S. Gaudi, M. D. Albrow, J. An, J.-P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, M. Dominik, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. M. Naber, J.-W. Pel, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, P. M. Vreeswijk, R. Watson, and A. Williams, "Microlensing Constraints on the Frequency of Jupiter-Mass Companions: Analysis of 5 Years of PLANET Photometry," ApJ, vol. 566, pp. 463–499, Feb. 2002.

[4] A. Einstein, "Lens-Like Action of a Star by the Deviation of Light in the Gravitational Field," *Science*, vol. 84, pp. 506–507, Dec. 1936.

[5] D. P. Bennett, C. Akerlof, C. Alcock, R. Allsman, T. Axelrod, K. H. Cook, K. Freeman, K. Griest, S. Marshall, H.-S. Park, S. Perlmutter, B. Peterson, P. Quinn, A. Rodgers, C. W. Stubbs, and W. Sutherland, "The First Data from the MACHO Experiment," in *Texas/PASCOS '92: Relativistic Astrophysics and Particle Cosmology*, pp. 612–+, 1993.

[6] B. Paczynski and A. Udalski, "Optical Gravtational Lensing Experiment (OGLE)," IAU Circ., vol. 5997, pp. 1–+, May 1994.

[7] E. Aubourg and et al., "Status of EROS (presented by M. MONIEZ)," in *Gravitational Lenses in the Universe*, pp. 493–+, 1993.

[8] A. Gould, "Microlensing search for planets [review article]," *New Astronomy Review*, vol. 49, pp. 424–429, Nov. 2005.

[9] G. E. Moore, "Cramming more components onto integrated circuits.," IAU Circ., vol. 38, 1965.

[10] A. Einstein, "The Foundation of the General Theory of Relativity," *Annalen der Physik*, vol. 49, 1916.

[11] D. Walsh, R. F. Carswell, and R. J. Weymann, "0957 + 561 A, B - Twin quasistellar objects or gravitational lens," Nature, vol. 279, pp. 381–384, May 1979.

[12] J. Pelt, R. Schild, S. Refsdal, and R. Stabell, "Microlensing on different timescales in the lightcurves of QSO 0957+561 A,B," A&A, vol. 336, pp. 829–839, Aug. 1998.

[13] B. Paczynski, "Gravitational microlensing by the galactic halo," ApJ, vol. 304, pp. 1–5, May 1986.

[14] C. Alcock, C. W. Akerloff, R. A. Allsman, T. S. Axelrod, D. P. Bennett, S. Chan, C. H. Cook, K. C. Freeman, K. Griest, S. L. Marshall, H. S. Park, S. Perlmutter, B. A. Peterson, M. R. Pratt, P. J. Quinn, A. W. Rodgers, C. W. Stubbs, and W. Sutherland, "Possible Gravitational Microlensing of a Star in the Large Magellanic Cloud," Nature, vol. 365, pp. 621–+, Oct. 1993.

[15] E. Aubourg, P. Bareyre, S. Brehin, M. Gros, M. Lachieze-Rey, B. Laurent, E. Lesquoy, C. Magneville, A. Milsztajn, L. Moscoso, F. Queinnec, J. Rich, M. Spiro, L. Vigroux, S. Zylberajch, R. Ansari, F. Cavalier, M. Moniez, J. P. Beaulieu, R. Ferlet, P. Grison, A. V. Madjar, J. Guibert, O. Moreau, F. Tajahmady, E. Maurice, L. Prevot, and C. Gry, "Evidence for Gravitational Microlensing by Dark Objects in the Galactic Halo," Nature, vol. 365, pp. 623–+, Oct. 1993.

[16] C. Alcock, R. A. Allsman, D. Alves, R. Ansari, E. Aubourg, T. S. Axelrod, P. Bareyre, J.-P. Beaulieu, A. C. Becker, D. P. Bennett, S. Brehin, F. Cavalier, S. Char, K. H. Cook, R. Ferlet, J. Fernandez, K. C. Freeman, K. Griest, P. Grison, M. Gros, C. Gry, J. Guibert, M. Lachieze-Rey, B. Laurent, M. J. Lehner, E. Lesquoy, C. Magneville, S. L. Marshall, E. Maurice, A. Milsztajn, D. Minniti, M. Moniez, O. Moreau, L. Moscoso, N. Palanque-Delabrouille, B. A. Peterson, M. R. Pratt, L. Prevot, F. Queinnec, P. J. Quinn, C. Renault, J. Rich, M. Spiro, C. W. Stubbs, W. Sutherland, A. Tomaney, T. Vandehei, A. Vidal-Madjar, L. Vigroux, and S. Zylberajch, "EROS and MACHO Combined Limits on Planetary-Mass Dark Matter in the Galactic Halo," ApJ, vol. 499, pp. L9+, May 1998.

[17] J. Binney, N. Bissantz, and O. Gerhard, "Is Galactic Structure Compatible with Microlensing Data?," ApJ, vol. 537, pp. L99–L102, July 2000.

[18] C. Thurl, P. D. Sackett, and P. H. Hauschildt, "Examining stellar atmospheres via microlensing," *Astronomische Nachrichten*, vol. 325, pp. 247–247, 2004.

[19] A. Udalski, M. Kubiak, M. Szymanski, J. Kaluzny, M. Mateo, and W. Krzeminski, "The Optical Gravitational Lensing Experiment. The Catalog of Periodic Variable Stars in the Galactic Bulge. I. Periodic Variables in the Center of the Baade's Window," *Acta Astronomica*, vol. 44, pp. 317–386, Oct. 1994.

[20] M. Dominik, M. D. Albrow, J.-P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. M. Naber, J.-W. Pel, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams, "The PLANET microlensing follow-up network: results and prospects for the detection of extra-solar planets," Planet. Space Sci., vol. 50, pp. 299–307, Mar. 2002.

[21] S. Mao and B. Paczynski, "Gravitational microlensing by double stars and planetary systems," ApJ, vol. 374, pp. L37–L40, June 1991.

[22] B. S. Gaudi, R. M. Naber, and P. D. Sackett, "Microlensing by Multiple Planets in High-Magnification Events," ApJ, vol. 502, pp. L33+, July 1998.

[23] D. P. Bennett, S. H. Rhie, A. C. Becker, N. Butler, J. Dann, S. Kaspi, E. M. Leibowitz, Y. Lipkin, D. Maoz, H. Mendelson, B. A. Peterson, J. Quinn, O. Shemmer, S. Thomson, and S. E. Turner, "Discovery of a planet orbiting a binary star system from gravitational microlensing," Nature, vol. 402, pp. 57–59, Nov. 1999.

[24] M. D. Albrow, J.-P. Beaulieu, J. A. R. Caldwell, M. Dominik, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. M. Naber, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, A. Williams, H. E. Bond, and I. M. van Bemmel, "Detection of Rotation in a Binary Microlens: PLANET Photometry of MACHO 97-BLG-41," ApJ, vol. 534, pp. 894–906, May 2000.

[25] C. Afonso, C. Alard, J. N. Albert, J. Andersen, R. Ansari, É. Aubourg, P. Bareyre, F. Bauer, J. P. Beaulieu, A. Bouquet, S. Char, X. Charlot, F. Couchot, C. Coutures, F. Derue, R. Ferlet, J. F. Glicenstein, B. Goldman, A. Gould, D. Graff, M. Gros, J. Haissinski, J. C. Hamilton, D. Hardin, J. de Kat, A. Kim, T. Lasserre, É. Lesquoy, C. Loup, C. Magneville, J. B. Marquette, É. Maurice, A. Milsztajn, M. Moniez, N. Palanque-Delabrouille, O. Perdereau, L. Prévot, N. Regnault, J. Rich, M. Spiro, A. Vidal-Madjar, L. Vigroux, S. Zylberajch, C. Alcock, R. A. Allsman, D. Alves, T. S. Axelrod, A. C. Becker, K. H. Cook,

A. J. Drake, K. C. Freeman, K. Griest, L. J. King, M. J. Lehner, S. L. Marshall, D. Minniti, B. A. Peterson, M. R. Pratt, P. J. Quinn, A. W. Rodgers, P. B. Stetson, C. W. Stubbs, W. Sutherland, A. Tomaney, T. Vandehei, S. H. Rhie, D. P. Bennett, P. C. Fragile, B. R. Johnson, J. Quinn, A. Udalski, M. Kubiak, M. Szymański, G. Pietrzyński, P. Woźniak, K. Zebruń, M. D. Albrow, J. A. R. Caldwell, D. L. DePoy, M. Dominik, B. S. Gaudi, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. M. Naber, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams, "Combined Analysis of the Binary Lens Caustic-crossing Event MACHO 98-SMC-1," ApJ, vol. 532, pp. 340–352, Mar. 2000.

[26] M. D. Albrow, J. An, J.-P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, M. Dominik, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams, "PLANET Observations of Microlensing Event OGLE-1999-BUL-23: Limb-darkening Measurement of the Source Star," ApJ, vol. 549, pp. 759–769, Mar. 2001.

[27] N. W. Evans, "The first heroic decade of microlensing," 2003.

[28] P. D. Sackett, M. D. Albrow, J.-P. Beaulieu, J. A. R. Caldwell, C. Coutures, M. Dominik, J. Greenhill, K. Hill, K. Horne, U.-G. Jorgensen, S. Kane, D. Kubas, R. Martin, J. W. Menzies, K. R. Pollard, K. C. Sahu, J. Wambsganss, R. Watson, and A. Williams, "PLANET II: A Microlensing and Transit Search for Extrasolar Planets," in *IAU Symposium*, pp. 35–+, June 2004.

[29] N. J. Rattenbury, "Planetary microlensing: From prediction to discovery," *Mod. Phys. Lett.*, vol. A21, pp. 919–934, 2006.

[30] A. Gould, "The New Era of Precision Microlensing," in *ASP Conf. Ser. 289: The Proceedings of the IAU 8th Asian-Pacific Regional Meeting, Volume I*, pp. 453–560, May 2003.

[31] D. P. Bennett and S. H. Rhie, "Simulation of a Space-based Microlensing Survey for Terrestrial Extrasolar Planets," ApJ, vol. 574, pp. 985–1003, Aug. 2002.

[32] D. P. Bennett *et al.*, "The microlensing planet finder: Completing the census of extrasolar planets in the milky way," 2004.

[33] H. Asada, "Images for a binary gravitational lens from a single real algebraic equation," A&A, vol. 390, pp. L11–L14, July 2002.

[34] F. Abe *et al.*, "Search for low-mass exoplanets by gravitational microlensing at high magnification," *Science*, vol. 305, pp. 1264–1266, 2004.

[35] M. Jaroszyński and S. Mao, "Predicting the second caustic crossing in binary microlensing events," MNRAS, vol. 325, pp. 1546–1552, Aug. 2001.

[36] M. Dominik, "Theory and practice of microlensing light curves around fold singularities," MNRAS, vol. 353, pp. 69–86, Sept. 2004.

[37] H. Asada, "A Parametric Representation of Critical Curves and Caustics for a Binary Gravitational Lens," *Progress of Theoretical Physics*, vol. 110, pp. 425 - 432, Sept. 2003.

[38] H. Asada, T. Kasai, and M. Kasai, "Algebraic Properties of the Real Quintic Equation for a Binary Gravitational Lens," *Progress of Theoretical Physics*, vol. 108, pp. 1031–1037, Dec. 2002.

[39] J. Yoo, D. L. DePoy, A. Gal-Yam, B. S. Gaudi, A. Gould, C. Han, Y. Lipkin, D. Maoz, E. O. Ofek, B.-G. Park, R. W. Pogge, A. Udalski, I. Soszyński, Ł. Wyrzykowski, M. Kubiak, M. Szymański, G. Pietrzyński, O. Szewczyk, and K. Zebruń, "OGLE-2003-BLG-262: Finite-Source Effects from a Point-Mass Lens," ApJ, vol. 603, pp. 139–151, Mar. 2004.

[40] M. C. Smith, S. Mao, and B. Paczyński, "Acceleration and parallax effects in gravitational microlensing," MNRAS, vol. 339, pp. 925–936, Mar. 2003.

[41] B. S. Gaudi and A. O. Petters, "Gravitational microlensing near caustics ii: Cusps," *Astrophys. J.*, vol. 580, pp. 468–489, 2002.

[42] R. Kayser, S. Refsdal, and R. Stabell, "Astrophysical applications of gravitational micro-lensing," A&A, vol. 166, pp. 36–52, Sept. 1986.

[43] R. R. Bourassa and R. Kantowski, "The theory of transparent gravitational lenses," ApJ, vol. 195, pp. 13–21, Jan. 1975.

[44] H. J. Witt, "Investigation of high amplification events in light curves of gravitationally lensed quasars," A&A, vol. 236, pp. 311–322, Sept. 1990.

[45] C. Alcock, R. A. Allsman, D. Alves, T. S. Axelrod, D. Baines, A. C. Becker, D. P. Bennett, A. Bourke, A. Brakel, K. H. Cook, B. Crook, A. Crouch, J. Dan, A. J. Drake, P. C. Fragile, K. C. Freeman, A. Gal-Yam, M. Geha, J. Gray, K. Griest, A. Gurtierrez, A. Heller, J. Howard, B. R. Johnson, S. Kaspi, M. Keane, O. Kovo, C. Leach, T. Leach, E. M. Leibowitz, M. J. Lehner, Y. Lipkin, D. Maoz, S. L. Marshall, D. McDowell, S. McKeown, H. Mendelson, B. Messenger, D. Minniti, C. Nelson, B. A. Peterson, P. Popowski, E. Pozza, P. Purcell, M. R. Pratt, J. Quinn, P. J. Quinn, S. H. Rhie, A. W. Rodgers, A. Salmon, O. Shemmer, P. Stetson, C. W. Stubbs, W. Sutherland, S. Thomson, A. Tomaney, T. Vandehei, A. Walker, K. Ward, and G. Wyper, "Binary

Microlensing Events from the MACHO Project," ApJ, vol. 541, pp. 270 297, Sept. 2000.

[46] A. Cassan, J. P. Beaulieu, S. Brillant, C. Coutures, M. Dominik, J. Donatowicz, U. G. Jørgensen, D. Kubas, M. D. Albrow, J. A. R. Caldwell, P. Fouqué, J. Greenhill, K. Hill, K. Horne, S. Kane, R. Martin, J. Menzies, K. R. Pollard, K. C. Sahu, C. Vinter, J. Wambsganss, R. Watson, A. Williams, C. Fendt, P. Hauschildt, J. Heinmueller, J. B. Marquette, and C. Thurl, "Probing the atmosphere of the bulge G5III star OGLE-2002-BUL-069 by analysis of microlensed Hα line," A&A, vol. 419, pp. L1–L4, May 2004.

[47] V. Bozza, "Trajectories of the images in binary microlensing," A&A, vol. 374, pp. 13–27, July 2001.

[48] K. Chang and S. Refsdal, "Flux variations of QSO 0957+561 A, B and image splitting by stars near the light path," Nature, vol. 282, pp. 561–564, Dec. 1979.

[49] K. Griest and N. Safizadeh, "The Use of High-Magnification Microlensing Events in Discovering Extrasolar Planets," ApJ, vol. 500, pp. 37–+, June 1998.

[50] P. Schneider and A. Weiss, "The two-point-mass lens - Detailed investigation of a special asymmetric gravitational lens," A&A, vol. 164, pp. 237 259, Aug. 1986.

[51] H. Erdl and P. Schneider, "Classification of the multiple deflection two point-mass gravitational lens models and application of catastrophe theory in lensing," A&A, vol. 268, pp. 453–471, Feb. 1993.

[52] A. Gould and A. Loeb, "Discovering planetary systems through gravitational microlenses," ApJ, vol. 396, pp. 104–114, Sept. 1992.

[53] D. P. Bennett and S. H. Rhie, "Detecting Earth-Mass Planets with Gravitational Microlensing," ApJ, vol. 472, pp. 660–+, Nov. 1996.

[54] J.-P. Beaulieu, D. P. Bennett, P. Fouqué, A. Williams, M. Dominik, U. G. Jorgensen, D. Kubas, A. Cassan, C. Coutures, J. Greenhill, K. Hill, J. Menzies, P. D. Sackett, M. Albrow, S. Brillant, 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," Nature, vol. 439, pp. 437–440, Jan. 2006.

[55] S. J. Peale, "Probability of Detecting a Planetary Companion during a Microlensing Event," ApJ, vol. 552, pp. 889–911, May 2001.

[56] P. Schneider and A. Weiss, "A gravitational lens origin for AGN-variability? Consequences of micro-lensing," A&A, vol. 171, pp. 49–65, Jan. 1987.

[57] J. Wambsganss, H. J. Witt, and P. Schneider, "Gravitational microlensing - Powerful combination of ray-shooting and parametric representation of caustics," A&A, vol. 258, pp. 591–599, May 1992.

[58] H. J. Witt, *Phd Thesis*. PhD thesis, Hamburg, 1991.

[59] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2nd ed., 1992.

[60] C. S. Kochanek, "Selection effects in optical surveys for gravitational lenses," ApJ, vol. 379, pp. 517–531, Oct. 1991.

[61] N. Nethercote and J. Seward, "Valgrind: A program supervision framework.," *Electr. Notes Theor. Comput. Sci.*, vol. 89, no. 2, 2003.

[62] P. Vermaak, "The effects of resolved sources and blending on the detection of planets via gravitational microlensing," MNRAS, vol. 319, pp. 1011–1019, Dec. 2000.

[63] M. Dominik, "A robust and efficient method for calculating the magnification of extended sources caused by gravitational lenses," A&A, vol. 333, pp. L79–L82, May 1998.

[64] C. Han, "The Effect of Luminous Lens Blending in Gravitational Microlensing Experiments," ApJ, vol. 500, pp. 569–+, June 1998.

[65] R. Di Stefano, "On the Nature and Location of the Microlenses," ApJ, vol. 541, pp. 587–596, Oct. 2000.

[66] P. Wozniak and B. Paczynski, "Microlensing of Blended Stellar Images," ApJ, vol. 487, pp. 55–+, Sept. 1997.

[67] A. Gould, J. Miralda-Escude, and J. N. Bahcall, "Microlensing Events: Thin Disk, Thick Disk, or Halo?," ApJ, vol. 423, pp. L105+, Mar. 1994.

[68] A. Gould, "Resolution of the macho-lmc-5 puzzle: The jerk-parallax microlens degeneracy," *The Astrophysical Journal*, vol. 606, p. 319, 2004.

[69] M. Dominik, "Galactic microlensing with rotating binaries," A&A, vol. 329, pp. 361–374, Jan. 1998.

[70] S. Dong, A. Udalski, A. Gould, W. T. Reach, G. W. Christie, A. F. Boden, D. P. Bennett, G. Fazio, K. Griest, M. K. Szymanski, M. Kubiak, I. Soszynski, G. Pietrzynski, O. Szewczyk, L. Wyrzykowski, K. Ulaczyk, T. Wieckowski, B. Paczynski, D. L. DePoy, R. W. Pogge, G. W. Preston, I. B. Thompson, and B. M. Patten, "First space-based microlens parallax measurement: Spitzer observations of ogle-2005-smc-001," *The Astrophysical Journal*, vol. 664, p. 862, 2007.

[71] M. Dominik, "The binary gravitational lens and its extreme cases," A&A, vol. 349, pp. 108–125, Sept. 1999.

[72] M. D. Albrow, J. An, J.-P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, M. Dominik, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams, "A Short, Nonplanetary, Microlensing Anomaly: Observations and Light-Curve Analysis of MACHO 99-BLG-47," ApJ, vol. 572, pp. 1031–1040, June 2002.

[73] M. D. Albrow, J.-P. Beaulieu, J. A. R. Caldwell, M. Dominik, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. M. Naber, J.-W. Pel, K. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, A. Williams, and M. S. Sahu, "Limb Darkening of a K Giant in the Galactic Bulge: PLANET Photometry of MACHO 97-BLG-28," ApJ, vol. 522, pp. 1011–1021, Sept. 1999.

[74] S. Mao and R. Di Stefano, "Interpretation of gravitational microlensing by binary systems," ApJ, vol. 440, pp. 22–27, Feb. 1995.

[75] C. M. Bishop, *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press, 1996.

[76] I. Witten and E. Frank, *Data Mining*. Morgan Kaufmann, 2000. WIT i 02:1 1.Ex.

[77] P. Vermaak, "Rapid analysis of binary lens gravitational microlensing light curves," *Monthly Notices of the Royal Astronomical Society*, vol. 344, no. 2, pp. 651–656, 2003.

[78] J. Scheffer, "Dealing with missing data."

[79] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441 and 498–520, 1933.

[80] H. Vafaie and K. D. Jong, "Genetic algorithms as a tool for feature selection in machine learning," 1992.

[81] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press, 1992.

[82] Kubas, *Phd Thesis*. PhD thesis, Universität Potsdam, 2005.

[83] G. Paperin. "www.jaga.org," 2004.

[84] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning software library," 2002. IDIAP Research Report 02-46, Martigny, Switzerland. (see also www.torch.ch).

[85] I. VA Linux Systems, "Joone - java object oriented neural engine," 2001.

[86] E. W. Weisstein, "B-spline from mathworld," 2006. A Wolfram Web Resource.

[87] P. Dierckx, *Curve and surface fitting with splines*. New York, NY, USA: Oxford University Press, Inc., 1993.

[88] M. Jaroszynski, A. Udalski, M. Kubiak, M. K. Szymanski, G. Pietrzynski, I. Soszynski, K. Zebrun, O. Szewczyk, and L. Wyrzykowski, "Mass Estimates for Some of the Binary Lenses in OGLE-III Database," *Acta Astronomica*, vol. 55, pp. 159–175, June 2005.

[89] A. Udalski, "The Optical Gravitational Lensing Experiment. Real Time Data Analysis Systems in the OGLE-III Survey," *Acta Astronomica*, vol. 53, pp. 291–305, Dec. 2003.

[90] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++." [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004. [Accessed on 31 Jan. 2005.].

[91] S. H. Rhie *et al.*, "Observations of the binary microlens event macho-98-smc-1 by the microlensing planet search collaboration," 1998.

[92] A. Udalski, M. Szymanski, J. Kaluzny, M. Kubiak, M. Mateo, W. Krzeminski, and B. Paczynski, "The Optical Gravitational Lensing Experiment. The Early Warning System: Real Time Microlensing," *Acta Astronomica*, vol. 44, pp. 227 234, July 1994.