Gravitational lens modeling with iterative source deconvolution and global optimization of lens density parameters

by

Adam Rogers

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of
DOCTOR OF PHILOSOPHY

Department of Physics and Astronomy
University of Manitoba
Winnipeg

Abstract

Strong gravitational lensing produces multiple distorted images of a background source when it is closely aligned with a mass distribution along the line of sight. The lensed images provide constraints on the parameters of a model of the lens, and the images themselves can be inverted providing a model of the source. Both of these aspects of lensing are extremely valuable, as lensing depends on the total matter distribution, both luminous and dark. Furthermore, lensed sources are commonly located at cosmological distances and are magnified by the lensing effect. This provides a chance to image sources that would be unobservable when viewed with conventional optics.

The semilinear method expresses the source modeling step as a least-squares problem for a given set of lens model parameters. The blurring effect due to the point spread function of the instrument used to observe the lensed images is also taken into account. In general, regularization is needed to solve the source deconvolution problem. We use Krylov subspace methods to solve for the pixelated sources. These optimization techniques, such as the Conjugate Gradient method, provide natural regularizing effects from simple truncated iteration. Using these routines, we are able to avoid the explicit construction of the lens and blurring matrices and solve the least squares source optimization problem iteratively. We explore several regularization parameter selection methods commonly used in standard image deconvolution problems, which lead to previously derived expressions for the number of source degrees of freedom.

The parameters that describe the lens density distribution are found by global optimization methods including genetic algorithms and particle swarm optimizers. In general, global optimizers are useful in non-linear optimization problems such as lens modeling due to their parameter space mapping capabilities. However, these optimization methods require many function evaluations and iterative approaches to the least squares problem are beneficial due to the speed advantage that they offer. We apply our modeling techniques to a subset of gravitational lens systems from the Sloan Lens ACS (SLACS) survey, and are able to reliably recover the parameters of the lens mass distribution with both analytical and regularized pixelated sources.

Acknowledgements

I would like to thank my advisor and friend Jason D. Fiege for overseeing my thesis work, and providing the software that made this project possible. Jason gave me the freedom to choose the problem I was most interested in at the outset of this work and for that I am most grateful. The work proved to be more interesting and challenging than either of us initially expected.

Thanks to Peter Loly for providing encouragement during my undergraduate years. Peter has been my mentor and gave me the support I needed during a personally difficult and challenging period in my life. If it were not for Peter, it is likely I would never have continued on to this point.

I am extremely grateful to Kaushala Bandara for providing the data I used to test my lens modeling code. This data allowed me the opportunity to model real gravitational lens systems, and a large part of this work is dependent on Kaushi's contribution.

Deepest thanks to my Mother and Father for encouraging me in every possible way and driving my curiosity in astronomy since I was a young boy. I also thank my family for consistently standing by me. Cheers to all of my friends who have helped me through the writing of this thesis.

Finally, thanks to my wife Mel. Wthout her, none of this would have been possible.

Contents

List of Figures

List of Tables

List of Acronyms

Advanced Camera for Surveys	ACS
Conjugate Gradient method for Least Squares problems	CGLS
Digital Source Reconstruction	DiSoR
Fast Fourier Transform	FFT
General Relativity	GR
Generalized Cross Validation	GCV
Genetic Algorithm	GA
Hubble Space Telescope	HST
Least Squares algorithm	LSQR
Maximum Entropy Method	MEM
Minimum Residual Norm Steepest Descent	MRNSD
Particle Swarm Optimizer	PSO
Point Spread Function	PSF
Pseudo Singular Isothermal Potential	PSIEP
Singular Isothermal Ellipse	SIE
Singular Isothermal Sphere	SIS
Singular Value Decomposition	SVD
Sloan Lens ACS Survey	SLACS
Steepest Descent	SD
Truncated Singular Value Decomposition	TSVD
Very Large Array	VLA

For Mel, Mom and Dad

"Faithless is he who says farewell when the road darkens."
J. R. R. Tolkien

Preface

The original work of this thesis is contained in Chapters 1, 3, 4, which are based on published refereed articles. A submitted manuscript forms the basis of Chapter 5. Sections 1.1 to 1.5 contain text based on an editor-refereed article published in the Journal of the Royal Astronomical Society of Canada (Friesen et al. 2011). The material in this article is the result of collaboration between myself and Jordan Friesen, for whom I acted as direct supervisor and Dr. Jason D. Fiege as secondary supervisor (working at arm's length) over the period of May 1, 2011 to September 1, 2011. The original article was developed from an end of term project that began in an undergraduate class instructed by Dr. Fiege. The project was planned by me, numerical calculations and code were developed by Friesen. Figures 1.2, 1.3, 1.4 and 1.5 were also generated by Friesen. I wrote the text of the original paper in its entirety, and editing was handled collaboratively between Dr. Fiege and myself.

The text in Sections 3.1 to 3.4 is based on an article published in The Astrophysical Journal (Rogers & Fiege 2011a). The original Appendix to this paper is reprinted in Section D.8. Sections 4.1 to 4.4 are closely based on material published in The Astrophysical Journal (Rogers & Fiege 2011b), with minor modifications to refer to previous sections and equations contained in this thesis. Appendix E contains the published Appendix to this paper. Chapter 5 is based on an article that has been submitted to The Astrophysical Journal. The text and figures in these Chapters are reproduced with permission from the American Astronomical Society.

The original ideas developed in Chapters 3, 4 and 5 are my own. I performed all of the numerical and analytical calculations, and all of the algorithms and codes were written by me except for the Qubist Global Optimization Toolbox, which was developed by Dr. Fiege. I wrote each of the original articles in their entirety, except for Section D.8, which was written by Dr. Fiege, who also contributed to editing. The preparation of the articles for submission and peer review process was handled collaboratively. All other sections of this thesis were written by myself. Dr. Fiege contributed to editing.

Chapter 1

Introduction

In this thesis, a number of tools are developed to study and numerically model strong gravitational lens systems. Chapter 1 describes the simplest lens model, the point lens, and applies this model to simulate the deflection of starlight during a solar eclipse. Several eclipses, both past and future, are simulated. A short history of the phenomenon from the first eclipse observation to the modern day is given. The theory necessary to describe more complicated lens models is developed in Chapter 2.

Chapter 3 contains the main contribution to the thesis. We extend the range of applicability of a versatile lens modeling technique, and test the resulting code on a number of simulated data sets. This work is novel in that the linear system describing the lensed source is solved for each lens model without an explicit construction of the matrices involved in the problem, while maintaining the linear least squares formulation. This allows us to use linear optimization methods on larger gravitational lens source models with more complicated blurring effects than previously possible in a linear context. The parameters of analytical lens models are simultaneously found by means of global optimization procedures. This work represents a signficant contribution and unique approach to the modeling of strong gravitational lens systems.

In Chapter 4, we consider the inclusion of spatially variant blurring effects. This functionality has never before been included in the source deconvolution step. This modification is possible due to our matrix-free method developed in the previous Chapter, and should be useful in modeling large lensed images. Thus, our work provides a useful advance toward including more realistic and complicated blurring effects than previously considered in the gravitational lens literature. The code is extensively tested in this chapter, including a test on a simulated large-scale system.

We apply our matrix-free method to real data in Chapter 5. We include a number of modifications, such as an initial step that avoids unphysical, trivial solutions automatically. This is a powerful feature that allows our global optimization methods to determine the optimal region of parameter space more easily than previously possible. Furthermore, the code is shown to perform well in modeling real data using analytical lens models and pixelated sources. Our code reliably recovers the optimal parameters of previously studied lens systems, proving the viability of our method on real data.

The Appendices provide the background for several topics crucial to the arguments in the thesis, including the weak field limit of general relativity, the details of the background cosmological model used in our code, the operational details of two common linear iterative optimization schemes, and the details of the global optimization routines used in the body of the text. I have tried to keep these derivations as transparent as possible.

1.1 The Development of Gravitational Lensing

The first mathematical description of gravity was provided by Sir Isaac Newton in 1687. Newton's law of universal gravitation describes the force of gravity F acting between two objects having mass M and m, separated by a distance between their centers r. This force is then written

$$F = \frac{GMm}{r^2}, \tag{1.1}$$

where G is a universal constant. By using this simple equation, Newton was able to predict the orbits of the planets and the dynamical behavior of the solar system with great success. However, Newton had many interests, including optics. He believed that light was made up of infinitesimal material bodies, each of which was presumed to have a tiny mass. It seemed natural for Newton to be the first to ask if it is possible for the gravitational force to act on light. If this were the case, it would appear to a distant observer that a light ray would bend around a massive object.

The first quantitative description of light deflection due to gravitation was carried out by the German physicist Johann von Soldner in 1804 (Soldner 1804). Soldner used Newton's law of universal gravitation, Equation 1.1, and found the amount by which light should be deflected if the path of the ray just grazed the surface of the Sun. The deflection angle at the edge of the Sun, denoted as α, was calculated to be $\alpha = 0.88$ arcseconds. This calculation would remain a mere curiosity until a century later.

In 1905 Albert Einstein described the theory of special relativity (Einstein 1905). This theory forever changed our view of the universe, describing space and time unified in one four-dimensional entity: spacetime. Using this framework, Einstein was able to show that the laws of electromagnetism require that the speed of light is constant for all observers. This has some shocking consequences: time and distance are no longer constant for all observers. Rather, it is the state of relative motion of two observers that determines the passage of time and distances measured between them. Since the speed of light is so central to his theory and to the mechanics of spacetime, Einstein calculated the effect that the Sun would have on a light ray. In this first calculation, Einstein used an as-yet incomplete description of spacetime to model the effect. Though Soldner's work was unknown to him at this time, Einstein's initial calculation agreed with Soldner's result.

Following his inital successes, Einstein began to work on a more complete description of spacetime, known as general relativity (GR; Einstein (1915)). This theory was Einstein's masterwork and was published in 1915. Once again Einstein demolished commonly accepted notions of the universe. In GR, gravity is a consequence of spacetime "curving" around a massive object.

GR is a remarkable theory because it includes all of the results of special relativity (found when spacetime is not curved at all) and Newton's law of gravitation (found when spacetime is only gently curved, like on the surface of the Earth and far from the Sun). This means that all of the successes of both Einstein and Newton's former works are contained in GR. The theory also makes some predictions that are far from our everyday experience, describing the physics of black holes, the expanding universe and the big bang.

Once his description of spacetime was complete, Einstein returned to the question of light deflection due to gravity. This time, Einstein used the full theory of GR, taking into account the spacetime curvature around the Sun, to calculate the deflection angle, α. Surprisingly, he found that the solution was twice Soldner's previously calculated result. This prediction became one of the first tests of GR, and provided a crucial piece of evidence in favour of Einstein's theory of gravitation.

The next sections derive details of the simplest lens model, the point mass lens, and develop the basic properties of the lens effect. We then review the details of the original 1919 eclipse expedition used to verify GR, and use the point mass lens to simulate the deflection of star light in the upcoming eclipses of 2012 and 2019. We conclude this introduction by discussing modern observations of gravitational lens phenomena over a variety of scales.

1.2 Testing General Relativity

Gravitational lens effects are described by a transformation between coordinate systems that specify the positions of the source and lensed images. The link between these coordinate systems is the deflection angle field produced by the lens mass. Our discussion of lensing theory follows the approach and notation in Schneider, Ehlers & Falco (1992), Narayan & Bartelmann (1995) and Petters et al. (2001).

Consider a coordinate system centered on the lens, such that the z direction defines the optic axis and $\boldsymbol{\xi} = (\xi_x, \xi_y)$ is a position vector in the plane orthogonal to the z direction. We refer to the $\boldsymbol{\xi}$ plane as the lens plane, and consider background objects on the source plane (Refsdal 1964). We denote the distance to the lens plane along the optic axis as D_d and the distance to the source plane D_s. Since these distances are large, most of the deflection occurs near the lens itself. Due to the small size of the lens with respect to the light path, we use a thin lens approximation to describe the light deflection. This approximation treats the mass of the lens as a two dimensional distribution on the lens plane. The deflection angle produced by a thin lens is a vector field $\hat{\boldsymbol{\alpha}}$, given as a function of the lens potential in Equation A.66:

$$\hat{\boldsymbol{\alpha}} = \frac{2}{c^2} \int \boldsymbol{\nabla}_{\perp} \phi \, dz. \tag{1.2}$$

The gradient in this equation is the two dimensional gradient in the lens plane with respect to $\boldsymbol{\xi}$ (Schneider 1985):

$$\boldsymbol{\nabla}_{\perp} = \hat{\boldsymbol{\xi}}_x \frac{\partial}{\partial \xi_x} + \hat{\boldsymbol{\xi}}_y \frac{\partial}{\partial \xi_y}, \tag{1.3}$$

where we have used $\hat{\boldsymbol{\xi}}_x$ and $\hat{\boldsymbol{\xi}}_y$ as unit vectors in the x and y directions on the lens plane, respectively. The Newtonian potential of a point lens with mass M is written in terms of the lens plane coordinates:

$$\phi(\boldsymbol{\xi}, z) = -\frac{GM}{(\xi_x^2 + \xi_y^2 + z^2)^{\frac{1}{2}}}. \tag{1.4}$$

Equation 1.2 gives the deflection angle of the point lens

$$\hat{\boldsymbol{\alpha}} = \frac{4GM}{c^2 \xi} \hat{\boldsymbol{\xi}}, \tag{1.5}$$

where $\hat{\boldsymbol{\xi}}$ is a unit vector in the direction of the lens. Since we observe positions on the plane of the sky, it is simpler to refer to the lens mass and the positions of lensed images in terms of the angular Cartesian coordinates $\boldsymbol{\theta} = (\theta_x, \theta_y)$, such that $\boldsymbol{\xi} = D_d \boldsymbol{\theta}$. Angular positions on the source plane are denoted by $\boldsymbol{\beta} = (\beta_x, \beta_y)$ (Narayan & Bartelmann 1995).

Consider the geometrical relationship between the image and source positions in Figure 1.1. It is apparent from the figure that

$$D_s \boldsymbol{\beta} = D_s \boldsymbol{\theta} - D_{ds} \hat{\boldsymbol{\alpha}}, \tag{1.6}$$

where we have written the distance between the lens and source plane D_{ds}. Equation 1.6 can be simplified by dividing through by the source distance D_s, such that we define the reduced deflection angle

$$\boldsymbol{\alpha} = \frac{D_{ds}}{D_s} \hat{\boldsymbol{\alpha}}. \tag{1.7}$$

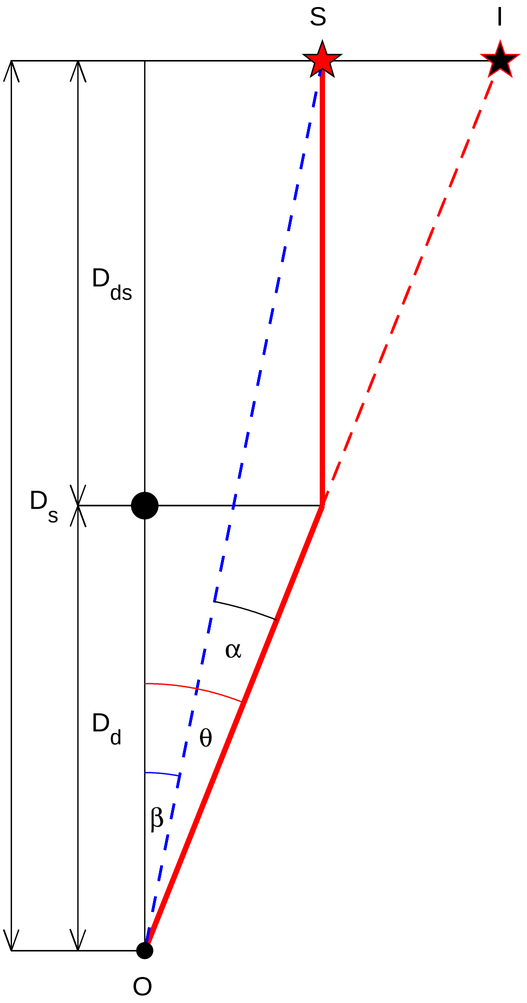

Figure 1.1: The Geometry of the Lens Mapping

The geometry of the lens mapping. An observer at O receives light rays (bold red line) emitted from an object S as the rays pass by a lensing mass. The observer sees an image of S at the image point, labelled I. The angular diameter distances between source, deflecting mass and the difference between them are D_s, D_d and D_{ds} respectively.

The thin lens equation can be written

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \boldsymbol{\alpha}\left(\boldsymbol{\theta}\right). \tag{1.8}$$

The lens equation is nonlinear since the deflection angle depends on the image coordinates (Schneider, Ehlers & Falco (1992); Petters et al. (2001); Narayan & Bartelmann (1995)). Due to this nonlinearity, many image locations $\boldsymbol{\theta}$ may satisfy Equation 1.8 for a single source position $\boldsymbol{\beta}$. This is the explanation for multiple imaging in gravitational lens systems. Note that when describing the deflection angle of the Sun the distance to the deflector is much smaller than the distance to the background stars, such that $D_d \ll D_s$ and $\boldsymbol{\alpha} \approx \hat{\boldsymbol{\alpha}}$ by Equation 1.7.

Noting the symmetry of the lens, we consider the scalar deflection angle due to a point source using angular Cartesian coordinates, such that Equation 1.5 becomes

$$\alpha(\theta) = \frac{4GM}{c^2 D_d \theta}. \tag{1.9}$$

Given the mass of the Sun M and the solar radius $r = D_d\theta$, we find that the deflection angle at the limb of the Sun is $\alpha = 1.75''$. When the deflection angle is inserted in the thin lens equation, we find a quadratic relationship in terms of the image positions

$$\theta^2 - \beta\theta - \frac{4GM}{c^2 D_d} = 0, \tag{1.10}$$

which can then be solved by application of the quadratic formula:

$$\theta_{\pm} = \frac{1}{2}\left(\beta \pm \sqrt{\beta^2 + \frac{16GM}{c^2 D_d}}\right). \tag{1.11}$$

This expression gives the resulting deflected position of the star as a function of undeflected position β. Note that the equation has two solutions, θ_+ and θ_-. In the case of lensing by the Sun, one solution (θ_-) is near the center of the Sun's disk and cannot be observed. Therefore, during an eclipse the position of a background star will be deflected to a new position θ_+. Though the magnitude of the deflection is small, Einstein knew that if the lensing effect was observed by the amount he calculated it could be used to validate his theory of GR along with all of its exotic predictions. However, the measurements are difficult in practice for a variety of reasons.

In order to observe gravitational lensing of light by the mass of the Sun, it is necessary to accurately measure the apparent positions of the background stars. This can be done by observing the sky during a solar eclipse. With the moon occulting the Sun's disk at totality, enough sunlight is blocked so that stars are visible and the stars nearest the Sun are deflected the most severely. The separation between the stars can then be measured in the sky when the Sun is absent, and the difference is due to the gravitational lens effect. From Equation 1.5 we can see that the deflection angle decreases as $1/\theta$ as we move away from the Sun. The maximum deflection possible is therefore $1.75''$, but smaller deflections are observed in practice since observations require bright stars be present near the limb of the Sun during totality of the eclipse. Furthermore, the observations have to be made in the path of totality, so an expedition would usually have to be mounted and astronomical equipment moved into place at the site.

The first observer to undertake the challenge of testing GR was the British astronomer Arthur Eddington. Eddington, along with the Astronomer Royal, Frank Dyson, planned an expedition to sites in Africa (Principe) and Brazil (Sobral), to observe the 1919 eclipse (Dyson, Eddington & Davidson 1919). The expedition was beset by difficulties. While there, one of the telescopes malfunctioned due to the heat, and produced only blurry images (Almassi 2009). Furthermore, the Principe site was shrouded in cloud cover for hours before totality. The clouds only parted a half hour before the critical moment, allowing for the team to scramble to produce reliable observations (Kennefick 2007).

In order to illustrate the difficulty of these observations, we have produced graphics using the MATLAB programming language (http://www.mathworks.com) that simulates the positions of the stars Eddington observed during the eclipse. We found the dates and locations of totality of solar eclipses using the NASA Eclipse website (http://eclipse.gsfc.nasa.gov). We then find the undeflected positions β of the background stars near the Sun at this time and location on Earth using planetarium software (Starry Night Pro version 6.0). Using these positions in Equation 1.11, we calculate the resulting positions θ_+ as they would appear under the effect of gravitational lensing by the Sun.

Though the value 1.75 arcseconds is often used casually, it is difficult to comprehend how tiny an angle this truly is in practice. To demonstrate this, consider Figure 1.2. The large upper panel shows the stars' undeviated position and observed deflected position under the gravitational lens effect, which appears at first glance non-existant! However, a closer inspection of the stars (lower panels) show that there is actually a deflection of each of these stars though their deviations are extremely small. Nevertheless Eddington's 1919 expedition managed to find a solar deflection angle of $\alpha = 1.61'' \pm 0.30$ at Principe and $\alpha = 1.98'' \pm 0.12$ at Sobral, versus the GR value of $1.75''$. These observations favor a large deflection angle as found from GR rather than half this value, which would have suggested the Newtonian value Soldner arrived at. Eddington's observations validated GR and forever burned Einstein's name into the public consciousness.

There remains a significant amount of controversy over Eddington's results. The plates that were discarded from the initial analysis were obtained with the 13" Astrographic telescope of the Royal Observatory at Sobral. These plates were said to show a deflection that was much smaller than Einstein predicted, $\alpha = 0.93''$. This deflection angle is more in line with the Newtonian prediction than the prediction of GR. If these plates had been included in the original analysis the measurements would have significantly reduced Eddington's observed value of the deflection angle. The problem with these plates is that the telescope focus seems to have changed significantly between reference image and eclipse observation due to the heat on site and was therefore discounted. Since Eddington's observations were a strong piece of evidence in support of Einstein's theory this would have been a serious setback for GR. However, re-analysis of the surviving Sobral plates in 1979 (Harvey 1979) has shown that when properly referenced using modern means, the Sobral plates give a result consistent with the other observations, yielding a revised deflection angle of $\alpha = 1.52'' \pm 0.34$.

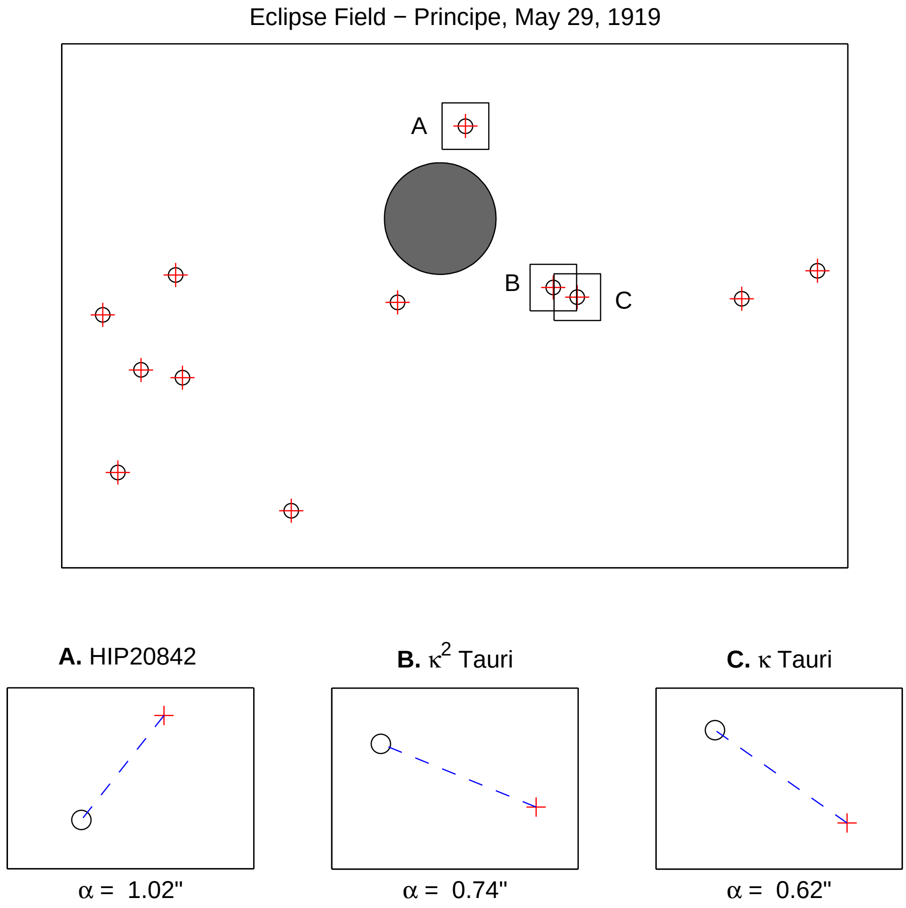

Figure 1.2: The 1919 Eclipse - Principe, Africa

Top: Configuration of stars during the 1919 eclipse as seen from Principe, Africa. Circles represent the true positions of background stars, and image positions are represented by a cross. Bottom: The lower set of figures represent close-up views of the stars indicated in the top panel.

1.3 Historical Development

Since Eddington's historical expedition, the observation of the gravitational lens effect has been repeated a number of times. Efforts were made to observe subsequent eclipses in 1922, 1929, two expeditions in 1936, 1947, 1952 and 1973 (Will 1993). Subsequent efforts to measure the effect provided similar results with little improvement. The measured values of the deflection angle lie typically between 0.75 to 1.5

times Einstein's predicted value. Even the 1973 expedition achieved only a modest accuracy of 0.95 ± 0.11 times the predicted value.

Extremely high accuracy is required to experimentally validate GR by the deflection of starlight during an eclipse using visible light, and the measurement is difficult to make even with modern instrumentation. In fact, the 1973 eclipse is the last time the measurement was attempted optically. Astronomers had found a better way to perform these kinds of observations using the fact that GR predicts gravitational lensing affects all wavelengths of light equally.

A much more accurate measurement of the lensing effect can be made using radio telescopes and quasars rather than observations of stars in the visible spectrum. Quasars are cosmologically distant active galaxies. These galaxies are highly luminous ($\approx$ 100 times that of the total light of average galaxies like the Milky Way), which allows them to be seen from cosmological distances. About one in ten quasars are also radio loud and can be observed in daylight at radio wavelengths; thus a solar eclipse is not required. These radio loud quasars provide excellent targets for verifying the lensing effect provided that several of them can be observed simultaneously near the limb of the Sun. In fact, this is exactly what was found with the quasar pair $3C273$ and $3C279$. Every October, the Sun passes this pair of quasars in the sky, and the separation between them can be readily measured using radio interferometry (Will 1993). By making a series of measurements the deflection angle as a function of distance from the Sun can be determined, and the results have shown that a light ray passing near the limb of the Sun behaves as GR predicts. These observations were carried out annually between 1969 and 1975, and the uncertainty in the measurements was gradually reduced from the 20% error found in optical light deflection measurements to 0.01% using lensed groups of quasars (Fomalont and Sramek 1975). Over time the technique was improved to μarcsecond - a millionth of an arcsecond - accuracy (Robertson, Carter and Dillinger 1991). In fact, due to these high-precision methods, the gravitational lensing of background radio sources by the planet Jupiter was first carried out in 1991. Though the observations only had a precision of 50%, this result is impressive because the lensing effect at the limb of Jupiter comprises an angle of only 17 milliarcseconds, a hundred times smaller than that of the Sun

(Treuhaft & Lowe 1991).

1.4 Future Eclipses

Using MATLAB and planetarium software, we investigated the first attempt to measure the gravitational lens effect by Eddington during the 1919 eclipse, shown in Figure 1.2. It is fortunate that the first such measurement was carried out in 1919, as the largest stellar deflection seen was $\alpha = 1.02''$, whereas successive attempts would not be as ideal. Our approach was also used to simulate the lensing of starlight during a number of other eclipses, including the eclipse of 1979 which saw totality over Winnipeg, Manitoba (see Figure 1.3), and the upcoming eclipse of 2012, which will be visible from Cairns, Australia (Figure 1.4). Note that the maximum deflection angles for these eclipses are $\alpha = 0.60''$ and $\alpha = 0.72''$ respectively, significantly less than the fortuitous 1919 eclipse. We have also calculated maximum deflection angles for all solar eclipses for the next 10 years, and found that a large stellar deflection should be seen during the eclipse of July 2, 2019 which will occur over the Pacific ocean. Our calculation shows that the nearest star to the Sun during this eclipse has magnitude 6.5, and should show a displacement of $\alpha = 1.30''$, a larger deflection than was seen during the 1919 eclipse. Since this occurs 100 years after Eddington's initial expedition, there may be some interest in performing a similar observation. For optical deflection measurements, the July 2019 eclipse will provide an ideal situation, provided that this star is not obscured by the Sun's corona. This simulation is shown in Figure 1.5. Our eclipse results are summarized in Table 1.1.

Table 1 - Undeflected Stellar Information					
Eclipse	Star	R. A.	Dec.	Mag.	α (”)
1919	HIP20842	4h 28.004m	21° 37.254’	5.71	1.02 ± 0.05
	κ^2 Tauri	4h 25.406m	22° 12.061’	5.25	0.74 ± 0.04
	κ Tauri	4h 25.359m	22° 17.092’	4.18	0.62 ± 0.03
1979	HIP111761	22h 38.367m	−7° 53.854’	6.21	0.60 ± 0.03
	HIP111577	22h 36.283m	−7° 39.865’	7.00	0.45 ± 0.02
	HIP111414	22h 34.344m	−9° 18.310’	8.46	0.39 ± 0.02
2012	HIP74728	15h 16.328m	−18° 37.749’	8.15	0.72 ± 0.04
	TYC6174	15h 19.450m	−18° 17.308’	7.59	0.34 ± 0.02
	HIP74593	15h 14.467m	−18° 25.717’	6.75	0.26 ± 0.01
2019	HIP32431	6h 46.173m	23° 22.288’	6.50	1.30 ± 0.07
	HIP32367	6h 45.389m	23° 38.774’	7.15	0.70 ± 0.04
	HIP32285	6h 44.411m	22° 34.744’	8.03	0.62 ± 0.03

Table 1.1: Summary of Eclipse Modeling Results

Stellar positions are found directly from Starry Night Pro+ v6.0, which has a stated accuracy of $0.5''$. Using this value and the angular size of the solar radius we have calculated errors for the deflection angles obtained from equation 1.11. Deflection angles are calculated at the location and time of greatest eclipse as stated on the NASA Eclipse website.

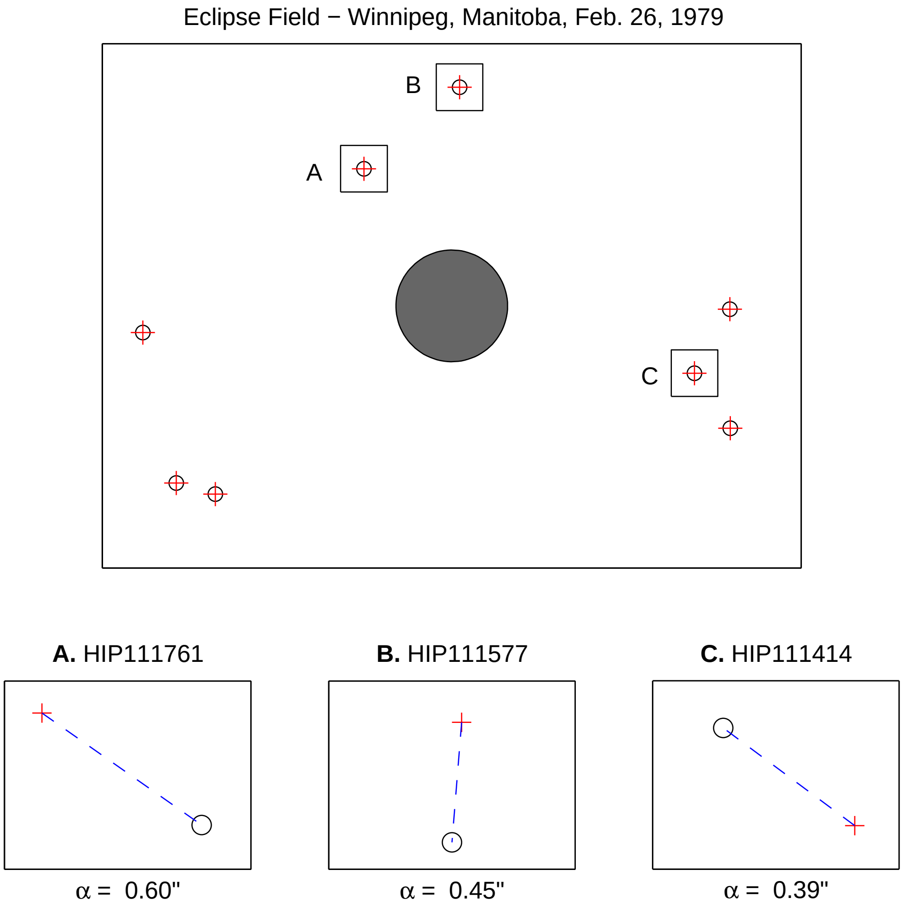

Figure 1.3: The 1979 Eclipse - Winnipeg, Canada

Configuration of stars during the 1979 eclipse as seen from Winnipeg, Manitoba, Canada. Note the small size of the deflection angles compared to the 1919 eclipse.

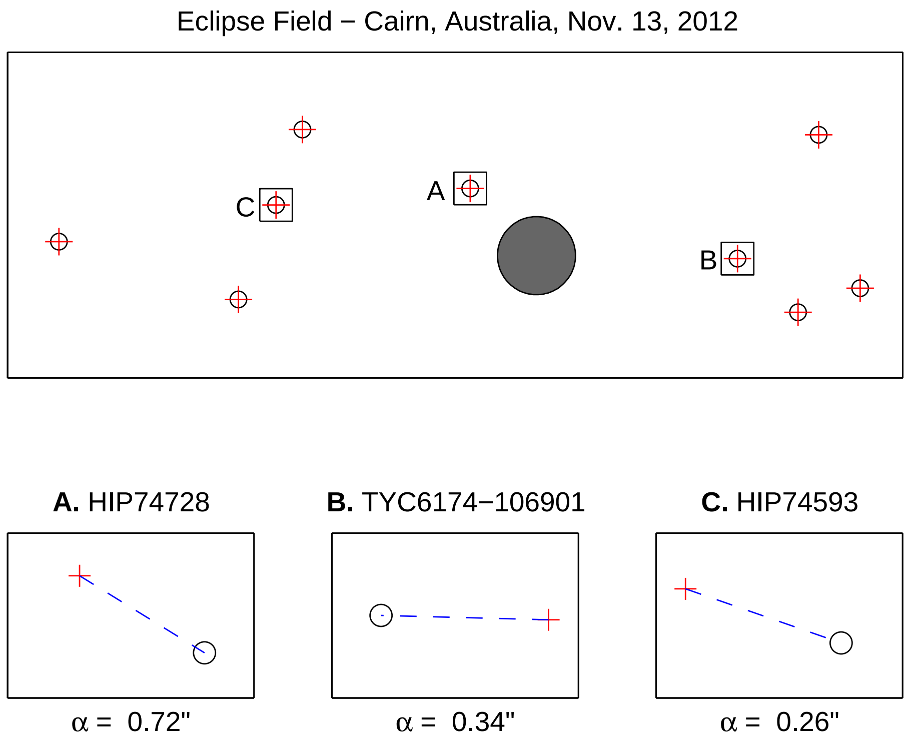

Figure 1.4: The 2012 Eclipse - Cairns, Australia

Configuration of stars during the 2012 eclipse as seen from Cairns, Australia.

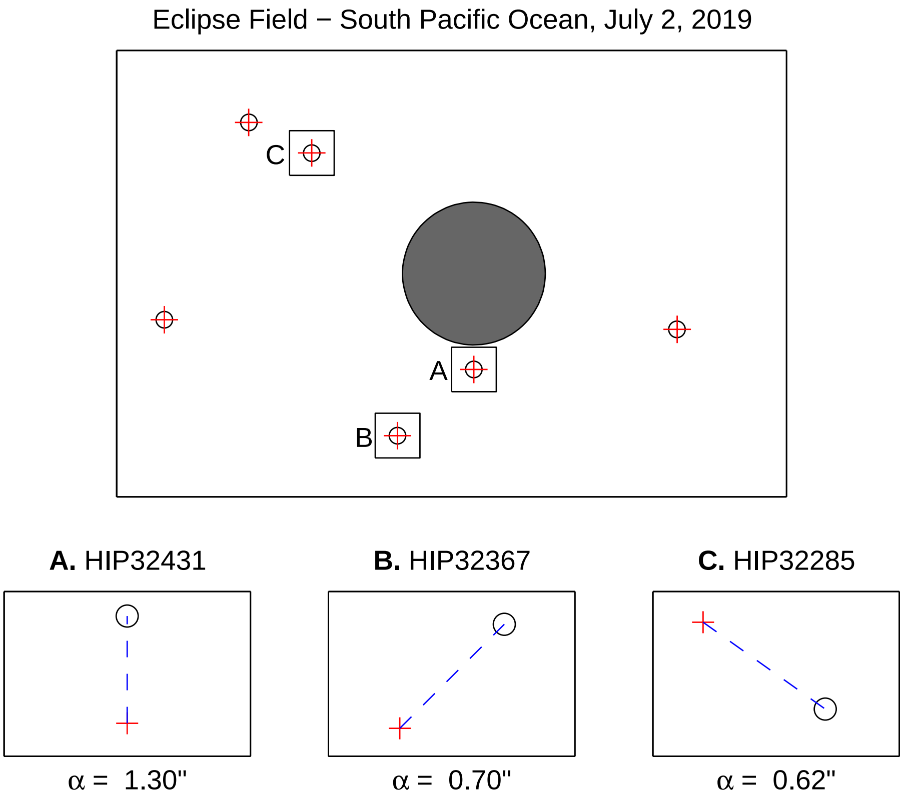

Figure 1.5: The 2019 Eclipse - Pacific Ocean

Configuration of stars during the 2019 eclipse as seen from the Pacific Ocean. Note the large deflection of star "A".

1.5 Gravitational Lensing Occurs on Many Scales

After the gravitational lens effect of the Sun was measured by Eddington, interest in the phenomenon grew and other lens configurations were considered. Since the lens mapping is non-linear, light rays traveling along separate paths around the lens cause multiple images of a single source to occur. In fact, with the absence of any significant perturbing mass along the line of sight, the exact alignment of background and foreground stars with respect to an observer produces an image that would appear as a full ring (Chwolson (1924); Einstein (1936)), found by setting $\beta = 0$ in Equation 1.11. When the background star is slightly offset from the line of sight, an observer would see the images as a pair of partial arcs. While the size scale of individual stars prohibit direct observation of multiple imaging due to the extremely small deflection angles involved (Einstein 1936), the lens effect provides magnification that significantly increases the amount of light reaching an observer. Paczynski (1986) showed that the apparent intensity of background sources changes in a characteristic way when foreground masses pass in front of background stars. By consistently monitoring a field of many stars on a regular basis it is possible to detect this change in flux (Udalski et al. (1992); Alcock et al. (1993); Sackett (1995)), which is known as microlensing. The microlensing effect has been used to detect the lensing of stars in the Large Magellanic Cloud and the bulge of the Milky Way by objects within our galaxy, and searches have also been used to observe planets orbiting distant stars (The MOA and OGLE Collaborations 2011). Furthermore, these techniques have been applied to successfully identify the presence of massive compact halo objects (MACHOs) that populate the halo of the Milky Way. Detecting these objects is difficult because they emit little to no radiation, such as quiescent compact objects and dim, low-mass stars. It has been estimated that MACHOs contribute up to 20% of the dark matter content of our galaxy (Alcock 2000). Highly magnified microlensed stars have been used to estimate the radius of these stars (Alcock et al. 1997b) and the first observations of limb darkening on stars other than the Sun have been made using microlensed sources (Witt 1995).

Despite the difficulty in directly observing gravitational lens effects due to individual stars, Zwicky (1937) postulated that far more dramatic examples could be observed when a background galaxy is lensed by a foreground galaxy or cluster of galaxies. Zwicky showed that when this is the case, the huge mass of the lens should provide a large enough gravitational field to significantly bend light and form multiple images of a single source on a scale that might one day be observed. Since then, strong gravitational lensing has grown from curiosity into the realm of precision science. The first galaxy-scale lens system was observed by Walsh, Carswell & Weymann (1979), when dual images of the radio bright galaxy $QSO0957+561$ were subsequently confirmed by spectroscopy to be separate views of the same object. The first full radio ring, $MG1131+0456$ (Hewitt et al. 1988), was modeled using an analytical lens that provided evidence for the interpretation of the system as a lensed object (Kochanek et al. 1989). Hundreds of strong lens systems are now known through the work of surveys dedicated to finding new examples of this phenomenon (Kochanek et al. (1998); Koopmans & Treu (2002); Koopmans et al. (2006)). These systems provide some of the most visually impressive examples of gravitational lensing, forming complex image configurations (Bolton et al. 2008) and magnifying the received flux of distant background galaxies (Zwicky (1937b); Brewer & Lewis (2005); Negrello et al. (2010)).

Gravitational lensing effects are sensitive to the total mass distribution of the lens. Therefore, the multiplicity and morphology of the images allow strong constraints to be imposed on mass models of the deflector. By comparing these mass models with the observed surface brightness of the lens galaxy, the distribution of dark matter in the deflector can be studied. This fact has tremendous implications because the nature and composition of dark matter is presently unknown. Strongly lensed images also offer a wealth of information about the sources that create them. Due to magnification, the lensing effect allows us to study the surface brightness of extremely distant, faint galaxies on sub-kiloparsec scales (Marshall et al. 2007). Lensed images of these primordial galaxies allow for spectroscopic studies that can provide information about star formation rates and permit the estimation of chemical abundances in galactic building blocks (Stark et al. 2008). The observation of lensed quasars provides a view into the cores of these systems, revealing details of the supermassive black holes at their centers (Schneider et al. 2006) as well as the accretion disks responsible for their massive luminosities (Kochanek 2004). These systems offer a unique opportunity to study the links between galaxy development and black hole growth at high redshift (Hopkins et al. 2009). Entire galaxy clusters have been observed acting as lenses (Bonnet, Meiller & Fort 1994), providing a magnified view of many background objects simultaneously. These cluster lenses can produce giant arcs, stretched images of background galaxies that are many times longer than wide (Lynds & Petrosian 1986). Observations of giant arc systems provide evidence of the presence of dark matter in the cluster cores (Paczynski 1987) and impose strong constraints on models of the cluster mass distribution.

There is significant overlap between galaxy scale strong lensing and the microlens regime. Consider the potential of a galaxy halo which may be divided into a large scale, smooth potential superimposed with small-scale perturbations of lesser magnitude. The general configuration of images depends on the bulk potential of the galaxy, while the magnifications of the images are sensitive to the presence of the perturbations. The change in magnification results in time delays that vary from what would be expected from a smooth model alone. Time delays have been measured for lensed quasars (Kochanek et al. (1998); Mao & Schneider (1998); Saha et al. (2006)) and provide a method of studying the substructure of galactic halos in detail. The morphology of these dark matter halos has far reaching implications for our understanding of the involvement of dark matter in the formation of structure in the universe (Kravtsov 2010).

Far from the centers of galaxies and galaxy clusters, the density of matter is not sufficient to produce multiple images. However, weak lensing effects manifest as a systematic distortion in the shapes of background sources (Tyson et al. 1990). Since galaxies generally posess complex morphologies, the magnitude and direction of this shear must be measured by observing the correlated distortions of large groups of sources. Measurements of cosmic shear are then used to reconstruct the mass density distribution. This effect probes mass distributions when strong lensing is not observed and cannot provide any constraints on the lensing mass (Kaiser & Squires (1993); van Waerbeke & Mellier (2003)). Weak lensing has produced some of the most striking evidence for the existence of dark matter in the universe (Markevitch et al. 2004) and is now widely used to model the distribution of matter on large scales (Brainerd et al. (1996); van Waerbeke et al. (2000)).

A common theme among these distinct regimes is the unique tool that lensing provides to map the distribution of matter on a variety of scales. However, gravitational lenses are also valuable due to their inherent connection to cosmological parameters (Dobke et al. 2009). Observable lens properties depend on the ratios of angular diameter distances, which are sensitive to cosmological models (Chae (2007); Oguri et al. (2008); Suyu et al. (2010)). Refsdal (1964) showed that there is a time delay due to path length and time dilation effects between the strongly lensed images of a source. A measurement of these time delays for variable sources constrain the value of the Hubble constant. The fraction of lensed systems in a sample depends on the number, size and redshift distribution of lens populations, so lens statistics can be used to study the growth of structure in the universe (Schneider et al. 2006). Due to these varied applications, gravitational lensing provides a versatile approach to study a host of significant subjects in modern astrophysics.

We focus on the strong lensing regime in this work. To describe lensing by individual galaxies, a more thorough treatment of the lens formalism is needed, and this theory is developed in Chapter 2. Chapter 3 outlines the development of algorithms to model gravitational lens systems and presents our unique approach to lens modeling. Chapter 4 presents a generalization of our modeling algorithm that allows for the inclusion of spatially variant point spread functions (PSFs), and Chapter 5 details the application of our modeling code to a subsample of galaxies from the Sloan Lens ACS (SLACS) group.

Chapter 2

Gravitational Lens Theory

In section 1.2, we derived the deflection angle of a point mass lens, which is useful when lensing effects occur on the scale of individual stars. However, we are interested in studying the strong lensing regime where more general asymmetric lens models are needed to describe the mass distributions of galaxy scale objects. Consider a three dimensional matter distribution centered on the lens plane, with density $\rho(\boldsymbol{\xi}, z)$ and potential $\phi(\boldsymbol{\xi}, z)$. By the thin lens approximation, it is appropriate to represent the lens plane as a two dimensional mass sheet. To find the projected two dimensional surface density distribution, we simply integrate along the z direction. Using the thin lens approximation, the limits of integration are taken to be from $-\infty$ to ∞:

$$\Sigma(\boldsymbol{\xi}) = \int \rho(\boldsymbol{\xi}, z) dz. \tag{2.1}$$

It is useful to perform a similar projection on the corresponding Newtonian potential (Schneider, Ehlers & Falco 1992). Defining the angular gradient operator $\boldsymbol{\nabla}_\theta = D_d \boldsymbol{\nabla}_\perp$, and making use of Equations 1.7 and 1.2, we have

$$\psi(\boldsymbol{\theta}) = \frac{D_{ds}}{D_s D_d} \frac{2}{c^2} \int \phi(D_d \boldsymbol{\theta}, z)\ dz, \tag{2.2}$$

such that

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \boldsymbol{\nabla}_\theta \psi(\boldsymbol{\theta}). \tag{2.3}$$

A projected Poisson equation can be found using the lens potential. Let us return to the three dimensional Newtonian potential, $\phi(\boldsymbol{\xi}, z)$. Poisson's equation integrated over z gives

$$\int \boldsymbol{\nabla}^2\phi(\boldsymbol{\xi}, z)dz = 4\pi G \int \rho(\boldsymbol{\xi}, z)dz = 4\pi G\Sigma(\boldsymbol{\xi}), \tag{2.4}$$

using the definition of $\Sigma(\boldsymbol{\xi})$ in Equation 2.1. We write $\boldsymbol{\nabla}_\theta$ in terms of the full 3D Laplacian operator such that

$$\boldsymbol{\nabla}^2_\theta = D^2_d\left(\boldsymbol{\nabla}^2 - \frac{\partial^2}{\partial z^2}\right). \tag{2.5}$$

Using this expression with Equations 2.2 and 2.4 gives

$$\boldsymbol{\nabla}^2_\theta\psi(\boldsymbol{\theta}) = \frac{2}{c^2}\frac{D_{ds}D_d}{D_s}4\pi G\Sigma(\boldsymbol{\theta}) \tag{2.6}$$

where $\phi(\boldsymbol{\xi}, z) \to 0$ as $z \to \pm\infty$. We define the critical density of the lens (Petters et al. (2001); Schneider, Ehlers & Falco (1992)) as

$$\Sigma_c = \frac{D_s}{D_d D_{ds}}\frac{c^2}{4\pi G}, \tag{2.7}$$

which is an important quantity in lensing theory that we return to in Section 2.1. For now we use this constant as a normalization term to simplify the projected Poisson equation (Schneider 1985):

$$\boldsymbol{\nabla}^2_\theta\psi(\boldsymbol{\theta}) = 2\kappa(\boldsymbol{\theta}) \tag{2.8}$$

where the convergence κ is defined as

$$\kappa(\boldsymbol{\theta}) = \frac{\Sigma(\boldsymbol{\theta})}{\Sigma_c}. \tag{2.9}$$

The Poisson equation is a linear second order differential equation, and a consequence of this linearity is that the superposition principle holds for its solutions $\psi(\boldsymbol{\theta})$. This is expected since we are working in the linearized weak field limit of GR (See Appendix A), when the lens velocity is small with respect to the speed of light $v \ll c$ and the gravitational fields are weak, such that $|\phi| \ll c^2$. This implies that lens potentials can be added together and the total deflection angle field can be easily calculated. To illustrate this, consider the infinitesimal deflection of a mass element $dm = \Sigma(\boldsymbol{\theta})d^2\theta$ using Equation 1.9. The total deflection angle is simply the sum of the contributions of each mass element over the entire lens plane:

$$\alpha(\boldsymbol{\theta}) = \frac{1}{\pi}\int \kappa(\boldsymbol{\theta})\frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{|\boldsymbol{\theta} - \boldsymbol{\theta}'|^2}d^2\boldsymbol{\theta}, \tag{2.10}$$

using the definition of κ in Equation 2.9. From Equation 2.10 with the identity

$$\boldsymbol{\nabla} \ln |\boldsymbol{\theta}| = \frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|^2}, \tag{2.11}$$

we are led to the general form of the lens potential

$$\psi(\boldsymbol{\theta}) = \frac{1}{\pi} \int \kappa(\boldsymbol{\theta}) \ln |\boldsymbol{\theta} - \boldsymbol{\theta}'| d^2\boldsymbol{\theta}' \tag{2.12}$$

which is a solution of the two-dimensional Poisson equation.

2.1 Critical Density

The physical interpretation of the critical lens density Σ_c defined in Equation 2.7 can be understood by considering a lens density distribution with circular symmetry that has constant density Σ within an area of radius $D_d\theta$. The total mass of such a lens is $M(\theta) = \pi D_d^2 \theta^2 \Sigma$, where we now treat θ as a one-dimensional quantity due to the symmetry of the mass distribution. The reduced deflection angle, Equation 1.7, using a symmetric lens (Equation 1.9) becomes

$$\alpha(\theta) = \frac{D_d D_{ds}}{D_s} \frac{4\pi G \Sigma}{c^2} \theta. \tag{2.13}$$

Note that the coefficient multiplying the surface density is the reciprocal of the critical density. The lens equation in this case, (Equation 1.8), can be expressed as

$$\beta = \theta \left(1 - \frac{\Sigma}{\Sigma_c}\right). \tag{2.14}$$

Placing a point source directly behind this lens at $\beta = 0$ implies that a ring-like image should be formed just as it was for a point mass due to the circular symmetry of the lens. The ring occurs for non-trivial values of Σ that satisfy

$$\kappa = \frac{\Sigma}{\Sigma_c} = 1. \tag{2.15}$$

This condition shows that multiple imaging occurs if the surface mass density along the line of sight is equal to the critical density.

2.2 Magnification

Gravitational lensing redistributes light rays emitted by a source and does not change the total number of rays. Therefore the surface brightness of the source must be conserved. Consider a resolved source that has an intensity I_ν at frequency ν, such that the surface brightness on the source plane is described by $I_\nu(\boldsymbol{\beta})$. The intensity in the image plane is then given by

$$I_\nu(\boldsymbol{\theta}) = I_\nu(\boldsymbol{\beta}(\boldsymbol{\theta})). \tag{2.16}$$

However, the cross sectional area of a bundle of light rays is affected by the deflection. To illustrate this effect, consider a bundle of light rays with a circular cross section that passes by a lensing mass distribution. The deflection depends on the distance of each ray from the lens center, so that neighbouring rays are deflected by different amounts. Therefore the cross section of the light ray bundle is distorted from the undeflected case. Since the lensing effect alters the apparent area of a source, it also changes the corresponding flux such that the surface brightness is conserved.

To determine the degree of magnification, suppose that the source in the absence of lensing subtends a solid angle Ω_β on the sky, and let the solid angle of the corresponding image be Ω_θ (see Figure 2.1). The flux of the source is

$$F_\beta = I_\nu \Omega_\beta \tag{2.17}$$

and the image has flux

$$F_\theta = I_\nu \Omega_\theta, \tag{2.18}$$

so that the magnification is given by the ratio of these two expressions

$$\mu = \frac{F_\theta}{F_\beta} = \frac{\Omega_\theta}{\Omega_\beta}. \tag{2.19}$$

The distortion of area elements is given by the Jacobian $\boldsymbol{A}(\boldsymbol{\theta})$ of the lens equation. Using the definition of the projected lens potential and defining the partial derivatives as $\psi_{ij} = \partial^2\psi/\partial\theta_i\partial\theta_j$, we write the components of the Jacobian as

$$A_{ij} = \delta_{ij} - \psi_{ij}. \tag{2.20}$$

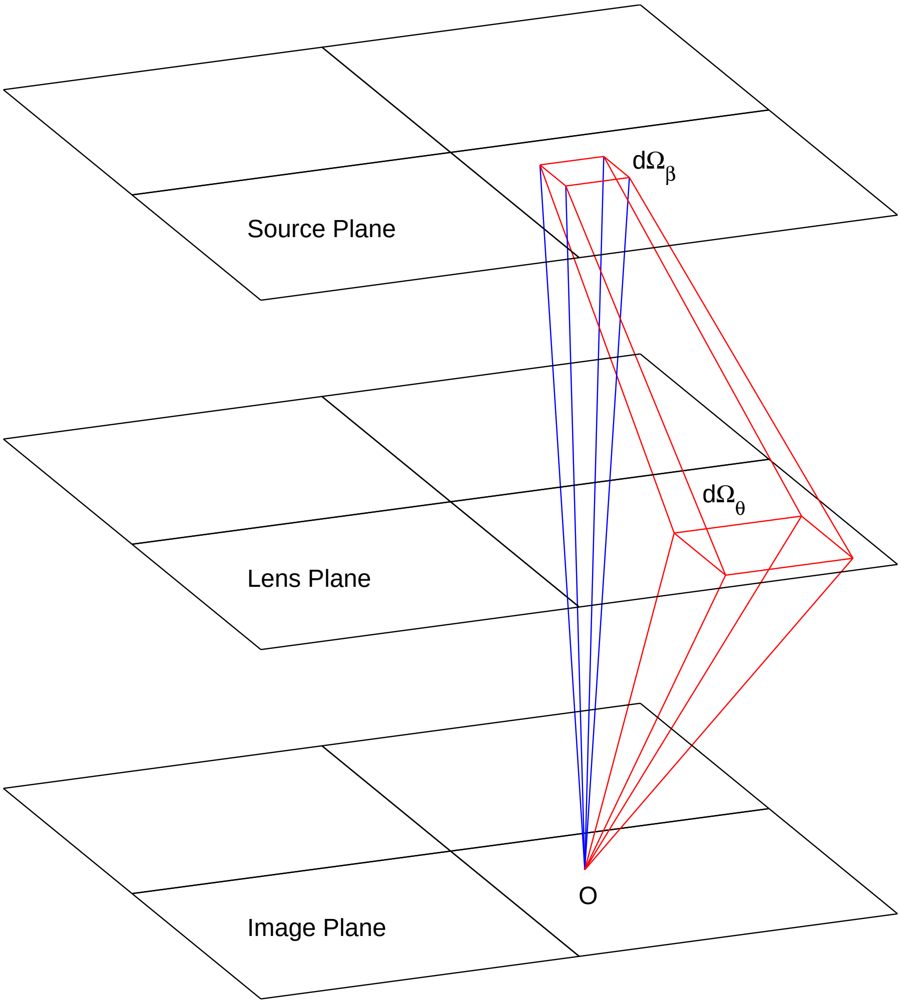

Figure 2.1: Illustration of Gravitational Lens Magnification

Detected flux from an extended object in the source plane. The blue solid angle $d\Omega_\beta$ represents the light rays collected by the observer in the absence of lensing. The red solid angle $d\Omega_\theta$ shows the apparent size of the source under the influence of the lensing effect.

The magnification factor for a general lens surface density distribution is found by calculating the inverse of the Jacobian determinant

$$\mu(\boldsymbol{\theta}) = \frac{1}{\det(\boldsymbol{A}(\boldsymbol{\theta}))}. \tag{2.21}$$

The flux of an infinitesimal source with an image at $\boldsymbol{\theta}$ is changed by a factor of magnitude $|\mu|$. In general, the magnification factor μ can change sign over the image plane. Negative magnifications represent areas of reversed parity where the lensed image is mirrored with respect to the orientation of the source. As we shall see in Section 3, the conservation of surface brightness is the foundational principle behind several important methods for modeling strong lens systems.

It is also possible for the Jacobian determinant to vanish at certain points on the image plane, such that $\mu \to \infty$. In general the set of these critical points lie on closed curves in the image plane, which are called caustic curves when mapped to the source plane. Caustic curves define closed boundaries on the source plane enclosing regions where a source produces a constant number of multiple images. When a source crosses a caustic from "outside" to "inside", from low to high image multiplicity, an image crosses the corresponding position on the critical curve and splits in two. When sources pass from high to low image multiplicity, two images are seen to merge at a critical curve in the image plane.

We can obtain a more intuitive understanding of the magnification effect by returning to the definition of the inverse magnification tensor, Equation 2.20. We have already introduced the lens convergence, $\kappa(\boldsymbol{\theta})$ in Equation 2.9, such that

$$\kappa = \frac{1}{2}\left(\psi_{11} + \psi_{22}\right). \tag{2.22}$$

Let us also define the following combinations of derivatives (Narayan & Bartelmann 1995):

$$\begin{aligned} \gamma_1(\boldsymbol{\theta}) &= \tfrac{1}{2}\left(\psi_{11} - \psi_{22}\right) \\ \gamma_2(\boldsymbol{\theta}) &= \psi_{12} = \psi_{21}. \end{aligned} \tag{2.23}$$

The quantities $\gamma_1(\boldsymbol{\theta})$ and $\gamma_2(\boldsymbol{\theta})$ are the components of the shear tensor. These definitions allow us to express Equation 2.20 as a function of the convergence and shear:

$$\boldsymbol{A}(\boldsymbol{\theta}) = \begin{bmatrix} 1-\psi_{11} & -\psi_{12} \\ -\psi_{21} & 1-\psi_{22} \end{bmatrix} = \begin{bmatrix} 1-\kappa-\gamma_1 & -\gamma_2 \\ -\gamma_2 & 1-\kappa+\gamma_1 \end{bmatrix}. \tag{2.24}$$

Now define $\gamma(\boldsymbol{\theta}) = \sqrt{\gamma_1^2 + \gamma_2^2}$ and angle $\delta(\boldsymbol{\theta})$. We can then write the inverse magnification tensor in a simple form given by

$$\boldsymbol{A}(\boldsymbol{\theta}) = (1-\kappa) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \gamma \begin{bmatrix} \cos(2\delta) & \sin(2\delta) \\ \sin(2\delta) & -\cos(2\delta) \end{bmatrix}. \tag{2.25}$$

This expression shows the role of the convergence and shear in gravitational lens imaging. Consider a circularly symmetric source with unit radius. The first term in Equation 2.25 depends only on the convergence and produces an image that is an isotropically scaled image of the source. The second term is the shear tensor that describes a shear with magnitude γ in the direction $\delta(\boldsymbol{\theta})$ with respect to the θ_x axis. Note that the shear tensor is symmetric, so that the eigenvectors are guaranteed to be orthogonal. The shear tensor transforms a circular source into an ellipse with major and minor axes given by the eigenvalues of the shear matrix, along the direction of the eigenvectors. Taken together, these two effects combine to produce an elliptical image of a circular source.

This notation is useful for studies of weak lensing, since the convergence and shear components are both expressed as linear combinations of the potential. Using the relationship between these three quantities, it has been shown (Kaiser & Squires 1993) that a measurement of the shear field $\boldsymbol{\gamma}$ can be used to derive the corresponding surface mass density $\Sigma(\boldsymbol{\theta}) = \Sigma_c\kappa$.

2.3 Lens Models

Using the theory developed in the previous section, we now explore the properties of some useful analytical lens models. We begin by returning to the example of the point mass lens and discuss the properties of the singular isothermal sphere. After we have established the effects of symmetrical lenses, more realistic elliptical lens models are introduced.

2.3.1 Magnification of The Point Mass Lens

The deflection angle for a point mass is given by Equation 1.9, and was used to determine the deflection of starlight at the limb of the Sun in Section 1.2. We showed that the lens equation reduces to a quadratic for this lens model (Equation 1.10) with solutions:

$$\theta_{\pm} = \frac{1}{2}\left(\beta \pm \sqrt{\beta^2 + 4\theta_E^2}\right), \tag{2.26}$$

where we have used the Einstein radius

$$\theta_E = \sqrt{\frac{D_{ds}}{D_d D_s}\frac{4GM}{c^2}}, \tag{2.27}$$

with the image position $\theta = \sqrt{\theta_x^2 + \theta_y^2}$, and source position $\beta = \sqrt{\beta_x^2 + \beta_y^2}$.

Consider a source directly behind the point mass such that $\beta = 0$. When this is the case, we find one solution to Equation 2.26 given by $\theta = \theta_E$. This radius defines the Einstein ring in the image plane. Therefore the Einstein radius is also the critical curve of the point mass lens. The only caustic is a single point in the source plane directly behind the deflector. When $\beta \to \pm\infty$, the lens equation reduces to $\theta\left(\theta - \beta\right) \approx 0$, resulting in an image at the lens center $\theta \approx 0$ and an image at the source position $\theta \approx \beta$.

The magnification of the point mass lens can also be found analytically. Consider the ratio of solid angle elements of the image and source given by Equation 2.19. We write the solid angle of a circularly symmetric source $\beta d\beta$ and for the image $\theta d\theta$. The ratio of these quantities is the magnification,

$$\mu = \frac{\theta}{\beta}\frac{d\theta}{d\beta}. \tag{2.28}$$

Using the lens equation for a point mass lens for β and the corresponding derivative, we find the magnification of the images $\theta_{\pm}$:

$$\mu_{\pm} = \left(1 - \frac{\theta_E^4}{\theta_{\pm}}\right)^{-1}. \tag{2.29}$$

From this equation we see that $\mu \to 0$ as $\theta \to 0$, so the image that approaches the origin is unobservable since the flux vanishes. As the source moves far from the lens so that $\beta \gg \theta_E$, we find that $\theta \approx \beta$ and $\mu \to 1$, such that the source is not magnified at all (Paczynski 1986).

2.3.2 The Singular Isothermal Sphere

The singular isothermal sphere (SIS) is a commonly used model in Galactic astronomy (Binney & Tremaine (1988); Shapiro et al. (1999)). A spherically symmetric gravitational potential is assumed, and the particles of an ideal gas represent the masses that comprise the galaxy. The hydrostatic equilibrium conditions are given by

$$\frac{1}{\rho}\frac{dP}{dr} = -\frac{GM(r)}{r^2} \tag{2.30}$$

$$\frac{dM}{dr} = 4\pi r^2 \rho, \tag{2.31}$$

where ρ is the mass density, P the pressure of the gas and $M(r)$ the mass inside radius r. We assume an equation of state of the gas given by

$$P = \frac{\rho k T}{m}, \tag{2.32}$$

where m is the mass of each particle in the gas, T is the gas temperature and k is Boltzmann's constant. For an isothermal gas in equilibrium, the velocity dispersion σ_v and the temperature T are constant such that:

$$m\sigma_v^2 = kT. \tag{2.33}$$

Assuming a solution of the form $\rho = Cr^{-B}$ with B and C constant, the hydrostatic equilibrium conditions give the SIS density distribution as

$$\rho(r) = \frac{\sigma_v^2}{2\pi G}\frac{1}{r^2}. \tag{2.34}$$

It has been observed that galaxies have flat rotation curves in general (Rubin et al. 1962), which show that the velocities of stars orbiting the center of galaxies are constant out to large radii. These velocity profiles are recovered by the SIS, which is an attractive feature of the model. The velocity of a particle on a circular orbit within the SIS potential is given by

$$v_r^2 = 2\sigma_v^2. \tag{2.35}$$

Despite this fact, the continuous SIS is not a physically realizable model since the mass diverges as r as seen in Equation 2.31 unless truncated at some radius r_t.

Nevertheless the SIS is a widely used density distribution due to its simplicity and useful dynamical properties.

The surface density of the SIS is found by projecting the three dimensional mass density along the line of sight. Once again, we make use of the symmetry of the lens to write $\xi = D_d\theta$, so that the surface density becomes:

$$\Sigma(\theta) = \frac{\sigma_v^2}{2G}\frac{1}{D_d\theta}. \tag{2.36}$$

The potential produced by this density distribution is given by

$$\psi(\theta) = \frac{D_{ds}}{D_s}\frac{4\pi\sigma_v^2}{c^2}\theta, \tag{2.37}$$

which results in the reduced deflection angle given by Equation 2.10:

$$\alpha = \frac{D_{ds}}{D_s}\frac{4\pi\sigma_v^2}{c^2}. \tag{2.38}$$

Therefore, the lens equation becomes

$$\beta = \theta\left(1 - \frac{\theta_E}{\theta}\right), \tag{2.39}$$

with the Einstein radius

$$\theta_E = \frac{D_{ds}}{D_s}\frac{4\pi\sigma_v^2}{c^2}. \tag{2.40}$$

The lens equation for the SIS (Equation 2.39) can be solved analytically, just as in the case of the point mass lens. The positions of the images of a lensed source can be found by solving the polynomial defined by the square of Equation 2.39,

$$\theta^2 - 2\theta_E\theta + (\theta_E^2 - \beta^2) = 0. \tag{2.41}$$

The image positions can be found for three cases depending on the source position $\boldsymbol{\beta}$. The SIS lens permits two solutions when the source is within the Einstein radius θ_E, with one inside and the other outside this radius. The images are colinear with the source position and the lens center. When the source is outside the Einstein radius, $\beta > \theta_E$, only one image exists.

The SIS can be "softened" by adding a core radius that replaces the central singularity with a finite density. This requires an additional parameter s to be added to the radial coordinate, resulting in deflection angle

$$\alpha = \frac{D_{ds}}{D_s}\frac{4\pi\sigma^2}{c^2}\frac{\theta}{\left(s^2+\theta^2\right)^{\frac{1}{2}}}. \tag{2.42}$$

The effect of adding a finite core size causes a third image to appear with non-zero magnification at $\boldsymbol{\theta} = \mathbf{0}$. We make use of the SIS as an example when exploring the effects of spatially variant PSFs in Chapter 4.

2.3.3 Elliptical Lens Models

Spherically symmetric lenses are useful for describing microlensing effects and provide illustrative examples of image formation by gravitational lenses. An attractive property of the models presented in Sections 2.3.1 and 2.3.2 is that they are sufficiently simple to permit full analytical solutions of the lens equation. However, in the case of strong lensing by galaxies, spherically symmetric models are too simple to describe real lens systems. Galaxies are observed to have a wide range of ellipticities, suggesting that more realistic lens models must include the effects of ellipticity. The magnitude of the ellipticity and the orientation of the lens add two more degrees of freedom to lens models, which permit more complicated image morphologies. Although we can find analytical expressions for the deflection angle fields of these elliptical models, they are complex enough that we are forced to abandon a fully analytical solution to the lens equation itself. Due to the increased complexity of elliptical lenses, numerical methods must be used to find solutions of the lens equation. We discuss this difficulty further in Chapter 3.

Several approaches have been used to include ellipticity to describe realistic lens models. The simplest elliptical lens is found by including the ellipticity in the potential of Equation 2.37 directly (Blandford & Kochanek (1987); Schramm (1990)). Using the angular Cartesian coordinates $\boldsymbol{\theta}$, we write the potential

$$\psi\left(\boldsymbol{\theta}\right) = \frac{D_{ds}}{D_s}\frac{4\pi\sigma_v^2}{c^2}\left(s^2+(1-\epsilon)\theta_x^2+(1+\epsilon)\theta_y^2\right)^{\frac{1}{2}}, \tag{2.43}$$

where we have included the ellipticity ϵ and a finite core size s, which acts to remove the singularity at the origin, just as in the case of the softened SIS. This pseudo-singular isothermal elliptical lens potential (PSIEP) lens is useful due to its simplicity. However, the elliptical potential produces unphysical dumbbell shaped isodensity contours for large values of ϵ, and so the value of this parameter must be restricted to a small range, corresponding to approximately $0 < \epsilon < 0.3$. A set of isopotential curves is shown for various values of the ellipticity in Figure 2.2.

A more realistic elliptical lens model builds the ellipticity into the projected density directly (Kassiola & Kovner (1993); Kormann, Schneider and Bartelmann (1994); Keeton and Kochanek (1998)). An elliptical lens density model does not suffer from the unphysical dumbell shaped isodensity curves of the PSIEP, and is therefore not restricted to low ellipticity ϵ. These elliptical density models have been shown to represent isolated early-type (E and S0) galaxies well (Bolton et al. 2008). For these reasons, the pseudo-singular isothermal elliptical mass distribution is more versatile than the PSIEP lens, although this usefulness comes at the cost of analytical simplicity.

We define the quantity $\Psi = \sqrt{q^2\left(s^2 + \theta_x^2\right) + \theta_y^2}$, where q is the axis ratio defined by

$$q = \sqrt{\frac{1-\epsilon}{1+\epsilon}}, \tag{2.44}$$

so that $q = 1$ ($\epsilon = 0$) corresponds to a spherical mass density, in keeping with the notation used in Equation 2.43. Using b as the Einstein radius of a SIS (Equation 2.40), the scaled projected density of the elliptical mass distribution is given by

$$\frac{\Sigma}{\Sigma_c} = \frac{bq}{2\Psi}. \tag{2.45}$$

This density distribution gives the deflection angle $\boldsymbol{\alpha} = (\alpha_x, \alpha_y)$:

$$\begin{aligned} \alpha_x &= \frac{bq}{\sqrt{1-q^2}} \tan^{-1}\left(\frac{\theta_x\sqrt{1-q^2}}{\Psi + s}\right) \\ \alpha_y &= \frac{bq}{\sqrt{1-q^2}} \tanh^{-1}\left(\frac{\theta_y\sqrt{1-q^2}}{\Psi + q^2 s}\right), \end{aligned} \tag{2.46}$$

The potential that gives rise to this deflection angle is quite complicated when expressed as a function of θ_x and θ_y. A simpler form is obtained by writing the lens

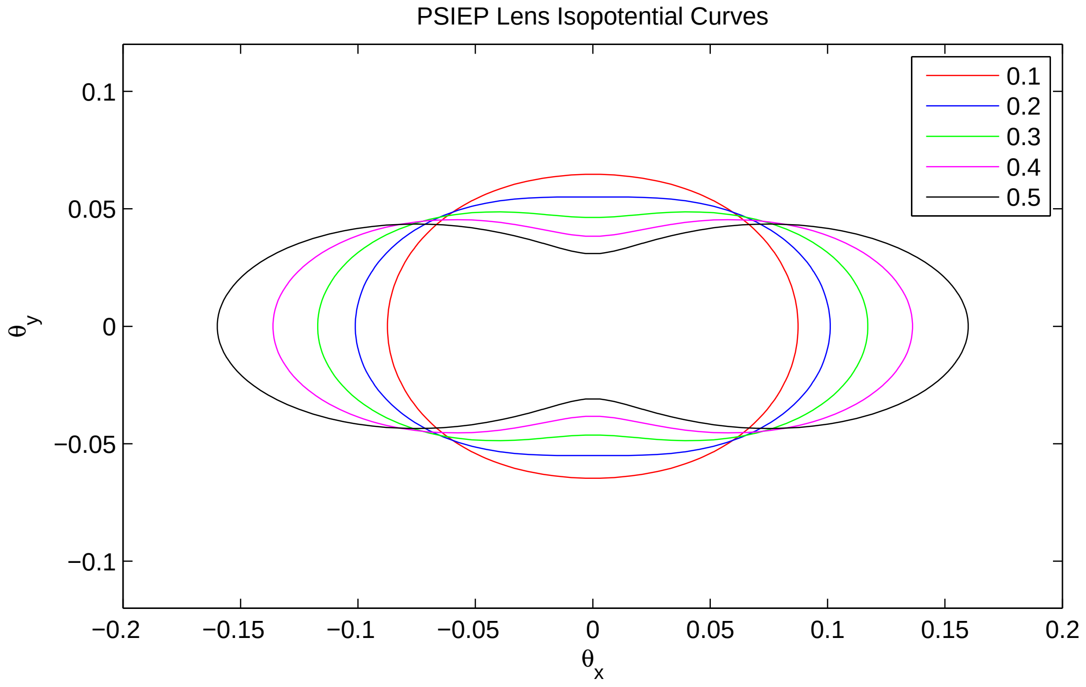

Figure 2.2: Pseudo-Singular Isothermal Elliptical Potential Density Contours

PSIEP lens density contours for a variety of ellipticity values. The lens parameters used to generate this plot were $\sigma_v = 200$ km s^{-1}, with the lens centered at the origin and position angle 0. The source and lens planes were set to $D_s = 0.7$ and $D_d = 0.3$ with the cosmological parameters as defined in Appendix B.

potential in terms of the deflection angle (Keeton and Kochanek 1998), such that

$$\psi(\boldsymbol{\theta}) = \theta_x\alpha_x + \theta_y\alpha_y - bqs\ln\left[(\Psi+s)^2 + (1-q^2)\theta_x^2\right]^{\frac{1}{2}} + bqs\ln\left[(1+q)s\right]. \tag{2.47}$$

The magnification produced by this lens has the following form:

$$\det(\boldsymbol{A}) = \mu^{-1} = 1 - \frac{bq}{\Psi} + \frac{b^2q^2s}{\Psi\left((\Psi+s)^2 + (1-q^2)\theta_x^2\right)}. \tag{2.48}$$

In the limit of a singular spherical mass distribution, $q \to 1$ and $s \to 0$, the above expressions simplify to those of the SIS. Due to it's close connection with the SIS, the elliptical mass density lens model is typically referred to as a singular isothermal ellipse (SIE) lens in the literature. Note that for a finite core size $s > 0$, the lens is softened in analogy with Equation 2.42.

2.3.4 External Shear

The previous analysis considered the path of light rays around an isolated density distribution. However, the morphologies of lensed images are also affected by matter concentrations near the line of sight. The effect of these adjacent mass distributions are taken into account by adding constant convergence and shear terms to the lens potential (Schneider, Ehlers & Falco 1992). Consider the potential

$$\psi(\boldsymbol{\theta}) = \frac{\kappa_c}{2}\left(\theta_x^2 + \theta_y^2\right) - \frac{\gamma_1}{2}\left(\theta_x^2 - \theta_y^2\right) - \gamma_2\theta_x\theta_y \tag{2.49}$$

with γ_1 and γ_2 the constant components of the shear in the θ_x and θ_y directions, and constant convergence κ_c. These effects are modeled after the expressions in Equation 2.25, except that the external convergence and shear are taken to be constant, in contrast to the previous expressions. The deflection angle field is given by

$$\begin{aligned} \alpha_x(\boldsymbol{\theta}) &= \kappa_c\theta_x - \gamma_1\theta_x - \gamma_2\theta_y \\ \alpha_y(\boldsymbol{\theta}) &= \kappa_c\theta_y + \gamma_1\theta_y - \gamma_2\theta_x. \end{aligned} \tag{2.50}$$

External convergence and shear are useful to model the effects of lensing through dense environments.

In general, shear introduces an effective ellipticity in the lens potential and can produce complicated image morphologies when used with a spherical lens model. A point source and shear model, called a Chang-Refsdal lens (Chang & Refsdal (1979); Subramanian & Chitre (1984)), has been used to describe a stellar mass microlens perturbed by the potential of a background galaxy. It has been shown that an isolated circularly symmetric lens with an external shear produces images that are comparable with those found from elliptical lens models (Schneider, Ehlers & Falco 1992).

The extra degrees of freedom provided by external shear are not always necessary when fitting strong lens systems. For instance, the lens galaxies in the SLACS survey can be effectively modeled as isolated galaxies using SIE lens models and the inclusion of external shear was not found to greatly improve the fits (Koopmans et al. (2006); Bolton et al. (2008)). We fit a sample of the SLACS lens systems using a SIE lens model without including shear in Chapter 5.

Chapter 3

The Development of Gravitational Lens Modeling

Several methods of varying complexity have been developed to model strong gravitational lens systems. We begin by discussing analytical lens models that rely on a set of lens parameters $\boldsymbol{p}_L$. As we have seen, the lens equation is non-linear in general, and multiple solutions of the lens equation $\boldsymbol{\theta}$ exist for each source position $\boldsymbol{\beta}$. The complexity of the deflection angle field $\boldsymbol{\alpha}$ determines the form of the lens equation. Therefore, the lens equation can be solved analytically for only the simplest cases, such as the point mass and SIS lenses that have a high degree of symmetry. Most realistic lenses do not permit a fully analytical treatment of the lens equation. The solution of the lens equation requires an inversion since we wish to find all images corresponding to a given source position. The direct (and naive) approach is to use a non-linear root-finding algorithm to search for solutions to the lens equation. However, this is difficult in general since there may be many solutions that satisfy the lens equation and there is no objective way to determine whether all have been found. The lensing of extended sources introduces another problem as the lens equation must be inverted at each point within the source. This is a numerically intensive task, so ray-tracing methods are often used instead.

The ray-tracing procedure in gravitational lens modeling requires the generation of pixelized grids on the lens and image planes $\boldsymbol{\theta}$. For a given lens model, we can use the lens equation to map points from the image plane to positions in the source plane using the deflection angle field of a given lens model. This procedure avoids the inversion of the lens equation since we essentially work backward. Rather than determining the resulting images from the source position $\boldsymbol{\beta}$, we cast the points of the pixelated image grid $\boldsymbol{\theta}$ back to the source plane grid. This ray-tracing procedure can be used to find the lensed images of an analytical source intensity function easily and quickly using the conservation of surface brightness (Equation 2.16), thus providing the basic approach of parameterized source modeling (Schramm & Kayser (1987); Kayser and Schramm (1988)). A surface brightness function I_s describes intensity across the source plane and the lensed image intensities (I_i) are easily found since

$$I_i(\boldsymbol{\theta}) = I_s(\boldsymbol{\beta}(\boldsymbol{\theta})) \tag{3.1}$$

by the conservation of surface brightness. Analytical models of the source are sometimes used because they ensure smoothness and positivity when used to model the source intensity distribution. However, the correct parameterization is not always clear for such models, and the choice of a specific parametric model biases the lens and source solutions. Authors have attempted to partially overcome this drawback by using complex but flexible parametric models specified by large parameter sets. The most extreme example is Tyson et al. (1998), who used an elaborate source model with more than 200 parameters to fit the gravitational lens $CL0024 + 1654$. In such cases, it may be simpler to use pixelized source models, which treat all pixels on the source independently. Convolution with a point spread function (PSF) that describes the instrumental blurring of the observations can then be taken into account and a goodness of fit statistic (χ^2) can be calculated to quantify how well the source intensity function and lens parameters $\boldsymbol{p}_L$ match the observed data. An optimization routine can be used to determine the optimal source and lens parameters by minimizing the χ^2 statistic. This approach is still widely used today (Bolton et al. (2008); Marshall et al. (2007); Brewer & Lewis (2011)).

It was also suggested that a variation on the method could be used to determine the surface brightness of a pixelated source from a set of observed data. Kayser and Schramm (1988) introduced the Digital Source Reconstruction (DiSoR) method that treats the source plane as a pixelated grid. The positions of image pixels are traced back to the source plane, and each source plane pixel is assigned an intensity based on the back-traced image pixels that are contained within it. This pixelated source method can be useful since complicated sources may not be easily described by simple analytical functions.

The DiSoR method was first used to model the radio ring $MG1131+0456$ (Hewitt et al. 1988) by Kochanek et al. (1989). This approach, called the Ring Cycle, coupled an iterative optimization routine to the DiSoR method to simultaneously determine the optimal lens parameters. The optimization proceeds in two distinct steps: an "outer loop" searches for optimal lens parameters, and for each lens parameter set an "inner loop" is used to find the best fitting intensity distribution of the pixelated source. Kochanek et al. (1989) also approximated parameter errors and added terms to the merit function to promote models with the maximal number of multiply imaged pixels possible. This is an argument for parsimony in the solution as simpler sources are allowed when multiple imaging is maximized. The basic loop structure of the Ring Cycle as well as the use of the conservation of surface brightness have been incorporated into virtually every successive lens optimization scheme.

Despite the apparent success of these early pixelated source methods, they are limited in their applicability. Equation 3.1 requires all multiply imaged pixels associated with a given source pixel to posess the same intensity. However this is never exactly true due to the fact that real data are noisy. Source pixel intensities must be found from averaging image intensities over rays that trace to the same pixel. This is a problem for the algorithm since the image noise cannot be properly accounted for in the inversion process. Another serious problem with mapping data pixels to determine source plane intensities is that the data are affected by the PSF of the observing instrument. This means that nearby image pixels are no longer truly independent since the blurring effect smears intensity over adjacent source pixels. The Ring Cycle algorithm does not correct for this PSF blurring, and this tends to bias the reconstructions. In general, artificial structure in the recovered sources will be found in order to accomodate for the lack of a description of the PSF. Since all astronomical observations are affected by blurring to some degree, this presents a major difficulty for algorithms based on the Ring Cycle.

The shortcomings of the Ring Cycle algorithm can be avoided by including a deconvolution step in the source modeling routine. Deconvolution problems are well known in the image processing literature (Nagy & Palmer (2003); Hansen et al. (2006); Vogel (1987)), where the goal is to find a deblurred approximation of an image given a blurred observation that is degraded by noise. Image deconvolution is fundamentally an ill-posed inverse problem (Hansen 1997) and is subject to instability due to the noise present in the data (Hansen 2010). Regularization methods are often used to overcome the amplification of noise in the data (Press et al. 2007). Regularization is an attempt to stabilize the inverse problem of image deconvolution by modifying the χ^2 minimization problem to prefer a smooth deblurred (source) image. Regularization and the associated numerical methods are discussed at length in Sections 3.1 and 3.1.3.

The first approach to include a deblurring step in lens modeling involved an implementation of the CLEAN algorithm (Högbom 1974) to compensate for the finite resolution of radio observations. The resulting method, called LensCLEAN (Kochanek & Narayan (1992); Wucknitz (2004)) was first used to model the radio ring $B0218+357$ (Patnaik et al. 1993). The CLEAN algorithm models the source intensity as a set of point sources, called the CLEANed source. The algorithm starts by placing a point source at a location on the source plane, such that the lensed images of this point source (found by ray-tracing) are convolved with the PSF and subtracted from the brightest part of the image. If the remaining residuals are still above the noise level, another CLEANed source is added, and the process repeated. In this way the source distribution is built up by the addition of more CLEANed components each iteration. Once the residuals are comparable to the noise level, the CLEANed source is smoothed by a Gaussian, and the residuals are added back in.

After the CLEAN algorithm was used to model gravitational lens systems, the maximum entropy method (MEM, Cornwell (1982); Skilling and Bryan (1984); Narayan & Nityananda (1986)) was adapted for use with the lensing problem. The resulting algorithm is called LensMEM (Wallington et al. 1996) and was first used to model the VLA observed ring $MG1654+134$ (Ellithorpe et al. 1996). The most commonly used expression in the context of lens modeling is given by

$$J = -\sum_i s_i \log(s_i), \tag{3.2}$$

where s_i are image pixel intensities. The MEM is included in the source optimization step by adding an entropy-based regularization term to the fit statistic,

$$G = \chi^2 + \lambda \sum_i s_i \log(s_i), \tag{3.3}$$

where χ^2 is the standard image fit statistic and λ is an undetermined Lagrangian multiplier. The purpose of this type of regularization is to quantify the complexity of the source intensity term. This MEM fitness statistic restricts the source intensity distribution to positive values. The optimization begins with a large value of λ such that the entropy term dominates early in the run. This tends to pick out solutions that are very smooth. As λ is decreased, an increasing amount of information is included from the χ^2 term and the relative importance of the entropy term is decreased. The resulting source at the end of the procedure is as featureless as possible (the highest entropy source) that simultaneously satisfies the χ^2 criteria. This process of slowly decreasing the regularization constant is necessary to guide the minimization process. As with any method that uses an undetermined multiplier λ, a third loop must be used outside of the lens parameter optimization loop to slowly decrease the value of the regularization constant. The MEM has been included in many successful optimization schemes including the lensVIEW package (Wayth & Webster 2006). Both the MEM and CLEAN based algorithms have drawbacks. The behavior of the CLEAN algorithm is difficult to understand statistically. On the other hand, the MEM fitness function is not linear in $\boldsymbol{s}$ and requires more complicated non-linear optimization methods.

The optimization of extended sources via pixelized intensity distributions is simplified using the versatile semilinear method developed by Warren & Dye (2003) and later expanded upon by a number of authors (Treu & Koopmans (2004); Dye & Warren (2005); Suyu et al. (2006)). The semilinear method uses a pixelized source, and also incorporates the blurring due to the PSF of the instrument used to obtain the data. Additive noise in the observed data is also taken into account by this scheme. The semilinear method describes the source deconvolution routine as a least squares problem (Koopmans (2005); Press et al. (2007)), which is well understood and can be solved using straightforward matrix methods or linear optimization.

In the following sections we detail Mirage, a gravitational lens modeling code written in MATLAB and C. The present version of Mirage is designed to optimize the parameters of analytical lens models and pixelized sources, but work is underway to extend the code to handle non-parametric lens models as well. A modified version of the semilinear method forms the backbone of our lens modeling program. We use sophisticated global optimization methods to fit the lens parameters, and the semilinear method to determine the corresponding source light profile that best matches the data. As a final step, we employ the method of Brewer & Lewis (2005) to enforce the positivity of the source while keeping the nonlinear lens parameters constant. This affords a method of comparison between distinct lens density models because the number of degrees of freedom is well-defined and fixed (Brewer & Lewis 2006). The global optimizers studied in this work consist of a sophisticated genetic algorithm (GA), called Ferret (Fiege et al. 2004), and an enhanced particle swarm optimizer (PSO), Locust, which are both components of the Qubist Global Optimization Toolbox by Fiege (2010). This Chapter discusses a robust method for gravitational lens reconstructions, highlights the benefits of both types of optimization routines, and compares their performance.

In Section 3.1 we will review the semilinear method and our new matrix-free approach to lens modeling. In Section 3.2 we discuss the details of the GA and PSO, as well as a variety of simulated data tests. Section 3.3 presents our results using these methods, and our conclusions are summarized in Section 3.4. The following sections are based on material published in The Astrophysical Journal (Rogers & Fiege 2011a).

3.1 The Semilinear Method

The semilinear method provides a way to solve for optimal source intensities by the direct inversion of a lens matrix, for a given set of lens parameters. However, the search for optimal lens parameters is nonlinear in general, and requires more sophisticated nonlinear optimization methods, such as those discussed in Section 3.2.1. Fast execution of the matrix inversion part of the problem is crucial, because a linear system of equations must be solved for each set of lens parameters tested by the nonlinear optimizer during the search for optimal lens models. Many sets of lens parameters must be evaluated to search the parameter space thoroughly enough to determine the globally optimal solution.

Following Warren & Dye (2003), we label the image pixels $j = 1..J$ and treat the pixels in the source as independent free parameters $i = 1..I$. Given a set of lens parameters, the image of each source pixel is formed by ray-tracing assuming unit brightness $s_i = 1$, and convolved with the PSF. This transformation is encoded in a matrix $\boldsymbol{f}$=$\boldsymbol{BL}$. We assume linear blurring described by the blurring matrix $\boldsymbol{B}$ which accounts for the PSF. The matrix $\boldsymbol{L}$ performs ray-tracing via the lens equation (1.8). The problem is then reduced to finding a set of source pixel scaling factors s_i that minimize the reduced χ^2 statistic between the model image and the observed data. Using the set of source pixel intensities, the lensed image of a source is found easily:

$$b_j = \sum_i s_i f_{ij}, \tag{3.4}$$

where f_{ij} are the elements of the matrix $\boldsymbol{f}$. The χ^2 statistic between the lensed image and the data is:

$$\chi^2 = \sum_j \frac{\left(\sum_i s_i f_{ij} - d_j\right)^2}{\sigma_j^2} \tag{3.5}$$

where d_j is the observed intensity in each image pixel, and σ_j is the standard deviation error associated with pixel j. After differentiating this equation with respect to the source pixel intensities s_i, we define $\boldsymbol{F} = f_{ij}/\sigma_j$, $\hat{\boldsymbol{d}} = d_j/\sigma_j$ and we obtain the relation

$$\boldsymbol{F}^T \boldsymbol{F} \boldsymbol{s} = \boldsymbol{F}^T \hat{\boldsymbol{d}}, \tag{3.6}$$

from which it follows that the source pixel scalings can be determined by linear inversion:

$$\boldsymbol{s} = \boldsymbol{M}^{-1}\hat{\boldsymbol{d}}' \tag{3.7}$$

where $\hat{\boldsymbol{d}}'$=$\boldsymbol{F}^T\hat{\boldsymbol{d}}$ and $\boldsymbol{M}$=$\boldsymbol{F}^T\boldsymbol{F}$. This inversion determines the optimal set of source pixel scalings necessary to reproduce the observed data for a given lens model. Further details of this derivation can be found in Warren & Dye (2003) and Treu & Koopmans (2004). In the standard semilinear method, the system matrix $\boldsymbol{M}$ is very large, where the linear size of the matrix scales as the number of source pixels used in the inversion. The matrix is very sparse when the PSF is narrow, but a greater fraction of matrix elements are non-zero for increasingly broad PSFs. Mirage uses sparse matrix methods to minimize memory usage.

Warren & Dye (2003) originally presented the semilinear method as a ray-shooting algorithm that performs nearest neighbor interpolation. Treu & Koopmans (2004) modified the lens matrix to accommodate bilinear interpolation of the source. This consists of using the four source pixels surrounding a back-traced image pixel with appropriate weighting to assign a brightness value to each image pixel. Mirage currently implements nearest neighbor, bilinear, and bicubic source plane interpolation. Higher order interpolation schemes are possible, but they are computationally more expensive and result in a lens mapping matrix that is less sparse. It is also possible to use more elaborate schemes to grid the source plane, including the Delaunay tesselation scheme used by Vegetti & Koopmans (2009). We restrict the source models to rectilinear grids in this paper, but plan to explore other such options.

In practice, regularization is necessary to stabilize the matrix inversion due to the presence of noise in the data (Treu & Koopmans 2004). This regularization term makes the system matrix $\boldsymbol{M}$ more diagonally dominant and hence better conditioned, which has the effect of increasing the smoothness of the source light distribution. In general, we add a regularization matrix to the system matrix, to give

$$\boldsymbol{M}' = \boldsymbol{M} + \lambda\boldsymbol{H}^T\boldsymbol{H}, \tag{3.8}$$

where λ is an adjustable regularization parameter. It is then possible to control the smoothness of the derived solution by adjusting the regularization parameter, with the unregularized case recovered as $\lambda \to 0$. Zeroth order regularization has $\boldsymbol{H}=\boldsymbol{I}$, which suppresses noise in the inversion by preferring sources with less total intensity (Warren & Dye 2003). It is also possible to introduce more complicated forms of regularization, typically based on finite difference representations of two-dimensional derivative operators.

It has been shown that different regularizing terms produce qualitatively similar results (Treu & Koopmans 2004), and the behavior of a host of linear regularization schemes has been studied in great detail by Suyu et al. (2006) in the framework of Bayesian analysis. An important drawback of regularization is that it introduces dependencies between source plane pixels, which makes it difficult to characterize the effective number of degrees of freedom required to compute the reduced χ^2 (χ_r^2). Therefore, direct comparison of different models is more difficult in regularized schemes than without regularization. Dye & Warren (2005) use an adaptive mesh in the source plane to overcome the problem of calculating the number of degrees of freedom in the problem.

An important advantage of the semilinear method is that errors of the source intensity parameters can be easily determined from the lensing matrix, as seen from the relation

$$M_{ik} = \frac{1}{2}\frac{\partial^2(\chi^2)}{\partial s_i \partial s_k}. \tag{3.9}$$

This equation expresses the lensing matrix as half of the Hessian matrix of the reduced image χ^2 statistic. Warren & Dye (2003) use this relationship to find the covariance matrix $\boldsymbol{C}=\boldsymbol{M}^{-1}$, thus determining the source plane errors automatically during application of the semilinear method. When regularization is used, the covariance matrix cannot be found in this way, but Warren & Dye (2003) proposed a Monte Carlo method as an alternative method to estimate errors.

Despite its conceptual elegance, there are significant practical limitations and drawbacks to the semilinear method. The number of non-zero matrix elements of $\boldsymbol{M}$ scales linearly with the number of pixels in the source and depends on the source interpolation method used. Direct inversion quickly becomes impractical for very large images, or those with large PSFs. This sparsity requirement can be fulfilled by representing the PSF by a simpler function, for example a Gaussian, and setting small values to zero. This thresholding helps to control the potentially poor conditioning of the blurring matrix. However, realistic PSFs may contain significant low-level structure, and fitting a simple analytical function to it may not be desirable in such cases. The semilinear scheme is therefore most practical when the image is small and the PSF is narrow, as in typical optical data.

Another problem with the semilinear method is that it does not enforce the positivity of source pixel intensities. Optimal source solutions derived using this algorithm may contain negative pixel intensities, due to noise in the data, since bounds cannot be enforced in the matrix inversion step. Moreover, there is no form of linear regularization that is guaranteed to prevent this behavior. We note, however, that it is possible to enforce positivity in other lens modeling schemes, such as the MEM, explored by Wallington et al. (1996).

In summary, semilinear inversion provides a convenient method for modeling strongly lensed extended sources because it states the gravitational lens modeling problem using a least-squares approach that is solved by direct matrix inversion. The inversion step guarantees that the globally optimal solution is found for unbounded source pixel values. However, the method is computationally expensive for large images and PSFs. In such cases even building the transformation matrix $\boldsymbol{f}$, incorporating both lensing and blurring effects, is an expensive computation and the inversion step may be time-consuming or impractical due to the poor sparsity and size of the matrix. We show in Section 3.1.1 that it is possible to derive a ‘matrix-free’ formulation that avoids the explicit construction of the matrix and dramatically improves the efficiency of solving for the linear parameters by employing local optimization methods to solve the least squares problem. We also compare results from this technique with the semilinear inversion method and show that both methods produce solutions of similar quality. A final refinement step ensures positivity of the source pixels, thus rectifying a limitation of the standard semilinear method.

3.1.1 Matrix-free modeling of lensed images

The main goals of Mirage are generality, flexibility and sufficient efficiency to allow thorough exploration of lens model parameter space by global optimization techniques, which typically require the evaluation of $10^4 - 10^5$ lens models. We therefore require a fast code to solve the linear part of the problem, which is able to function with arbitrarily complicated PSFs and data of high resolution. Mirage implements the semilinear method, using direct matrix inversion, but also extends this method by using faster and less memory intensive local iterative optimization routines that avoid the need for explicit construction of the lens and blurring operators. The iterative methods in Mirage are not intended as a replacement for the semilinear method, which is the preferred approach when it is computationally practical. However, memory requirements and long run times for the global lens parameter search may practically restrict the semilinear method to source intensity distributions and PSFs that can be built on a small mesh to limit the matrix size and maintain its sparsity. Our technique is intended to augment the least-squares approach of the semilinear method by providing a complement of algorithms capable of modeling large lens images quickly, even if the PSF is also large. We avoid the direct inversion of large matrices, but maintain the least-squares formulation, allowing the use of any linear optimization algorithm suited to the solution of large-scale problems. Since this paper focuses on solving the full nonlinear lens modeling problem, we largely make use of matrix-free methods for the remainder of this work except for comparisons with the direct semilinear method.

Given the parameters of a lens model and assuming a source intensity distribution, matrix multiplication with $\boldsymbol{L}$ results in the unblurred lensed image. This image can also be found by the conservation of surface brightness, as given by Equation 3.1. By storing the positions of the back-traced image pixels on the source plane, we perform an interpolation on the source plane directly, which allows us to find the unblurred lensed image without the need for an explicit representation of the lens matrix. Similarly, a separate algorithm in Mirage has an effect equivalent to multiplying by the transpose of the lens matrix, which works by carefully keeping track of the positions of back-traced image pixels.

The lens mapping magnifies portions of the image plane by differing amounts such that square image pixels mapped back to the source plane may no longer remain square. This effect is especially pronounced when image pixels are traced back to the source plane near caustic curves. In general, the distortion in shape of the back-traced image pixels may cause portions of the back-traced pixels to lie within separate source pixels. To account for this effect, we split each image pixel into N_p subpixels, and trace each of these subpixels to the source plane independently. By interpolating each of these subpixels on the source mesh, we can then average over their intensities in the image plane to find a better estimate of the lensed intensity profile. N_p can be any size, but the execution time of the code increases as we include finer subpixel resolution. This approach improves the accuracy of the transformation from the lens to source plane. In addition, we find that it improves the smoothness of the χ^2 surface, which helps with the global search for optimal lens parameters. Image pixels that do not map to the source plane are not included in the local optimization and are assigned no intensity.

To successfully model a realistic blurred observation, we incorporate the blurring effect of the PSF, which is usually described by the blurring matrix $\boldsymbol{B}$. This matrix is of size $N_i^2 \times N_i^2$ where N_i is the number of image pixels which map back to the source plane. Since the lens matrix $\boldsymbol{L}$ is sparse, avoiding its construction does not directly increase the speed of the code, but it allows us to sidestep the explicit construction of the blurring matrix. Since PSFs may describe a significant blurring effect, the product of the lens and blurring matrices $\boldsymbol{f}$ may have a large number of non-zero entries, decreasing the sparsity of the system. To avoid building the blurring matrix, we convolve with the PSF in Fourier space, which is computationally inexpensive, even for large images. Convolution without a blurring matrix is common in the image processing community and was used by Nagy et al. (2002) to solve least-squares problems in the context of the standard image deconvolution problem. By direct extension, this 'matrix-free' lensing method allows us to solve the least-squares problem described by Warren & Dye (2003) without the need for explicit representation of the matrices involved using local optimization. This approach not only provides a substantial increase in speed, but also allows for the use of large data sets with complicated PSFs.

In principle the matrix-free approach can be extended to spatially variant PSFs using the techniques described by Nagy et al. (2002). See Chapter 4 for details on using spatially variant PSFs with the semilinear method.

The local linear optimization algorithms considered in this paper require an initial guess of the solution. We begin with a blank source prior, and each successive iteration adds increasingly higher spatial frequency detail to this initial guess. Local optimizers such as the conjugate gradient method for least squares problems (CGLS; (Björck 1996)) and steepest descent (SD) require one matrix multiplication and one matrix transpose multiplication per iteration, so that the least squares problem can be solved without explicit representation of the lens matrix. We show that such a procedure converges in practice to a source model that is very close to the solution found through matrix inversion, given a sufficient number of iterations. Moreover, an explicit regularization term is not required in general since iterative optimization techniques have been shown to have an automatic regularizing effect on the problem (Vogel 1987), allowing Equation 3.6 to be minimized directly.

Iterative schemes have been used previously in the strong lensing literature in application to nonlinear regularization, for example by Wallington et al. (1996) in the lensMEM method. A similar approach is used in the LENSVIEW code by Wayth & Webster (2006), which utilizes the MEM discussed by Skilling and Bryan (1984). The semilinear method is restricted to linear regularization terms of the type detailed in Suyu et al. (2006). Nonlinear regularization like the MEM can also be used with Mirage, although these techniques require more complicated nonlinear optimization schemes to solve for the source intensity distribution.

3.1.2 A Small-scale Test

As a small scale test problem, we generated a 120 × 120 pixel image of an analytical source intensity profile. The image pixel size used in this test is 0.03 arcsec. The test source is defined by a two-armed spiral test function, given by Bonnet (1995):

$$S(r,\omega) = \frac{S_0}{r_c^2 + r^2} \exp\left[-2\ \sin^2\left(\omega - \omega_0 - \tau r^2\right)\right], \tag{3.10}$$

where S_0 is the maximum brightness in arbitrary units and core radius r_c. The tightness of the arms about the central bulge is controlled by τ, and ω_0 controls the orientation of the spiral, in standard polar coordinates (r, ω). The lensed image of this function can be complicated, since the function contains significant structure, and therefore provides a good test of the level of detail that we are able to recover using our lens inversion algorithm. A test image is generated using the approach detailed above, with each image pixel composed of a 10 × 10 grid of subpixels. The high subpixel resolution mimics the smooth structure of a natural image. We blur the resulting image by convolving with a Gaussian PSF, with a FWHM of 5 pixels on a 30 × 30 grid, and add Gaussian noise to construct our final test image, as shown in Figure 3.1. Our simulated data set has a peak signal-to-noise ratio $S/N = 8$, where we define the S/N as the maximum image intensity divided by the standard deviation of the additive Gaussian noise. The lens used to produce this image is the six-parameter singular isothermal ellipse (SIE; Keeton and Kochanek (1998)) which has Cartesian deflection angles given by Equation 2.46, restated here for clarity:

$$\begin{aligned}\alpha_x &= \frac{bq}{\sqrt{1-q^2}} \tan^{-1}\left(\frac{x\sqrt{1-q^2}}{\Psi + s}\right) \\ \alpha_y &= \frac{bq}{\sqrt{1-q^2}} \tanh^{-1}\left(\frac{y\sqrt{1-q^2}}{\Psi + q^2 s}\right),\end{aligned} \tag{3.11}$$

where $\psi^2 = q^2(s^2 + x^2) + y^2$ and $q = \sqrt{(1-\epsilon)/(1+\epsilon)}$, and b is the corresponding Einstein radius in the limit of a spherical model with $q = 1$. The parameter b is related to the velocity dispersion σ_v by

$$b = 4\pi \left(\frac{\sigma_v}{c}\right)^2 \frac{D_{ds}}{D_s} \tag{3.12}$$

where D_{ds} and D_s are the angular distances between lens and source and observer and source respectively, and c is the speed of light. The actual parameters used to construct Figure 3.1 are as follows: The velocity dispersion is $\sigma_v = 260$ km s^{-1} resulting in $b = 1.35$ arcsec, ellipticity $\epsilon = 0.4$, lens center $(x, y) = (0, 0.12)$, orientation angle $\theta_L = \pi/2$. We keep the core size fixed at $s = 0$. In addition to these six parameters, we assume that the redshift of the lens and source are $z_d = 0.3$ and $z_s = 1.0$ respectively. For convenience we measure angular distances with respect to the “flat”

Friedmann metric with $k = 0$ (See Appendix B for a discussion of the background cosmology).

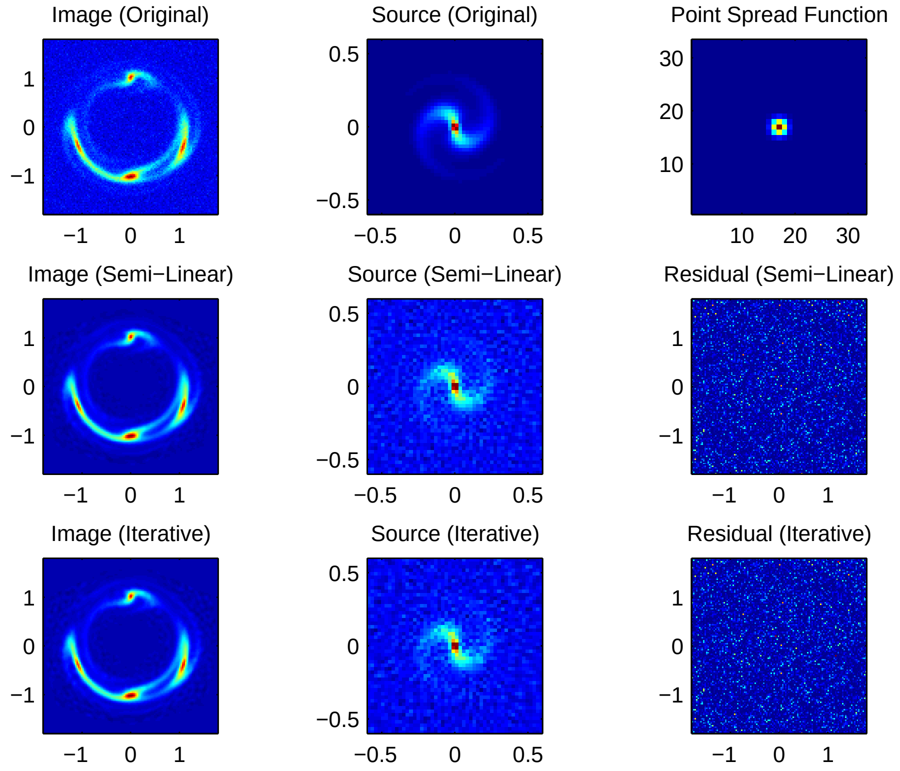

Figure 3.1: Semilinear Method and Iterative Optimization Comparison

Top row, from left: artificial data, source intensity distribution and Gaussian PSF used to generate the observation. The source was built on a 50×50 grid, and the lensed image is defined on a 120×120 grid. Middle row, from left: model observation of resulting source intensity reconstruction, source intensity profile as found by the semilinear method, and the resulting image residuals. Zeroth order regularization with a regularization constant $\lambda = 2.5 \times 10^{-3}$ was used to reconstruct the source. Bottom row, from left: resulting model image, model source and image residuals as determined by the CGLS algorithm after 40 iterations. Note the similarity between the semilinear and iterative solutions with respect to the derived source. Although both of these models contain back-traced noise, the real features of the source are reproduced and clearly visible in the reconstructions.

We model the data using the semilinear method and matrix free methods with subpixel grids of size 2×2. The sub-pixel resolution used for modeling is lower than that used to produce the data, which makes the test more realistic. The source plane is defined on a 40 × 40 grid, with source pixels of size 0.024 arcsec. The original simulated-data image is shown in the first row of Figure 3.1, the semilinear reconstruction on the second row, and a reconstruction using the CGLS algorithm on the third row. CGLS was chosen as the linear optimizer because of its speed and popularity as a local optimization scheme, but in practice all of the optimizers included in the Mirage package produce similar results. All of the local optimization algorithms tested are able to recover the details of the original source function well. Figure 3.2 shows that the rate of convergence varies between optimization algorithms, but they all settle down to the minimum reduced image χ_r^2 values to within 5% by generation 40. The semilinear method is displayed on this plot simply as a constant χ_r^2 because it is a direct method. Note that all of the source reconstructions show noise back-traced from the image, which is unavoidable using pixel mapping techniques on data containing noise. All local optimization algorithms converge in approximately 2 s, while the semilinear method required approximately 16 s. This test was conducted on a 2.4 GHz dual-core Intel machine with 3 GB of memory. Memory usage was monitored and did not exceed the hardware memory limit at any time.

In general, the lensed model image becomes increasingly well matched to the data the longer an iterative optimizer runs, but the usefulness of the solutions eventually starts to degrade as the algorithm begins fitting to the noise in the data. Therefore, the corresponding noise level in the source reconstruction rises as iterations continue, which we can quantify for this test problem because we know the true solution in the absence of noise as shown in Figure 3.3. In effect, the number of iterations of the local optimizer acts as a regularization parameter (Fleming 1990). Thus, it is possible to introduce regularization by carefully controlling the number of iterations during local optimization. In general, implicit regularization is present whenever these local optimizers are used in the context of deblurring problems, which implies that suitable stopping criteria must be established to find the optimally regularized solution (Hansen et al. 2006). It is noteworthy that the semilinear method suffers from a related problem since the regularization constant is a free parameter, and therefore the associated ambiguity is equivalent to the problem of choosing a stopping criteria

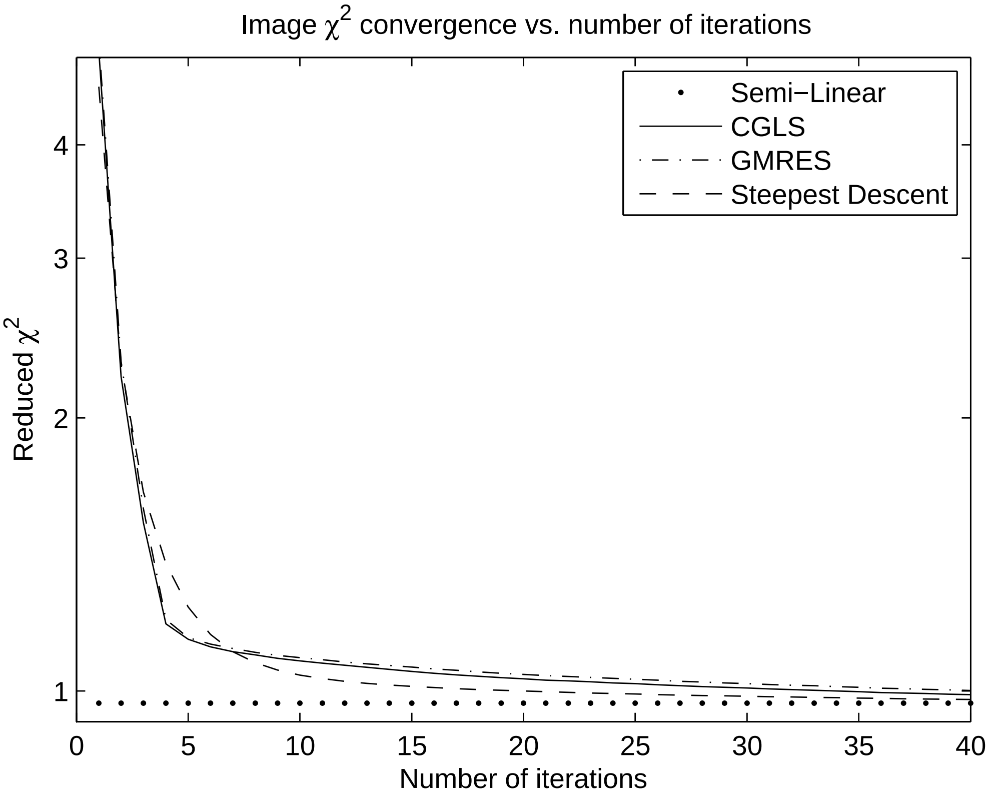

Figure 3.2: Convergence Properties of Iterative Optimizers - Image Model

Convergence properties of several local optimization routines. The CGLS and LSQR (Björck 1996) algorithms exhibit identical behavior, and the performance of the GMRES (Saad & Schultz 1986) algorithm is similar. The steepest descent algorithm converges more slowly but attains a slightly lower image χ^2 value. The difference between the local optimization routines and the semilinear method is emphasized on this plot due to the logarithmic scale. The semilinear method result was obtained using a zeroth order regularization constant $\lambda = 2.5 \times 10^{-3}$, and iterative algorithms were terminated after 40 iterations.

in iterative methods. Techniques have been developed to deal with this problem using Bayesian analysis for the semilinear method, as discussed by Brewer & Lewis (2006) and Suyu et al. (2006). For iterative methods, the issue of a stopping criteria is a non-trivial problem that has no unique solution for local optimization, although many methods exist to deal with this problem, such as Generalized Cross Validation (Wahba et al. 1979) and the L-curve criterion (Engl et al. 2000). We discuss a novel approach in Section 3.1.5 that uses the L-curve analysis to estimate the optimal regularization parameter (stopping condition) in conjunction with global parameter search methods.

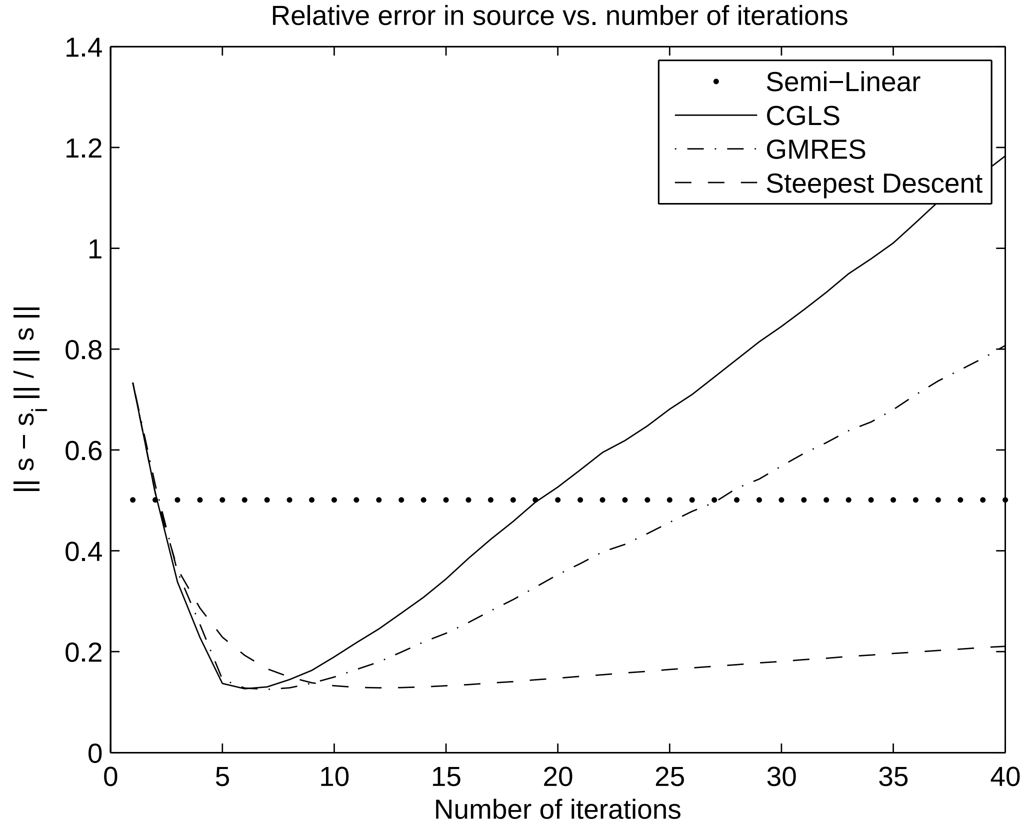

Figure 3.3: Convergence Properties of Iterative Optimizers - Source Model

Convergence properties of the source intensity distribution. This figure plots the relative error between the source found by a given method as a function of iteration $\boldsymbol{s}_i$ and the true source intensity distribution $\boldsymbol{s}$. All of the iterative optimization algorithms display semi-convergence behavior. The semilinear method result was obtained using a zeroth-order regularization constant $\lambda = 2.5 \times 10^{-3}$, and iterative algorithms were terminated after 40 iterations. The relative error of the solution found by the SD algorithm increases more slowly past convergence than the error for any of the other iterative schemes shown here.

3.1.3 Iterative Optimization as Implicit Regularization

To see how iterative schemes produce implicit regularization, consider a system $\boldsymbol{b}$=$\boldsymbol{Bx}$+$\boldsymbol{n}$, where $\boldsymbol{n}$ describes the noise added to the true image. Suppose that the blurring matrix $\boldsymbol{B}$ is ill-posed (Hansen 1997), and the "true" solution is the unblurred image, represented as a vector $\boldsymbol{x}$. Given the blurring matrix and the noisy data $\boldsymbol{b}$, we can formally write an approximate solution to the inverse problem as $\boldsymbol{x}$=$\boldsymbol{B}^{-1}\boldsymbol{b}$. However, this proves to be difficult in practice because of the poor conditioning of $\boldsymbol{B}$ and the noise contained in the data. The resulting solution is the sum of two terms, $\boldsymbol{x}_n$ =$\boldsymbol{x}$+$\boldsymbol{B}^{-1}\boldsymbol{n}$. The second term can dominate the first in this expression, which results in poor recovery of the true solution, $\boldsymbol{x}$. To overcome this problem, regularization schemes seek a solution to the system

$$\boldsymbol{x}_\lambda = argmin \left(||\boldsymbol{b} - \boldsymbol{Bx}||^2 + \lambda ||\boldsymbol{H}\left(\boldsymbol{x} - \boldsymbol{x}_0\right)||^2 \right) \tag{3.13}$$

where $\boldsymbol{H}$ is the regularization matrix, $\boldsymbol{B}$ is the system matrix and b is a vector of data to be fit. The regularized solution is $\boldsymbol{x}_\lambda$, and the default solution is $\boldsymbol{x}_0$, which is found when $\lambda \to \infty$. By requiring the derivative of Equation 3.13 to vanish we derive the following expression

$$\left(\boldsymbol{B}^T\boldsymbol{B} + \lambda \boldsymbol{H}^T\boldsymbol{H}\right)\boldsymbol{x} = \lambda \boldsymbol{H}^T\boldsymbol{H}\boldsymbol{x}_0 + \boldsymbol{B}^T\boldsymbol{b} \tag{3.14}$$

provided that the regularization is linear in nature. Note that the first term on the right depends on the default solution $\boldsymbol{x}_0$, which represents a bias in general. For the remainder of this report we set the default solution to zero, which is reasonable since most astronomical images are largely composed of pixels corresponding to blank sky (Brewer & Lewis 2006).

The solution to this system can also be found by considering the singular value decomposition (SVD) of a matrix $\boldsymbol{B}$=$\boldsymbol{U\Sigma V}^T$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal $N \times N$ matrices (Golub and Reinsch 1970). The matrix $\boldsymbol{\Sigma}$ is diagonal, containing the non-increasing singular values $\nu_1 \geq \nu_2 \geq ... \geq \nu_n$. The columns of $\boldsymbol{U}$ are a set of orthogonal vectors $\boldsymbol{u}_i$, and the orthogonal columns of $\boldsymbol{V}$ are denoted by $\boldsymbol{v}_i$, which leads us to the expression

$$\boldsymbol{x} = \sum_{i=1}^{N} \frac{\boldsymbol{u}_i^T\boldsymbol{b}}{\nu_i}\boldsymbol{v}_i. \tag{3.15}$$

The small singular values (large values of i) correspond to the addition of high frequency noise, and the terms involving the smallest singular values ν_i tend to dominate the solution. The singular values ν_i and the expansion coefficients $|\boldsymbol{u}_i^T\boldsymbol{b}|$ as a function of the number of terms are shown in Figure 3.4. These plots are called Picard plots and show that an increased contribution to the noise in the reconstruction is found as the singular values become smaller than the magnitude of the expansion coefficients.

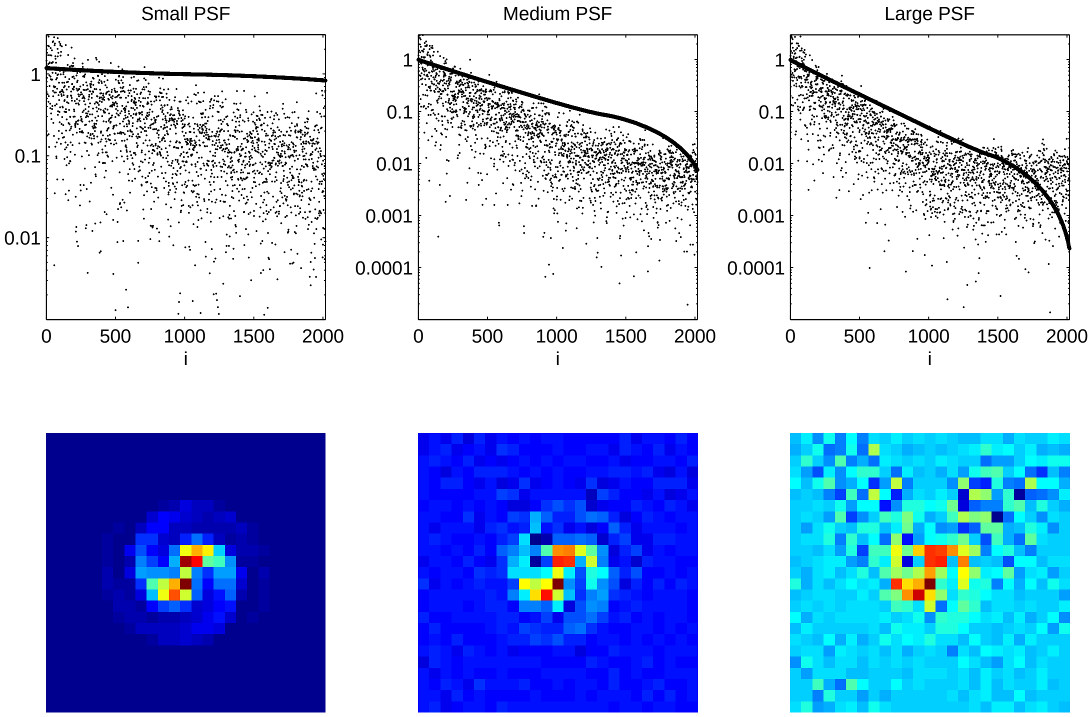

Figure 3.4: Picard Plot for Gaussian Point Spread Functions

Picard plot for Gaussian PSFs with full width at half-maximum of 0.94, 1.64, and 2.35 pixels, respectively. Top row: The points on the black curve are the singular values ν_i and the small dots are the expansion coefficients $|u_i^T b|$. As the PSF becomes increasingly large, the singular values drop below the expansion coefficients more quickly. Bottom row: The drop-off of the singular values signifies an increased contribution of high frequency noise in the recovered solutions, shown for each PSF.

The goal of a regularization scheme is to limit the amount of noise that contributes to the solution. In principle, the simplest scheme is to truncate Equation 3.15 for sufficiently large values of i to limit the amount of high frequency noise in the solution. Truncation of the SVD expansion can be accomplished by multiplying the terms of Equation 3.15 by a "filter factor" ϕ_i (Vogel 1989) defined by

$$\boldsymbol{x}_{filt} = \sum_{i=1}^{N} \phi_i \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\nu_i} \boldsymbol{v}_i, \tag{3.16}$$

where ϕ_i takes the form of a Heaviside function such that $\phi_i = 1$ for singular values below the cutoff point k, and $\phi_i = 0$ for terms with $i > k$. In this way, the contribution of high-frequency noise to the solution can be controlled. However, this regularization scheme is somewhat artificial because of the sharp cut off in the filter factors. A more natural scheme was developed by Tikhonov (1963), which introduces a regularization parameter λ to solve

$$\left(\boldsymbol{B}^T \boldsymbol{B} + \lambda \boldsymbol{I}\right) \boldsymbol{x} = \boldsymbol{B}^T \boldsymbol{b}. \tag{3.17}$$

The Tikhonov solution for $\boldsymbol{x}$ is expressed as the standard SVD expansion with modified filter factors

$$\phi_i = \frac{\nu_i^2}{\nu_i^2 + \lambda}. \tag{3.18}$$

The solution of this system corresponds to the solution of Equation 3.13 with the regularization matrix equal to the identity and the prior solution $\boldsymbol{x}_0 = \boldsymbol{0}$.

When $\nu_i \gg \lambda$, $\phi_i \approx 1$. For large i, $\nu_i \ll \lambda$ such that $\phi_i \approx \nu_i^2/\lambda$. Note that the eigenvalues of the $N_s \times N_s$ system matrix $\boldsymbol{B}^T\boldsymbol{B}$ are the squares of the singular values, $\mu_i = \nu_i^2$. The sum of the Tikhonov filter factors is then

$$\gamma = \sum_{i=1}^{N_s} \frac{\mu_i}{\mu_i + \lambda}. \tag{3.19}$$

This expression agrees with Equation 21 in Suyu et al. (2006), who show that the sum γ represents the number of source degrees of freedom in the problem when Tikhonov regularization is included.

Iterative methods effectively add consecutive terms to the sum in Equation 3.15 with each step of the algorithm, such that the number of iterations itself acts as a regularization parameter (Hanke 1995). In order to find the best solution from an iterative optimizer, it is necessary to stop it near the optimal iteration, before the contributions due to noise in the solution grow too large. As can be seen in Figure 3.3, the CGLS algorithm converges significantly faster than the SD method. However, the noise also rises more quickly past convergence, which makes the CGLS solution more sensitive to the stopping condition. The SD method is generally considered a slower and less sophisticated local optimization algorithm than CGLS, but performance issues are outweighed by SD's more stable behavior past convergence. Nagy & Palmer (2003) first noted that optimization schemes based on SD do not require as precise a stopping criterion as other methods, which makes it easier to find an approximation to the optimally regularized solution. In the absence of the L-curve criteria, we choose SD. When using a stopping condition based on the L-curve, CGLS is recommended due to the algorithms speed in obtaining better solutions.

3.1.4 Monte Carlo Estimate of the Effective Degrees of Freedom

The iterative optimizers we have considered in this paper can be expressed in terms of the SVD expansion, Equation 3.16, with unique expressions for the filter factors ϕ_i. As in the case of Tikhonov regularization, we associate the sum of these filter factors γ with the number of effective degrees of freedom in the problem (Vogel 1987). For the case of the CGLS algorithm, these filter factors are recursive in the singular values (Hansen 1997). This poses a problem because we use the CGLS scheme without explicitly building the matrices, and the solution of the singular values presents difficulty when using large data sets. Furthermore, the recursive scheme can become unstable (Hansen 1994). To circumvent this problem, we use a Monte Carlo scheme to estimate the sum of the filter factors. In essence, this scheme introduces a Gaussian random vector $\hat{\boldsymbol{b}}$ with zero mean and unit standard deviation that contains the same number of elements as the data vector $\boldsymbol{b}$. While iteratively solving for the solution vector x using the conjugate gradient method, we simultaneously solve a second system with noise vector $\hat{\boldsymbol{b}}$ using the same CGLS coefficients ($\bar{\alpha_k}$ and $\bar{\beta_k}$ in standard notation) to derive a corresponding vector $\hat{\boldsymbol{x}}$. Hanke and Hansen (1993) and Girard (1989) show that $\hat{\boldsymbol{b}}^T(\hat{\boldsymbol{b}}-\boldsymbol{A}\hat{\boldsymbol{x}})$ provides an estimate of the number of degrees of freedom in the original system with data vector $\boldsymbol{b}$. Note, however, that this estimate approximately doubles the computational overhead of the standard CGLS method. We perform this calculation during each call to the CGLS algorithm, allowing an estimate of the reduced χ^2 for each set of lens parameters.

3.1.5 L-curve Analysis

The iterative optimization algorithms used in the local optimization step (the inner loop of our optimization scheme) converge to lower spatial frequencies faster than higher frequencies, and therefore the high-frequency noise present in the source reconstructions can be suppressed by controlling the number of iterations of the local optimizer. In general, we wish to find a balance between the image χ^2 and the amount of source regularization (Press et al. 2007). Since the regularizing effect of iterative optimizers is implicit, we need a metric to evaluate the amount of regularization introduced at each iteration. For simplicity, we use zeroth-order regularization (Warren & Dye (2003); Suyu et al. (2006)) in this paper which sums the squares of source pixel intensities, in order to quantify the regularizing effects of the local optimizers. By calculating the image χ^2 and regularization measure, $\sum_i^{N_s} s_i^2$ at each iteration of the local optimizer, we can form an L-curve (Hansen and O'Leary 1993) for each solution. In the standard image deblurring problem, the point associated with the "corner" of the L-curve represents the solution that best balances the image fitness and the amount of regularization introduced in modeling the source. This solution is found by determining the point on the trade-off curve with maximum curvature. We parameterize the L-curve by arclength $(x(s), y(s))$, where x and y represent the regularization term and image χ^2, respectively, and fit a cubic spline curve to x and y. The derivatives of the cubic spline curves with respect to the arclength can be calculated analytically, which provide a simple method to calculate the curvature κ.

The point of maximum curvature is found using the curvature formula:

$$\kappa = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{\frac{3}{2}}}. \tag{3.20}$$

We show a sample of L-curves in Figure 3.5 and corresponding source solutions in Figure 3.6, including the solution located at the point of maximum curvature. In general, the solution found by the L-curve analysis agrees with the maximum Bayesian evidence solution to approximately 10%. The solution corresponding to the point of maximum curvature of the L-curve is used to evaluate the fitness of each set of lens parameters.

3.1.6 A Large-scale Test

We form the gravitationally lensed image of a large source to demonstrate the efficiency of our iterative matrix-free approach. The source is a square image of $M51$, of dimension 512 × 512, shown in Figure 3.7, obtained from the NED online data archive (Kennicut et al. 2003). The lensed image is generated using an SIE lens defined on a 640 × 640 grid, with Einstein radius b=3 arcsec, $\epsilon = 0.4$, and $\theta_L = \pi/4$, with the lens centered at the origin. To demonstrate the behavior of the code with a complicated PSF, we used a PSF composed of a radial sinc function multiplied by an elliptical Moffat PSF (Moffat 1969), which is shown in the figure. The resulting function provides a non-symmetric PSF that contains significant low-level structure. Such a large PSF would require a very large non-sparse blurring matrix, whose linear size must necessarily match the number of image pixels which map to the source, in this case 3.34×10^5 square. After adding Gaussian white noise, the peak S/N of the blurred observation is $S/N = 20$. The solution shown in the figure was computed by the CGLS method and has a reduced image $\chi^2_r = 0.9954$ and was found in 35 iterations that took 86.4 seconds using a single 2.4 GHz Intel processor.

The next section discusses the global optimizers, Ferret and Locust, which we use to solve for the lens model parameters. Both are parallel codes that require approximately $10^4 - 10^5$ lens parameter sets to be evaluated for a thorough search, optimization, and mapping of the parameter space. Assuming 5×10^4 evaluations,

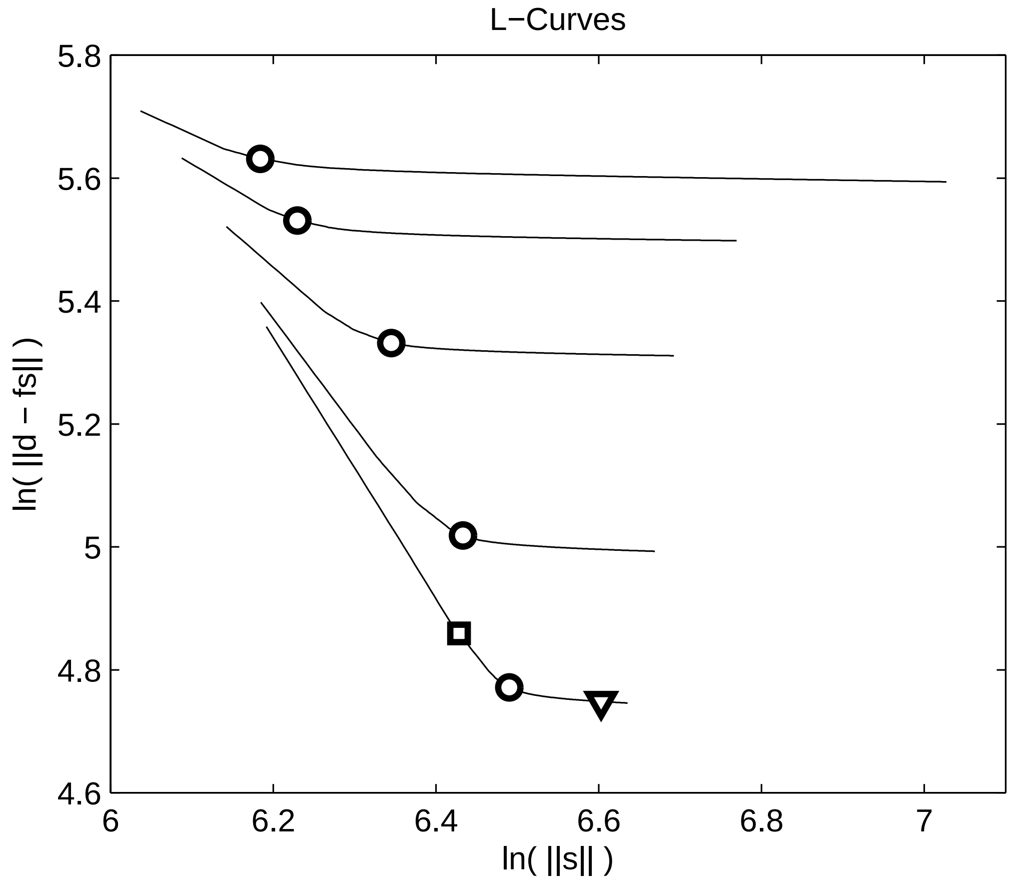

Figure 3.5: Examples of L-curves for Gravitational Lens Systems

Variety of L-curves for the system shown in Figure 3.1. The curve marked with the square, circle and triangle is the true solution, with the parameters described in Section 3.1.2. Each successive curve has the same parameters as the true solution except for the velocity dispersion, which takes the values 260, 262.5, 265, 267.5 and 270 km s^{-1}, respectively. The location of the optimally regularized solution balances the residual and solution norms, denoted by the corner of the curve and marked by a circle. The optimally regularized solution for the true set of lens parameters has reduced $\chi^2 = 0.998$, $||s|| = 658.4$, found after 7 CGLS iterations.

the lens parameters could be solved for this large-scale test problem in approximately six days on an eight-core computer. Such a large-scale problem would be impractical using a matrix inversion scheme due to the large size of the matrices involved.

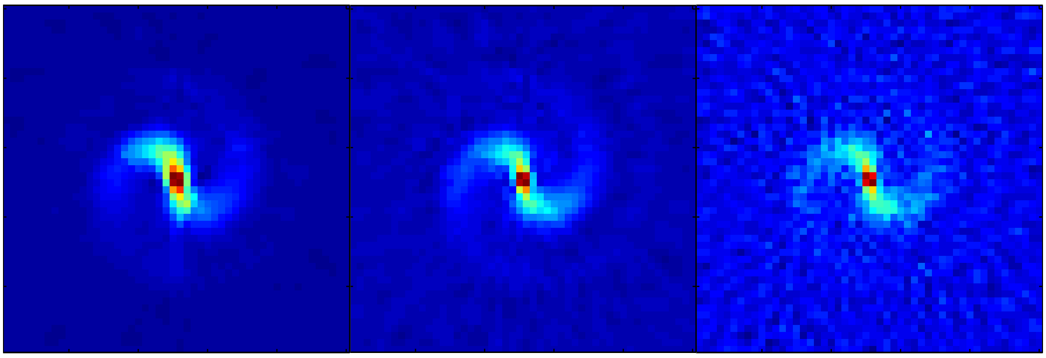

Figure 3.6: Regularization Effects on the Source Intensity Distribution

Sources corresponding to the solutions marked in Figure 3.5. Left: oversmoothed solution (square), 3 CGLS steps, reduced $\chi_r^2 = 1.228$, $||s_i|| = 613.2$. The middle panel shows the optimally regularized solution in Figure 3.5 (circle) after 7 CGLS steps, reduced $\chi_r^2 = 0.998$, $||s_i|| = 658.4$. The panel on the right shows the solution (triangle) after 18 CGLS steps, reduced $\chi_r^2 = 0.965$, $||s_i|| = 740.4$.

3.2 The Full Optimization Problem

Section 3.1 focused mainly on the linear least squares reconstruction of the source, for a known lens mass distribution. However, the full problem must also determine the optimal set of lens parameters. The lens parameters are solved as an 'outer loop' optimization problem, which calls the semilinear method, or alternatively our iterative approach, as an inner loop optimization for each set of nonlinear lens parameters evaluated. The inner loop optimizes the lensed source by executing an arbitrary number of iterative steps (in our examples, 40) of a local optimizer like the CGLS algorithm. The L-curve for each lens parameter set is built, and the optimally regularized solution that lies nearest the corner of this curve is found. The χ^2 value of this optimally regularized solution is used to evaluate the fitness of the corresponding set of lens parameters. During the inner loop optimization, a statistical estimate of the number of degrees of freedom for the optimally regularized solution is made and used to determine the reduced image χ^2 during the analysis at the end of the run. In this paper, the outer loop problem is solved by the Ferret GA and Locust PSO from the Qubist Global Optimization Toolbox (Fiege 2010). However, the Mirage code is not limited to either of these optimizers and can make use of any external nonlinear optimization scheme.

Both Ferret and Locust are able to map out "fuzzy" optimal sets defined by an inequality. In this case, we request a distribution of solutions with $\chi^2 \leq \chi^2_{min} + N_u$, where χ^2_{min} is the lowest image χ^2 value found and N_u is an upper limit selected at the start of the run. The upper limit N_u is chosen to be large enough so as to include solutions within the 99% confidence interval. The members of the optimal set, along with the estimates for the number of degrees of freedom, allow us to determine solutions within the 99%, 95% and 68% confidence intervals by the standard method (Press et al. 2007). Thus, we can easily estimate errors for the nonlinear lens parameters, since these global optimizers determine the form of the χ^2 surface in the neighborhood of the global minimum.

The source intensities may contain negative values since bounds cannot be imposed in direct matrix inversion, and are not enforced in our iterative schemes. A final

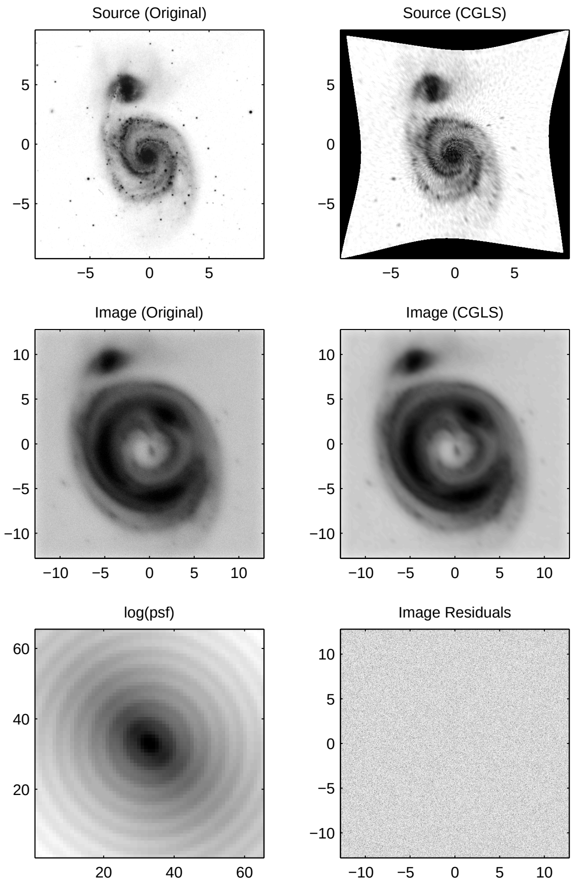

Figure 3.7: A Large Scale Test of Mirage

Large scale test of Mirage. Top left: original image of $M51$ used to generate a large-scale test problem by forming a lensed image, blurred by the PSF and including additive noise. Top right: model source obtained with the CGLS algorithm after 40 iterations. The original image was obtained from NED, originally 700 × 700, cropped to 512 × 512 pixels. The effect of the lens mapping can be seen as the source plane is not completely covered by the back-traced image. Middle left: lensed image of $M51$ as seen through an SIE lens used as artificial data. The image is comprised of 640×640 pixels, over an area of 25.5 arcsec^2. Middle right: model image of $M51$ as produced by the CGLS algorithm after 40 iterations. Bottom left: PSF used to blur the observation, shown in logarithmic intensity to highlight the low-level structure. The function is a 65 × 65 pixel PSF. Bottom right: residuals obtained from comparing original and model images. The residuals are featureless and have a maximum 10^{-3} of the original image maximum. The reconstruction has a reduced image $\chi_r^2 = 0.9954$.

source refinement step, discussed in Section 3.2.4, uses the GA and PSO as bounded optimizers to find the optimal positive definite source distribution, with the lens parameters held fixed at their previously optimized values.

3.2.1 Global Nonlinear Optimization

The Qubist Global Optimization Toolbox contains five global optimizers in total, all of which are designed to be interchangeable. Ferret and Locust are the most powerful and well-tested optimizers in the package, which makes them well-suited for our problem. Qubist includes more than 50 test problems, some of which are discussed in its user's guide (Fiege 2010).

GAs and PSOs differ greatly from local optimization routines such as CGLS and SD, which require an initial guess and then search iteratively along a deterministic trajectory through the parameter space. Such methods are prone to becoming trapped in local minima. Moreover, these methods are usually implemented to solve unbounded optimization problems, which may be less useful than bounded optimization when there are physical constraints on the parameters, such as the positivity of source pixel values in the lens reconstruction problem (see Section 3.1).

GAs and PSOs search the parameter space in parallel, making use of the collective behavior of numerous interacting "agents" - a population of individuals for a GA or a swarm of particles in the case of a PSO. These optimizers distribute agents randomly throughout the parameter space initially, which subsequently interact using heuristic rules that aim to search the space thoroughly, and encourage the improvement of the population or swarm as a whole. In both types of algorithm, these heuristic rules are partly deterministic and partly stochastic. The resulting optimization algorithms are more powerful and robust than purely deterministic methods and vastly more efficient than random search. In general, only a single agent must find the high-performance region in the vicinity of the true global solution for the algorithm to succeed. Once such a solution is discovered, it is rapidly communicated to all other individuals or particles, which will accumulate near the global minimum and refine it.

3.2.2 Genetic Algorithms

GAs are an important class of algorithms for global optimization that work in analogy to biological evolution. Evolution is biology's optimization strategy of choice, in which organisms evolve and continually improve their own designs as they struggle to survive. GAs are normally discussed using biological terminology, such that each "individual" is a trial solution, whose parameters are encoded on "genes". The set of individuals is a "population", and individuals search the parameter space in parallel as they evolve over multiple "generations". A basic GA requires three genetic operators, which are mutation, crossover, and selection (Goldberg 1989). The role of mutation is to apply occasional random perturbations to individuals, which helps them to explore new regions of the parameter space. Crossover mixes together two parent solutions to produce offspring that are intermediate between the parent solutions. The role of the selection operator is to choose which solutions propagate to the next generation, based on the Darwinian notion of survival of the fittest. Various types of selection operators are possible, but tournament selection has the advantage that it is insensitive to the scaling of the fitness function (Goldberg 2002).

Ferret is a parallel, multi-objective GA, which has been under constant development since 2002, and is the most sophisticated optimizer in the Qubist package. The current version is the fourth major version of the code, and earlier versions were used by Fiege et al. (2004) to model magnetized filamentary molecular clouds, and by Fiege (2005) to model submillimeter polarization patterns of magnetized molecular cloud cores. Ferret extends the basic GA paradigm in several important ways, as discussed below.

Multi-objective optimizers like Ferret emphasize the thorough exploration of parameter spaces and the ability to map trade-off surfaces between multiple objective functions, which allows the user to understand the compromises that must be made between several conflicting objectives. A core feature of a multi-objective GA is the ability to spread solutions approximately evenly over an extended optimal set of solutions, which Ferret accomplishes using a niching algorithm similar to the one discussed by Fonseca and Fleming (1993). Even for single-objective problems, Ferret's multi-objective machinery is well-suited to explore and map out χ^2 intervals in the neighborhood of the global minimum. We see in Section 3.3 that it is especially useful for degenerate cases where multiple disconnected islands of solutions exist within the parameter space.

Ferret's most novel and powerful feature is its 'linkage-learning' algorithm (Goldberg 2002), which is designed to reduce a complex, multi-parameter problem to a natural set of smaller sub-problems, whenever such a reduction is possible. These simpler sub-problems are discovered experimentally by Ferret during the process of optimization, and sub-problems evolve almost independently during a run. Ferret regards two parameters A and B as linked if finite variations of A and B are discovered, which result in worsening of a solution when applied independently, but the same variations applied together result in improvement. In such cases, it is clear that A and B should be linked so that they are usually traded together during crossovers, to preserve gains made by varying the parameters together. A novel extension of Ferret's linkage-learning algorithm is its ability to search entire sets of parameters $\{A_i\}$ and $\{B_i\}$ for linkage in parallel, which is assigned probabilistically to the parameters within these sets. Thus, Ferret treats linkage as a matrix of probabilities that co-evolves with the population during the search. Parameters that appear linked at the start of a run may not appear linked at the end, when most solutions may be nearly optimal. Conversely, new links can also arise as the code explores previously uncharted regions of parameter space.

The ability to partition a complicated problem into natural sub-problems is crucial to the successful optimization of large problems. A difficult 100 parameter problem with many local minima is often unsolvable on its own, but it becomes quite tractable if it can be partitioned into (say) 10 sub-problems (or building blocks) with 10 parameters each. A particularly interesting feature of Ferret's linkage-learning system is that the linkages discovered are entirely insensitive to scale. Two sub-problems (building blocks) that are orders of magnitude different in importance are discovered at the same rate, so that Ferret can solve all of the sub-problems correctly and simultaneously, rather than one at a time in order of significance. This ability allows Ferret to discover the true, globally optimal solution or solution set, even when applied to problems with very poorly scaled building blocks.

Ferret contains an algorithm that monitors its progress and uses this information to automatically adapt several of its most important control parameters, including the mutation scale, size scale of crossover events, and several others. If these parameters are set poorly by the user, Ferret quickly and dynamically adapts them to improve the search. This algorithm provides an extra layer of robustness to the code, which helps Ferret to adapt as different regions of the fitness landscape are discovered.

Ferret, and the other global optimizers of the Qubist toolbox, place considerable emphasis on visualization. The analysis window displayed at the end of a run contains various graphics options to tease out interesting features from the optimal set. These features include two and three-dimensional scatter plots, image plots, contour plots, and user-defined graphics. It is possible to 'paint' interesting regions of the parameter space and select different two and three-dimensional projections to explore and visualize where the painted solutions reside in a high-dimensional parameter space.

Modeling a gravitational lens system is a computationally intensive task that requires approximately $10^4 - 10^5$ parameter sets to be evaluated for a single run. GAs are well suited to parallel computing because each individual in the population represents a single parameter set, which can be evaluated independently. Ferret is designed with built-in parallelization to take advantage of multi-CPU computers and inexpensive clusters. Parallel jobs are managed with a graphical "node manager" tool, and no changes are required to the implementation of the user's fitness function. It is notable that Ferret does not require MATLAB's parallel computing toolbox, or use any other third-party parallel computing software. Appendix D discusses some additional details of the Ferret algorithm.

3.2.3 Particle Swarm Optimizers

Locust is a parallel multi-objective PSO in the Qubist toolbox. PSOs are biologically inspired global optimizers, which search the parameter space using a swarm of interacting particles. PSOs are often discussed in terms of the dynamics of flocks of birds, schools of fish, or swarms of social insects searching for food. The commonality is that intelligent search behavior emerges as a property of the system as a whole, even if the component parts are modeled as relatively simple automata that interact with each other through simple rules. Kennedy & Eberhart (2001) provides a good introduction to the PSO technique.

PSOs are similar to GAs in that they sample many points in the search space simultaneously, with a swarm of particles moving through the parameter space following simple dynamical equations. Each particle in a simple PSO is simultaneously attracted to its own "personal best" solution, which is the best solution that the particle has personally discovered, and the "global best" solution, which is the best solution that the entire swarm has ever encountered. The law of attraction follows a simple spring law: $F \propto |\Delta \mathbf{x}|$, where $|\Delta \mathbf{x}|$ is the distance between a given particle and either the personal best solution $\mathbf{x}_p$ or the global best $\mathbf{x}_g$. Assuming that the force and velocity are approximately constant over a time step, the new velocity and position of particle i after a time step Δt are given by

$$\begin{aligned} \mathbf{v}_i(t+\Delta t) &= \mathbf{v}_i(t)\,(1-\Delta t/t_{damp}) + \\ &\quad [c_p\xi_p(\mathbf{x}_p-\mathbf{x}_i)+c_g\xi_g(\mathbf{x}_g-\mathbf{x}_i)]\,\Delta t \\ \mathbf{x}_i(t+\Delta t) &= \mathbf{x}_i(t)+\mathbf{v}_i(t)\Delta t, \end{aligned} \tag{3.21}$$

where c_p and c_g play the role of spring constants for the personal and global best solutions respectively. The equations include a damping term to decrease the velocity magnitude in approximately time t_{damp}, which helps the swarm settle down as it zeros in on the optimal region. Damping also serves to prevent runaway growth in so-called 'particle explosions', which can occur as a result of accumulated errors in Equation 3.21. Some randomness is added via the uniform random variables ξ_p and ξ_g, which are typically drawn from the range [0,1]. The stochastic terms play a role similar to the mutation operator in a GA; they add randomness to the search, which helps the particles to explore previously unexplored parts of the parameter space. The roles of the personal and global best solutions are clear. The personal best solution represents a particle's memory of the best region of parameter space that it has seen, and the global best solution represents the entire swarm's collective memory. In effect, the global best solution allows indirect communication between particles to encourage collective behavior.

Particle swarm optimization is a young and rapidly changing field of research that still has many open questions, which are discussed in a recent review by Poli et al. (2007). Equation 3.21 is perhaps the simplest set of swarm equations, but many alternative implementations are possible, which strive to balance thorough exploration of the parameter space against the need to exploit high performance regions when they are found.

Equation 3.21 is equivalent to a simple Euler integration scheme for a dynamical system of equations that move each particle every time step. However, Locust uses an exact solution to the swarm equations, which is easily obtained by solving Equation 3.21 analytically, in the limit $\Delta t \to 0$. Numerical experiments with Locust, and alternate schemes that use Euler integration, show that the exact solution results in a more stable and reliable PSO (Fiege 2010). It is possible that the exact solution eliminates the build-up of errors in the orbits, which would result from applying Equation 3.21 directly with a finite Δt. The exact solution is slightly more costly to evaluate than the Euler approximation, but this extra computational expense is insignificant for any realistic problem, where the computational time is normally dominated by the evaluation of the fitness function.

Determining $\mathbf{x}_p$ is straightforward because it represents the personal best solution (often denoted *pbest*) that any particle has encountered. Thus, each particle simply keeps track of the position where it encounters the lowest value of the fitness function $F(\mathbf{x})$, following Ferret's convention that lower values of F correspond to more desirable solutions.

The most common particle swarm implementation is the simple PSO discussed above, where the global best solution $\mathbf{x}_g$ (*gbest*) is evaluated over the entire swarm. This swarm topology can be thought of as a fully connected graph, where each particle in the swarm communicates with every other particle via the *gbest* solution. Other swarm topologies are possible, where the network of communication between swarm members is less densely connected, so that each particle only communicates with a few other particles in its neighborhood. In this case, the *gbest* solution is replaced by a set of local best, or *lbest* solutions, such that each *lbest* solution is assigned to a subset of the swarm. This scenario is referred to as a static *lbest* topology when the network connecting particles do not change throughout the run. Dynamic *lbest* topologies are also possible, where the network co-evolves with the swarm as the run progresses. Swarms based on sparsely connected networks can be thought of as being divided into sub-swarms, where each sub-swarm shares a common *lbest* solution. Such a topology is better able to avoid local minima because the sub-swarms explore the space in parallel. On the other hand, the fully connected *gbest* topology is best for exploiting a single isolated solution late in a run, because it focuses the efforts of the entire swarm on the region of parameter space in the vicinity of the *gbest* solution.

Locust requires some non-standard techniques designed to thoroughly explore parameter spaces containing sets of solutions that are equally good. Extended solution sets are also possible when a fuzzy tolerance is specified for a single objective problem, which often represents the χ^2 error tolerance of a data-modeling problem. Locust emphasizes the mapping of spatially extended solution sets, and therefore it makes sense to define particle neighborhoods dynamically, based on their spatial location within the swarm. The code keeps track of the Euclidean distances between all particles, and assigns neighborhoods based on the nearest *lbest* particle. Moreover, Locust implements a novel algorithm that allows neighborhoods, and hence sub-swarms, to merge and divide as required to map out the structure of the optimal set. This dynamic swarm topology is quite different from other topologies discussed in the literature, and has the benefit that it essentially self-optimizes. A large number of neighborhoods will generally be preserved to map a spatially extended solution set, but the swarm topology will correctly collapse to a single neighborhood late in a run if only a single solution exists, thus reducing the algorithm to a simple *gbest* approach. In practice, this technique represents a good balance between exploration of the parameter space and exploitation of the optimal set; the parallel action of many sub-swarms evade local minima early in the run for all problems, and many are retained to the end when the focus is on mapping an extended solution set, but swarms reduce to the maximally exploiting *gbest* algorithm late in the run for problems where only a single best solution exists.

Locust uses the same visualization system as Ferret. It uses a simpler setup file than Ferret, but it can read Ferret's setup files and translate them. Moreover, the formats for the initialization, fitness, and custom graphics functions are identical. This makes it easy to swap optimizers for comparison purposes. Appendix D discusses additional details about Locust.

3.2.4 Source Refinement Routine

We use a two-step process to solve the full inversion problem. In the first step, we determine the nonlinear lens parameters as described in Section 3.2. In the second refinement step, we hold the best set of lens parameters constant and allow the global optimizer to fit the source brightness distribution. We treat each source plane pixel as a free parameter and judge the fitness of solutions based on the image χ_r^2 statistic. This type of pixelized source fitting using a GA was outlined by Brewer & Lewis (2005). The Qubist global optimization routines are bounded, so positivity conditions on the source reconstruction are easily enforced in this step. Since the intensity of each source pixel is independent, this approach does not produce a regularizing effect and the number of degrees of freedom in the problem is well defined, allowing direct comparisons of lens models. Therefore, this two-step method allows an estimation of the errors on both the lens and source intensity parameters.

The bounds used in the refinement step can significantly speed up this optimization. Figure 3.8 shows a sequence of solutions with a lower bound of 0 and an upper bound equal to 1.1 times the maximum pixel intensity in the source. These bounds ensure that the source is strictly positive but can significantly slow the optimization due to the large volume of the parameter space that is searched. Both of the global optimizers used in this report can include a user-defined solution in the first generation. Therefore, a more practical optimization strategy is to consider the absolute value of the optimally regularized solution found by the iterative optimization process, and define a "window" of acceptable pixel intensity values for each source pixel. In our tests, a tolerance of ±25% of the pixel intensities is usually sufficient to bracket the true intensities. Pixels with negative intensities in the optimally regularized solution should always have a lower bound of 0 to prevent artifacts in the source solution. The upper bound of these pixels is taken to be the absolute value of the pixel intensity plus 15%. Practically, this reduces the volume parameter space to be searched and generally allows a solution to be found more quickly.

3.3 Demonstrations of the Full Optimization Problem

In this section, we show results from several illuminating test problems that solve the full lens reconstruction problem and characterize the behavior, performance, and limitations of the global optimizers.

3.3.1 Trivial Solutions and the Problem of Dimensionality

Consider a lens model based on a singular isothermal sphere, which provides a simple analytical model with a circularly symmetric deflection angle given by Equation 3.12 in the radial direction. This deflection angle is used to form the synthetic data with velocity dispersion $\sigma_v = 500$ km s^{-1}, centered at the origin $(x, y) = (0, 0)$. We construct artificial data where the Einstein ring has radius $b = 1$ arcsec, assuming source redshift $z_d = 0.2$, deflector redshift $z_s = 1.5$. For convenience, we again measure angular distances with respect to the Friedmann metric with $k = 0$. The lensed image is defined on a 120×120 rectangular mesh with an image pixel size of 0.015 arcsec. A 3×3 subsampling per pixel is used to construct the lensed image. The source is perfectly aligned with the lens center and forms a full Einstein ring due to the symmetry of the mass distribution. We have blurred the image using a Gaussian PSF with an FWHM of 2.35 image pixels defined on a 33×33 grid. The test source is also a Gaussian model on a 50×50 square mesh from -3 to 3 arcsec in both directions.

In the following discussion, we hold the x and y coordinates of the lens center constant, using the actual values from the artificial data, and plot χ_r^2 as a function of b in Figure 3.9. As the size of the Einstein radius (velocity dispersion) is varied, the corresponding χ_r^2 statistic becomes double peaked, with the true solution between the

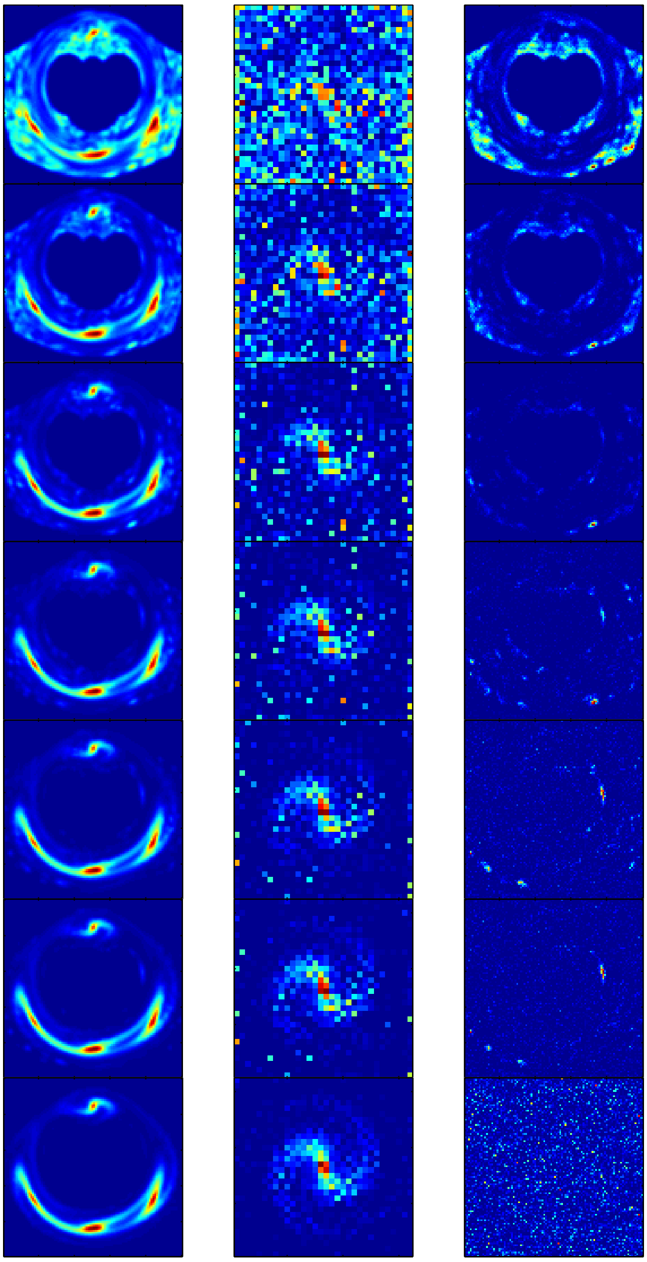

Figure 3.8: Genetic Algorithm Optimized Source Intensity Distribution

The lowest χ_r^2 solution at 100, 250, 500, 750, 1000, 1250, 5000 generations. Left column: model image. Middle column: source brightness distribution. Note the presence of reconstructed noise. Right column: image residuals. At 5000 generations, a model image with $\chi_r^2 = 1.05$ was found. Each image is independently scaled to highlight image features.

peaks. The area to the left of the peaks, the region of low b, contains trivial solutions that map the source almost straight through the lens, reproducing the image almost exactly with minimal distortion. In fact, the $b = 0$ source does not include any gravitational lens effect at all, thus reducing the problem to a conventional image deconvolution exercise. Note that this trivial solution results in reduced $\chi_r^2 = 0.973$, even though the reconstructed source is physically unrealistic. The χ^2 surface varies smoothly as we approach the 'true' value of b with $\chi_r^2 = 0.986$, and increases with b beyond this value. When b is large, we again begin to see a decrease in χ_r^2, to the asymptotic value of $\chi_r^2 = 1.15$, as the structure of the source becomes increasingly complex to compensate for the distortion introduced by the lens. Typically such high b solutions give rise to spurious images in which some pixels lie outside the boundaries of the image plane. In general, these images will be missed by the ray-tracing operation, and will thus not contribute to the χ^2, providing the apparently low vales of the fit statistic at high b. As the lensing effect grows sufficiently strong, these spurious images occupy a smaller portion of the image plane, and the χ^2 decreases correspondingly. This is a shortcoming of ray-tracing methods due to the fact that we are ray-tracing over a finite image plane when the lens parameters are not restricted. In general, we wish to avoid the very low and very high b solutions, since they do not correspond to physical solutions of the problem.

With the lens center fixed, the above example is a simple one parameter problem, which can be easily solved by a global optimization routine designed to map a range of χ^2 values near the global minimum. However, analogous examples may exist in more complicated systems with more parameters, where the parameter space can become dominated by trivial solutions. The problem becomes especially difficult when false solutions, such as the trivial ones in Figure 3.9, occupy a region of space whose dimensionality is greater than the true solutions. In such cases, GAs and PSOs can fail when the number of search agents is too small for the problem, since the entire population or swarm may be drawn into the region of trivial solutions and never discover the class of true solutions that occupy a region of lower dimensionality. Even if the high-performance region containing the true solution is discovered, both Ferret and Locust are designed to spread solutions evenly over the optimal region, which

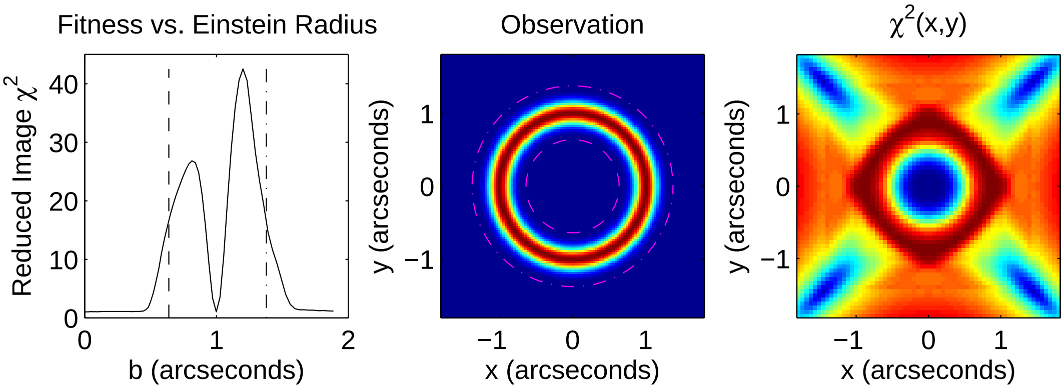

Figure 3.9: Image Fitness Landscape of the Singular Isothermal Sphere

Left: fitness as a function of Einstein radius for a symmetric lensed image produced by a background Gaussian source intensity distribution. The Einstein radius was varied manually with the lens center fixed at the origin. The dashed line indicates the lower limit used to model the system, and the dash-dotted line is the upper limit used to restrict the value of the Einstein radius. The region between these two limits is the region in which the true solution is located. Middle: Einstein radius limits are shown superimposed on the artificial data. Right: fitness as a function of lens center. The lens center position was varied over a 64×64 grid and the lens normalization fixed at the true value, $b = 1$ arcsec. Trivial solutions populate the corners of the image. The true solution lies at the center of the image.

contains the trivial solutions if the goal is to map the solution set within $\Delta\chi^2$ of the χ^2 minimum. Thus, the population or swarm may become diluted by spreading out over the trivial region, which has higher dimensionality. The right panel of Figure 3.9 shows the results of keeping the Einstein radius b fixed at its true value and varying the lens center (x, y). The true solution point, with $\chi^2_r = 0.986$, is surrounded by a ring of poor solutions, which signifies multiply imaged solutions. In this projection, trivial solutions occupy a two-dimensional plane at large radius and have $\chi^2_r = 0.978$. In order to overcome the complication of trivial solutions, an estimate of the range of acceptable parameter values is made. By imposing such parameter restrictions the algorithm is guaranteed to find a non-trivial solution to the optimization problem. Notably, Ferret also implements a novel algorithm that promotes the speciation of the population into isolated clusters, which may help to overcome this difficulty.

3.3.2 A Realistic Test

For a more realistic and complicated test, consider the SIE lens model presented in Section 3.1.2. We use the same parameters to solve a test system, with the source intensity profile as given by Equation 3.10. We fix the redshift of the deflector and source as in the previous example, and model the parameters of the lens density model using both Ferret and Locust. The fitness objective to be minimized is the standard χ^2. The parameters of the best solutions are summarized in Table 3.1. Both algorithms automatically map the region of parameter space near the minimum by heavily populating this region of parameter space. The effective number of degrees of freedom for each model is estimated and saved during the course of the run. By using these quantities we are able to estimate confidence intervals and the errors of the lens parameters. The structure of the global χ^2 surface is calculated at the end of the run using the members of the optimal set, saved from each generation (Ferret) or time step (Locust). Figure 3.10 shows that Ferret more thoroughly explores the parameter space than the Locust algorithm.

We find that the GA and PSO converge approximately at the same rate. Figure 3.11 compares the performance of the algorithms by plotting the fitness of the best solution as a function of the number of function evaluations, while Figure 3.10 shows the distribution of solutions in the parameter space. Figure 3.10 shows that Ferret

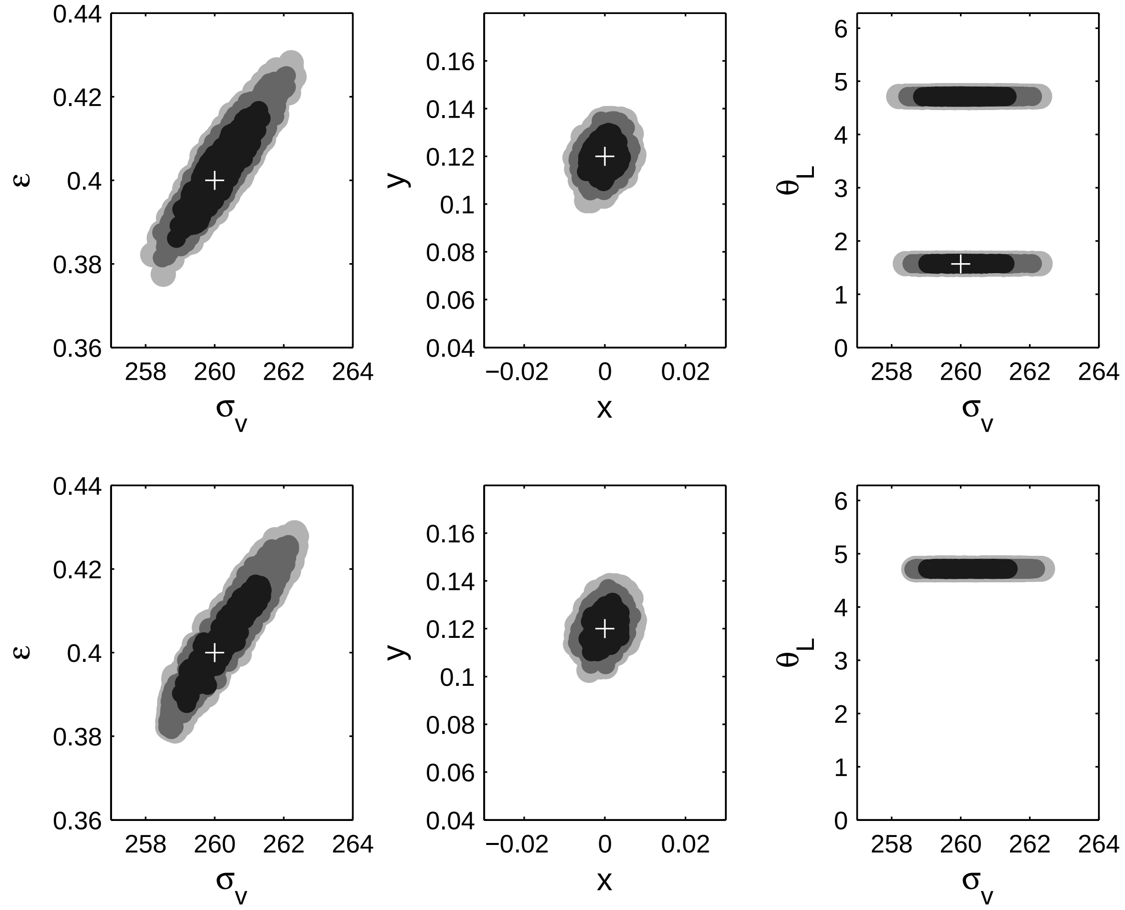

Figure 3.10: Parameter Space Plots of a Test Function

Parameter space plots of the SIE example in Section 3.3.2. The optimal set of solutions determined by global optimization marked by points shaded according to position within the 99%, 95% and 68% confidence intervals, represented by light gray, medium gray and black respectively. The location of the true solution is marked with a white cross. Top row: the Ferret GA optimal set. Bottom row: the Locust PSO optimal set. Left column: ellipticity ϵ vs. velocity dispersion σ_v, middle column: lens centre coordinates y vs. x, and right column: orientation angle θ_L vs. σ_v. The PSO does not explore the structure of the parameter space as thoroughly as the GA. Note the rightmost column in which the orientation angle degeneracy of the system is detected by the Ferret GA but no corresponding solution group is present in the Locust PSO optimal set of solutions.

correctly discovers a pair of equally good degenerate solutions symmetric in orientation angle, but Locust picks out only one of these groups, which reflects Ferret's

Solution	Ferret GA	Locust PSO	True	Lower	Upper
χ_r^2	1.010	1.012	0.998	-	-
σ_v	260.002	259.892	260.000	250.000	280.000
ϵ	0.401	0.399	0.400	0.200	0.500
x	-2.101×10^{-4}	-2.123×10^{-4}	0.000	-0.500	0.500
y	0.119	0.120	0.120	-0.500	0.500
θ_L	4.713	4.713	$\pi/2$	0.000	2π

Table 3.1: Comparison of the Ferret GA and Locust PSO on a Test Problem

Optimal lens parameters found by the Qubist optimizers for the example given in the text using the SIE lens. We restrict the range of the lens parameters to prevent convergence to a trivial solution. Performance of the GA and PSO routines is similar as can be seen from the reduced image χ^2.

greater emphasis on mapping the parameter space. Baran (2009) used these same optimizers to estimate the system temperature of the DRAO synthesis array and noted that Locust found solutions significantly faster on average. We do not find the same behavior of the PSO in this problem.

3.3.3 Source Refinement

Once we have determined the lens parameters, we hold them constant and begin the final source refinement step of the optimization, which involves 2500 parameters for the case shown. The source refinement results in an optimal non-negative source intensity distribution, as discussed in Section 3.2.4. The image is of size 120 × 120, while the source plane is defined as a 50 × 50 grid. The solutions at the beginning of this step appear to be comprised purely of noise, but an approximation to the true solution becomes increasingly well defined as the run progresses, and the image residuals gradually become featureless. Figure 3.8 shows an evolutionary sequence of the lowest χ_r^2 solution, where the final solution has $\chi_r^2 = 1.05$. The search is a bounded linear problem, which is mathematically simpler than the nonlinear search for lens parameters. However, the large number of parameters complicates the optimization and the GA converges in a few thousand generations. Source refinement is the most computationally expensive part of this problem, requiring approximately five days on

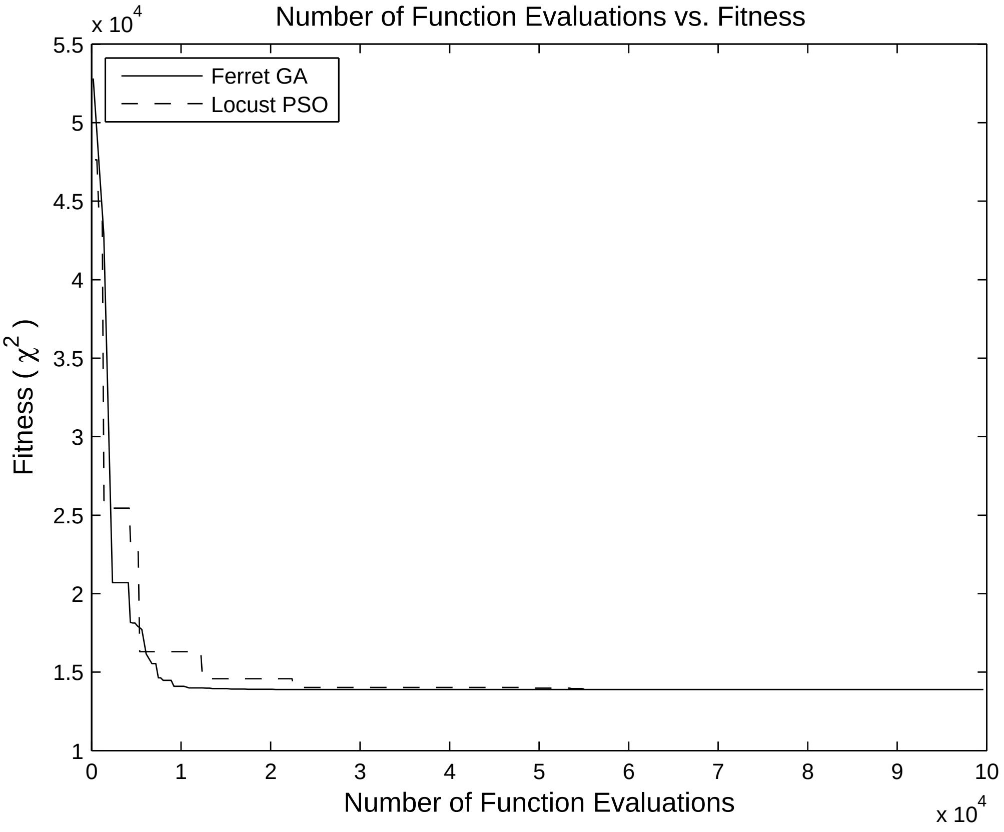

Figure 3.11: Convergence History of the Ferret GA and Locust PSO

Average convergence history of the Ferret GA (solid line) and the Locust PSO (dashed line) as a function of the number of function evaluations for the test described in Figure 3.10. The convergence of the GA is more stable, as the PSO tends to converge in a series of steps as the parameter space is explored. The figure is plotted over 1.0×10^5 function evaluations. We have averaged over four runs of the PSO and four runs of the GA.

an eight core machine. The most useful aspect of this intensive search is to estimate errors on the source plane pixels determined by the best fit lens model.

Ferret's convergence on the source refinement problem is shown in Figure 3.12. The smooth convergence curve is a hallmark of linear or other easy problems. We have noted that the source reconstructions begin fitting to noise in the target image slowly, so it is generally quite easy to find an acceptable termination criteria for the algorithm. Since each pixel in the source is independently treated by the GA, this problem cannot be expressed in terms of the SVD expansion in the same way that the solutions to a linear optimization step can. However, to quantitatively ensure that overfitting to noise is prevented, we once again form the L-curve between the image χ^2 and a linear regularization measure $\sum s_i^2$ to quantify the amount of noise in the source. In practice the L-curve analysis in the final analysis step is of limited use due to the smooth convergence of the algorithm and the slow rise in noise in the reconstructed sources. Ferret is able to converge to a source near the location of the true solution for all situations that we have tested. The best solution typically agrees with the true source to within 15% though we have noticed variation in the details of the derived sources from run to run, which is expected considering the large number of parameters involved in the optimization. It is interesting, and perhaps surprising, that the Locust PSO is unable to solve this problem, despite its linearity. We conclude that a GA is a more robust and efficient approach than particle swarm optimization for both of these optimization problems. When the problem is small, a PSO can often find the solution in a comparative amount of time as a GA.

3.4 Conclusions

The semilinear method provides an elegant way to describe gravitational lens inversion in terms of a least squares problem, but is limited to relatively small images and a narrow PSF. This is due to the fact that the semilinear method requires the inversion of a large matrix whose size increases as the fourth power of the number of source pixels, and the sparsity of this matrix is reduced as larger PSFs are used. Solving for lens parameters is a nonlinear optimization problem, which can be solved by global optimization techniques. We applied and compared the Ferret GA and Locust PSO to determine the nonlinear parameters of the lens model. The global optimization of lens parameters requires a lens inversion for each set of lens parameters tested, and $10^4 - 10^5$ such evaluations are required for a thorough exploration of the parameter space and mapping of the optimal region. This reinforces the need for fast lens inversion techniques that scale well with the size of the image and PSF.

We addressed the need for a fast lens inversion algorithm by developing a matrix-

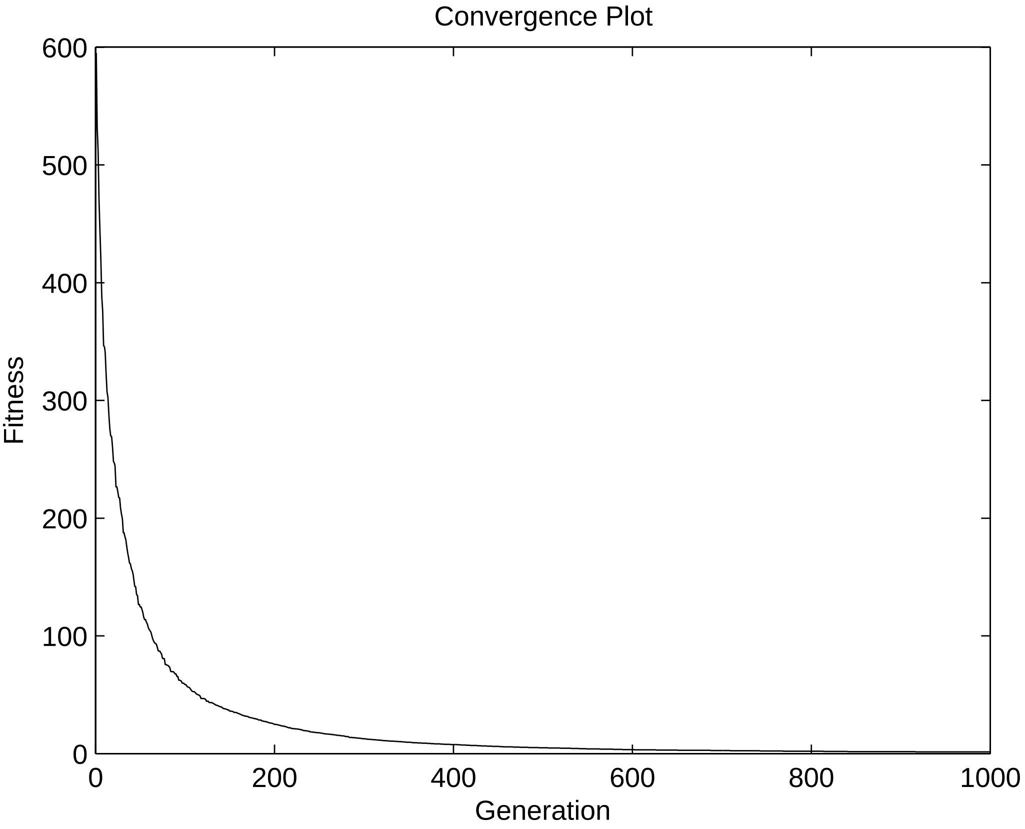

Figure 3.12: Converence History of Ferret in Source Intensity Optimization
Convergence history of the linear parameters during source refinement stage using the Ferret genetic algorithm.

free approach to solve the least squares lensing problem, based partly on recent developments in the image deblurring literature, which solves the problem without the need to explicitly build the lens or blurring operators. This novel approach is intended to complement the semilinear method when speed is of the essence, or when large images and broad, highly structured PSFs are used. We note that our approach can be extended to the case of a spatially variant PSF. Our analysis evaluated the convergence behavior of a matrix-free method using several local optimization methods. We found that the CGLS method is fastest to converge, but all linear optimization schemes suffer from over-fitting of noise if the optimization is not stopped at the critical iteration, which cannot be predicted *a priori.* We showed that steepest descent methods are more robust against over-fitting to noise at the expense of the speed of convergence. This work extends the applicability of the semilinear method and represents a unique and significant contribution to strong gravitational lens modeling.

The number of degrees of freedom in the iterative optimization step is estimated using a Monte Carlo method, allowing us to draw connections to the work of Suyu et al. (2006) that estimate the number of degrees of freedom using Bayesian statistics. We derived a formula for the number of degrees of freedom based on the filter factors of the Tikhonov regularization problem, which agrees with the expression found by Suyu et al. (2006) using Bayesian analysis.

We developed a novel method that computes the optimally regularized solution for each set of lens parameters by finding the point of maximum curvature in the trade-off curve between χ^2 and a measure of the amount of regularization in the solution, which we took to be the sum of the squares of source pixel intensities. The ambiguity of choosing a regularization parameter or stopping criteria is removed, because we automatically determine the optimal number of iterations (regularization constant) using the L-curve. We evaluate the fitness of lens parameter sets using the image χ^2 statistic.

The convergence and parameter space mapping properties of the Ferret GA and the Locust PSO schemes were compared, and we determined that the GA explores the parameter space more thoroughly than the PSO. The GA obtained a more detailed optimal set of solutions, highlighting the degeneracy in the position angle of a Singular Isothermal Elliptical lens model due to the rotational symmetry of the lens. Both methods converge at a similar rate.

As a final refinement step in the image reconstruction our approach uses the GA or PSO to directly solve for pixel intensities. This addition has the important benefit that the non-negativity of the source intensity profile can be enforced. It is notable that the Ferret GA was able to solve this bounded linear solution refinement problem, but the Locust PSO failed due to the high dimensionality of the search ($\sim$2500 parameters). This analysis step shows stable convergence, and noise is introduced to the source very slowly. In practice this routine is relatively insensitive to stopping criteria.

This work serves as a foundation for future explorations, which will apply the techniques discussed here to data, and expand them to include non-parametric lens models, such as those used by Vegetti & Koopmans (2009) and Saha et al. (2007). Just as the intensity distribution of the source can be described by a pixelated model, a similar approach can be used to describe the lens density distribution (Diego et al. (2005); Liesenborgs et al. (2007); Suyu & Blandford (2006)). In fact, the use of GAs to optimize pixelated lens density distributions has been previously investigated in the literature (Liesenborgs et al. (2006); Liesenborgs et al. (2007); Liesenborgs et al. (2009)). Non-parametric lens density models are extremely valuable, since dark matter haloes may contain significant substructure (Koopmans 2005) that is not taken into account by smooth analytical lens models. This added flexibility reduces the bias that is introduced by assuming a specific analytical form for the lens. Vegetti & Koopmans (2009) used smooth analytical lenses and added a pixelated perturbing potential to the models to simulate more realistic lenses. In general, the gross morphology of gravitational lens images are recovered well using smooth analytical functions to describe the lens density, but more realistic descriptions of the lens density distributions are expected to produce models with smoother, increasingly featureless residuals. Systems of lensed quasars have also been modeled using pixelated mass maps (Saha & Williams (1997); Saha et al. (2006)). The Java code PixeLens (Read 2003) has been used to model variable lensed quasars and obtain estimates of the Hubble constant H_0.

For the remainder of this thesis we make use of analytical lens mass distributions with our lensing code due to the intuitive nature of the parameters in these lenses and the speed of evaluating such lens models. However, pixelated lens models could be used in conjunction with the semilinear method to model complicated lens density distributions and reveal the details of cosmologically distant extended sources.

Chapter 4

Spatially Variant Point Spread Functions

One of the main computational problems in developing models of gravitational lens systems using the semilinear method is the deconvolution step to account for the PSF. In Section 3 we discussed several schemes to include this deconvolution step. The least squares form of the semilinear method treats the source deconvolution as a linear problem. In the previous sections, we showed that appropriate subroutines can be used in place of matrix operations to compute the lensed and blurred images, and that this approach can be extended to SD and CGLS optimization schemes while retaining the linear form of the problem. This allows us to avoid the explicit construction of these large matrices. Though we have only considered galaxy-galaxy lensing to this point, our code is applicable to strong lensing phenomena on all scales, including cluster lenses. Our code therefore complements the semilinear method, which handles small scale problems well. Furthermore, we noted that the matrix-free approach would lend itself to including the effects of complicated and possibly spatially varying PSFs.

Image deconvolution problems are usually handled by approximating the PSF as constant over the entire image. This is true for the basic semilinear scheme as well. However, the PSFs of a variety of astronomical instruments are position dependent, such that point sources at various locations in the image will be blurred by different amounts. A great deal of consideration has gone into dealing with this issue, and a particularly elegant method to deal with spatially variant PSFs is based on the work of Nagy et al. (2002).

The matrix-free method of gravitational lensing can accomodate the spatially variant nature of the PSF for large-scale modeling applications. In the next section, we will show how this can be done using an implementation of the Nagy et al. (2002) approach. Although small scale images such as those of galaxy lenses should not be affected much by spatially variant effects, these effects may be seen in large scale images of giant arcs in galaxy clusters. This section develops a technique to incorporate spatially variant PSFs in our version of the semilinear method. We test our code using artificial data constructed by distorting a portion of the Hubble Deep Field using an elliptical potential model, which is representative of a galaxy cluster lens. Despite the large size of the problem and complicated variation of the PSF over the sample data, we are able to recover the details of the source distribution accurately. The following Sections are based on the text of Rogers & Fiege (2011b), originally published in The Astrophysical Journal.

4.1 Introduction

Spatial dependence of the PSF is not considered in most conventional deconvolution problems. This simplifies the construction of the blurring matrix $\boldsymbol{B}$, since only one PSF is taken into account. However, it is well known that the PSF cannot always be treated as constant over an image in cases of astronomical interest. For example, spatially variant PSFs have been studied in the context of adaptive optics (Lauer (2002); Gilles et al. (2002)) and the PSF of astronomical instruments, such as the Hubble Advanced Camera for Surveys (ACS), can be extremely position dependent (Bandara et al. 2009). Several schemes have been designed to deal with this variability (Boden et al. (1995); Biretta (1994); Adorf (1994); Lauer (2002)). Describing a spatially variant PSF is much more complicated than for the invariant case, since each row of the blurring matrix $\boldsymbol{B}$ will be derived from a unique PSF in general. The position of a pixel in the image determines the amount by which it is blurred.

We illustrate the effect of spatially dependent PSFs on gravitationally lensed images in Figure 4.1. Consider the lensing effect produced by a singular isothermal sphere (SIS), which has three parameters: velocity dispersion σ_v and lens center (x, y). The deflection angle due to a SIS lens has a simple analytical form most conveniently described in standard polar coordinates (r, ω):

$$\alpha\left(r\right) = 4\pi \left(\frac{\sigma_v}{c}\right)^2 \frac{D_{ds}}{D_s}, \tag{4.1}$$

where D_{ds} and D_s are the angular distances between lens and source and observer and source respectively, and c is the speed of light. We model the blurring in Figure 4.1 with $\sigma_v = 265$ km s^{-1} and use source redshift $z_s = 1.5$ and lens plane redshift $z_l = 0.12$. This model was calculated using cosmological parameters $H_0 = 70$ km s^{-1} Mpc^{-1}, $\Omega_0 = 0.3$ and $\Lambda_0 = 0.7$ which we adopt for the remainder of this study. The source is comprised of a set of circular disks in the source plane as shown in the left-hand panel of Figure 4.1 and the gravitationally lensed image is shown in the center panel. The lensed image is then blurred by a spatially variant PSF and is shown in the right panel of Figure 4.1. The distortion used to create this image varies from a delta function in the lower-left corner (negligible blur) to a Gaussian with standard deviation $\sigma_g = 6.0$ pixels in the upper right corner. Each PSF is defined on an arbitrary 33×33 grid and is normalized to unity sum. The source and image plane size are 240×240 pixels.

Unlike constant PSFs, spatially variant PSFs cannot be described by a simple convolution operation. Fortunately, numerical methods have been devised to handle them, including sectioning methods (Trussel & Fogel 1992), which deconvolve each PSF independently and form the source from the sum of the reconstructions. Nagy & O'Leary (1998) devised a clever method to model the effects of spatially variant PSFs within the framework of the standard image deconvolution problem. This approach differs from sectioning methods in that the separate PSFs are used to build an approximation to the blurred image of a given source, and a single iterative deconvolution operation is needed to solve for the source intensity distribution. The method was implemented in Nagy et al. (2002) and represents the spatial dependence of the PSF as a summation of piecewise blurring matrices, each of which applies over

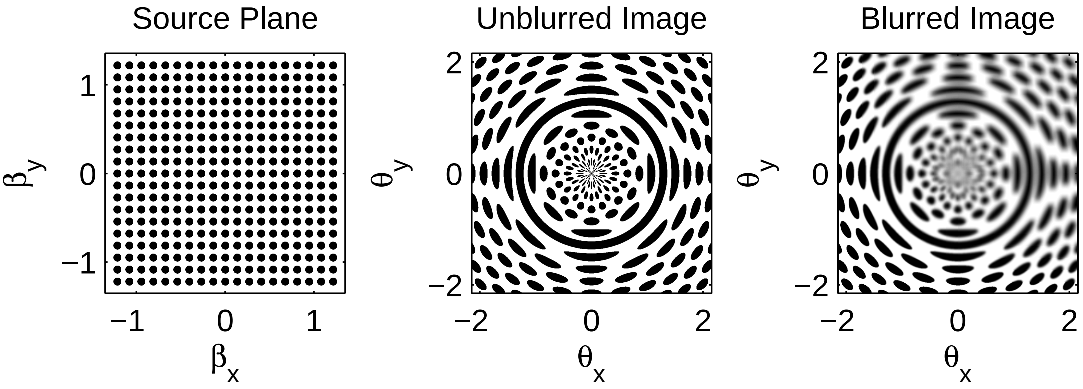

Figure 4.1: Example of Spatially Variant Blurring

Example of spatially variant blurring. Left: a set of regular disks with radius 0.268 tile the source plane. Center: the circular disks are seen under the lensing effect of a Singular Isothermal Sphere (SIS) lens model. The SIS distorts the background circles into arcs, and the disk at the center of the SIS becomes a complete ring. Right: the same disk pattern under the effect of the SIS lens, with a spatially variant PSF blurring the observation. The blur is described by a delta function in the lower left hand corner to a Gaussian with standard deviation $\sigma_g = 6.0$ pixels in the upper right corner, introducing a significant blur.

a limited area of the image. In this study, we use the method of Nagy et al. (2002) to incorporate spatially variant blurring into our gravitational lens modeling code. We briefly review the method here and discuss the numerical procedure in detail in Appendix E.

To include the effects of spatially variant blurs, the image of the unblurred lensed source is padded to enforce a boundary condition (Hansen et al. 2006). We focus on the use of reflexive boundary conditions, in which the image is padded by symmetric reflections of itself. Reflexive boundary conditions tend to reduce ringing artifacts if a significant amount of structure is located near the edges of the image. The image is then divided into a square grid, where the PSF is assumed constant in each region. These image regions and the PSFs are then padded to match in size. The two-dimensional fast Fourier transform (FFT) is used to calculate the resultant blurred image regions independently, resulting in an effective piecewise convolution. By substituting efficient algorithms for the explicit matrix and matrix-transpose multiplications in Equation 3.6, the least squares form of the problem is preserved and the system can be solved efficiently.

In principle spatially variant blurring can be described by a blurring matrix compatible with the semilinear method. However, in practice there are several problems with the matrix approach. First, the size of the blurring matrix is $N_{pix} \times N_{pix}$, so the matrix quickly becomes large as the image resolution is increased. Second, since the PSFs vary over regions of the image, it is possible that $\boldsymbol{B}$ may contain a large number of small but non-zero entries, particularly for large, complicated PSFs that are not well approximated by Gaussians or other simple analytical functions. This complicates the optimization because $\boldsymbol{M}{=}\boldsymbol{F}^T\boldsymbol{F}$ must be inverted in the semilinear scheme. It is generally required that $\boldsymbol{M}$ is sparse in order to store and invert this large matrix. The sparsity requirement helps to reduce computation time and reduces the amplification of noise in the reconstructed source. In practice the semilinear method requires regularization to control the amount of noise present in the solution of Equation 3.6. The details and effects of several distinct regularization methods used with the semilinear method were studied in detail by Suyu et al. (2006).

Our previous work (Rogers & Fiege (2011a); Chapter 3), compared the semilinear method with several iterative methods to solve the least-squares problem (Equation 3.6). Iterative schemes have the advantage that time is saved by avoiding the explicit construction of the lens and blurring matrices. This is done using direct interpolation on the source plane under the effect of the lens equation (1.8).

Rogers & Fiege (2011a) used the Qubist Optimization Toolbox (Fiege 2010) to map the χ^2 surface over the space of the nonlinear lens parameters using the Ferret Genetic Algorithm (GA) and Locust Particle Swarm Optimizer (PSO). Since this mapping requires a large number of function evaluations ($\approx 10^5$) over the course of a run, speed is of the essence when choosing an inner loop optimization to determine the source plane parameters.

Using the techniques introduced by Nagy et al. (2002) as a foundation, we have added the capability to include spatially variant PSFs to our gravitational lens modeling code using piecewise constant PSFs. This new capability is the subject of the current exploration.

4.2 A Small-Scale Test

In this section, we provide an example of modeling an extended source under the effects of a spatially variant PSF. We generate the lensed image of an analytical function that describes a spiral source intensity distribution, given by Equation 3.10. This artificial "galaxy", originally described by Bonnet (1995), serves as a convenient test pattern. To draw comparisons between our results for spatially invariant PSFs (Rogers & Fiege 2011a), we will again make use of a SIE lens with deflection angle components given by Equation 2.46. The parameters used in this test are velocity dispersion $\sigma_v = 265$ km s^{-1} with $z_d = 0.3$ and $z_s = 1.05$ giving an equivalent Einstein ring of $b = 1.32$ arcsec, ellipticity $\epsilon = 0.35$, lens center $(x, y) = (0.11, 0)$, core size s, and orientation angle $\theta_L = \pi/4$ measured counterclockwise from the right of the image. We set $s = 0$, resulting in a singular mass distribution.

We used the lens equation (1.8) to form the lensed image of the source (Equation 3.10) using the SIE deflection angle formulae. We generated a 20×20 grid of spatially variant Gaussian PSFs where each PSF is defined by a 33×33 pixel mesh and has a FWHM ranging from 2.35 to 4.8 pixels, shown in Figure 4.2. This grid of PSFs was used to blur the gravitationally lensed image and additive Gaussian white noise with standard deviation $\sigma_g = 1.05$ was added after the blurring operation, resulting in the artificial data shown in Figure 4.3. We define the peak signal-to-noise ratio (PSNR)

$$PSNR = \frac{I_{max}}{\sigma_g}, \tag{4.2}$$

giving $PSNR = 105.83$. To illustrate the effect of varying the number of PSFs, we model the data using smaller grids of 3 × 3, 5 × 5, 7 × 7, 10 × 10, and 20 × 20 PSFs. As shown in Figure 4.4, the best reconstruction with the lowest reduced χ^2 is obtained using a grid of 20 × 20 PSFs, which is the same number used to generate the data. This source and corresponding model image after 20 iterations are also shown in Figure 4.3. The 3 × 3 and 5 × 5 image residuals show significant structure, which is not present in the finer approximations. The residuals using a grid of 20 × 20 PSFs appear featureless. This demonstrates the improvement in image reconstruction as we include successively more information characterizing the blur. Note that we have used Gaussian white noise in this example for the purpose of testing the algorithm, though our method is not limited to this situation. In general, the specific details of the image noise is included in the solution of the semilinear least squares problem.

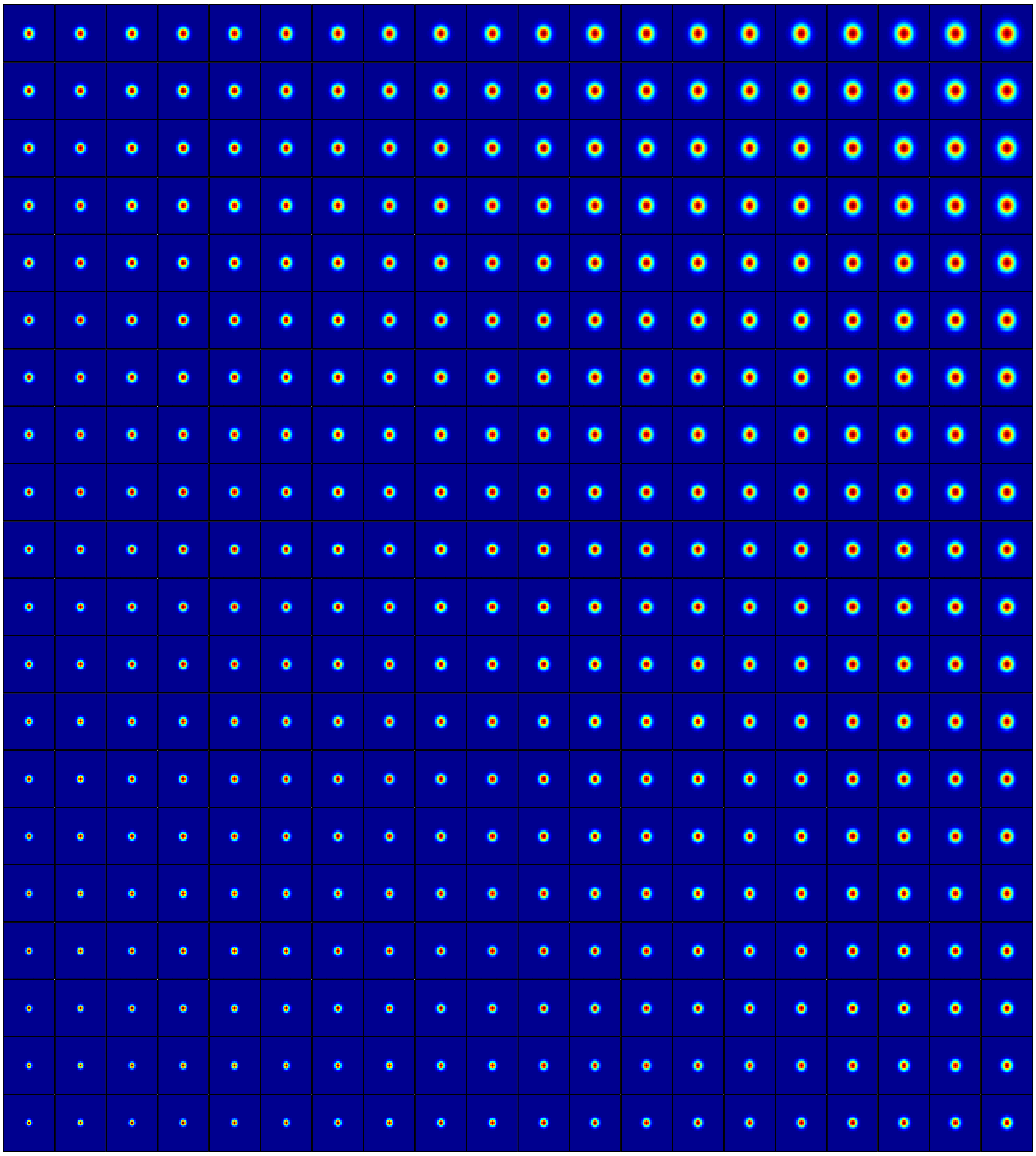

Figure 4.2: Grid of PSFs Used In Spatially Variant Blurring Example

Grid of PSFs used in Figure 4.3. The PSFs vary from a Gaussian of FWHM of 2.35 pixels in the lower-left corner producing a modest blur to a Gaussian with FWHM 4.75 pixels in the upper right corner.

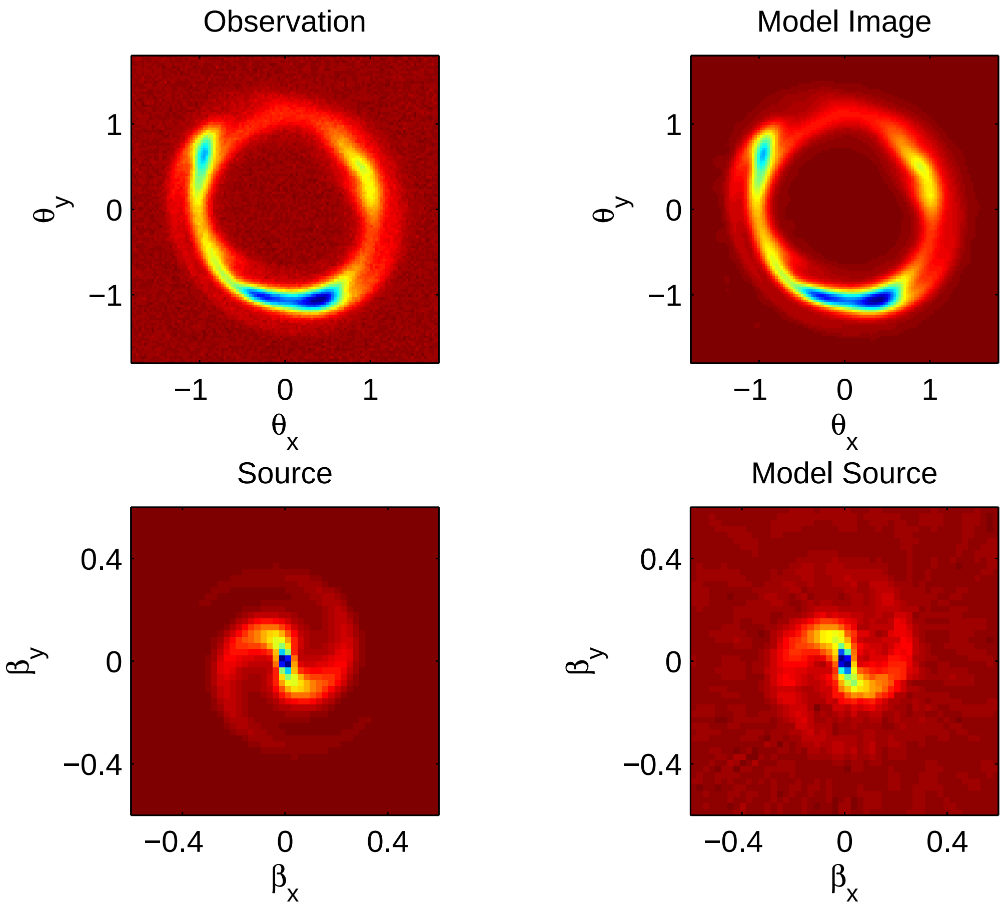

Figure 4.3: Small Scale Test with a Spatially Variant PSF

Top left: artificial data on a 120 × 120 grid. Bottom left: artificial source on a 50 × 50 grid. Top right: model observation. Bottom right: model source. The results after 20 iterations are shown. Note the presence of reconstructed noise in the source. The model has a reduced $\chi^2 = 0.998$.

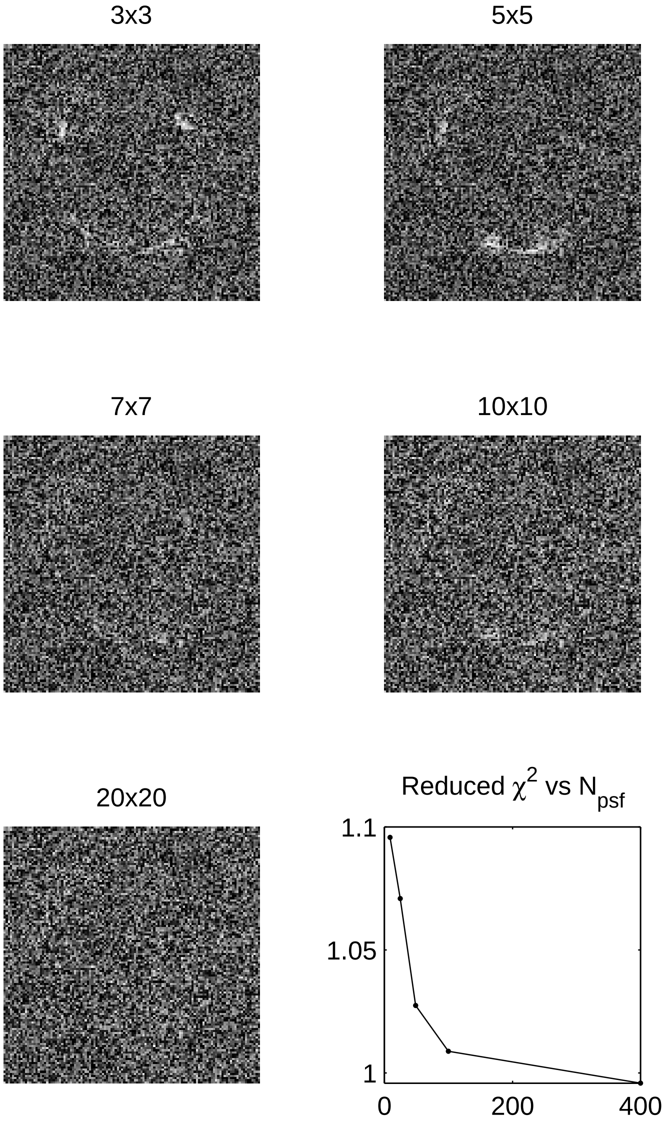

Figure 4.4: Image Residuals for A Variety of Tests

Image residuals for a 3×3, 5×5, 7×7, 10×10, and 20×20 PSF grids after 20 CGLS iterations. For a small number of PSFs there is a significant amount of residual structure, but these artifacts are reduced as the grid of PSFs is enlarged. The reduced χ^2 is shown as a function of the number of PSFs (N_{psf}) used in the inversion.

Figure 4.5 shows the relative error between the model source and the true solution as a function of iteration. For all PSF grid sizes, we find that the solutions display semi-convergence behavior such that the relative error between the model solution and the true solution improves until a minimum is reached and then begins to increase. This is due to the properties of the local optimizer used to determine the optimal source, and arises in the deconvolution step due to noise in the observed image. Regularization methods are generally used to control the increase of noise in the reconstructed source found by the semilinear method (Suyu et al. 2006). Several optimization methods have been applied to problems with spatially variant blur including Landweber iteration (Nocedal & Wright (1999); Fish et al. (1996); Trussel & Hunt (1978)), Richardson-Lucy deconvolution (Faisal et al. 1995), and Lanczos-Tikhonov hybrid methods (Chung et al. 2008) in the context of the standard image deconvolution problem. Following Rogers & Fiege (2011a), we focus on the iterartive CGLS and SD methods. Figure 4.6 shows the convergence history of the SD algorithm. As in the invariant PSF case discussed in Rogers & Fiege (2011a), the SD solution converges more slowly than CGLS and therefore it is less sensitive to the stopping criteria. When using an iterative method for local optimization, the number of iterations itself acts as a regularization parameter. The optimal stopping iteration of these local optimizers is at the minimum of the relative error curve for a given set of lens parameters and PSF tiling. This critical iteration represents a balance between the reduced image χ^2 and the amount of regularization used (Press et al. 2007). Established methods exist to determine this critical iteration, including the L-Curve criterion (Hansen and O'Leary 1993) and Generalized Cross Validation (Golub et al. 1979). In previous work (Rogers & Fiege 2011a) we made use of the L-Curve criterion but Generalized Cross Validation is also implemented in our software. These selection methods are explored further in Chapter 5.

We find that the execution time of the problem including a spatially variant PSF increases approximately linearly with the number of separate PSFs used in the inversion as shown in Figure 4.7. This suggests that significant gains could be made in the efficiency of the routine by parallelizing the implementation, since each image region is independent. By splitting up the problem over several processors, the runtime for

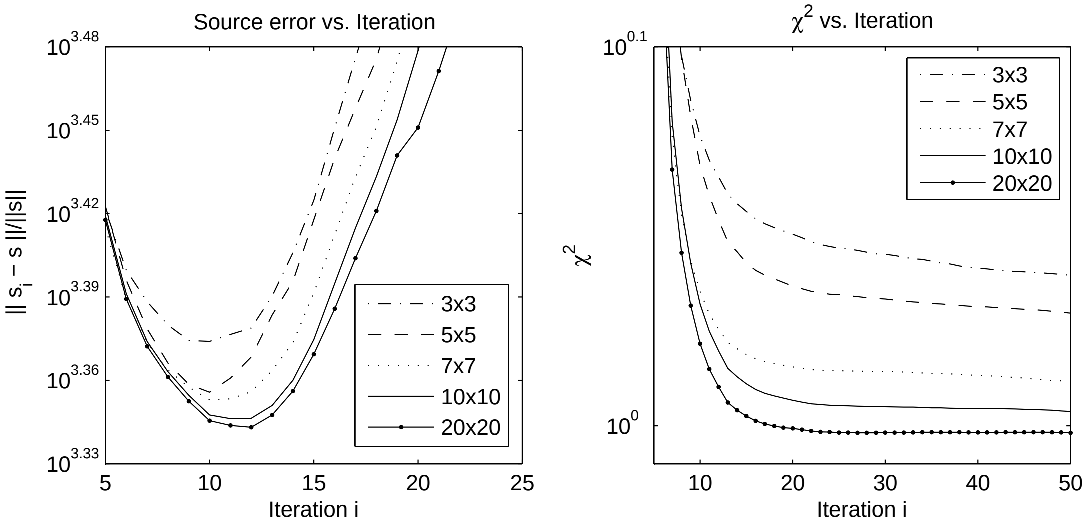

Figure 4.5: Source Convergence History with the CGLS Algorithm

Left: source convergence history using the CGLS algorithm. Right: corresponding Image convergence history. Note that the source displays semi-convergent behavior. The disagreement between model and actual source reaches a minimum before increasing. The critical iteration changes as the PSF grid is enlarged.

very large PSF grids can become feasible.

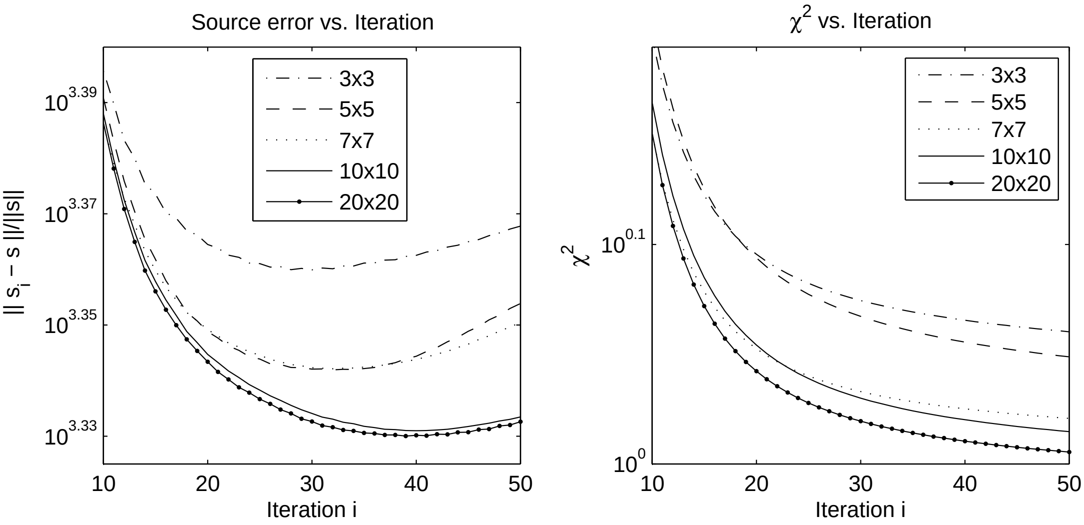

Figure 4.6: Source Convergence History with the Steepest Descent Algorithm
Left: source convergence history using the SD algorithm. Right: corresponding Image convergence history. The semi-convergent behavior of the source is less extreme than for the CGLS algorithm, shown in the left panel of Figure 4.5.

4.3 A Large-Scale Test

To demonstrate the code in operation on a large scale problem, we simulate the lensing effect of the mass distribution of a galaxy cluster on a portion of the Hubble deep field using an elliptical potential. This test is intended as a demonstration of

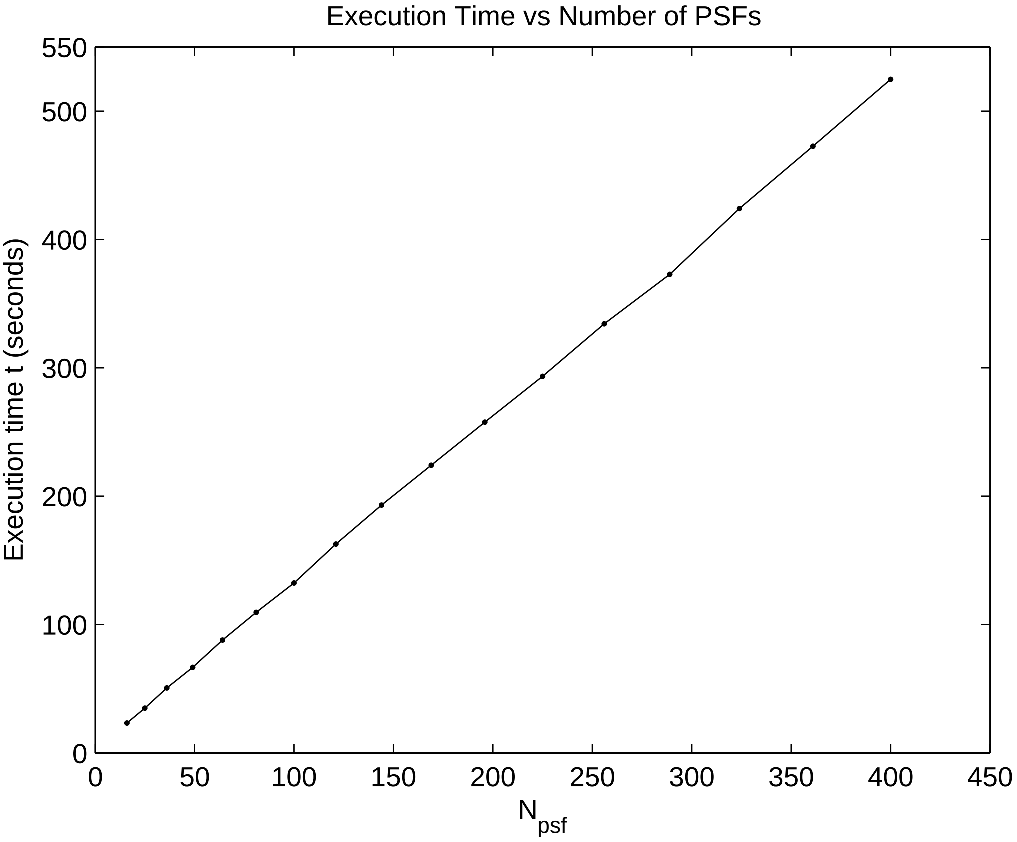

Figure 4.7: Timing Results for a grid of N PSFs

Timing results for the CGLS algorithm using N_{psf} as the number of PSFs to approximate the blurring effect. The plot illustrates the runtime for 4×4 to 20×20 square PSF grids in seconds. Each CGLS run was terminated at 20 iterations.

the feasibility and efficiency of our method on a problem that would be difficult using the semilinear method while including a spatially dependent PSF. Problems of this size are realistic for a number of practical modeling situations. For example, Alard (2009) has modeled the lensed system SL2SJ021408-053532, which produces a set of large arcs. This system has a lens that is comprised of a small group of six galaxies. Due to the large size of the lensed arcs, the scope of the source modeling prohibited the direct application of the semilinear method.

We form the lensed image of a portion of the Hubble deep field (Williams et al. 1996) by applying the elliptical potential of Blandford & Kochanek (1987) which was used by Link & Pierce (1998) to simulate the lens effect of the dark matter distribution of galaxy clusters. This lens is closely related to the PSIEP lens developed in Section 2.3.3, with a potential function given by

$$\psi(\boldsymbol{\theta}) = \frac{b^{2(1-q)}}{2q}\left[s^2 + (1+\epsilon_c)\theta_x^2 + 2\epsilon_s\theta_x\theta_y + (1-\epsilon_c)\theta_y^2\right]^q, \tag{4.3}$$

which results in deflection angle $\boldsymbol{\alpha}(\boldsymbol{\theta}) = \nabla_\theta\psi(\boldsymbol{\theta})$. The elliptical potential depends on seven parameters: b is the equivalent Einstein radius in the limit of vanishing core radius s, ellipticity ϵ, and power law index q, where $0 \leq q \leq 0.5$. The position angle of the lens ϕ determines the functions $\epsilon_c = \epsilon\cos\phi$ and $\epsilon_s = \epsilon\sin\phi$. We use the Einstein radius $b = 9$, power law index $q = 0.25$, $\phi = \pi/4$, position $(\theta_x, \theta_y) = (0, 0)$ and $s = 0.5$. The lens and source redshifts are $z_d = 0.12$ and $z_s = 1.5$, respectively. We used an array of 25 PSFs arranged on a 5×5 grid to blur the image. This set of PSFs has been used to test image restoration schemes for Hubble Space Telescope (HST) images and represents the spatially variant nature of the aberrations affecting the HST before it was repaired (Katsaggelos et al. (1994); Nagy & O'Leary (1998)). The size of each PSF is 60×60 pixels, and the source and image plane used to generate our lensed image are 800×800 pixel^2. Gaussian white noise was added with standard deviation $\sigma_g = 1.37$, giving the image $PSNR = 138.4$.

The image after 100 iterations is shown in Figure 4.8, and a reduced $\chi^2 = 0.995$ was found. The system was solved using the CGLS algorithm with all 25 PSFs using the lens parameters defined above. The model took approximately 7 minutes to solve using a single 2.4 GHz CPU core. An approximation to the nonlinear lens parameters could be found using global optimization methods if one of the following strategies were employed: (1) a low-resolution approximation to the data could be used early during the lens parameter optimization, with successive refinement occurring later during the run; (2) a global optimizer could be used to roughly approximate the lens parameters, shifting to a faster local optimization scheme once solutions are localized to a small region of parameter space; or (3) global optimization could be used for the entire problem making use of large-scale parallelization.

4.4 Conclusions

We have developed a novel and unique method to include the effects of a spatially variant PSF in gravitational lens modeling. Including these effects in the standard semilinear method would be difficult due to the complicated blurring matrix required. These complications can be overcome easily by incorporating the method of Nagy et al. (2002) with the matrix-free method. Our approach can accommodate large lensing problems like the case studied by Alard (2009), which limits the applicability of the direct semilinear approach. Techniques to include the effects of spatially variant PSFs are important, as the response varies over the detector area for many astronomical instruments. Our algorithm allows this effect to be included in lensing problems, thus improving the quality of reconstructions when the variability of the PSF is significant. The CGLS and SD algorithms allow a regularized inversion to be found quickly by truncated iteration.

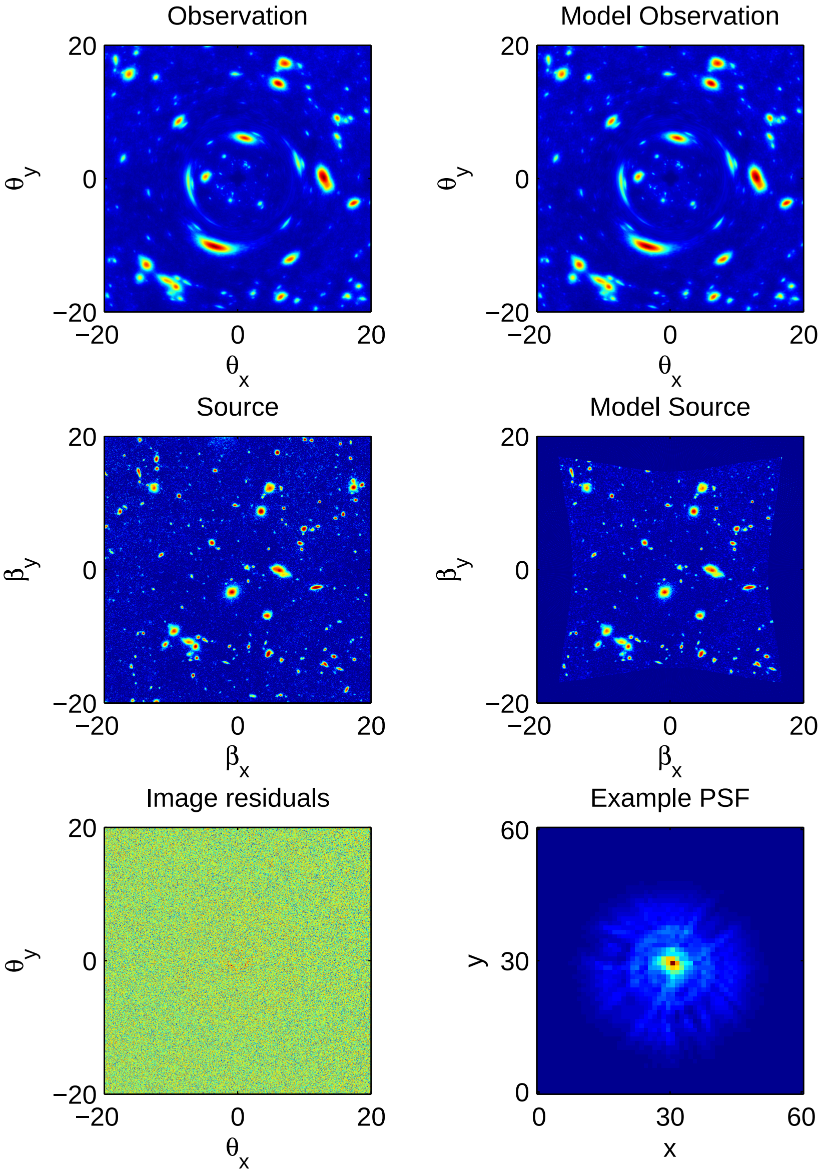

Figure 4.8: A Large Scale Spatially Variant PSF Test

Top row: observation and model image. Middle row: actual and model source. The image and source plane are both 800 × 800 pixels. These results are shown for 100 iterations. Bottom row: image residuals and an example of one of the 25 large PSFs used to generate the observations. Both of these images are plotted in logarithmic intensity to emphasize low level structure. Runtime for this large-scale test is approximately 7 minutes.

Chapter 5

Application to Data - The SLACS Lenses

In this chapter, we apply our method to model a subset of gravitational lenses from the Sloan Lens ACS Survey (SLACS). This survey was undertaken with the ACS instrument aboard the Hubble Space Telescope (HST). The lenses have been previously modeled by other studies (Koopmans et al. (2006); Bolton et al. (2008)), and therefore provide a useful set of systems to test the ability of our code to recover the lens parameters and source morphologies.

Several approaches exist to model gravitational lens systems. In this study, we apply global optimization methods to find the optimal set of lens parameters using a genetic algorithm. We treat the full optimization procedure as a two-step process: an analytical description of the source plane intensity distribution is used to find an initial approximation to the optimal lens parameters. The second stage of the optimization uses a pixelated source plane with the semilinear method to determine an optimal source. Regularization is handled by means of an iterative method and the Generalized Cross Validation function that is commonly used in standard image deconvolution problems. This approach simultaneously estimates the optimal regularization parameter and the number of degrees of freedom in the source. Using these techniques, we are able to justify an empirical estimation of the number of source degrees of freedom found in previous work. We test our approach by applying our code to a subset of the lens systems included in the SLACS survey.

5.1 Introduction

Methods for modeling gravitational lens systems are divided into a broad dichotomy between schemes that require a parameterized analytical model for the source intensity distribution, and schemes that assume only a pixelated source with no underlying model. Methods that parameterize the source intensity distribution are often quite easy to implement, but assume *a priori* knowledge of the source structure. Schemes that make use of a pixelated source are generally more complex, but offer greater flexibility since no parametric form is assumed for the source. This paper makes use of both parameterized and pixelated source models, exploiting the benefits provided by each.

Lens inversion schemes based on analytical source models assume an intensity distribution $I_s(\boldsymbol{\beta})$ in the source plane $\boldsymbol{\beta}$. A model of the lens density is then used to calculate a ray-tracing from the image plane $\boldsymbol{\theta}$ to the source plane using the thin lens equation (1.8). Since gravitational lensing conserves surface brightness (Kayser and Schramm 1988), the lensed image intensity is easily found by Equation 3.1,

$$I_i(\boldsymbol{\theta}) = I_s(\boldsymbol{\beta}(\boldsymbol{\theta})) \tag{5.1}$$

for an assumed parametric source intensity function I_s. The resulting lensed image $I_i(\boldsymbol{\theta})$ is then convolved with a point spread function (PSF) and compared with the data. The χ^2 statistic is minimized over the combined set of lens and source parameters using non-linear methods for parameter search and global optimization.

Sérsic profiles (Sérsic 1968) are widely used for galaxy scale sources, as defined by the equation

$$I_s(r) = I_0 \exp\{-k(n)[(r/r_0)^{-n} - 1]\}, \tag{5.2}$$

which assumes intensity I_0 at the scale length r_0 and shape index n. The shape index controls the curvature of the profile, where most galaxies have profiles with $0.5 < n < 10$. The de Vaucouleurs (1948) profile is recovered for $n = 4$, and the exponential disk is found by setting $n = 1$. The scaling factor $k(n)$ is used to normalize the distribution such that half the total luminosity is within r_0.

Due to their flexibility and simple physical interpretation, Sérsic functions are commonly used to model lensed sources (Bolton et al. (2008); Marshall et al. (2007); Brewer & Lewis (2011)). However, more complicated analytical source functions have also been used to approximate the varied and complex morphologies of galaxies and can include hundreds of parameters in extreme cases (Tyson et al. 1998). In general, analytical models are used because they are typically fast to evaluate and provide an intuitive understanding of the resulting source.

As useful as analytical models are, they may not be flexible enough to describe complex sources and may bias the lens parameters during χ^2 minimization to compensate for the artificial constraints imposed by their assumed analytical form. Pixelated source models were introduced to move past this limitation. This approach represents the source plane intensity as a set of basis functions, each having an adjustable parameter that represents the surface brightness of the source plane at a given pixel. The semilinear method treats each pixel as a basis function and minimizes the mismatch between model and data by manipulating the brightness of each source pixel s_j independently (Warren & Dye (2003); Treu & Koopmans (2004); Suyu et al. (2006)).

The semilinear method divides the lens modeling problem into a a non-linear "outer loop" problem that solves for lens parameters, and an "inner loop" problem that solves for the pixelated source, assuming a fixed set of lens parameters. An important benefit of this approach is that the inner loop problem is linear and therefore does not require complicated nonlinear optimization routines. The blurring and lensing effects are expressed by the matrix $\boldsymbol{f} = \boldsymbol{BL}$. Here, the lensing matrix $\boldsymbol{L}$ encodes the ray tracing operation from the image plane to the source plane, and blurring matrix $\boldsymbol{B}$ describes the effect of the PSF on the resulting lensed image. By minimizing the χ^2 statistic with respect to the source plane intensities s_j, the least-squares form of the problem is exposed:

$$\boldsymbol{F}^T\boldsymbol{F}\boldsymbol{s} = \boldsymbol{F}^T\hat{\boldsymbol{d}}, \tag{5.3}$$

where $\boldsymbol{F}$ is the lens matrix divided by the errors in the data, $F_{ij} = f_{ij}/\sigma_i$, and

$\boldsymbol{s}$ is a "flattened" image vector containing the intensities of the source plane pixels (Warren & Dye (2003); Koopmans (2005)). The vector $\hat{d_i} = d_i/\sigma_i$ is the data vector $\boldsymbol{d}$ normalized by the noise σ_i. This type of problem has been well studied in the context of the standard image deconvolution problem (Golub et al. (1979); Hansen (1994); Nagy et al. (2002); Vogel (2002)), which seeks to remove the distortion introduced by a blurring function (PSF).

In general, the solution of Equation 5.3 requires regularization to stabilize the inversion of the system matrix $\boldsymbol{F}^T\boldsymbol{F}$ (Koopmans 2005). The modified matrix is then given by

$$\boldsymbol{M} = \boldsymbol{F}^T\boldsymbol{F} + \lambda\boldsymbol{H}^T\boldsymbol{H}, \tag{5.4}$$

where $\boldsymbol{H}$ is a regularization matrix and λ a multiplier that controls the amount of regularization added to the problem. The simplest case, zeroth order regularization, assumes that $\boldsymbol{H} = \boldsymbol{I}$. This scheme regularizes the problem by seeking the solution $\boldsymbol{s}$ that has minimal intensity over the source plane. Higher order regularization schemes are also commonly used, such as curvature regularization that uses the second order derivatives of $\boldsymbol{s}$ to smooth the solution by minimizing the curvature over the source plane. Regularization schemes seek to impose physicality constraints on the source intensity to select a smoothly varying and physically realistic solution from the many alternatives that exist to solve the ill-posed system. Linear regularization schemes were studied in depth by Suyu et al. (2006).

Following our previous work (Rogers & Fiege (2011a); Chapter 3), we use the Qubist Optimization Toolbox (Fiege 2010) to find the nonlinear lens parameters varied in the outer loop of the lens inversion problem. The Qubist Toolbox contains several non-linear global optimization routines including Ferret, an advanced GA, and Locust, a PSO. In the inner loop, we solve the least squares problem of the semilinear method using Krylov subspace methods (Björck 1996). Krylov subspace methods are well known in the image deblurring community and have been studied in the context of deconvolution problems at length (Hansen (1994), Nagy et al. (2002)). This class of optimization routines include the CGLS and the SD methods. Krylov methods are attractive because they naturally regularize ill-posed problems and are efficient at solving large scale problems. We previously studied the performance of the GA and PSO methods on test problems using simulated lens data (Rogers & Fiege (2011a); Chapter 3). In that work we found that the GA explored the parameter space more thoroughly than the PSO, although the PSO was slightly faster to converge.

In this work, we will explore parameter selection methods to determine an appropriate value for the regularization constant in the semilinear method, and use the Ferret GA with our lens code to model data from the SLACS survey. We use a two stage approach to the lens modeling problem: we begin the optimization with analytical sources to estimate the approximate position of the globally optimal lens parameters, and switch to a pixelated source for further model refinement once the global optimizer has converged.

5.2 Gravitational Lens Source Deconvolution

The semilinear method with regularization describes gravitational lens modeling in the context of a least squares problem, where we seek a vector $\boldsymbol{s}$ that minimizes

$$g = ||\boldsymbol{F}\boldsymbol{s} - \hat{\boldsymbol{d}}||^2 + \lambda||\boldsymbol{H}\boldsymbol{s}||^2. \tag{5.5}$$

The first term in this sum is the χ^2 between the model and observed images, while the second term quantifies the strength of the regularization.

The most direct method to solve the least squares problem is to decompose $\boldsymbol{F}$ using the singular value decomposion (SVD; Golub and Reinsch (1970)),

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{5.6}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix composed of a set of non-zero, non-increasing elements $\nu_1 \geq \nu_2 \geq, ..., \geq \nu_N$. These diagonal elements are the singular values of $\boldsymbol{F}$, defined as the eigenvalues of $\boldsymbol{F}^T\boldsymbol{F}$ and $\boldsymbol{F}\boldsymbol{F}^T$, both of which have identical characteristic polynomials. The $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices are orthogonal, and have columns denoted as $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, the left and right singular value basis vectors. These vectors are the set of eigenvectors of the square matrices $\boldsymbol{F}\boldsymbol{F}^T$ and $\boldsymbol{F}^T\boldsymbol{F}$, respectively. In general $\boldsymbol{F}$ can be expressed in terms of the basis vectors and represented as a sum:

$$\boldsymbol{F} = \sum_{i=1}^{N} \frac{\boldsymbol{u}_i \boldsymbol{v}_i^T}{\nu_i}. \tag{5.7}$$

It is straightforward to write the solution to the system defined by Equation 5.5 from the matrix inverse $\boldsymbol{s} = \boldsymbol{F}^{-1}\hat{\boldsymbol{d}}$ in the absence of regularization, when $\boldsymbol{H} = \boldsymbol{0}$ in Equation 5.5. Using the results of the SVD analysis, the solution is

$$\boldsymbol{s} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\hat{\boldsymbol{d}} = \sum_i \frac{\boldsymbol{u}_i^T\boldsymbol{d}}{\nu_i}\boldsymbol{v}_i. \tag{5.8}$$

In this equation, we have written the SVD in terms of sums over the orthogonal columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, and the entries of $\boldsymbol{\Sigma}^{-1}$, which is trivial to compute since it is diagonal. This expansion allows us to express $\boldsymbol{s}$ as an expansion over the orthogonal basis $\boldsymbol{v}_i$.

The matrix $\boldsymbol{F}$ will have small singular values such that $\nu_i \to 0$ if the problem is ill-posed. These vanishingly small singular values cause the corresponding terms in Equation 5.8 to become large. The solution $\boldsymbol{s}$ may then become corrupted by the noise contained in the data vector $\hat{\boldsymbol{d}}$. This amplification of noise due to small singular values is the reason why regularization is required in Equation 5.5.

The simplest regularization scheme simply truncates the terms that arise from small singular values from the sum in Equation 5.8. Since the singular values form a non-increasing set, this corresponds to discarding all terms $i \geq k$, where k is the truncation threshold. Early termination of the sum removes the high frequency components of the basis vectors $\boldsymbol{v}_i$. This is known as the truncated singular value decomposition, or TSVD:

$$\boldsymbol{s}_\phi = \sum_i \phi_i \frac{\boldsymbol{u}_i^T\boldsymbol{d}}{\nu_i}\boldsymbol{v}_i, \tag{5.9}$$

where ϕ_i are a set of constants called the filter factors that are equal to 1 for terms $i \leq k$ and 0 for all terms higher than this threshold. However, terminating the summation abruptly may discard too much high frequency information. A more general choice is to gradually decrease the contribution of small singular value terms to the sum. This approach is called Tikhonov regularization, which amounts to a modification of the filter factors (Tikhonov 1963):

$$\phi_i = \frac{\nu_i^2}{\nu_i^2 + \lambda} \tag{5.10}$$

where λ is the regularization constant. Note that $\phi_i \approx 1$ when $\nu_i^2 \gg \lambda$, which occurs for small i. When ν_i is smaller than the regularization constant (large i), the filter factors damp the corresponding terms of Equation 5.8 as $\phi_i \approx \nu_i^2/\lambda$. Thus, λ must be assigned a value between the maximum and minimum singular values ν_1 and ν_N. This regularization scheme corresponds to setting the matrix $H = I$ in Equation 5.5 (Twomey (1963); Tikhonov (1963)). Regularization modifies the system that we are attempting to solve so that the inverse of Equation 5.6 becomes

$$\boldsymbol{F}_\phi^{-1} = \boldsymbol{V\Phi\Sigma}^{-1}\boldsymbol{U}^T, \tag{5.11}$$

where $\boldsymbol{\Phi}$ is the diagonal matrix of filter factors.

Note that neither of these schemes specifies how much regularization should be included for a given problem. The strength of the regularizing effect in Tikhonov regularization is controlled by the value of the regularization constant λ and by the truncation index k in the TSVD scheme. The regularization constant is a "hyper-parameter" which must be selected *a priori*. Fortunately, several methods exist to estimate the optimal regularization parameter for a given problem (Hansen 2010).

5.2.1 Regularization Parameter Selection Methods

A widely used technique to select a regularization parameter is the L-curve criterion (Hansen 1992), which we used in Rogers & Fiege (2011a). The L-curve is a plot of the residual versus the regularization term that appears in Equation 5.5, and is named for the characteristic shape of the resulting curve. The L-curve is parameterized by the regularization constant λ and the position on the plot with the largest curvature represents a balance between the image χ^2 and regularization term (Press et al. 2007). The position on the L-curve with the largest curvature provides an estimate of the optimal regularization parameter.

Another well-known regularization selection method is Generalized Cross Validation (GCV; (Golub et al. 1979)). This is a statistical method that aims to minimize the mean square error, $||\boldsymbol{F}\boldsymbol{s}_\phi - \boldsymbol{d}||$, where $\boldsymbol{s}_\phi$ is the optimally regularized solution. We now define the GCV function:

$$G(\lambda) = \frac{||\boldsymbol{F}\boldsymbol{s}_\phi - \boldsymbol{d}||^2}{\text{trace}(\boldsymbol{I}_N - \boldsymbol{F}\boldsymbol{F}_\phi^{-1})^2}, \tag{5.12}$$

where N is the number of source pixels involved in the inversion and $\boldsymbol{I}_N$ is the $N \times N$ identity matrix. This equation is based on statistical arguments that consider a solution to be properly regularized when it can predict elements of the data vector that have been omitted (Hansen 1997). The trace term in the denominator can be dramatically simplified given the definition of $\boldsymbol{F}_\phi^{-1}$ in terms of the SVD (Equation 5.11). The denominator of the GCV function becomes:

$$\text{trace}(\boldsymbol{I}_N - \boldsymbol{F}\boldsymbol{V}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T) = \text{trace}(\boldsymbol{I}_N - \boldsymbol{U}\boldsymbol{\Phi}\boldsymbol{U}^T) \tag{5.13}$$

using the SVD expansion of $\boldsymbol{F}$ (Equation 5.6). With the orthogonality of $\boldsymbol{U}$ the trace term simplifies dramatically. We are left with trace$(\boldsymbol{I}_N - \boldsymbol{\Phi})$ such that

$$\text{trace}(\boldsymbol{I}_N - \boldsymbol{F}\boldsymbol{F}_\phi^{-1}) = N - \sum_i \phi_i. \tag{5.14}$$

This sum represents the number of degrees of freedom in the problem. Putting these arguments together, the GCV function becomes

$$G(\lambda) = \frac{||\boldsymbol{F}\boldsymbol{s}_\phi - \boldsymbol{d}||^2}{(N - \sum_i \phi_i)^2}. \tag{5.15}$$

Wahba et al. (1979) showed that when the errors in the data vector are unbiased white noise with covariance matrix $\boldsymbol{C} = \sigma^2 \boldsymbol{I}_N$, and satisfy the discrete Picard condition (Kress (1989); Engl et al. (1996)), the minimum of the GCV function corresponds to a regularization parameter that is a good estimator of the optimal λ and approaches this value asymptotically as $N \to \infty$. The convergence results between the true solution of a test problem and the GCV-regularized solution have been also been thoroughly explored when these conditions are not satisfied (Vogel (2002); Lukas (1993)).

The denominator of the GCV function has special significance for gravitational lens modeling. Lens modeling schemes that pixelate the source plane have been criticized for relying on regularization since smoothing causes the number of degrees of freedom in the source to become undetermined (Kochanek et al. 2004). Suyu et al. (2006) give an estimate for the number of effective degrees of freedom based on Bayesian arguments. In that work the authors construct a variety of possible expressions for the number of degrees of freedom (NDF), and chose NDF $= N - \gamma$ with N the number of image pixels, and

$$\gamma = \sum_{i=1}^{N_s} \frac{\nu_i^2}{\nu_i^2 + \lambda}. \tag{5.16}$$

Their analysis was based on empirical tests that showed that this expression produced a reduced χ^2 nearest to 1 for a set of test problems (See Table 1, Suyu et al. (2006)). In fact, γ is simply the sum of the filter factors from Tikhonov regularization. The GCV function gives a statistical argument for choosing this value based on the nature of an optimally regularized source inversion.

Iterative methods complicate the calculation of the GCV function since we do not know the filter factors *a priori*, nor do we have the decomposition of $\boldsymbol{F}$, which can be expensive due to the sparsity and size of the matrix. In this case, we estimate the denominator by a Monte Carlo method (Girard 1989). This allows two advantages: we approximate the number of source degrees of freedom while simultaneously finding an approximation to the optimal regularization parameter. Using an iterative method, we find these quantities simultaneously while solving for the source intensity distribution. This is accomplished by running iterations on both $\hat{\boldsymbol{d}}$ and $\tilde{\boldsymbol{d}}$ simultaneously, where the vector $\tilde{\boldsymbol{d}}$ is composed of random elements drawn from a normal distribution with mean 0 and standard deviation σ_0.

We form the product $\tilde{\boldsymbol{d}}^T \tilde{\boldsymbol{r}}$, where $\tilde{\boldsymbol{r}} = \tilde{\boldsymbol{d}} - \boldsymbol{F}\tilde{\boldsymbol{s}}$. This quantity approximates the denominator of the GCV function and therefore the number of degrees of freedom in the iterative problem (Girard (1989); Hansen (1997)). This calculation requires twice the work during the iterative process and therefore effectively doubles the execution time of the code to solve for the source intensity function. However, since we generally require only a small number of iterations to solve a gravitational lens system, this extra work is acceptable due to the amount of information the calculation provides. By using this Monte Carlo estimate, we find the number of effective degrees of freedom at each iteration of the optimization process and therefore also the denominator of the GCV function, allowing an evaluation of Equation 5.15 at each iteration. Once we have evaluated an arbitrary number of iterations, we find the minimum of the GCV function and therefore we can select the critical number of iterations necessary to produce an optimally regularized source. The residual at this iteration is used to evaluate the χ^2 of the lens model.

Rogers & Fiege (2011a) explored the L-curve method for the selection of regularization parameters in gravitational lens modeling, arguing that the L-curve provides a useful parameter selection method that yields results which are easy to interpret. However, using this selection criterion can be difficult due to the curvature calculation, which requires spline fitting of the points on the L-curve and the curvature of the resulting smoothed curve. This calculation is non-trivial and results can be somewhat sensitive to the details of the fitting procedure. The GCV function requires more involved statistical arguments but provides a more robust selection method, since the function is calculated at each iteration simultaneously with the linear optimization. We find that the GCV and L-curve methods provide similar measures of the regularization parameter in practice, indicating that both can be used effectively to determine the optimal termination condition for the iterative solver. However, we prefer the GCV method for the reasons outlined above and focus on the GCV method in this study.

5.3 The SLACS Survey

The Sloan Lens ACS Survey (SLACS; www.slacs.org) was conducted using the Hubble Space Telescope ACS instrument (Bolton et al. 2006). The survey has detected 70 early type galaxies with definite lensed sources in the redshift range $z = 0.06$ to $z = 0.33$. The candidate systems were chosen by spectral analysis of galaxies in the luminous red galaxy (LRG; Deng et al. (2007b)) and main samples (Deng et al. 2007a) of the Sloan Digital Sky Survey (SDSS; www.sdss.org). Potential gravitational lens candidates were discovered when two distinct galaxy redshifts were seen within a single SDSS spectrum. We use reduced SLACS data from Bandara et al. (2009), who modeled the surface brightness of the E/S0 lens galaxies using the sum of two components, a Sérsic bulge (Equation 5.2) and an exponential disk. The PSF model from the ACS library was used in the surface brightness subtraction (Bandara et al. 2009), making use of the GIM2D code (Simard et al 2002). All of the data are F814W I-band images. See Bandara et al. (2009) for more details on the reduction procedure.

5.4 Results

Bolton et al. (2008) modeled the SLACS gravitational lens systems using analytical Sérsic and Gaussian source models to describe the intensity distribution in the source plane. A subset of 15 of these systems were further investigated using the semilinear method (Koopmans et al. 2006). We focus on six of the SLACS lens systems in this paper, and plan to model more of them in the future. Since they have been well studied using several established methods, the SLACS galaxies provide a useful consistency check for verifying the results of our lens modeling code.

The SLACS systems are modeled using a normalized singular isothermal elliptical mass density (SIE). We define a distance $\psi = \sqrt{qx^2 + y^2/q}$, such that the deflection angle $\boldsymbol{\alpha} = (\alpha_x, \alpha_y)$ is given by

$$\alpha_x = \frac{b}{q_f} \tan^{-1}\left(\frac{q_f x}{\psi}\right) \tag{5.17}$$

$$\alpha_y = \frac{b}{q_f} \tanh^{-1}\left(\frac{q_f y}{\psi}\right), \tag{5.18}$$

with $q_f = \sqrt{1/q - q}$, and Einstein radius b. In the limit $q \to 1$, the model corresponds to a singular isothermal sphere with Einstein radius

$$b = 4\pi \frac{\sigma_v^2}{c^2} \frac{D_{ds}}{D_s}, \tag{5.19}$$

where σ_v is the velocity dispersion, c the speed of light, D_{ds} the distance between the deflector and the source, and D_s the distance between the observer and the source.

These distances depend on the corresponding redshifts z_d and z_s and determine angular diameter distances that depend on the cosmological model used. We assume a standard cosmology with Hubble constant $H_0 = 70$ km s^{-1} Mpc^{-1}, matter density $\Omega_0 = 0.3$ and cosmological constant $\Lambda_0 = 0.7$. Following Bolton et al. (2008), we adopt the intermediate-axis normalization of the SIE (Kormann, Schneider and Bartelmann 1994). This normalization fixes the mass within given isodensity contours for constant b, and is implemented in the deflection angles above.

Koopmans et al. (2006) showed that the SIE is a useful model of early type isolated galaxies because the lens density ellipticity and orientation were found to align well with the surface brightness of the lens galaxies, indicating that light closely traces mass for these galaxies. No significant external shear was found to improve the fits. We therefore follow Koopmans et al. (2006) and adopt the SIE as a good lens model to represent isolated early type E/S0 galaxies.

We cropped out the residuals left over from the surface brightness subtraction of the lens in the F814W SLACS data, and cropped the field of view to the region of interest. We subtracted the surface brightness of the satellite companion in the SDSS J0956+5100 system using two Sérsic profiles, but performed no rebinning or other manipulation of the data in any way. Our lens models use the same ACS PSF that was used for the lens galaxy subtraction. Although it is known that the ACS PSF is position dependent (Bandara et al. 2009), we simplify our treatment by assuming a constant PSF over the region of interest to facilitate comparison with previously published results, though we have developed methods to include spatially variant PSFs in the gravitational lens problem (Rogers & Fiege (2011b); Chapter 4). We output the sigma image from the GALFIT code (Peng et al. 2010) that corresponds with the region of interest to estimate the errors on the image plane. We emphasize that the main focus of this work is to study the regularizing properties of the CGLS method on the derived solutions with the GCV scheme to select the optimal level of regularization.

Our analysis initially solves for the parameters of an analytical source model, which we use as an approximate solution to a more refined model that uses a pixelated source. We start by treating the source plane intensity distribution as a sum of

Sérsic profiles, using the same number of analytical source components to model each system as in Bolton et al. (2008). The SIE lens is used to find the lensed image of the source plane, which is convolved with the appropriate ACS PSF. We search for the global minimum of χ^2, using the Ferret GA (Fiege 2010) to fit both the lens and source parameters. Once we find an approximation to the global minimum, we select a volume of lens parameter space in the neighbourhood around the best fit lens model. Noting that Ferret is used predominantly as a bounded optimizer, this neighbourhood becomes the search volume in the next step of our method, which replaces our analytical source model with a pixelated source. The optimization of a pixelated model requires a new Ferret run, which begins with the search volume found in the previous step populated initially by random lens models. Normally, we expect the lowest χ^2 model to reside within this volume; however, we configure the optimizer using 'soft' boundaries, which allows the GA to move outside of the predefined search volume if the initial approximation is bounded too tightly. This option allows Ferret to expand the search space if a large fraction of the GA population occupies positions close to the boundaries of the parameter space. In general, the lens parameters of our pixelated sources were found to reside within these search volumes and agree well with the analytical approximations. We compute our best refined model by optimizing the lens and source plane parameters using a pixelated source and regularizing iteration selected by the GCV function.

In addition to the regularizing effect of truncated iteration, we have found that enforcing non-negativity in the source solutions dramatically improves the quality of the reconstruction and tends to further smooth remaining structure in the image residuals. As a final step, we have modeled the set of best-fit lens models with the modified residual norm steepest descent algorithm (MRNSD; Kaufman (1993); Nagy & Strakoš (2000); Bardsley (2006)). This algorithm is a bounded SD optimization routine that seeks sources with $s_j \geq 0$. It is well known that the MRNSD method can be difficult to use with the standard regularization parameter selection methods since the L-curve and GCV functions are not guaranteed to be smooth when non-negativity is enforced (Favati et al. 2010). However, we plan to explore the regularizing properties of several iterative non-negative reconstruction schemes in a future

Lens Model Parameters						
SDSS System	z_d	z_s	σ_v (km s^{-1})	q	b (")	Reduced χ^2
J0037-0942	0.1955	0.6322	286	0.825	1.55	0.95
J0216-0813	0.3317	0.5235	351	0.783	1.18	0.96
J0737+3216	0.3223	0.5812	292	0.630	1.00	1.06
J0912+0029	0.3240	0.1642	341	0.561	1.59	0.98
J0956+5100	0.2405	0.4700	318	0.620	1.33	1.05
J1402+6321	0.2046	0.4814	292	0.843	1.34	0.96

Table 5.1: SLACS Lens Modeling Results

Lens model parameters for a subset of the SLACS systems found by the Ferret GA with source reconstruction by the CGLS routine.

study.

Combining analytical and pixelated sources greatly improves the efficiency of the search, since analytical models can be evaluated very quickly. Searching using pixelated sources is a more intensive process, and time can be saved by adopting the semilinear method only once we have a good approximation to the lens parameters corresponding to the minimum χ^2. Rogers & Fiege (2011a) noted that a set of trivial pixelated solutions exist when global optimization methods are used to model lensed systems. These trivial solutions are found when the effect of the lens is reduced, resulting in sources that closely resemble the data. The two-stage optimization process is useful since the initial analytical sources are generally not as flexible as pixelated sources, and thus provide a natural method for avoiding exploration of the trivial regions of the parameter space. The analytical stage of the algorithm terminates once the GA has converged and we no longer see improvement in the population. Typically, convergence requires only $50 - 100$ generations using a population of 300 individuals for the analytical portion of the optimization, and approximately 100 iterations for the second semilinear optimization stage.

The final velocity dispersion σ_v, axis ratio q, and Einstein radius b of our models are shown in Table 5.1. The reduced data, model image, recovered non-negative source and residuals are shown in Figures 5.1 and 5.2. Our results agree with the SLACS lens models for each system to within 3% in velocity dispersion σ_v. Both the pixelated and analytical source plane intensity distributions agree with one another in all cases. Our lens modeling results agree with the parameters in Bolton et al. (2008) very well. The reduced χ^2 statistic for all systems is very close to unity.

We find the largest discrepancy in the ellipticity of SDSS $J0737 + 3216$. The Einstein radius (velocity dispersion) of the system is similar to the results from both Bolton et al. (2008) and the F814W data analyzed by Marshall et al. (2007). However, the ellipticity of the lens recovered by the GA is significantly lower than the models found by the previous studies by $\approx 5\%$. We found this lower ellipticity from both the initial analytical source fit and by pixelated source modeling. The SDSS $J0912+0029$ data is heavily contaminated with noise, although it is adequately fit by our GCV regularized solution, and our analytical and pixelated sources agree. Of all of the systems, SDSS $J0956 + 5100$ and SDSS $J0737 + 3216$ show the most structure in the residuals, although the magnitude of these residuals are small ($< 1\%$) compared to the intensities of the image pixels. In fact, the largest systematic effects present in most of the residual images in Figures 5.1 and 5.2 are produced from the subtraction of the intensity profile of the lens galaxy.

We have used the GCV approach with both CGLS and SD, and find similar results for both of these algorithms. The SD routine takes longer than the CGLS method to converge, although it is in general more a more stable approach to regularization and has been suggested as a superior routine for image deblurring problems due to its reduced sensitivity to stopping criterion (Nagy & Palmer 2003). The convergence properties of the CGLS and SD routines were discussed at length in Chapter 3. The bounded non-negative MRNSD routine helps to reduce structure in both the reconstructed sources and the image residuals. The best fit MRNSD solutions are found by comparing the solution at each iteration to the optimally regularized CGLS solution. This comparison minimizes

$$z = \frac{||x^{\rm CGLS} - x_i^{\rm MRNSD}||}{||x^{\rm CGLS}||} \tag{5.20}$$

where $x^{\rm CGLS}$ is the optimally regularized CGLS solution and $x_i^{\rm MRNSD}$ the non-negative solution at the i^{th} iteration of the MRNSD algorithm. The reduced χ^2 values of the optimally regularized CGLS solution and the selected MRNSD solution were found to

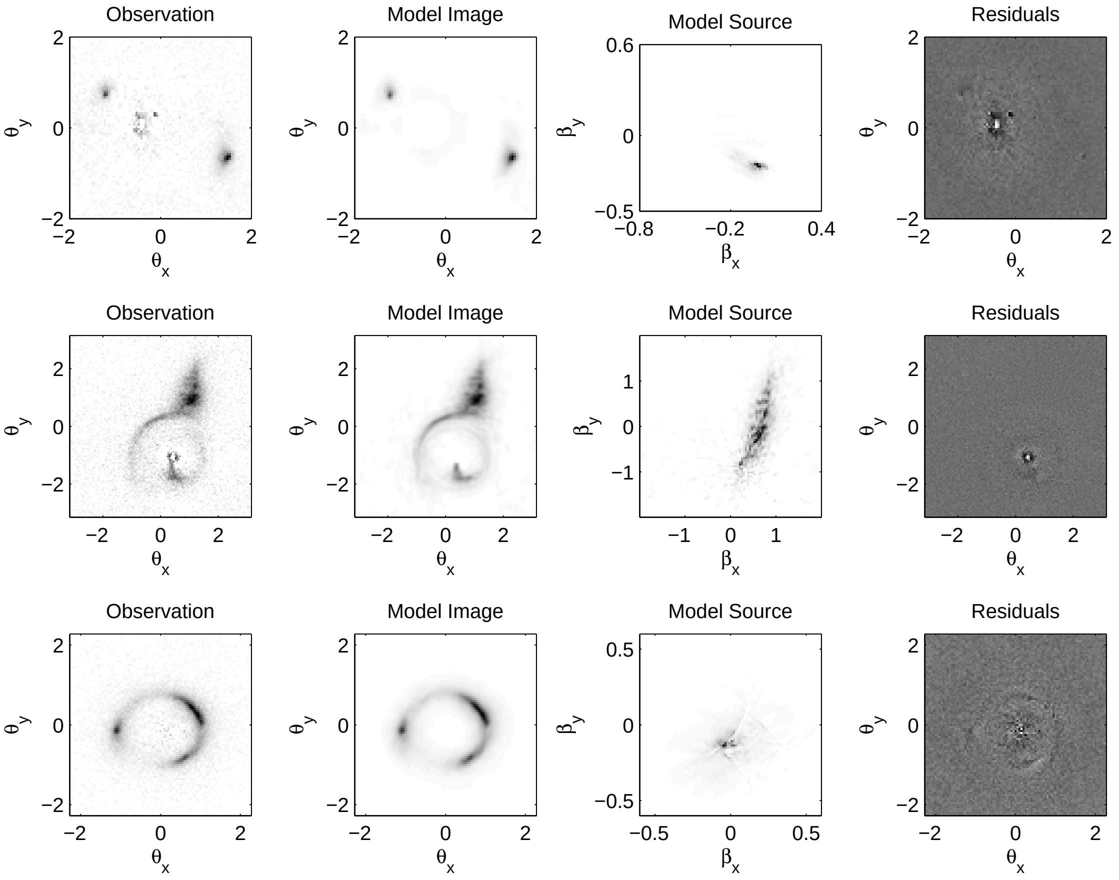

Figure 5.1: A Variety of SLACS Lens Models, I

A selection of SLACS gravitational lenses. The sources are non-negative and found using the MRNSD algorithm as the final polishing step. The columns show the data $\boldsymbol{d}$, image model, source model $\boldsymbol{s}$ and residual $\boldsymbol{r}$ respectively. The model parameters are given in Table 5.1. Top row: SDSS J0037-0942, second row: SDSS J0216-0813, bottom row: SDSS J0737+3216.

vary by less than 1% for a given system, though the MRNSD residuals are smoother than the residuals of the CGLS models. This is due to the reconstruction of back-traced noise present in the CGLS solutions. The filter factors of the CGLS method are found by a recursion relation that depends on all of the singular values (Hansen 2010). Even though CGLS tends to suppress high frequency noise at the beginning of the optimization process, the high frequency components are not completely damped

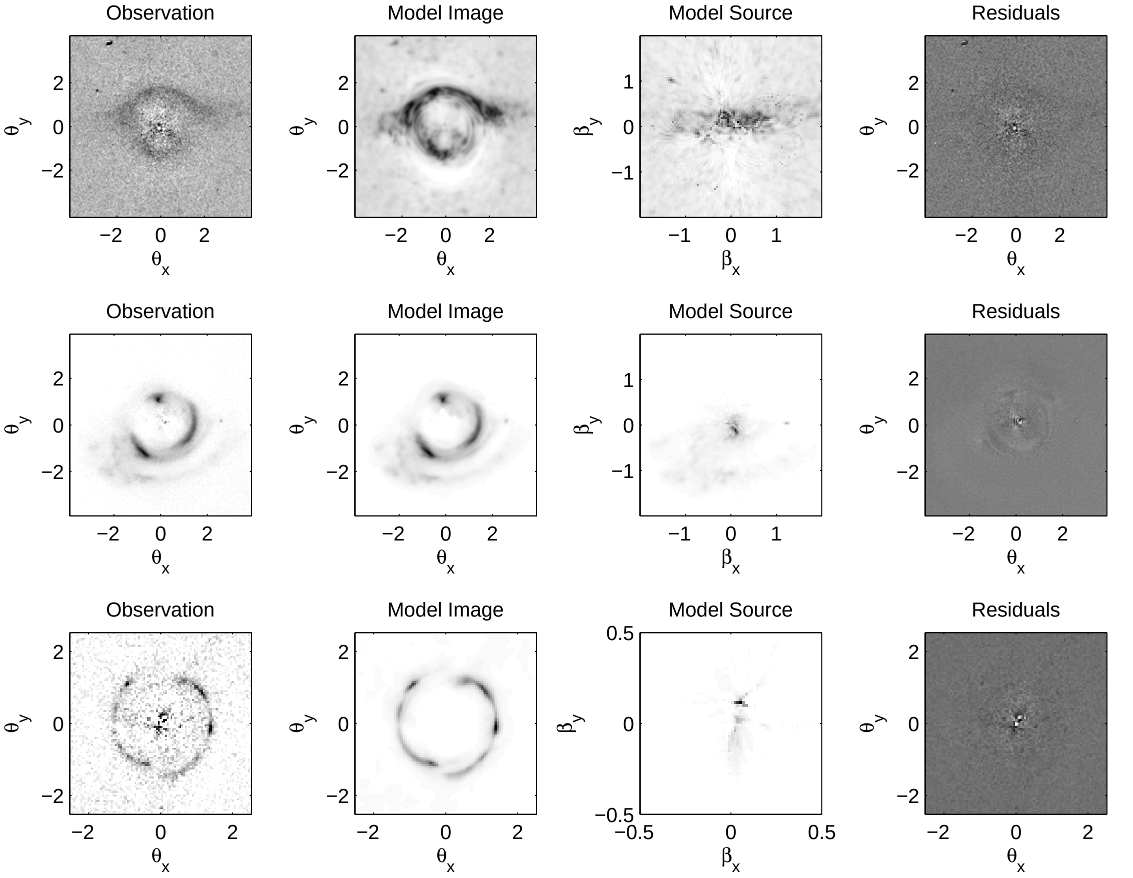

Figure 5.2: A Variety of SLACS Lens Models, II

Top row: SDSS J0912+0029, second row: SDSS J0956+5100, bottom row: SDSS 1402+6321

out at any given iteration and build up over the course of a run. Hence, even the optimally regularized solution still contains some high-frequency components that correspond to back-traced noise. The MRNSD algorithm seems to be more robust to the propagation of high-frequency noise in the recovered non-negative solutions, thus producing images that are naturally smoother than the corresponding CGLS sources.

Regularization by truncated iteration in the context of Krylov optimization is the simplest of many regularization methods that can be used. Truncated iteration regularization produces solutions (figures 5.1 and 5.2) which are less smooth than higher order regularization schemes, such as the second order (curvature) regularization used in Koopmans et al. (2006). It has been suggested that the LSQR algorithm (Björck 1996) can generally accomodate more complicated regularization schemes. We previously tested LSQR in the context of gravitational lens modeling using simulated data with the L-curve method (Rogers & Fiege (2011a); Chapter 3), and plan to further investigate this scheme in future work.

Overall we are encouraged by our results since we were able to recover the SLACS lens parameters and general source morphologies. The results could be improved slightly by including a final local optimization step to 'polish' the results returned from the GA. We did not detect any parameter space degeneracies except for the expected position angle degeneracy of the elliptical lens model.

We illustrate the behavior of GCV and the L-curve criteria in Figure 5.3. This figure shows the logarithm of the GCV curve on the left as a function of iteration k. The right-hand panel shows the spline-smoothed L-curve for the same problem, using the sum of pixel intensities $\sum_i s_i^2$ to quantify regularized solutions on a log-log scale. Logarithmic scaling emphasizes the structure of these curves. Note the difference in the scales of the L-curve. In our experience the GCV function used with the CGLS algorithm always shows a well-defined minimum. However, significant smoothing is needed to find the corner of the the L-curve, which may affect the accuracy of its determination. Using SDSS J0216 $-$ 0813 as an example, the optimally regularized L-curve and GCV solutions are marked with a triangle and circle respectively. In this case, both of these regularization parameter selection methods produce similar results. However, we often observe that the L-curve can show false curvature maxima when the iterative optimizers make rapid progress early in the run, leading to dramatically over-regularized solutions. The GCV function avoids this problem.

Combined with the statistical arguments used to derive the GCV function and its more robust behavior, we conclude that GCV is a more useful parameter selection method for solving the least-squares source deconvolution problem for gravitational lens systems. We have tested both of these selection methods against the Bayesian regularization method developed by Suyu et al. (2006) with the semilinear method and zeroth order regularization. In most cases the GCV function selects a regularized solution that is closer to the optimal Bayesian solution than solutions selected by the L-curve method. The iterative approaches tend to find smoother solutions than zeroth order regularization. This is not surprising since the filter factors of the CGLS and SD approaches differ from the standard Tikhonov approach.

We have marked two additional points on the left-hand panel on the GCV curve in Figure 5.3. These points signify over-regularized and under-regularized solutions. The sources corresponding to the solution of the SDSS J0216$-$0813 system are shown in Figure 5.4 using both CGLS and MRNSD algorithms. The critically regularized solutions balance the image χ^2 and regularization terms. As shown in this figure, the over-regularized solutions are over-smoothed, and the under-regularized solutions include too many high frequency components. The corresponding MRNSD solutions were found by terminating iteration when Equation 5.20 is minimized.

5.5 Conclusions

We have used iterative methods to model a subset of the SLACS lenses using Generalized Cross Validation to select the optimal regularizing iteration. By making use of the GCV function we addressed the problem of the number of effective degrees of freedom in the source and explained an empirical choice in Suyu et al. (2006) based on a parameter choice method that is commonly used in standard image deconvolution problems. The GCV function sheds light on the concept of optimally regularized sources and provides an efficient method to select regularization parameters for iterative methods. A non-negative bounded iterative algorithm is found to significantly improve the quality of the reconstructed sources. This approach provides non-negative solutions through linear optimization, which is significantly simpler to implement than other constrained optimization techniques such as the maximum entropy method (Skilling and Bryan (1984); Wayth & Webster (2006)) that require the use of more complicated non-linear optimization schemes.

The lens parameters recovered by the Ferret GA are similar to previously published results found by Bolton et al. (2008) and we find consistency between analytical approximations to the source plane intensity based on a sum of Sérsic profiles. We plan to model the rest of the SLACS lenses in the future and explore other local optimization methods to solve the least squares problem with more complicated regularization schemes.

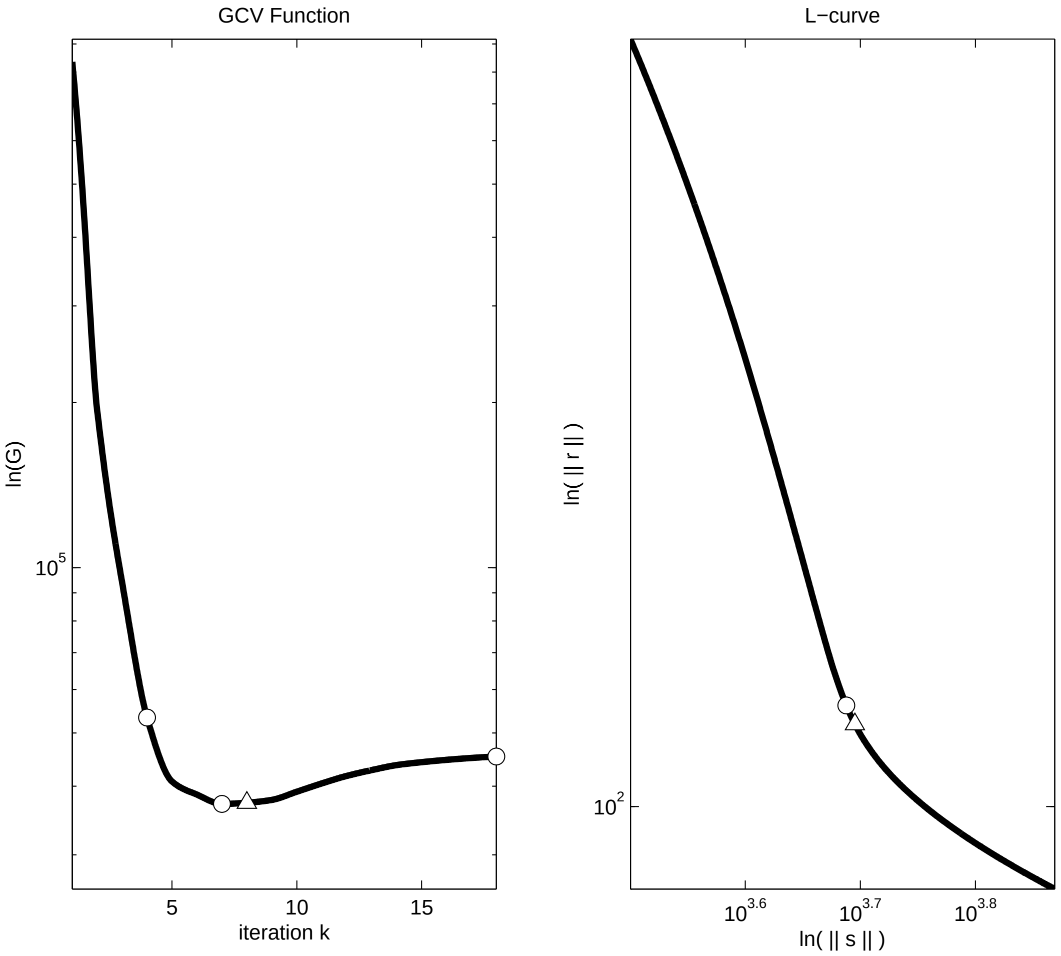

Figure 5.3: GCV and the L-curve as a function of iteration

Left: GCV as a function of iteration k for regularizing the SDSS J0216-0813 lens system. Right: Spline-smoothed L-curve of residuals vs. sum of source intensities. Note the difference in vertical and horizontal axes of the L-curve plot. In general, we find that the GCV function always has a well-defined minimum, while the L-curve is more sensitive to fluctuations in the behavior of the iterative optimization methods. The maximum curvature L-curve solution is marked on both figures with a triangle, and the minimum of the GCV function with an open circle (right). The three points on the left-hand panel represent over-regularized, critically-regularized and under-regularized solutions, respectively.

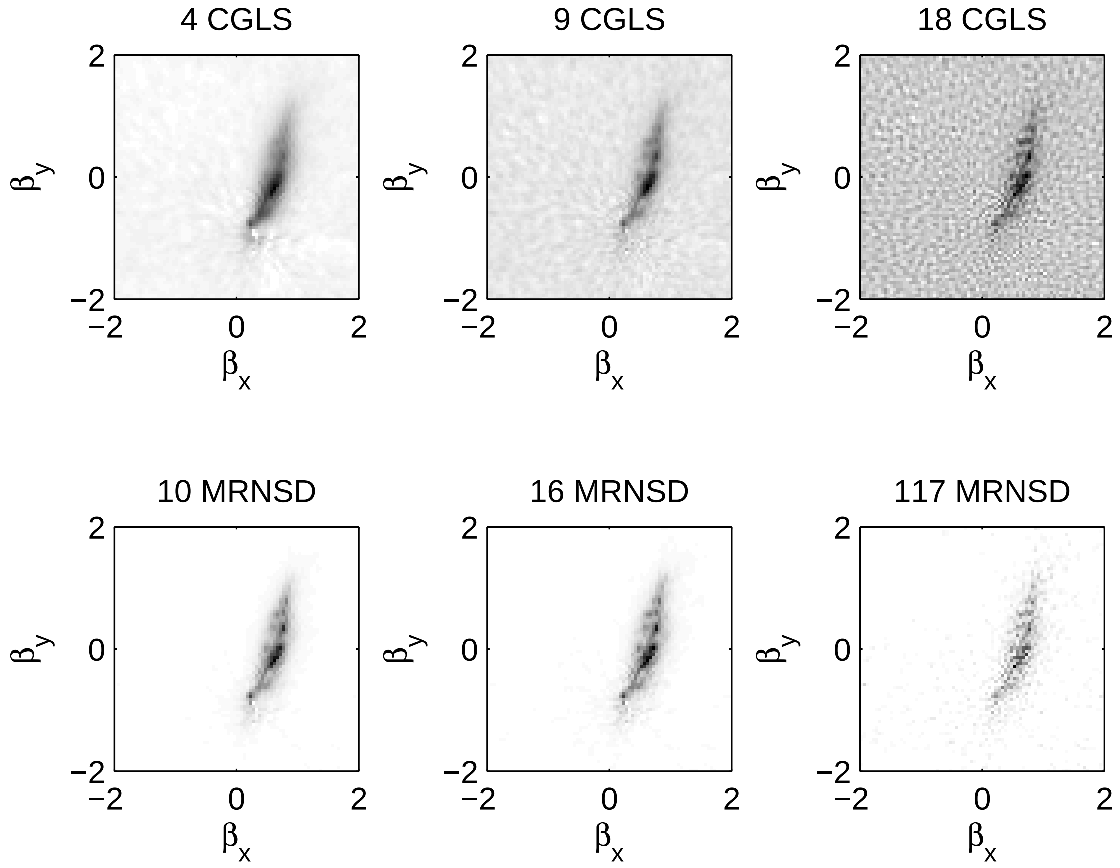

Figure 5.4: An Example of Regularization Effects on a SLACS Lens

Three solutions marked in the left-hand panel of Figure 5.3. These solutions correspond to over-regularized (left), critically-regularized (middle) and under-regularized solutions (right) as selected by the GCV function. Note the emphasis on back-traced noise in the under-regularized CGLS solution and the excessive smoothing of the over-regularized solution. Non-negative MRNSD solutions are shown on the second row. The title of the plots represent the number of iterations and methods used to obtain the solutions.

Chapter 6

Conclusions

The semilinear method provides the foundation for our studies on gravitational lensing. We have shown that the least squares problem can be efficiently solved using Krylov subspace methods such as the CGLS method, and by gradient descent using the SD method. These iterative approaches are advantageous since they allow us to avoid explicitly building the lens and blurring matrices, which can be large and potentially ill-posed depending on the PSF. Iterative methods allow us to substitute an algorithmic description of the matrix multiplication and transpose multiplication processes in the source optimization step rather than the full matrices. Thus, we are able to solve large scale problems quickly, allowing us to make efficient use of global optimization routines. These optimization algorithms require many function evaluations, so execution time of the code is paramount.

By making use of regularization selection methods commonly used in the standard image deconvolution problem, we made a connection between the number of spatial frequencies present in the SVD expansion and the number of source degrees of freedom in the source reconstruction step. This expansion shows that the sum of the filter factors represent the number of source degrees of freedom. We made use of both the L-curve criteria and the GCV function to select regularization parameters. The dependence of the GCV function on source degrees of freedom justifies the sum of the filter factors and the role of this sum in selecting an optimally regularized solution. We used a Monte Carlo approach to estimate the sum of the filter factors using CGLS and SD. Our previous work showed that SD is more stable than CGLS when applied to the standard image deconvolution problem, and is less sensitive to the stopping criteria in general.

Our lens density models are optimized by global optimization methods, including GAs and PSOs. The GA approach is generally more thorough at exploring the parameter space of test problems, and performed consistently and reliably on real data. The PSO provides a faster search in general, but is more limited in its ability to discover parameter degeneracies in the lens models.

Using the framework of the semilinear method, we were able to include spatially variant PSFs in our lensing code. Such spatially variable PSFs can also be built into the standard semilinear approach, but would require large complicated blurring matrices, causing instability in the lens inversion. However, such spatially variant PSFs are practical in our method, due to the efficiency of our matrix-free approach and the use of FFTs in the PSF convolution step. This spatially variant blurring effect was used on a large scale blurring problem, in which we simulated a heavily blurred cluster lens using a portion of the Hubble deep field and a grid of PSFs used in modeling the spatially dependent blurring of the HST (Nagy et al. 2002).

Finally, we applied our modeling code to a sample of SLACS lenses. We used a two step process in the optimization of the lens parameters using the Ferret GA. The first step of this approach used analytical functions to approximate the source intensity, and the second step switches to more general pixelated sources. We set the GA to search a volume of parameter space around the best lens approximation found in the first step. In general we find good agreement with previously published results, demonstrating the utility of our modeling code.

The concepts developed in this work are significant due to the matrix-free formulation of the semilinear method, which can be used on large problems since explicit construction of the lens and blurring matrices is not necessary. As mentioned in Chapter 4, this approach allows us to use linear optimization methods on large problems that prohibit the use of the standard semilinear method (Alard 2009). Our exploration of parameter selection methods simplify the determination of the number of degrees of freedom in the problem while determining the optimal amount of regularization to include. This is an important modification since the evaluation of the matrix inverse and its eigenvalues can be a time consuming calculation.

We have discovered that imposing non-negativity on the source reconstructions can provide a significant improvement in the solutions. There are a variety of iterative methods that enforce positivity and provide the semi-convergence behavior necessary for regularization by truncated iteration (Kaufman (1993); Nagy & Strakoš (2000); Bardsley (2006)). These linear non-negative optimization schemes are valuable approaches since they provide a simpler method for including non-negative solutions than the MEM, which requires more complicated non-linear optimization methods.

In Chapter 3 we noted that gravitational lens systems have been modeled occasionally in the literature using pixelated mass models in which the lens density is pixelated. These complicated numerical models can be combined with our pixelated source deconvolution routine, while including a global optimizer to find the optimal pixelated lens density distribution. The analysis of the regularizing properties of bounded local optimizers and the application of pixelated lens mass distributions will form the basis of future work.

In conclusion, the unique approach to lens modeling developed in this work simplifies and expands on the semilinear method. Linear iterative optimization methods allow the modeling of large scale systems and global optimization routines provide reliable approaches for exploring the parameter space of lens density models. When used together, these numerical techniques provide a powerful framework for modeling gravitational lens systems.

Appendix A

The Weak Field Limit of General Relativity and the Gravitational Lens Deflection Angle

The gravitational lensing effect on cosmological scales relies on the weak field limit of GR (Schneider 1985). This is due to the fact that the distances involved are on cosmological scales. In this section, we derive the weak field metric from first principles by considering a perturbation about the Minkowski metric of special relativity (Misner et al. 1973). The derivation in this section is a synthesis of the approaches presented in d'Inverno (1992) and Hobson et al. (2006).

In GR, the line element is written as

$$ds^2 = g_{ab}dx^a dx^b. \tag{A.1}$$

We begin by considering a metric with elements g_{ab} that is only slightly perturbed from the Minkowski metric in the standard coordinates (t, x, y, z):

$$g_{ab} = \eta_{ab} + h_{ab}, \tag{A.2}$$

where h_{ab} is a small perturbation about the Minkowski metric of flat spacetime given by

$$\boldsymbol{\eta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{A.3}$$

We adopt the boundary condition that space-time is asymptotically flat, such that if $r = \sqrt{x^2 + y^2 + z^2}$ denotes the radial distance, then

$$\lim_{r \to \infty} h_{ab} = 0. \tag{A.4}$$

Note that

$$(\eta_{ab} + h_{ab}) \left(\eta^{cb} - h^{cb} \right) = \delta^c_a \tag{A.5}$$

to lowest order in h_{ab}, where we identify

$$g^{ab} = \eta^{ab} - h^{ab} \tag{A.6}$$

to first order in h_{ab} (Misner et al. 1973). Since the weak field perturbation is small, we can effectively raise and lower tensor indices of small quantities f^{ab} using the Minkowski metric η_{ab}:

$$f^a_c = \eta_{bc} f^{ab}. \tag{A.7}$$

In general, the Christoffel symbols of the second kind are given by

$$\Gamma^a_{bc} = \frac{1}{2} g^{ad} \left(g_{dc,b} + g_{db,c} - g_{bc,d} \right), \tag{A.8}$$

where a comma followed by an index in the subscript denotes differentiation with respect to the corresponding coordinate; for example

$$\frac{\partial^2 f}{\partial x^i \partial x^j} = \partial_i \partial_j f = f_{,ij}. \tag{A.9}$$

The Riemann curvature tensor is defined in terms of the Christoffel symbols as

$$R_{abcd} = \Gamma^a_{bd,c} - \Gamma^a_{bc,d} + \Gamma^a_{kc} \Gamma^k_{bd} - \Gamma^a_{kd} \Gamma^k_{bc}. \tag{A.10}$$

Both of these important tensors can be simplified in the weak field limit. The Christoffel symbols can be rewritten (d'Inverno 1992)

$$\Gamma^a_{bc} = \frac{1}{2} \left(h^a_{c,b} + h^a_{b,c} - \eta^{ak} h_{bc,k} \right) \tag{A.11}$$

and the linearized Riemann tensor becomes

$$R_{abcd} = \frac{1}{2}\left(h_{ad,bc} + h_{bc,ad} - h_{ac,bd} - h_{bd,ac}\right). \tag{A.12}$$

The trace of the Riemann tensor is the Ricci tensor $R_{ab} = \eta^{cd} R_{cdab}$:

$$R_{ab} = \frac{1}{2}\left(h^{c}_{a,bc} + h^{c}_{b,ac} - \Box h_{ab} - h_{,ab}\right) \tag{A.13}$$

where $h = \eta^{cd} h_{cd} = h^{c}_{c}$ is the trace of h_{ab} in the last term, and $\Box$ is the d'Alembertian operator

$$\Box = \eta^{ab}\partial_a\partial_b = \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial z^2}. \tag{A.14}$$

Finally, the corresponding Ricci scalar R is the trace of the Ricci tensor:

$$R = \eta^{cd} R_{cd} = h^{cd}_{,cd} - \Box h. \tag{A.15}$$

The Einstein tensor is is related to the elements of the energy-momentum tensor T_{ab},

$$G_{ab} = R_{ab} - \frac{1}{2} g_{ab} R = \kappa T_{ab} \tag{A.16}$$

where the constant $\kappa = 8\pi G/c^4$. The linearized Einstein equations become

$$G_{ab} = \frac{1}{2}\left(h^{c}_{a,bc} + h^{c}_{b,ac} - \Box h_{ab} - h_{,ab} - \eta_{ab} h^{cd}_{,cd} + \eta_{ab}\Box h\right). \tag{A.17}$$

This equation in its current form is quite formidable, but fortunately can be simplified. Let us define a new quantity

$$\psi_{ab} = h_{ab} - \frac{1}{2}\eta_{ab} h. \tag{A.18}$$

This can be rewritten such that h_{ab} is expressed as a function of ψ_{ab}:

$$h_{ab} = \psi_{ab} - \frac{1}{2}\eta_{ab}\psi \tag{A.19}$$

which also implies $h = -\psi$. We can then substitute Equation A.19 into the expression for the linearized Einstein tensor (Equation A.17) to express it as a function of ψ_{ab}:

$$G_{ab} = \frac{1}{2}\left(\psi^{c}_{a,bc} + \psi^{c}_{b,ac} - \eta_{ab}\psi^{cd}_{,cd} - \Box\psi_{ab}\right). \tag{A.20}$$

While still quite daunting, the linearized field equations can be further simplified using a gauge transformation (d'Inverno 1992).

A.1 The Einstein Gauge

Consider a coordinate transformation from the standard Minkowski coordinates to a new coordinate system $\bar{x}^u = \Lambda^u_b x^b$ by a global Lorentz transformation. We find the metric in this new coordinate system by using the transformation property of covariant tensors,

$$\bar{g}_{ab} = \Lambda^x_a \Lambda^y_b g_{xy}. \tag{A.21}$$

With Equation A.2, we have

$$\bar{g}_{ab} = \Lambda^x_a \Lambda^y_b \left(\eta_{xy} + h_{xy}\right) = \bar{\eta}_{ab} + \bar{h}_{ab}. \tag{A.22}$$

Note that the components of the weak field perturbation transform just like the metric itself would, except that the components h_{ab} are only a perturbation of the Minkowski spacetime. This implies a freedom in the representation of the metric such that a coordinate transformation of the form used in Equation A.21 can be used to transform the weak field metric leaving the weak field condition $|h_{ab}| \ll 1$ unchanged.

Let us transform coordinates by adding a small displacement $d^a(x^b)$, such that the elements d^a are functions of position and $|d^a_{,b}| \ll 1$:

$$x^a \to \bar{x}^a = x^a + d^a(x^b). \tag{A.23}$$

Then we have

$$\Lambda^a_b = \frac{\partial \bar{x}^a}{\partial x^b} = \delta^a_b + d^a_{,b} \tag{A.24}$$

The metric transformation expression Equation A.21 to first order gives:

$$\bar{g}_{ab} = \eta_{ab} + h_{ab} - d_{a,b} - d_{b,a}. \tag{A.25}$$

This coordinate change redefines the weak field perturbation

$$h_{ab} \to \bar{h}_{ab} = h_{ab} - d_{a,b} - d_{b,a}. \tag{A.26}$$

The new h_{ab} is still small since the elements of the vector $\mathbf{d}$, d^a, and its derivatives were chosen to be small initially. This introduces additional freedom into the equations because we can choose the vector elements d^a. By analogy with gauge transformations of the electromagnetic 4-potential ϕ with scalar field χ,

$$\phi_a \to \phi_a + \partial_a \chi. \tag{A.27}$$

By analogy, we see that Equation A.26 corresponds to a gauge transformation. In fact, it can be shown that the linearized curvature tensor (Equation A.12) and its contractions (Equations A.13 and A.15) are invariant with respect to such gauge transformations. Then, just as in electromagnetism, the gauge can be fixed by imposing further conditions on the metric. We work in the Einstein gauge, also known as the de Donder, Hilbert or Fock gauge:

$$\psi^a_{b,a} = h^a_{b,a} - \frac{1}{2} h_{,b} = 0. \tag{A.28}$$

Under the coordinate transformation of the type in Equation A.23, we can show that Equation A.28 transforms as

$$\psi^a_{b,a} \to \bar{\psi}^a_{b,a} = \psi^a_{b,a} - \Box v_b. \tag{A.29}$$

Therefore, we can transform the equation into the Einstein gauge by choosing d_a to satisfy the wave equation $\Box d_a = \psi^b_{a,b}$. The Einstein gauge actually describes an entire family of infinitesimal transformations by adding small vectors ξ_a to d_a that obey the homogenous wave equation $\Box \xi_a = 0$. Dropping the barred notation, the Einstein tensor in this gauge has a very simple form:

$$G_{ab} = \frac{1}{2} \Box \psi_{ab} = -\kappa T_{ab}. \tag{A.30}$$

To solve the Einstein equations, the form of the energy-momentum tensor must be specified. As in most astrophysical applications, we describe the gravitating body as a perfect fluid, with matter density ρ, co-moving 4-velocity u^i and pressure p:

$$T^{ab} = (\rho + p)\, u^a u^b - p g^{ab}. \tag{A.31}$$

To simplify this expression, we assume the stationary source limit, such that the matter of the source moves slowly with respect to the coordinates x^a, so that $u^i =$

$dx^i/dt \ll 1$ and terms on the order of $|u|^2$ are neglected. Furthermore let us assume vanishing pressure within the body so that $p \approx 0$. Then in relativistic units $G = c = 1$ the elements of the energy-momentum tensor are:

$$T^{00} \approx \rho \tag{A.32}$$

$$T^{0i} \approx \rho u^i \tag{A.33}$$

$$T^{ij} \approx 0. \tag{A.34}$$

We use the slow-motion approximation, such that derivatives with respect to the time-like coordinate x^0 are small, of order ϵ, times the spatial derivatives:

$$\epsilon \frac{\partial f}{\partial x^a} = \frac{\partial f}{\partial x^0}. \tag{A.35}$$

Then to first order, the timelike component $a = b = 0$ of the linearized Einstein equations, Equation A.30, with the energy-momentum tensor, Equation A.32 is given by

$$\nabla^2 \psi_{00} = 16\pi\rho \tag{A.36}$$

Comparing this with Poisson's equation

$$\nabla^2 \phi = 4\pi\rho, \tag{A.37}$$

we identify the Newtonian potential ϕ,

$$\psi_{00} = 4\phi. \tag{A.38}$$

Note the trace of ψ_{ab} is $\psi = \psi_{00}$ since the diagonal components ψ_{11}, ψ_{22} and ψ_{33} vanish. Therefore, by Equation A.19 we have

$$h_{00} = h_{11} = h_{22} = h_{33} = 2\phi. \tag{A.39}$$

In general, the Newtonian potential can be expressed in terms of the density ρ. Since we are working in the stationary source limit we can safely neglect retarded time effects, so the solution to Equation A.36 can then be written as:

$$\phi(\boldsymbol{x}) = -\int \frac{\rho(\boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|} d^3\boldsymbol{y}, \tag{A.40}$$

where ρ enters from the energy-momentum tensor.

The mixed components with $a = 0$ and $b = i > 0$ depend on Equation A.33. The resulting form of Equation A.19 is

$$\nabla^2 \psi_{0i} = 4\pi \rho u_i. \tag{A.41}$$

We define the spatial vector $\boldsymbol{A}$ with the components ψ^{0i}, and the vector $\boldsymbol{u}$ with components u^i. Then we can rewrite Equation A.41 in the form

$$\nabla^2 \boldsymbol{A} = 4\pi \rho \boldsymbol{u}. \tag{A.42}$$

The solution to this equation is written component-wise as:

$$A^i(\boldsymbol{x}) = -4 \int \frac{\rho(\boldsymbol{y}) u^i(\boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|} d^3\boldsymbol{y}. \tag{A.43}$$

Then Equation A.19 gives

$$h_{0i} = A_i, \tag{A.44}$$

where we define the gravitational spatial vector potential A_i.

We can now write the metric of a weakly curved spacetime. The line element of the weak field metric with all factors of c replaced is given by Equation A.1

$$ds^2 = \left(1 + \frac{2\phi}{c^2}\right) c^2 dt^2 - 2\left(\frac{\boldsymbol{A} \cdot \boldsymbol{d\sigma}}{c}\right) cdt - \left(1 - \frac{2\phi}{c^2}\right) d\sigma^2, \tag{A.45}$$

with $d\sigma^2 = |\boldsymbol{d\sigma}|^2 = dx^2 + dy^2 + dz^2$. In general, the geodesics of this spacetime allow us to calculate the trajectories of test particles in the field of a non-relativistic source in the weak field limit. The metric does not require any assumptions about the nature or speed of the test particles. Therefore, setting $ds = 0$ allows us to find the null geodesics that define the paths of light rays.

The line element in Equation A.45 is more general than we need for the purpose of strong gravitational lens modeling. Static sources are defined as objects with constituent particles whose velocities can be completely neglected. When this is the case, all of the components of the perfect fluid energy momentum tensor vanish except for the timelike component, given by Equation A.32. This means that the vector potential also vanishes such that the metric is further simplified:

$$ds^2 = \left(1 + \frac{2\phi}{c^2}\right) c^2 dt^2 - \left(1 - \frac{2\phi}{c^2}\right) d\sigma^2 \tag{A.46}$$

This is often referred to as the line element in the Newtonian limit (Hobson et al. 2006). A variety of interesting phenomena can be described using this metric, including gravitational waves (Hulse & Taylor 1974), perihelion precession (Einstein 1916), time delay (Shapiro 1964), and gravitational lensing (Einstein 1936). With an expression for the weak field limit in hand, we can define the effective index of refraction of a gravitational field and from that result the deflection angle field due to massive objects.

A.2 Index of Refraction of a Gravitational Field

With the line element in the Newtonian limit, it is simple to derive the effect of weak gravitational fields on the paths of light rays. These paths through space-time are described as null geodesics, paths along which the line element (proper time) vanishes:

$$\left(1+2\frac{\phi}{c^2}\right)c^2dt^2 - \left(1-2\frac{\phi}{c^2}\right)d\sigma^2 = 0. \tag{A.47}$$

Defining $v = d\sigma/dt$, we find

$$v = c\left(\frac{1+2\frac{\phi}{c^2}}{1-2\frac{\phi}{c^2}}\right)^{-\frac{1}{2}}, \tag{A.48}$$

which we compare with the standard definition of the index of refraction n from classical optics which is determined by the relation $v = c/n$. We then have the expression

$$n = \left(\frac{1-2\frac{\phi}{c^2}}{1+2\frac{\phi}{c^2}}\right)^{\frac{1}{2}} \tag{A.49}$$

where the Newtonian potential ϕ is small. We expand this expression to first order in ϕ/c^2 to give our final expression for the index of refraction:

$$n = 1 - 2\frac{\phi}{c^2}. \tag{A.50}$$

Unlike the index of refraction of most materials, the gravitational index of refraction has no wavelength dependence. Note that $\phi \leq 0$ so that $n \geq 1$ always, which shows that light travels slower when traveling through a gravitational field than in free space.

A.3 Deflection Angle from the Weak Field Metric

In general, we can use the geodesic formula with the weak field metric to derive the gravitational lens deflection angle. However, a more illustrative derivation makes use of Fermat's principle in analogy with classical optics, using the gravitational index of refraction derived above (Schneider, Ehlers & Falco 1992). We note that for a static source, the gravitational index of refraction is a function of the spatial coordinates only, and in the weak field limit ϕ is small. Due to the large distances between the source, lens and observer, we approximate the path of a light ray as the piecewise path shown in Figure A.1. This approximation is applicable to all observable gravitational lens phenomena (Narayan & Bartelmann 1995).

Let us define a unit vector $\boldsymbol{e}$ that is tangent to the path of a light ray emitted from a source object S and received by an observer O. The lens is located at the origin on the lens plane, and the z axis is denoted by the unit vector $\hat{\boldsymbol{z}}$. Define the unit tangent vector to the undeflected ray $\boldsymbol{e}_S$ at the position of the source and $\boldsymbol{e}_O$ the tangent to the deflected light path at the observer. The deflection angle $\hat{\boldsymbol{\alpha}}$ is then given by

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{e}_S - \boldsymbol{e}_O. \tag{A.51}$$

This deflection angle vector approximates the amount of bending that the lens causes to the true path of the light ray (Petters et al. 2001). In general we seek an analytical description of $\hat{\boldsymbol{\alpha}}$ in the weak field limit. More details can be found in Petters et al. (2001) and Schneider, Ehlers & Falco (1992), the primary sources used in this derivation.

Let us consider the index of refraction of a weak gravitational field as defined in Equation A.50. As in classical optics, we define the optical path length as the physical path length multiplied by the index of refraction of the medium. Fermat's principle requires that this path is an extremum such that

$$\delta \int_A^B n(\boldsymbol{x}) dl = 0, \tag{A.52}$$

where $n(\boldsymbol{x})$ is the index of refraction as a function of position, and dl is a length element along the path. Let us write the path $\boldsymbol{x}$ as a curve parameterized by a

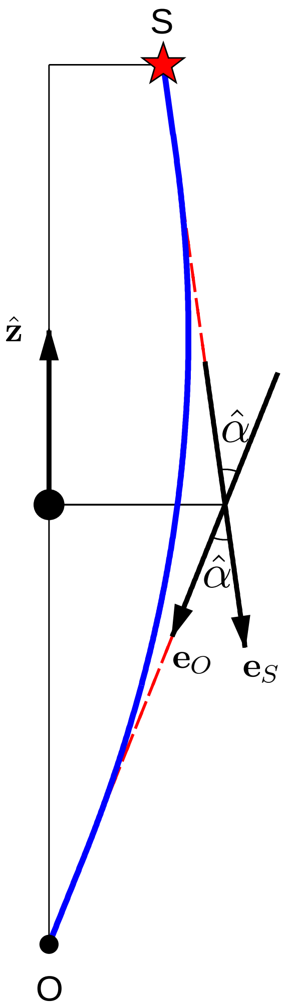

Figure A.1: Thin lens geometry

Thin lens geometry used to derive the deflection angle $\hat{\boldsymbol{\alpha}}$. The diagram is not to scale and the effect is greatly exaggerated to clearly denote the relevant quantities.

quantity λ such that

$$dl = \left|\frac{\boldsymbol{x}}{d\lambda}\right| d\lambda. \tag{A.53}$$

Equation A.52 then reduces to the parameterized integral

$$\delta \int_{\lambda_A}^{\lambda_B} n(\boldsymbol{x}(\lambda)) \left|\frac{d\boldsymbol{x}}{d\lambda}\right| d\lambda = 0, \tag{A.54}$$

from which we identify the Lagrangian

$$L(\dot{\boldsymbol{x}}, \boldsymbol{x}, \lambda) = n(\boldsymbol{x}) \left|\dot{\boldsymbol{x}}\right|, \tag{A.55}$$

where we denote $\boldsymbol{x}$=$\boldsymbol{x}(\lambda)$ and $\dot{\boldsymbol{x}} = d\boldsymbol{x}/d\lambda$, which is tangent to the path of the light ray.

We can then use the Euler-Lagrange equations

$$\frac{d}{d\lambda}\frac{\partial L}{\partial \dot{\mathrm{x}}} - \frac{\partial L}{\partial \boldsymbol{x}} = 0 \tag{A.56}$$

to find the extremum of Equation A.52, from which we determine

$$\frac{\partial L}{\partial \dot{\boldsymbol{x}}} = \frac{n\dot{\boldsymbol{x}}}{|\dot{\boldsymbol{x}}|} \tag{A.57}$$

and

$$\frac{\partial L}{\partial \boldsymbol{x}} = \nabla n \left|\dot{\boldsymbol{x}}\right|. \tag{A.58}$$

Putting Equations A.57 and A.58 together, we find

$$\frac{d}{d\lambda}\left(n\frac{\dot{\boldsymbol{x}}}{|\dot{\boldsymbol{x}}|}\right) - |\dot{\boldsymbol{x}}|\, \boldsymbol{\nabla} n = 0, \tag{A.59}$$

where we have used the definition of the gradient $\frac{\partial n}{\partial \boldsymbol{x}} = \boldsymbol{\nabla} n$. Let us now define the unit tangent vector $\boldsymbol{e} = \dot{\boldsymbol{x}}/\left|\dot{\boldsymbol{x}}\right|$, which we use to simplify Equation A.59:

$$\dot{\boldsymbol{e}} = \frac{1}{n}\dot{\boldsymbol{x}}\left[\nabla n - \boldsymbol{e}\left(\boldsymbol{e}\cdot\nabla n\right)\right]. \tag{A.60}$$

The second portion of the bracketed term is the gradient parallel to the path of the light ray. Therefore, we identify the perpendicular component of the gradient

$$\nabla_{\perp} n = \nabla n - \boldsymbol{e}\left(\boldsymbol{e}\cdot\nabla n\right), \tag{A.61}$$

from which it follows that

$$\dot{\boldsymbol{e}} = \frac{1}{n}\nabla_{\perp} n = \nabla_{\perp} \ln{(n)}. \tag{A.62}$$

We can Taylor expand the logarithm using Equation A.50 and the limit $\phi/c^2 \ll 1$, obtaining the expression

$$\dot{\boldsymbol{e}} \approx -\frac{2}{c^2}\left|\dot{\boldsymbol{x}}\right| \nabla_{\perp}\phi. \tag{A.63}$$

We integrate both sides of this expression from the source to the observer along the light ray, obtaining

$$\boldsymbol{e}_O - \boldsymbol{e}_S = -\frac{2}{c^2}\int_{\lambda_S}^{\lambda_O} \nabla_{\perp}\phi \left|\dot{\boldsymbol{x}}\right| d\lambda. \tag{A.64}$$

With the help of Equation A.51, this expression reduces to an equation for the deflection angle field in the weak field limit:

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{e}_S - \boldsymbol{e}_O = \frac{2}{c^2}\int_S^O \nabla_{\perp}\phi \; dl. \tag{A.65}$$

A more useful expression is obtained by recognizing that the lens is thin, which allows us to rewrite the path integral in Equation A.65 in terms of an integral along the line of sight z perpendicular to the lens plane (Petters et al. 2001):

$$\hat{\boldsymbol{\alpha}} = \frac{2}{c^2}\int_{-\infty}^{\infty} \nabla_{\perp}\phi \; dz. \tag{A.66}$$

Appendix B

Cosmology and Angular Diameter Distance

A thorough description of the background cosmology is required to model gravitational lens systems. Since the observed light rays from a source are emitted at a redshift z_s and the deflector is at redshift z_d, we must scale the deflection angle using the distances between these objects. To do this, we will make use of the Friedmann-Robertson-Walker (FRW) metric (Friedmann (1922); Robertson (1935); Walker (1937)) to describe the background cosmology. This metric allows us to derive a relationship between distance and the observable angular diameters of objects in curved spacetime on cosmological scales. The primary sources for these derivations are the discussions in Hobson et al. (2006), d'Inverno (1992) and Coles & Lucchin (2002).

The FRW metric is based on the cosmological principle that states the universe is globally homogeneous. If an observer's position were changed in a homogenous universe, on large scales their view of the cosmos would remain unchanged. In addition, the cosmological principle implies that spacetime at the largest scales should be isotropic, such that no privileged directions exist. These assumptions are supported by observations of the cosmic microwave background (CMB; Odenwald, Newmark and Smoot (1998); Spergel et al. (2003)), which provides evidence that homogeneity and isotropy are valid assumptions to include in cosmological models.

B.1 The Robertson-Walker Line Element

In order to include the cosmological principle in a quantitative model of the background metric, we divide spacetime into a series of space-like sheets in which $t = constant$ on each sheet. In relativistic units with $G = c = 1$, we write the line element of such a foliated spacetime as

$$ds^2 = dt^2 - h_{ab} dx^a dx^b, \tag{B.1}$$

where $h_{ab} = h_{ab}(t, \mathbf{x})$ such that a, b run from 1 to 3.

In fact, this metric can be further simplified by considering a simple physical argument. Suppose that we form a large triangle using three points at time t. At later times, the three points will describe a larger triangle. The cosmological principle requires that there are no unique points or directions in the spatial hypersurfaces, which implies that the two triangles must be geometrically similar. The factor by which the size of the triangle grows must also be independent of position or direction. Therefore, time can enter the metric only through a real-valued scale factor $b(t)$ so that the ratio of distances between points is the same at all times. Isotropy and homogeneity require the curvature of the space-like hypersurfaces to be constant in order for the triangles to remain geometrically similar for all times. Therefore, we introduce the scale factor into the spatial part of the metric such that

$$h_{ab} = b(t)^2 h'_{ab}(x^a). \tag{B.2}$$

With the metric in the form given by Equation B.2, it is possible to write the Riemann tensor describing the curvature of the spatial part of the line element in the form

$$R_{abcd} = K \left(g_{ac} g_{bd} - g_{ad} g_{bc} \right), \tag{B.3}$$

where K is a scalar curvature constant. The corresponding Ricci tensor is given by

$$R_{bd} = 2K g_{bd} \tag{B.4}$$

assuming spherical symmetry. These constraints determine the components of the spatial metric elements

$$
\begin{aligned}
d\sigma^2 &= h_{ab}dx^a dx^b \\
d\sigma^2 &= \tfrac{dr^2}{1-Kr^2} + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2
\end{aligned}
\tag{B.5}
$$

where we have adopted the standard spherical coordinates (r, θ, ϕ). There is a further simplification that can be made to the line element to remove the arbitrary magnitude of the curvature constant. We define $k = +1, 0, -1$ so that $K = |K|k$ and rescale the radial coordinate

$$r = \frac{\bar{r}}{|K|^{\frac{1}{2}}}. \tag{B.6}$$

The full line element is then written, dropping the barred notation, as

$$ds^2 = dt^2 - a(t)^2 \left(\frac{dr^2}{1-kr^2} + r^2 d\theta^2 + r^2 \sin{(\theta)} d\phi^2 \right), \tag{B.7}$$

where the scale factor is defined such that

$$a(t) = \frac{b(t)}{|K|^{\frac{1}{2}}} \tag{B.8}$$

and

$$a(t) = b(t) \tag{B.9}$$

for $K = 0$. The significance of the curvature constant k is discussed below in terms of the solutions to Friedmann's equations.

B.2 The Friedmann Equations

Now that we have defined the elements of the metric, we use Einstein's equations to derive a pair of equations that determine the evolution of the universe (Friedmann 1922). Consider Einstein's equations with a cosmological constant term added:

$$G_{ab} - \Lambda g_{ab} = \kappa T_{ab}, \tag{B.10}$$

where Λ is the cosmological constant. Equation B.10 is the general form of Einstein's field equations. In early treatments the cosmological constant was simply set to 0, but as cosmological models were explored it was found that this term significantly affects the evolution of the resulting solutions. The cosmological constant was first used by Einstein in order to produce a static universe, and acts in general as a repulsive term to counteract the effects of gravity on large scales (Einstein 1917). Hubble's law is an empirical relationship between the proper distance d and recessional velocity of galaxies v, such that

$$v = H_0 d \tag{B.11}$$

for small z, where H_0 is Hubble's constant. In view of Hubble's observations (Hubble 1936), Einstein realized that a static universe was impossible, and regarded the cosmological constant as his "greatest blunder". However, recent observations of supernovae suggest that a cosmological constant term must be present to explain the apparent acceleration of the universe (Riess et al 1998).

In order to solve the field equations with the FRW metric, we specify a momentum-energy tensor that describes the distribution of matter-energy in the universe. To describe this mass distribution we make use of Weyl's postulate, which states that matter in the universe follows time-like geodesics that diverge from a finite time in the past and possibly intersect at a time in the finite future. While the relative velocities between individual particles may be large, the particles themselves are essentially stationary relative to a comoving coordinate system that expands with the universe. To approximate the matter distribution, we consider galaxies interacting in analogy with particles in a perfect fluid. By Weyl's postulate we work in a comoving spherical polar coordinate system, with the velocity of each particle dominated by the expansion such that the comoving velocity components of the fluid particles are $[u^a] = (1, 0, 0, 0)$ (d'Inverno 1992). Putting these arguments together, the Einstein field equations can be reduced to the cosmological field equations:

$$\dot{a}(t)^2 = \frac{8\pi}{3}\rho a(t)^2 + \frac{\Lambda}{3}a(t)^2 - k, \tag{B.12}$$

$$\ddot{a}(t) = -\frac{4\pi}{3}\left(\rho + 3p\right) a(t) + \frac{\Lambda}{3}a(t). \tag{B.13}$$

The first of these equations is found from the G_{00} component of the field equations. The remaining spatial field equations give rise to degenerate expressions that are equal to the second equation above. These relationships are known as the Friedmann equations. Given values for the cosmological constant, matter density and curvature, these equations determine the behavior of the scale factor a as a function of cosmic time.

In general the Friedmann equations permit several types of solution. When $\Lambda = 0$, setting the curvature $k = 1$ produces a closed universe, which contains a physical singularity at points in the finite past and future. The $k = 0$ solution represents a flat universe with an ever decreasing expansion velocity. Solutions with $k = -1$ describe open universes that continue to expand forever. It is worth noting that all of these solutions have singularities at finite points in the past, though the cases with $k \neq 1$ do not contain corresponding singularities in the future. The presence of a non-vanishing cosmological constant introduces additional effects that alter the expansion of the universe by introducing a repulsive force that changes the overall behavior of the solution as a function of cosmic time.

The Friedmann equations can be written in a normalized form that depends on measurable quantities. Define the current scale of the universe $a(t_0) = a_0$ and write

$$R = \frac{a(t)}{a_0} \tag{B.14}$$

and the derivative of this quantity with respect to time,

$$\dot{R} = \frac{\dot{a}(t)}{a_0}, \tag{B.15}$$

then $R(t_0) = 1$ and $\dot{R}(t_0) = H_0$. The Friedmann equations become:

$$\dot{R}^2 = \frac{8\pi}{3}\rho R^2 + \frac{\Lambda}{3}R^2 - \frac{k}{a_0^2} \tag{B.16}$$

$$\ddot{R} = -\frac{4\pi}{3}\left(\rho + 3p\right)R + \frac{\Lambda}{3}R. \tag{B.17}$$

It is useful to restate the Friedmann equations in terms of energy density. To derive the equation of motion of the cosmological fluid, we eliminate $\ddot{R}$ in Equation B.17. This gives the following relation:

$$\dot{\rho} = -3\left(\rho + p\right)\frac{\dot{R}}{R} \tag{B.18}$$

in terms of the normalized scale factor R. This equation is known as the continuity equation, which is a consequence of the conservation of energy.

So far we have not explored the nature of the fluid density except to say that it behaves as a perfect fluid. In fact, there are several kinds of energy density that influence the evolution of the universe. The total density ρ contains contributions from matter, ρ_m, radiation ρ_r, and the energy of the vacuum itself, which we identify with the cosmological constant.

We derive the properties of matter and radiation by thermodynamic arguments. We adopt the equation of state $p = \frac{\gamma}{3}\rho$ where radiation has $\gamma = 1$ and matter $\gamma = 0$. We rewrite Equation B.18 in terms of the density ρ:

$$\frac{d}{dt}\left[\rho R^{3+\gamma}\right] = 0. \tag{B.19}$$

We then integrate this equation from t to t_0, the current time. For the matter density, this gives

$$\rho_m(t) = \frac{\rho_{0m}}{R^3}, \tag{B.20}$$

The radiation density is then given by

$$\rho_r(t) = \frac{\rho_{r0}}{R^4}. \tag{B.21}$$

The cosmological constant can be treated as a perfect fluid component with constant density and equation of state $p = -\rho$, corresponding to $\gamma = -3$. Equation B.19 ensures that this choice of γ does not introduce time dependence to the vacuum energy density and allows the cosmological constant to act like a fluid with negative pressure in the Friedmann equations. We define the energy density of the vacuum

$$\rho_\Lambda = \rho_{0\Lambda} = \frac{\Lambda}{8\pi}, \tag{B.22}$$

where the factor of 8π is introduced to simplify Equation B.12, and allows us to treat the cosmological constant term on equal footing with the other contributions to the total density such that $\rho = \rho_m + \rho_r + \rho_\Lambda$.

Now let us consider the density necessary to produce a flat universe. Setting $k = 0$ in Equation B.16, we can solve for the corresponding critical density at t_0,

$$\rho_c = \frac{3H_0^2}{8\pi}, \tag{B.23}$$

where H_0 is the value of the Hubble constant at the current epoch t_0. This critical density causes the gravitational attraction of matter and the expansion of spacetime to be exactly balanced, producing a flat universe. Therefore, this quantity makes a useful scaling factor that allows us to define the fractional density of matter $\Omega_m = \rho_m/\rho_c$, radiation, $\Omega_r = \rho_r/\rho_c$ and the cosmological constant energy density $\Omega_\Lambda = \rho_\Lambda/\rho_c$. Define the curvature density at t_0,

$$\Omega_{0k} = -\frac{k}{a_0^2 H_0^2}. \tag{B.24}$$

In terms of the critical density, Equation B.16 becomes

$$\dot{R}^2 = H_0^2 \left[\frac{\Omega_{0m}}{R^3} + \frac{\Omega_{0r}}{R^4} + \Omega_{0\Lambda} + \frac{\Omega_{0k}}{R^2}\right] R^2. \tag{B.25}$$

To simplify this notation, define

$$\Upsilon(R) = \sqrt{\frac{\Omega_{0m}}{R^3} + \frac{\Omega_{0r}}{R^4} + \Omega_{0\Lambda} + \frac{\Omega_{0k}}{R^2}} \tag{B.26}$$

so that the Friedmann equation takes on an even more compact form

$$\dot{R} = H_0 R \Upsilon(R). \tag{B.27}$$

The utility of this formulation is that the densities and Hubble constant H_0 are observable in the present day universe at time t_0. In fact, the currently accepted values of these parameters have been estimated using several methods, including observations of the CMB (Spergel et al. (2003); Jarosik et al. (2011)), high redshift supernovae (Riess et al (1998); Perlmutter et al. (1999)), and large scale lensing observations (Refregier et al. (2004); Contaldi et al. (2003); Grillo et al. (2008)). The cosmological parameters are now known to an accuracy of a few percent (Coles & Lucchin 2002), and the standard cosmological model uses $\Omega_{0m} \approx 0.3$, $\Omega_{0r} \approx 10^{-5}$ and $\Omega_{0\Lambda} \approx 0.7$ with $H_0 = 70$ km s^{-1}. The radiation density is negligible at t_0 but played an important role at early times in the evolution of the universe. We adopt these values for the cosmological parameters needed in our lens models.

Evaluating Equation B.25 at $t = t_0$, we find that

$$\Omega_{0m} + \Omega_{0r} + \Omega_{0\Lambda} + \Omega_{0k} = 1. \tag{B.28}$$

Since the measured cosmological parameters have $\Omega_{0m} + \Omega_{0r} + \Omega_{0\Lambda} \approx 1$, this implies $\Omega_{0k} \approx 0$ and by Equation B.24, the universe must be geometrically flat with $k \approx 0$.

B.3 Angular Diameter Distances

The most relevant method for determining cosmological distances with the FRW metric in the context of gravitational lens modeling are angular diameter distances (Peacock 1998). In the Euclidean case, the standard angular diameter distance formula holds for small angles:

$$d = D\Theta, \tag{B.29}$$

where D is the distance at which an object of diameter d will appear to subtend an angle Θ. In the curved spacetime of GR, we define this relationship to be true so that angles and distances are related in the usual way. To find the corresponding distance relationship using the FRW metric, consider two light rays that travel along radial null geodesics from a source of diameter d at redshift z separated by an angle Θ. Suppose that the rays are emitted at time t_E and observed at t_0. From the angular part of the FRW metric, we have

$$d = a(t_E)r\Theta. \tag{B.30}$$

By equating Equations B.29 and B.30, we find the distance in terms of the scale factor and the radial coordinate,

$$D = a(t_E)r. \tag{B.31}$$

Equation B.14 can be used to simplify this result. The normalized scale factor is related to the redshift of the object at the time of emission t_E. To determine the form of this relationship, consider the cosmological redshift effect. In general, the emission of radiation with frequency ν_E from an object at redshift z will be observed to have frequency

$$\nu_0 = \frac{\nu_E}{1+z}. \tag{B.32}$$

This redshift occurs due to the expansion of the universe linearly stretching the wavelength of the emitted light. Thus we can relate the emitted and observed wavelengths

$$\frac{\nu_0}{\nu_E} = \frac{\lambda_E}{\lambda_0}, \tag{B.33}$$

and due to the linear dependence of wavelength on scale factor (Hobson et al. 2006)

$$R(t_E) = \frac{a(t_E)}{a_0} = \frac{1}{1+z}. \tag{B.34}$$

Rewriting Equation B.31 gives

$$D = \frac{a_0}{1+z} r \tag{B.35}$$

for a source at redshift z and observer at the origin. We now need to find an expression for the radial distance r along the path of the rays. Since each of the geodesics is radial, we set $ds = 0$ and eliminate the angular portion of the line element in Equation B.7. From the line element we find:

$$\int_{t_E}^{t_0} \frac{dt}{a(t)} = \int_0^r \frac{dr'}{(1-kr'^2)^{\frac{1}{2}}}. \tag{B.36}$$

The right hand side of this expression can be solved for three cases that depend on the curvature constant k:

$$\int_{t_E}^{t_0} \frac{dt}{a(t)} = \begin{cases} \sin^{-1}(r), & k=1 \\ r, & k=0 \\ \sinh^{-1}(r), & k=-1. \end{cases} \tag{B.37}$$

Note that the left hand side of Equation B.36 depends on the specific form of the scale factor $a(t)$. In order to find the radial distance r along a null geodesic in terms of observable quantities, it is necessary to calculate this integral for a given curvature. To do this, we return to the definition of the normalized scale factor given by Equation B.14. Using this relationship we rewrite the left hand side of Equation B.36 in the form

$$\int_{t_E}^{t_0} \frac{dt}{a(t)} = \frac{1}{a_0} \int_{R(t_E)}^{R(t_0)} \frac{dR}{R(t)} \frac{1}{\dot{R}}. \tag{B.38}$$

We then use the Friedmann equation given by Equation B.27 to write

$$f(r) = \frac{1}{H_0 a_0} \int_{R(t_E)}^{R(t_0)} \frac{dR}{R^2 \Upsilon(R)}, \tag{B.39}$$

where we have defined $f(r)$ using Equation B.38. This relationship between scale factor and redshift given in Equation B.34 can be used to convert this integral to an integration over the redshift interval z, using the integration element

$$dz = -\frac{dR}{R^2}. \tag{B.40}$$

Then we have

$$f(r) = \frac{1}{H_0 a_0} \int_0^z \frac{dz}{\Upsilon(z)}, \tag{B.41}$$

where

$$\Upsilon(z) = \left[\Omega_{0m}(1+z)^3 + \Omega_{0r}(1+z)^4 + \Omega_{0\Lambda} + \Omega_{0k}(1+z)^2\right]^{\frac{1}{2}}. \tag{B.42}$$

The angular diameter distance, Equation B.31, can then be evaluated using the relationship in Equation B.41. In general we write the result in terms of arbitrary redshifts $z_1 < z_2$. For $k = 0$ and reinserting factors of c and G, we have

$$D = \frac{1}{1+z_2} \frac{c}{H_0} \int_{z_1}^{z_2} \frac{dz}{\Upsilon(z)} \tag{B.43}$$

by the result of Equation B.37. We can simplify the results for $k = \pm 1$ by rewriting a_0 in terms of the total matter density. From Equation B.24 we have

$$a_0 = \frac{c}{H_0 \sqrt{(\mp \Omega_{0k})}} \tag{B.44}$$

taking into account the sign difference between k and Ω_{0k}. For the $k = 1$ case we have

$$D = \frac{1}{1+z_2} \frac{c}{H_0 \sqrt{-\Omega_{0k}}} \sin\left(\sqrt{-\Omega_{0k}} \int_{z_1}^{z_2} \frac{dz}{\Upsilon(z)}\right) \tag{B.45}$$

and the $k = -1$ case gives

$$D = \frac{1}{1+z_2} \frac{c}{H_0 \sqrt{\Omega_{0k}}} \sinh\left(\sqrt{\Omega_{0k}} \int_{z_1}^{z_2} \frac{dz}{\Upsilon(z)}\right). \tag{B.46}$$

In general, these integrals must be solved numerically.

Gravitational lens modeling requires three angular diameter distances: the distance D_s between the observer at redshift $z_o = 0$ and the lensed source at z_s, the distance D_d between observer and the lens at z_d, and the distance between lens and source D_{ds}. We calculate these distances using the standard cosmological parameters in Equation B.43, where we have assumed a flat geometry ($k = 0$).

Appendix C

Conjugate Gradient Optimization Methods

Modeling the deconvolved source intensity distributions of strong gravitational lenses requires the inversion of large linear systems. In principle, these systems can be solved by direct inversion or factorized expansion such as SVD. However, in practice these direct approaches can be computationally intensive for large matrices. A more efficient approach to matrix inversion treats the inversion process as a minimization problem. This allows us to make use of fast linear iterative optimization routines. Furthermore, the matrices to be inverted may be ill-posed and require explicit regularization. This appendix discusses regularization methods that incorporate regularization automatically as convergence occurs. We focus our discussion on two of the most efficient optimization algorithms for large ill-posed problems that require regularization, the steepest descent (SD) and the conjugate gradient (CG) methods.

We consider the SD and CG methods in the context of symmetric positive definite matrices, but we show that these approaches can be extended to general linear systems without the need for these strict symmetry requirements by stating the inversion in the context of least-squares fitting. We follow the derivations of these routines closely from the treatments in Shewchuk (1994) and Björck (1996). Gravitational lens inversion using the semilinear method requires linear optimization, so we focus on these approaches. However, in general the methods discussed here can be extended to include non-linear functions as well.

C.1 Steepest Descent Method

Consider an $N \times N$ symmetric, positive definite matrix $\boldsymbol{A}$ such that $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} > 0$ for an $\boldsymbol{x} \in \mathbf{R}^N$. We seek a solution to the linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{C.1}$$

However, it may not be practical to invert $\boldsymbol{A}$ directly when $\boldsymbol{A}$ is large or poorly conditioned. In such cases, solutions can still be found by treating Equation C.1 as a minimization problem, which can be solved by efficient iterative methods. Consider the N dimensional quadratic function defined by

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x} + c, \tag{C.2}$$

where c is an arbitrary constant that we set to zero without loss of generality. The minimum of this function, $\boldsymbol{x}_f$ is the solution to the system in Equation C.1 such that $\boldsymbol{A}\boldsymbol{x}_f = \boldsymbol{b}$. The N dimensional gradient of this function with respect to the coordinate vector $\boldsymbol{x}$ can be written

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}. \tag{C.3}$$

Consider an iterative algorithm where we start at some point $\boldsymbol{x}_0$ and take m successive steps using local information to choose the next step direction and distance at each iteration. We visit the set of points $\boldsymbol{x}_i$, $i \in [0, m]$, with the goal to end up as close as possible to the minimum such that $\boldsymbol{x}_m \approx \boldsymbol{x}_f$ on the final iteration. In general the gradient at step i points in the direction of steepest increase of $f(\boldsymbol{x}_i)$. Therefore, the direction in which the function decreases the most rapidly is the direction opposite to the gradient.

We define the error vector of the i^{th} step as

$$\boldsymbol{e}_i = \boldsymbol{x}_i - \boldsymbol{x}_f, \tag{C.4}$$

which indicates how far we are from the true solution $\boldsymbol{x}_f$. Furthermore, let us define the residual vector

$$\boldsymbol{r}_i = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_i, \tag{C.5}$$

where the residual vector is related to the error vector by $\boldsymbol{r}_i = -\boldsymbol{A}\boldsymbol{e}_i$. From the definition of the residual, we have

$$\boldsymbol{r}_i = -\boldsymbol{\nabla} f(\boldsymbol{x}_i), \tag{C.6}$$

which shows that the residual therefore provides the direction of steepest descent. The step we take from the i^{th} position $\boldsymbol{x}_i$ must lie along the line defined by the gradient direction. However, we need to find the size of the step to take to find the next location $\boldsymbol{x}_{i+1}$, which must satisfy

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{r}_i. \tag{C.7}$$

We choose the value of α_i that minimizes $f(\boldsymbol{x}_i)$ along this line, which is found at the position

$$\frac{df(\boldsymbol{x}_{i+1})}{d\alpha_i} = \boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})^T \frac{d\boldsymbol{x}_{i+1}}{d\alpha_i} = 0 \tag{C.8}$$

using the chain rule. Combining this result with eq. C.7, we see that $\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})^T \boldsymbol{r}_i = 0$. Therefore α_i should be chosen such that the gradient $\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})$ and the residual $\boldsymbol{r}_i$ are orthogonal. We find the value of α_i analytically:

$$\boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})^T \boldsymbol{r}_i = \boldsymbol{r}_{i+1}^T \boldsymbol{r}_i = (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{i+1})^T \boldsymbol{r}_i = 0. \tag{C.9}$$

Using Equation C.7, we rearrange this expression to solve for the stepsize α_i in terms of the residual vector $\boldsymbol{r}_i$:

$$\alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_i^T \boldsymbol{A} \boldsymbol{r}_i}. \tag{C.10}$$

Equations C.5, C.7 and C.10 define the SD algorithm as an iterative procedure. We find the residual at the current step i from Equation C.5, calculate α_i using Equation C.10, and update the position $\boldsymbol{x}_{i+1}$ from Equation C.7. In general, two matrix-vector multiplications are needed per iteration for the SD method. However we can save one of these multiplication operations by finding the $i+1$ residual with the step size:

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A} \boldsymbol{r}_i. \tag{C.11}$$

C.2 Conjugate Gradient Method

There is no constraint on the search directions in the SD method since the direction of steepest descent is determined by the gradient of $f(\boldsymbol{x}_i)$ from Equation C.6. In practice, the SD method will repeat steps in directions that it has already searched through. The idea behind the Conjugate Gradient (CG) method (Hestenes and Stiefel 1952) is to avoid searching in directions that have already been explored. If we define a set of directions $\boldsymbol{d}_0$, $\boldsymbol{d}_1$, ... $\boldsymbol{d}_{N-1}$ such that these directions are conjugate to one another, then we will only need to take one step along each $\boldsymbol{d}$ to end up at the solution $\boldsymbol{x}_f$.

Suppose that the conjugate vectors $\boldsymbol{d}$ satisfy the $\boldsymbol{A}$-orthogonality condition

$$\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_j = 0 \tag{C.12}$$

for $i \neq j$. We find the new position at step $i+1$ using

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{d}_i, \tag{C.13}$$

which leads to the expression for the residual,

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A} \boldsymbol{d}_i. \tag{C.14}$$

To find the expression for the step size α_i, consider the derivative of $f(\boldsymbol{x}_{i+1})$ with respect to the step size, analogous to the step size calculation in the SD method. We find

$$\frac{df(\boldsymbol{x}_{i+1})}{d\alpha_i} = \boldsymbol{\nabla} f(\boldsymbol{x}_{i+1})^T \frac{d\boldsymbol{x}_{i+1}}{d\alpha_i} = -\boldsymbol{r}_{i+1}^T \boldsymbol{d}_i = 0. \tag{C.15}$$

We use Equation C.14 to solve for the step size, such that

$$\alpha_i = \frac{\boldsymbol{d}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_i^T \boldsymbol{A} \boldsymbol{d}_i}, \tag{C.16}$$

which is analogous to Equation C.10 for the SD method and represents the step size along each conjugate direction at iteration i. To recover the SD method, we simply replace $\boldsymbol{d}_i$ with $\boldsymbol{r}_i$ so that we once again search along residual directions rather than conjugate directions.

Let us briefly return to the error vector defined in Equation C.4, which is related to the residual $\boldsymbol{r}_i = -\boldsymbol{A}\boldsymbol{e}_i$. When we move along the conjugate directions $\boldsymbol{d}_i$, the residual is given by Equation C.14 and therefore the error vector has the form

$$\boldsymbol{e}_{i+1} = \boldsymbol{e}_i + \alpha_i \boldsymbol{d}_i. \tag{C.17}$$

This expression can be written as a sum in terms of the first error vector $\boldsymbol{e}_0$,

$$\boldsymbol{e}_{i+1} = \boldsymbol{e}_0 + \sum_{j=0}^{i} \alpha_j \boldsymbol{d}_j \tag{C.18}$$

with $\alpha_0 = 0$. Suppose that we consider a set of N conjugate steps along the set of $\boldsymbol{d}_i$ directions. The error is reduced after each step, as $\boldsymbol{x}_i$ gets closer to the solution $\boldsymbol{x}_f$. We want the error vector $\boldsymbol{e}_N$ to vanish after N steps. Using Equation C.18, we write the initial error vector as

$$\boldsymbol{e}_0 = -\sum_{j=0}^{N-1} \alpha_j \boldsymbol{d}_j. \tag{C.19}$$

Inserting this expression back into Equation C.18 gives

$$\boldsymbol{e}_i = -\sum_{j=i}^{N-1} \alpha_j \boldsymbol{d}_j, \tag{C.20}$$

which means that each error vector must be A-orthogonal to all of the previous error vectors. At each iteration, we proceed along a unique search direction, removing an A-orthogonal term of the sum in Equation C.19 and eventually reducing the error to 0 after N steps. This has significant implications for the set of residuals found when stepping along these search directions as well. Consider multiplying Equation C.20 by $\boldsymbol{d}_k^T \boldsymbol{A}$ with $i > k$:

$$\boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{e}_i = -\sum_{j=i}^{N-1} \alpha_j \boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_j = 0 \tag{C.21}$$

by the A-orthogonality of the search directions, which implies

$$\boldsymbol{d}_k^T \boldsymbol{r}_i = 0 \tag{C.22}$$

for $i > k$ since $\boldsymbol{r}_i = -\boldsymbol{A}\boldsymbol{e}_i$ by Equation C.21.

To find the set of conjugate search directions $\boldsymbol{d}_i$, we consider a conjugation process similar to the Gram-Schmidt orthogonalization procedure on the set of residual (gradient) vectors. For $i = 0$ we take $\boldsymbol{d}_0 = \boldsymbol{r}_0$, reducing the first move to a SD step. Subsequent mutually conjugate steps are chosen such that

$$\boldsymbol{d}_i = \boldsymbol{r}_i + \sum_{k=0}^{i-1} \beta_{ik} \boldsymbol{d}_k \tag{C.23}$$

with the projection coefficient β_{ik}. We note that the conjugate search directions are built from linear combinations of the previous residual vectors and search directions. If we consider the inner product between C.23 and a previous residual $\boldsymbol{r}_j$ with $i > j$, we find that

$$\boldsymbol{d}_i^T \boldsymbol{r}_j = \boldsymbol{r}_i^T \boldsymbol{r}_j + \sum_{k=0}^{i-1} \beta_{ik} \boldsymbol{d}_k^T \boldsymbol{r}_j, \tag{C.24}$$

which implies that the residuals must also form an orthogonal set along the search directions by Equation C.22 so that

$$\boldsymbol{r}_i^T \boldsymbol{r}_j = 0. \tag{C.25}$$

When $j = i$ in Equation C.24, we are left with the useful relation

$$\boldsymbol{d}_i^T \boldsymbol{r}_i = \boldsymbol{r}_i^T \boldsymbol{r}_i. \tag{C.26}$$

To find the projection factor β_{ij}, we take the inner product of Equation C.23 with $\boldsymbol{A}\boldsymbol{d}_j$ such that

$$\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_j = \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j + \sum_{k=0}^{i-1} \beta_{ik} \boldsymbol{d}_k^T \boldsymbol{A}\boldsymbol{d}_j. \tag{C.27}$$

By the conjugacy of the search directions all terms in the sum except for the $k = j$ term vanish. For $i > j$, this simplifies to

$$0 = \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j + \beta_{ij} \boldsymbol{d}_j^T \boldsymbol{A}\boldsymbol{d}_j. \tag{C.28}$$

Solving for β_{ij}, we find

$$\beta_{ij} = -\frac{\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j}{\boldsymbol{d}_j^T \boldsymbol{A}\boldsymbol{d}_j}. \tag{C.29}$$

This expression can be further simplified using the iterative form of the residual. Consider the inner product of $\boldsymbol{r}_i$ and Equation C.14:

$$\boldsymbol{r}_i^T \boldsymbol{r}_{j+1} = \boldsymbol{r}_i^T \boldsymbol{r}_j - \alpha_j \boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j. \tag{C.30}$$

For $j = i - 1$ this expression simplifies due to Equation C.25:

$$\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_{i-1} = -\frac{1}{\alpha_{i-1}} \boldsymbol{r}_i^T \boldsymbol{r}_i. \tag{C.31}$$

Note that this is the numerator of the projection coefficient. For all other values of $j \neq i$, we have

$$\boldsymbol{r}_i^T \boldsymbol{A}\boldsymbol{d}_j = 0 \tag{C.32}$$

due to the orthogonality of the residuals. Substituting Equation C.31 into Equation C.29 gives

$$\beta_{i,i-1} = \frac{1}{\alpha_{i-1}} \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_{i-1}^T \boldsymbol{A}\boldsymbol{d}_{i-1}} \tag{C.33}$$

and $\beta_{ij} = 0$ for $i > j + 1$. This choice of projection coefficient ensures that the conjugacy condition $\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_{i-1} = 0$ is always satisfied.

The conjugate gradient method makes an efficient iterative routine since the majority of these coefficients vanish. Using the residual and search directions to find the next step in a conjugate direction, we can carry out the Gram-Schmidt conjugation process without the need to store the entire set of search directions $\boldsymbol{d}$. With the definition of the step size α, Equation C.16, we are left with

$$\beta_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_{i-1}^T \boldsymbol{r}_{i-1}}, \tag{C.34}$$

where $\beta_i = \beta_{i,i-1}$.

The iterative CG method is stated by the following steps.

1.) To initialize the algorithm we pick an arbitrary starting point, $\boldsymbol{x}_0$. Using this solution we find the residual vector $\boldsymbol{r}_0$ and take this as the first search direction (making the first step a SD step):

$$\boldsymbol{d}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0. \tag{C.35}$$

For $i \geq 0$, we iterate the following steps.

2.) Find the step size in the current search direction $\boldsymbol{d}_i$,

$$\alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{d}_i^T \boldsymbol{A}\boldsymbol{d}_i}. \tag{C.36}$$

3.) Take a step α_i along the current search direction to update the current position

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{d}_i. \tag{C.37}$$

4.) Use the step size and search direction to update the residual vector

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\boldsymbol{d}_i. \tag{C.38}$$

5.) Find the new search direction $\boldsymbol{d}_{i+1}$ by Gram-Schmidt conjugation on the residual and the previous $\boldsymbol{d}$ vectors to find the new conjugate search direction:

$$\beta_{i+1} = \frac{\boldsymbol{r}_{i+1}^T \boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^T \boldsymbol{r}_i} \tag{C.39}$$

$$\boldsymbol{d}_{i+1} = \boldsymbol{r}_{i+1} + \beta_{i+1} \boldsymbol{d}_i. \tag{C.40}$$

6.) Return to the second step and iterate the procedure, which is complete after N iterations, where N is the dimension of $\boldsymbol{A}$.

C.3 A Two-Dimensional Example

As a simple example, consider the $N = 2$ dimensional system defined by $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ that has

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} \tag{C.41}$$

and

$$\boldsymbol{b} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} \tag{C.42}$$

with solution

$$\boldsymbol{x}_f = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{C.43}$$

To begin, we choose an initial starting position at

$$\boldsymbol{x}_0 = \begin{bmatrix} 3 \\ -3 \end{bmatrix}. \tag{C.44}$$

The system expressed as a quadratic function is shown in the contours of Figure C.1 which has a minimum at $\boldsymbol{x}_f$. The path of the SD algorithm is shown in red, which repeats steps in previous search directions as it closes in on the minimum. However, the CG routine follows the blue path, which is more direct. For simple linear examples like this one, the CG method can find the solution in exactly N steps.

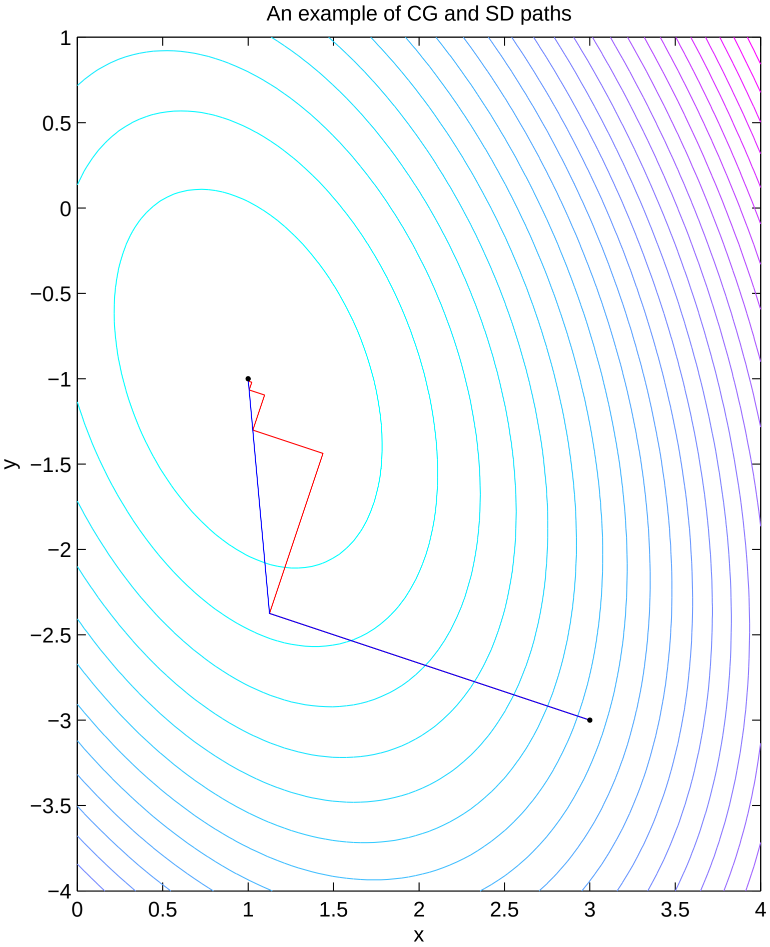

Figure C.1: An illustration of the SD and CGLS methods

Level curves of the quadratic function to be minimized are shown along with the SD path in red and the CG path in blue. For simple problems the CG path converges in N steps as illustrated.

C.4 Conjugate Gradient Method for Least Squares Problems

In developing the CG routine, we assumed that the matrix $\boldsymbol{A}$ is symmetric and positive definite. In fact, the method can be extended to systems in which these assumptions do not hold. Consider the least squares problem, where we seek the solution that minimizes

$$\min ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2. \tag{C.45}$$

Requiring that the derivative of this equation with respect to $\boldsymbol{x}$ vanish, we find the normal equations:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}. \tag{C.46}$$

The conjugate gradient method for least squares problems (CGLS) is simply the CG method applied to the normal equations. The matrix $\boldsymbol{A}^T\boldsymbol{A}$ is guaranteed to satisfy the positive definite and symmetry requirements of the CG method. In general, $\boldsymbol{A}$ does not even need to be square, as we are always guaranteed that $\boldsymbol{A}^T\boldsymbol{A}$ will be. In this case, the system may have many solutions and CGLS is not guaranteed to converge in exactly N iterations the way that CG did for a simple system.

The CGLS method is frequently used on ill-posed problems (Hansen 2010), where the determinant of $\boldsymbol{A}$ is very small. This is often the case in image processing applications where CGLS and associated methods are widely employed. To reduce numerical error in the calculations, the full matrix $\boldsymbol{A}^T\boldsymbol{A}$ does not need to be explicitly formed and matrix-vector multiplications are carried out in sequence instead. For example, rather than finding $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}$, we find $\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x})$. This allows us to sidestep explicitly forming $\boldsymbol{A}^T\boldsymbol{A}$ which is often less sparse than $\boldsymbol{A}$ and reduces numerical error in general.

Another convenient property of the CGLS algorithm is that it can be used even without an explicit representation of the matrix $\boldsymbol{A}$ since only matrix-vector multiplications are needed. If it is possible to describe the effect of $\boldsymbol{A}$ on an arbitrary vector with a subroutine, this can be used in place of an explicit multiplication resulting in a fast implementation if $\boldsymbol{A}$ is large and non-sparse. This technique has significance for image processing and gravitational lensing applications, which we exploit in Chapter

3 to build a gravitational lens inversion algorithm that is very efficient for large images. In addition to these convenient properties, CGLS has an advantage over other optimization schemes in solving ill-posed problems due to its convergence properties. The convergence behavior of the CGLS and SD methods are explored in the context of gravitational lensing problems in Chapter 3.

The CG and CGLS methods belong to a class of optimization methods called Krylov subspace methods (Björck 1996). In CG, each iteration of the optimization process searches along a new direction $\boldsymbol{d}_i$, and the residuals $\boldsymbol{r}_i$ are combinations of the previous residual and search direction. Therefore each iteration explores a new subspace D_i, which is spanned by the set of the previous search directions:

$$D_i = span\left(\boldsymbol{d}_0, \boldsymbol{A}\boldsymbol{d}_0, \boldsymbol{A}^2\boldsymbol{d}_0, ...\boldsymbol{A}^{i-1}\boldsymbol{d}_0\right). \tag{C.47}$$

The Krylov subspace D_i is formed by the repeated multiplication of $\boldsymbol{A}$ to the initial search direction $\boldsymbol{d}_0$. An illustration of the Krylov subspace is shown in Figure C.2. This subspace approach has a number of useful properties. First, the structure of the Krylov subspace simplifies the Gram-Schmidt process since it guarantees that each new residual is orthogonal to the previous search directions. Second, it provides a regularizing effect when solving the least squares problem using CGLS, which can be seen from the following argument. Let us assume an initial position $\boldsymbol{x}_0 = \boldsymbol{0}$. Since the first search direction in CGLS is a SD step we can write $\boldsymbol{d}_0 = \boldsymbol{A}^T\boldsymbol{b}$. The k^{th} iteration produces a solution $\boldsymbol{x}_k$ which has an expansion in the basis of the Krylov subspace D_k:

$$\boldsymbol{x}_k = c_1\boldsymbol{A}^T\boldsymbol{b} + c_2(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{A}^T\boldsymbol{b} + ... + c_k(\boldsymbol{A}^T\boldsymbol{A})^{k-1}\boldsymbol{A}^T\boldsymbol{b} \tag{C.48}$$

where the c factors are constant expansion coefficients. Now consider the SVD introduced in Chapter 3,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{C.49}$$

Using the orthogonality of $\boldsymbol{U}$ and $\boldsymbol{V}$, we have the relationship $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T$. Now we can rewrite the expansion in Equation C.48 in the form

$$\boldsymbol{x}_k = \boldsymbol{V}(c_1\boldsymbol{\Sigma}^2 + c_2\boldsymbol{\Sigma}^4 + ... + c_k\boldsymbol{\Sigma}^{2k})\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{b}. \tag{C.50}$$

We can rewrite this expression as a filtered SVD expansion by defining the matrix $\mathbf{\Phi}$, which has diagonal components ϕ_1, ϕ_2, ... ϕ_n. These filter factors are polynomials that depend on the expansion coefficients $c_1 ... c_k$ and the singular values. It has been shown that filter factors corresponding to early iterations damp the high frequency SVD components (Nagy & Palmer 2003) and therefore truncating the expansion process before completion naturally introduces regularization into the problem.

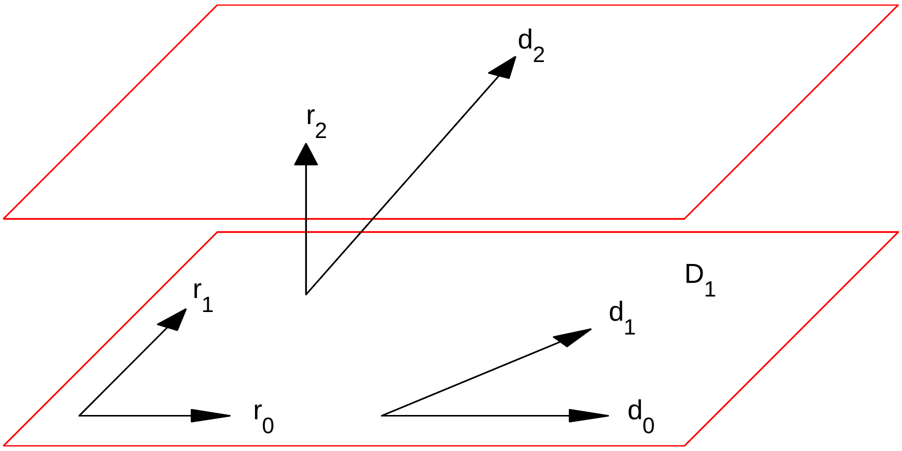

Figure C.2: A representation of the Krylov subspace

A representation of the Krylov subspace, adapted from (Shewchuk 1994). The residual vectors $\boldsymbol{r}$ are orthogonal to one another. The $(i+1)^{th}$ search direction $\boldsymbol{d}_{i+1}$ is constructed from the $(i+1)^{th}$ residual and i^{th} search direction $\boldsymbol{d}_i$.

Appendix D

Global Optimization Methods

Local optimization methods like SD and CG work well on optimizing simple linear functions and can be extended to include the optimization of non-linear functions. These local methods require functions that are smooth and differentiable (Björck (1996); Hansen (1997)). However, local optimization procedures display problematic behavior when applied to multi-modal functions that exhibit multiple extrema. Consider the following test function (Charbonneau 1995):

$$f(x,y) = -\left[16x(1-x)y(1-y)\sin(n\pi x)\sin(n\pi y)\right]^2 . \tag{D.1}$$

This function is plotted in Figure D.1 using $n = 9$ over $x = 0..1$ and $y = 0..1$, with 81 minima over the coordinate range. The global minimum $f = -1$ is located at $x = 0.5$, $y = 0.5$. By inspection, it is clear that an optimizer searching the fitness landscape using only local information may become trapped by a local minimum and fail to find the global solution. In general, this point is only a local minima, unless an initial position very near to the central minimum is chosen. Since we generally do not know the structure of the underlying function f (as in χ^2 fitting), this fine tuning of the starting point is not practical on a realistic problem. Smoothness and differentiability of the test function are also necessary for local optimizers using gradient-based search methods. There are a large number of multi-modal functions that are used in the literature that are known to be pathological to local iterative optimization schemes (Ackley (1987); Bäck (1996); Mühlenbein et al. (1991)). More complicated functions have also been used to test optimization routines, such as the Michalewicz function (Figure D.2; Michalewicz (1992)):

$$f(\boldsymbol{x}) = -\sum_{i}^{m} \sin(x_i) \sin\left(\frac{ix_i}{\pi}\right)^{2n} \tag{D.2}$$

where we have plotted the function in $2D$ ($m = 2$) with $n = 10$. The Langermann function in $2D$ goes beyond these tests in that symmetry is removed, further complicating the optimization (Figure D.3; Bersini et al. (1996)):

$$f(\boldsymbol{x}) = \sum_{i}^{m} c_i \exp\left(-\frac{(x-a_j)^2}{\pi} - \frac{(y-b_j)^2}{\pi}\right) \cos(\pi(x-a_j)^2 + \pi(y-b_j)^2) \tag{D.3}$$

with $m = 5$, $a = [3, 5, 2, 1, 7]$, $b = [5, 2, 1, 4, 9]$ and $c = [1, 2, 5, 2, 3]$. These functions are a challenge to local optimization routines because they have asymmetrically distributed local minima, and are difficult to optimize because of the extremely flat regions which can confound gradient-based methods. In fact, a large number of test functions have been developed to test the behaviour of optimization routines under given conditions and are widely used in the literature to benchmark optimization schemes (De Jong (1975); Mishra (2006); Zitzler et al. (2000)).

Fortunately, global optimization procedures exist that can find entire families of solutions that occupy the global minima of such problems. The Qubist optimization package, which is used in this thesis, has been thoroughly tested using many of the problems mentioned above (Fiege 2010). In this appendix we will discuss two of the global optimization schemes included in the Qubist toolbox: genetic algorithms (GAs) and particle swarm optimizers (PSOs). The results of the Ferret GA applied to the test problems (Equations D.1, D.2 and D.3) are shown in Figure D.4.

D.1 Genetic Algorithm Background

GAs (Holland 1975) are optimization methods based on the principles of biological evolution. Over time, evolution operates by driving individual solutions in a population toward forms that are better adapted to the environment, improving the survivability of the individual. These adaptations allow individuals a greater chance

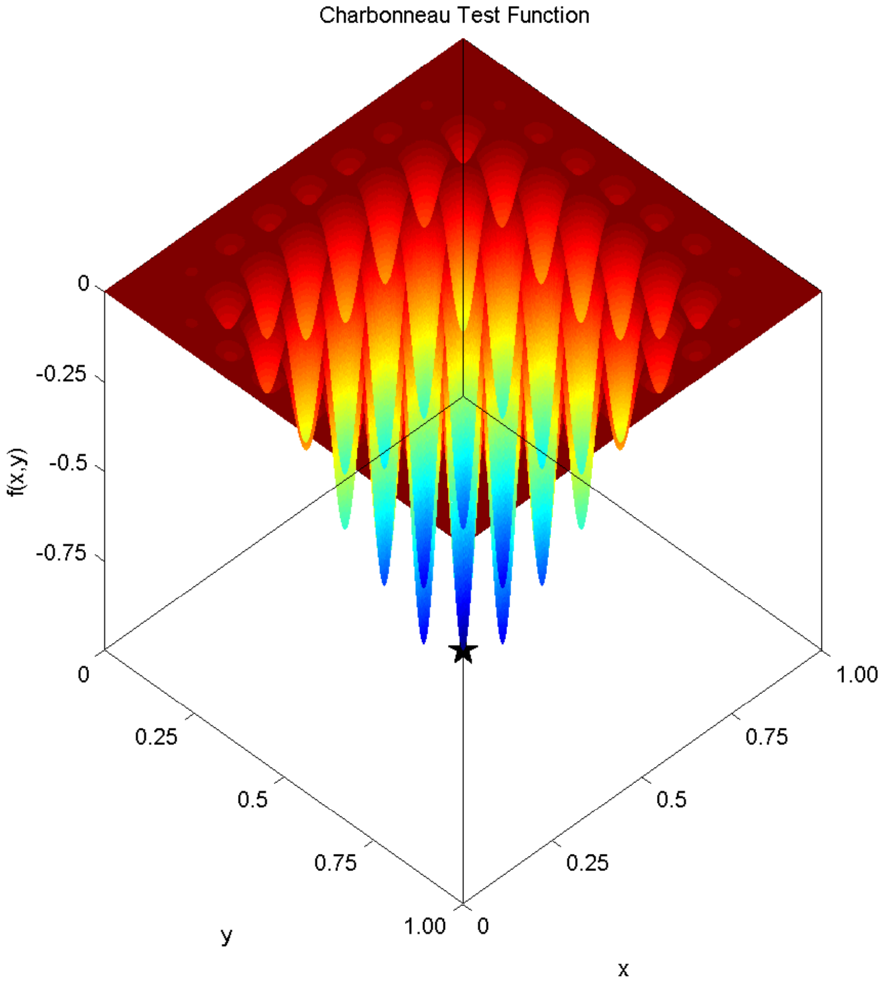

Figure D.1: Test function for global optimization

An example test function with many local extrema. The function is defined in Equation D.1.

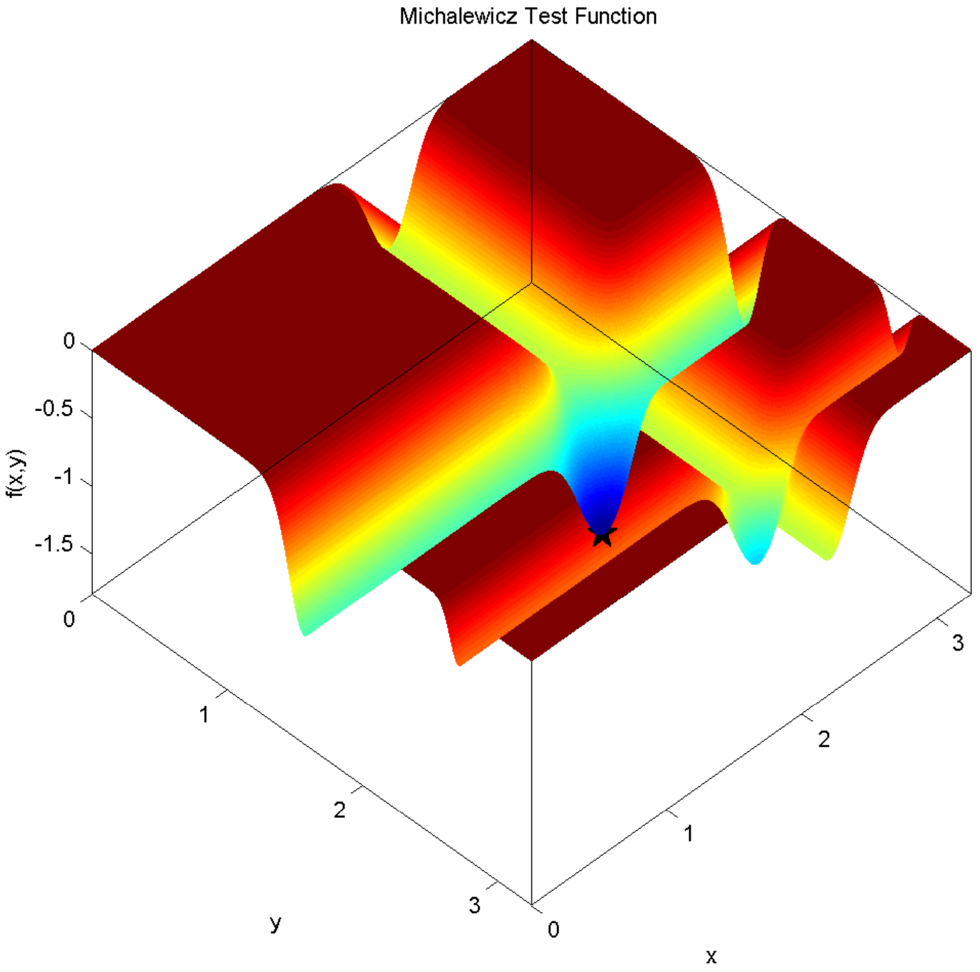

Figure D.2: Michalewicz test function

Michalewicz function. The function is defined in Equation D.2 and is a difficult test problem due to the flat areas, and deep channels. While the location of these channels may be easy to find, the exact position of the local minimum is difficult to find using most local optimization methods.

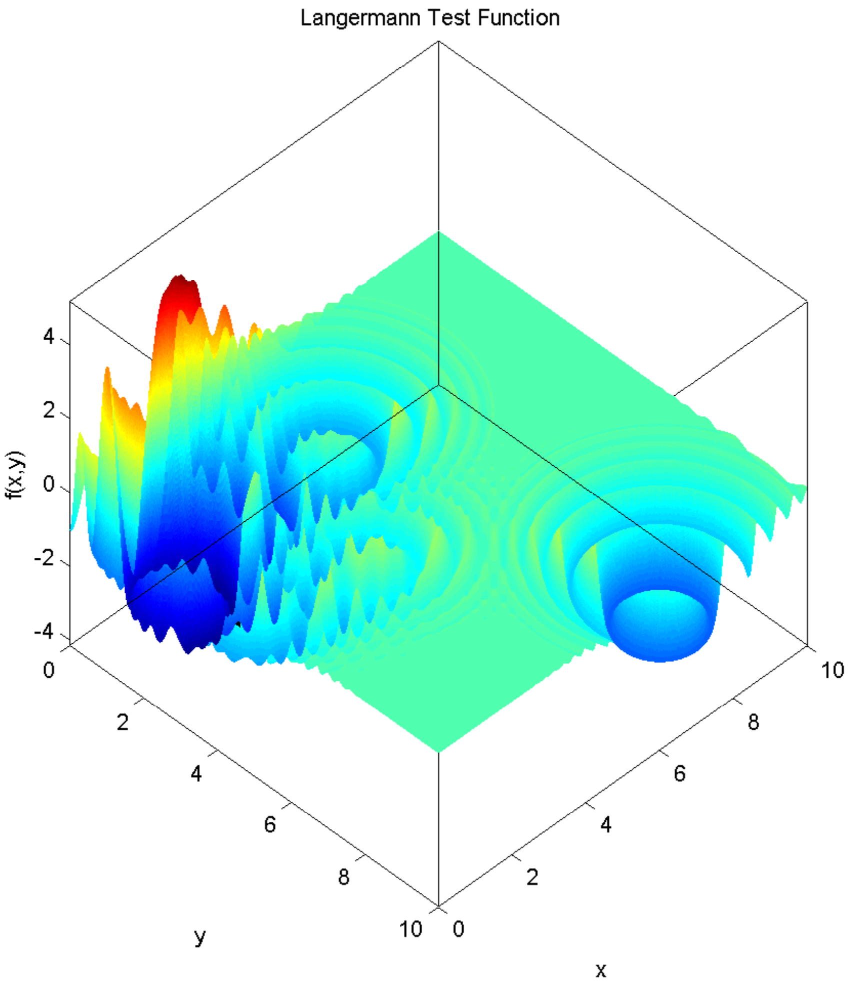

Figure D.3: Langermann test function

Langermann function. The function is defined in Equation D.3. The minima of this function are difficult to find due to the deceptive gradients in the flat areas. Furthermore, there is no symmetry in the function further complicating the optimization process.

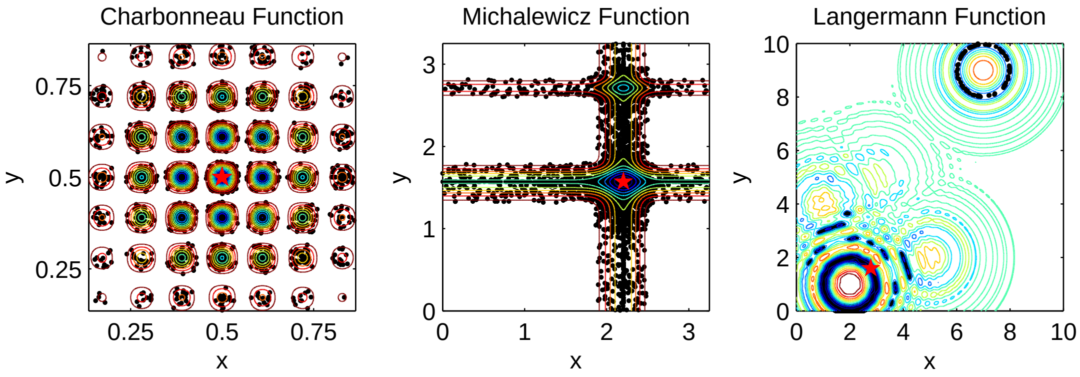

Figure D.4: Parameter space mapping of test problems

The three test functions (Equations D.1, D.2 and D.3) optimized by Ferret. Contours of the functions are shown overtop of the distribution of optimal solutions (black dots). Tolerance was set to highlight the structure of the functions. Results shown were discovered after 150 generations.

to propagate their genes throughout the population by producing more offspring. A variety of methods exist to implement analogues of these biological processes. To begin, we will describe a basic GA and the most common choices used in their operation. We will then discuss the specific implementation used in the Ferret GA, included in the Qubist suite of optimizers.

In the basic GA scheme (Goldberg 1989), each individual is represented by a set of N parameters in the optimization problem that defines the dimensionality of the search space. GAs operate on an encoded representation of an individual called a genotype. These genotypes represent the genetic material that comprises and determines the properties of each individual. In general, there are a variety of ways to define the genotype. The simple GA (Holland 1975) uses a binary representation. However, many other kinds of encodings have been used including base 10 (Charbonneau 1995) and real-valued (Fiege et al. (2004); Fiege (2010)) which is the scheme used by the Ferret GA.

The corresponding phenotype is defined by the set of observable properties of the decoded genotype. The phenotype of each solution includes quantities such as parameter values, the underlying model and the function of interest evaluated with the corresponding parameters. Each individual is also assigned a corresponding fitness measure that determines its likelihood of survival. The set of all such individuals defines the population (Holland (1975); De Jong (1975); Goldberg (1989)).

To illustrate the operation of a simple GA, let us consider finding the minimum of the test function given by Equation D.1. The GA operates on an abstract representation of the information required to evaluate the fitness function. This genotype is decoded by the GA to generate the coordinate position (x, y) for each individual. Together, the coordinate position and the fitness value $f(x, y)$ represent the phenotype.

Though many GAs are discussed as maximization routines in the literature, we will follow the convention of Ferret and consider minimization problems. This is a more natural approach for astrophysical model fitting in which the fitness is defined by the χ^2 statistic to quantify the goodness-of-fit of a model to data. A brief overview of the operation of a simple GA applied to function minimization is given below.

To initialize a simple GA, an initial population of individuals with random genotypes is generated. A simple GA uses binary encoding, so a set of genotypes are selected by generating a random binary number for each gene (Holland 1975). The fitness of each individual phenotype is then evaluated. Individuals are selected for inclusion in the next generation, and crossover and mutation operators are probabilistically applied to the group of selected individual genotypes. The crossover operator combines two selected "parent" genotypes to produce new "offspring" with the combined characteristics of each parent. The mutation operator acts on individuals and introduces random perturbations to the genotypes of each selected solution. When these operations are complete, the fitness values of the newly produced solutions are evaluated. After many iterations of this procedure, the population is comprised of solutions with better (in our case, lower) average fitness value than we began with. Over time, natural selection favours more fit solutions to the problem and the population moves toward the minima in the search space. Note that this scheme does not require any single initial solution to be chosen *a priori* as a starting point, in contrast to local optimization methods like SD and CG (Björck 1996).

D.2 Selection

Selection acts as a method of passing information from one generation to the next. The selection procedure promotes the propagation of individuals with superior (low) fitness values throughout the population. This represents the survivability of well adapted organisms. There are a number of methods by which to accomplish this selection, but the simplest types are roulette wheel and tournament selection.

The roulette wheel approach assigns a probability of selection to each individual based on fitness (Baker 1987). The selection probability is found by normalizing each individual solution:

$$p_i = \frac{1 - f_i}{\sum_i^{N_p}(1 - f_i)} \tag{D.4}$$

where f_i is the fitness of the i^{th} solution and N_p the population size such that $\sum_i^{N_p} p_i = 1$. The cumulative sum of these probabilities is then calculated. A random number

ξ is generated between 0 and 1, and a solution is selected by choosing the minimum i such that $p_i > \xi$. This procedure is analogous to the spin of a roulette wheel with a proportion p_i of the wheel assigned to each individual. As such, the highest fitness solutions have the highest likelihood of surviving to the next generation, participating in crossover operations or being acted on by the mutation operator (Mitchell 1996).

Roulette wheel selection is sensitive to the scaling of the fitness function. When the problem is poorly scaled such that min(f_i)<<mean(f_i), roulette wheel selection strongly favors the most fit individuals in the population, which places the GA at risk of premature convergence to a local minimum. In general, it is better to scale fitness values such that the selection pressure is more gentle, so that some individuals of moderate fitness are able to propagate alongside the most fit members of the population. The algorithm tends to exploit the solutions that it has already found rather than thoroughly explore the fitness space. As such, the global minimum may be missed. Similarly, a problem that is poorly scaled such that min(f_i)≈mean(f_i) results in the opposite problem of insufficient selection pressure, so that the GA may not converge to the correct solution. The search is unfocused and there is not enough exploitation of the discovered solutions to converge on a minimum. Both of these cases are problematic; thus an inherent drawback of roulette wheel selection is that fitness values must be carefully scaled which requires *a priori* knowledge of the fitness landscape.

Furthermore, when a GA first begins there is a large difference between the best and worst fitness individuals in the population. In general this fitness range F will shrink over time as the population converges toward the global minimum. Success requires good fitness scaling at each generation, which may be difficult to enforce. Therefore it is beneficial to remove the effect of the fitness range F altogether from the selection routine. Rather than basing the selection process on fitness, it is often useful to define the rank of a solution. The simplest method of ranking simply orders the solutions from best to worst fitness. Individuals are then selected based on their rank position rather than fitness. This is a useful scheme that overcomes the scaling problems of the fitness range F and helps to reduce the likelihood that an individual will dominate and generate the population and generate an excessive number of offspring (Bäck & Hoffmeister 1991). In general, roulette wheel fitness based on rank behaves more robustly than proportional fitness (Whitley 1989).

A better selection scheme that is widely used is tournament selection (Mitchell (1996); Goldberg & Deb (1991)). The most basic tournament selection operates by randomly picking k individuals at a time to compete in a tournament. The winner of the tournament is the solution with the best fitness from the k individuals chosen. This process is repeated to select more individuals, and typically individuals can be selected for more than one tournament. A tournament of one individual ($k = 1$), amounts to random selection.

Many variations of the tournament selection operator exist. For example, rather than simply choosing the best rank solution, winners of the tournament may be found by a finite probability p_s. Contestants in a tournament can be picked for more than one tournament or may be chosen without replacement (Sokolov & Whitley 2005). The size of the sample k can also be adjusted, increasing the selection pressure as k grows, although it has been suggested that tournaments with $k > 2$ increase the selection pressure too high and can lead to a loss of diversity in the population (Goldberg 1989). The tournament scheme is generally considered a better selection method since it is insensitive to the fitness range F. The winners of these tournaments are acted upon by the crossover and mutation operators with probability p_c and p_m respectively, and passed to the next generation.

D.3 Crossover

For each individual selected, there is a finite probability p_c that the crossover operation occurs. In the simple GA, this probability is a fixed strategy parameter set by the user before the optimization begins. Selected individuals that do not experience crossover or mutation are simply passed along to the next generation. For those individuals that have been selected for crossover, a second selection is made to find a "mate".

The simplest crossover procedure used in a simple GA is a one point crossover (Bäck et al. 1997). A single point on both individuals' genotypes are selected, and the genes beyond this point are swapped between individuals producing two new offspring. More complicated schemes can be constructed in which multiple point crossovers occur where discrete portions of the genome are interchanged between individuals.

To illustrate the one point crossover operation, let us consider two random binary genotypes A and B. Then we generate a random position to cut the genotypes (suppose this is between positions 5 and 6, for example). The crossover procedure generates the two offspring C and D, shown below.

$A:$	0	1	1	0	1	0	1	1
$B:$	1	0	1	1	0	1	1	1
$C:$	0	1	1	0	1	1	1	1
$D:$	1	0	1	1	0	0	1	1

An interesting alternative called uniform crossover uses a fixed mixing ratio between the two parent individuals (Syswerda 1989). In this scenario, rather than simply exchanging segments of the genome between two solutions, the contribution to the offspring from each parent is a probabilistic process. In general each genome can be split at any number of randomly chosen cross over points with the chance of each segment contributing to any given offspring (Spears & De Jong 1991).

D.4 Mutation

The probability that a selected solution undergoes mutation is given by a user defined probability p_m. When an individual is mutated, a random gene in the genotype is changed from its initial value (Goldberg 1989). To determine which gene is affected by a single point mutation, each gene is assigned a probability of being mutated as $g_i = 1/N_g$ where N_g is the number of genes in the genome, such that $\sum_i^{N_g} g_i = 1$. The cumulative sum is formed for each of the genes and a random number is generated that selects position in the genotype. This is the simplest kind of mutation, known as a one point mutation operator, however more complicated mutation operations exist which can affect more than one gene at a time. The simple GA represents genotypes in a binary encoding such that a finite number of states exist, so that the solutions are guaranteed to remain bounded in the case that genes encode numerical parameters. In real-valued GAs, it must be ensured that mutated solutions remain within the bounds of the search space.

The one point mutation operator is illustrated graphically below. Suppose that genotype A is mutated to produce genotype B, and that we have randomly selected the 3^{rd} gene to mutate:

$$\begin{array}{rcccccccc} A: & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ & & & \downarrow & & & & & \\ B: & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

These three basic operations define the simple GA. Information is processed using the crossover and mutation operators and filtered into subsequent generations by the selection process. The crossover operation directs the population toward the most successful individuals over time. Mutation keeps individuals from becoming too similar to one another and drives the population to explore new areas of the parameter space. It is the interplay of these two effects that results in the effectiveness of the GA as an optimization routine.

D.5 Beyond the basic GA

In order for natural selection to work, a population must posess a spectrum of fitness values. This variation must be maintained through generations to avoid becoming trapped in local minima. This premature convergence is due to the rapid loss of diversity during the evolutionary process and therefore encouraging diversity is important during the iterative procedure. Diversity can be lost when the selection pressure is too high, causing the genotype of a single very fit individual to dominate the population before the search space has been thoroughly explored.

Suppose that we wish to optimize a degenerate multi-modal function, such that there are m minima that have equally good fitness values f. In general, we would like to find all of the degenerate minima to map the solution space (Chen et al. 1999).

However in the standard GA scheme, the population will converge to a single point at the position of one of the minima. Even with mutation present, after a sufficient time the population converges toward a single arbitrary member of the population. The genes of this individual would spread through the population and the effectiveness of crossovers would no longer introduce novelty to the population since many of the members would share similar characteristics. This effect is called genetic drift and is the GA analogue of inbreeding. It is difficult for a GA to recover once diversity is eliminated from the population. In fact, this is the reason that GAs avoid constructing the next generation from only the best fit individuals.

Niching is one method designed to combat genetic drift in a population (Fonseca and Fleming 1993). Niching estimates the number of neighbors that each solution has, and selection preference is given to individuals with fewer neighbors to promote diversity in the population. Such isolated solutions represent separate individual ecological niches which are populated by separate species. Niching is usually discussed in the context of multi-objective GAs (Fonseca and Fleming (1993); Horn et al. (1993)), but it is also extremely valuable for mapping the parameter space of single objective problems. In general, niching can be done in parameter space or in objective space in the case of a multi-objective problem. In single-objective problems such as our lens modeling code, niching can only be done in parameter space.

To define an ecological niche, we define a niche radius σ_{share} that defines a distance in the parameters or fitness such that a niche count ϕ_i can be defined on the number of close neighbors that each solution has:

$$\phi_i = \sum_{j:j\neq i} \left(1 - min\left\{1, \frac{d_{i,j}}{\sigma_{share}}\right\}\right)^p , \tag{D.5}$$

where $p > 0$ defines the niche shape, and $d_{i,j}$ defines a distance between solutions i and j. The distance $d_{i,j}$ between solutions can be defined in terms of the genotype or phenotype (Deb (1989); Deb & Goldberg (1989)). In terms of the genotype, the Hamming distance is used, which defines the ‘distance’ between binary strings as the number of positions at which the corresponding symbols are different (Hamming 1950). The phenotype distance is simply defined in terms of Euclidean distance between sets of model parameters.

Usually, a GA population for a single-objective problem will contain a single individual at each generation with the lowest f_i, unless the fitness landscape is characterized by degenerate global minima in the form of a flat-bottomed valley. In the usual case of a single non-degenerate minimum (or isolated islands with the same minimum value at a finite number of well-separated points) it is often useful to be able to map the fitness landscape in the vicinity of the global minimum, for example in the context of mapping confidence intervals. To facilitate the mapping of distinct minima, we define a tolerance range τ. Solutions that have fitness values that are equal within this tolerance $f \pm \tau$ are considered tied during the tournament selection process. When a tie occurs, the niche count provides a method of determining the winner of the tournament. This is a valuable technique that favors isolated solutions and tends to promote the exploration of distinct ecological niches.

A common problem with the simple GA is that the best fitness solution can be lost through crossover or mutation when determining the next generation. Elitism is used in order to prevent this from occuring. The elites are a user defined fraction of the population with the best fitness (lowest f_i). These elite solutions are passed to the next generation unchanged to ensure that forward progress on the optimization problem is always made and that the best fit individuals of the next generation will be at least as good as the last. Elitism helps to accelerate the optimization behavior of GAs in general. Used in concert with niching, elitism allows the GA to make forward progress on optimization, yet still maintain diversity of the population. The elite fraction should be kept small ($\leq$ 10%) in order to promote diversity in the population.

D.6 The Ferret GA

The previous sections describe the basic operation of a basic and simple GA. The GA used in our work, Ferret (Fiege (2010); Fiege et al. (2004)), goes beyond this basic scheme in a variety of ways.

Ferret is a bounded real-valued multi-objective GA that searches an N dimensional volume in parameter space, determined by the parameter limits set by the user before the start of a run. The boundaries can be set as cyclic, indicating periodicity in a given parameter. The boundaries can also be set as soft, such that the search volume can be expanded if a significant fraction of the population has moved to a boundary. This feature is used in the gravitational lens modeling problem when making use of real data in Section 5.

Ferret uses real-valued genotype encoding, which is convenient for fitting real-valued physical parameters to astrophysical data. Binary encodings (Holland 1975) and base 10 encoding (Charbonneau 1995) are both limited for finite precision operations. Since we are operating on physical quantities real-valued genotypes are the most useful for our applications (Wright 1991).

Operations on real-valued genotypes lend themselves to geometric interpretations that are not obvious in the context of binary or base-10 encoded GAs. The crossover operation for a real-valued genotype amounts to choosing a point on the line connecting two parent solutions. Mutations in Ferret act on a given parameter in a genotype by adding a Gaussian perturbation in a random direction, with a distance determined by a user defined mutation strength parameter. Unlike the simple GA, Ferret evaluates the fitness of the population in multiple steps, after each crossover and mutation operation. This is done to ensure monotonicity in the fitness of the population.

Niching is easily understood in the context of real-valued GAs as well. The niche radius σ_{share} has an obvious geometrical interpretation, and the distinction between phenotype and genotype distance in Equation D.5 is unambiguous. In essence, we define a niche as an N dimensional hypersphere around a given solution. The solutions within this sphere determine the niche count for a given individual, with closer neighbors receiving greater weight. The interpretation of the distance factor in Equation D.5 is defined geometrically as the Euclidean distance or by a Holder metric. Due to the implementation of niching techniques, the Ferret GA is well-suited to effectively explore high dimensional parameter spaces and map confidence intervals in our data-modeling application.

The selection operation in Ferret is a binary tournament procedure. Ferret uses a prioritized selection scheme to select individuals. The criteria for selecting solutions are fitness and niche count. In the case of a tie in the fitness criteria, including ties due to fuzzy tournament selection, Ferret selects an individual based on the next priority. The niche count is a valuable property since solutions with fewer neighbors represent solutions in a less explored parameter space niche.

Elitism is also used within the Ferret GA. This is an important feature since we desire the best solutions to live on and propagate their genes to subsequent generations. The elites are defined as a user-defined fraction of the population that survive unchanged. Crossover and mutations are allowed to operate on the set of elites but Ferret will always pass along a copy of the elites to the next generation to ensure that the best solutions in the population live on unchanged.

Ferret also uses linkage learning to divide large problems into smaller linked groups called building blocks (BBs; Goldberg (2002)). Linkage learning is a method by which to split up complex problems into smaller, more managable ones. These BBs can evolve independently from each other, and the linkages can dynamically change during the course of a run. The process of detecting BBs can be extremely useful to the optimization of complicated functions but require more fitness evaluations each generation. To discover these linkages, Ferret applies variations to sets of genes individually and monitors the changes to the resulting fitness. When genes are linked, the effect of applying these variations to the genes individually tends to worsen the fitness. However, when variations applied to both genes simultaneously tend to improve the fitness, the genes should be considered linked, and kept together during crossovers. Ferret searches for these linkages in parallel and dynamically adjusts the linkage matrix as the run proceeds and the optimal region is discovered. In general a smaller number of linkages is considered to represent an easier problem. When optimizing analytical gravitational lens models we typically find a single linkage group. All lens density parameters typically become linked early in a run.

Ferret includes many other features that help to map parameter spaces and adjust the user-defined control parameters in the algorithm. Further details of these features are documented in the Qubist user guide (Fiege 2010). Ferret is well suited to find entire classes of solutions to single objective problems and is well suited to searching for the non-linear mass density parameters of gravitational lens models. Further details of the Ferret algorithm are explored in Chapter 3.

D.7 Particle Swarm Optimization

Particle Swarm Optimizers (PSOs) are a relatively new class of global optimizers (Kennedy & Eberhart 2001). PSOs are very simple global optimization schemes based on the group behavior of organisms such as insects, fish or birds. The social relationship between individuals allows the population, called the swarm in the language of PSOs, to converge on global extrema in optimization problems.

A swarm is initialized by randomly generating N_s positions within a bounded region of parameter space. Each of these points is the position of a constituent particle of the swarm. A corresponding randomly generated velocity is also assigned to each particle, which are scaled such that most particles are placed on initial orbits within the search space that do not collide with boundaries. Generally the aim is to use velocities that allow a large number of particles to explore the space but not so high that all the particles approach the boundaries of the search space.

The fitness function is evaluated for the current position i of each particle and a global best (lowest fitness) position $\boldsymbol{x}_g$ is determined from all of the positions the entire swarm has found throughout its history. The personal best position that each particle has individually encountered is also found. Each particle is attracted to both the current personal and global best solutions by a spring force via Hooke's law. The positions and velocities of the particles are updated by considering their motion in a potential composed of the sum of two separate harmonic oscillator potentials

$$U_i = \frac{1}{2} c_p \xi_p |(\boldsymbol{x}_p - \boldsymbol{x}_i)|^2 + \frac{1}{2} c_g \xi_g |(\boldsymbol{x}_g - \boldsymbol{x}_i)|^2, \qquad \text{(D.6)}$$

where $\boldsymbol{x}_i$ is the current position, c_p and c_g are the personal and global spring constants and $\boldsymbol{x}_p$, $\boldsymbol{x}_g$ the current personal and global best position vectors. The values ξ_p and ξ_g are randomly generated numbers between 0 and 1 in order to introduce a stochastic element to the search. These random quantities help the PSO explore the parameter space more thoroughly and act in analogy to the mutation operator in a GA, which encourages the explorative aspect of the optimization and aids in mapping the parameter space. After an update step, new personal and global best solutions are found and the process is iterated. As the trajectories of individuals in the swarm evolve over time, the swarm tends toward the global minima. The position and velocity of the i^{th} particle in the swarm is updated after a time Δt:

$$\begin{aligned} \boldsymbol{v}_i(t+\Delta t) &= \boldsymbol{v}_i(t)\,(1-\Delta t/t_{damp}) + \\ &\quad [c_p\xi_p(\boldsymbol{x}_p-\boldsymbol{x}_i)+c_g\xi_g(\boldsymbol{x}_g-\boldsymbol{x}_i)]\,\Delta t \\ \boldsymbol{x}_i(t+\Delta t) &= \boldsymbol{x}_i(t)+\boldsymbol{v}_i(t)\Delta t, \end{aligned} \tag{D.7}$$

where a damping term t_{damp} has been included in the velocity update step. This damping term follows the formulation in the Qubist user's guide, but is not standard in the PSO literature, as most references write the first term in Equation D.7 as $c_d\boldsymbol{v}_i$ where c_d is a damping coefficient. Equation D.7 is a discrete time equation, and in general there is a finite error associated with each update step. In the standard PSO, these errors can compound and cause a "particle explosion" in which the swarm may violently oscillate on nearly radial orbits, casting particles out to infinity if bounds are not stringently enforced (Clerc & Kennedy (2002); Shi & Eberhart (1998)). The damping term helps to overcome this effect by causing the swarm to settle down over time.

As in the operation of a GA, the optimization properties of the PSO scheme are a result of the interplay between separate simple effects that combine to produce powerful and interesting effects. The optimization process of a PSO is a result of the emergent social behavior of the swarm. The global best solution allows information to be communicated throughout the swarm, so the best solution found by the swarm is shared among the constituent particles (Clerc 2006). The advantage to this kind of global optimization is that very few user defined parameters are needed to initialize the algorithm. Similar to GAs, PSOs do not require a pre-selected unique starting point to begin the optimization procedure in contrast to the finely tuned specific starting point required by local iterative optimizers when optimizing complicated functions.

Locust overcomes the difficulties in the standard PSO scheme by solving for the swarm dynamics analytically, equivalent to solving Equation D.7 in the limit where $\Delta t \to 0$ (Fiege 2010). This helps to reduce error in the update steps, and suppresses the pathological behavior introduced by the finite precision in solving Equation D.7.

This scheme is computationally expensive compared to the single Euler integration scheme of Equation D.7; however, the extra cost is insignificant for problems such as ours (and most realistic astrophysical applications), where run times are completely dominated by the evaluation of the fitness function. Locust goes beyond the standard PSO by using a dynamic *lbest* approach in place of a single global best. This splits the swarm into local groups each with a best solution that takes the place of the global best in Equation D.7. These neighborhoods merge dynamically as the run proceeds by a binary tournament similar to that used in the GA. If one of the global solutions is significantly greater than another, the neighborhoods will merge. Similarly, groups can be split off of the swarm if there are multiple solutions within each neighborhood that are equally good. This dynamic *lbest* topology increases the explorative properties of the swarm and allows the swarm to settle when in the vicinity of an optimal position. This helps to balance the exploitation and exploration aspects of the swarm. Locust is described in further detail in Chapter 3. The performance of GAs and PSOs in the context of gravitational lens modeling is also compared in Chapter 3.

D.8 Configuration of Global Optimizers

This Section provides a few additional details about the Ferret and Locust optimizers used in this thesis and how their strategy parameters were set. The material in this section draws largely from the Qubist User's Guide (Fiege 2010), and was included as an appendix to Rogers & Fiege (2011a).

D.8.1 Ferret Genetic Algorithm Setup

Most GAs encode model parameters on binary strings (Holland 1975; Goldberg 1989), with mutations and crossovers defined as operators that work directly on these strings. For example, a mutation would typically flip a single bit, while a simple crossover would cut two binary strings at the same position and exchange the parts of the string to the right of the cut, effectively mixing together two individuals in the population. If these strings represent real valued parameters of a model, it is necessary to decode the binary representation into real numbers prior to evaluation. Ferret is specialized to work directly with genotypes specified by a list of real-valued parameters, thus side-stepping the conversion from binary strings to real numbers. An individual in Ferret is therefore represented by a point in an N-dimensional real vector space, where N is the number of parameters or "genes", which allows more elaborate mutation and crossover operators than can be defined on a simple binary string.

Ferret contains many options, which are controlled by "strategy parameters" that are encoded in a MATLAB structure called `par`. The strategy parameters are defined by a setup file, which is read at the start of a run. Ferret contains a default setup file, which fills in any strategy parameters not specified by the user. These default values are often adequate and the software is not usually very sensitive to the exact choice of strategy parameters. This robustness is achieved in part by an adaptive algorithm that automatically controls several of the most important control parameters, affective mutations and crossovers.

A Ferret run evolves `par.general.NPop` populations, where the size of each population is set by `par.general.popSize`. Ferret uses a single population by default, and it is recommended to set the population size in the range of $100 - 500$. Generally, this choice is guided by the computational expense of evaluating the fitness function, the complexity of the problem, and the user's experience solving it. Larger populations tend to explore the parameter space more thoroughly than smaller populations, but at greater cost. When `par.general.NPop > 1`, the populations interact weakly with each other by exchanging individuals with probability `par.immigration.PImmigrate` $\approx$ `0.01` each generation. This is beneficial for some very difficult problems because multiple populations explore the parameter space almost independently, thus increasing the probability of finding the global solution. Ferret often performs better on degenerate problems when the total number of individuals is divided into several populations rather than placing them all into a single population. However, we used a single population with `par.general.popSize=200`.

Ferret's mutation operator is defined as a perturbation in an N-dimensional real vector space, where the magnitude of the perturbation is drawn from an initially

Gaussian distribution, whose standard deviation is determined by a strategy parameter `par.mutation.scale=0.25` by default. The distribution of mutation scales is under adaptive control, and evolves during each run, as Ferret preferentially selects values that result in improved fitness. Ferret's default mutation rate is given by `par.mutation.PMutate=0.05`.

The role of crossover in a GA is to mix together two different solutions to produce offspring that are intermediate between the parents. Ferret contains two different crossover operators, which mix genes in fundamentally different ways. Ferret's "X-type" crossover operator is a geometry-based operator that can be shown to be analogous to the bit string operator found in traditional binary encoded GAs. X-type crossover is essentially an averaging operation, which draws a line between the parameter space coordinates of two individuals and selects a point between the individuals on that line. The fractional distance traveled along this line is drawn from a distribution, which was initialized to a Gaussian random distribution of standard deviation `par.XOver.strength=0.25` at the beginning of the run. The distribution of crossover strengths is under adaptive control and co-evolves with the population to prefer crossover strengths that tend to result in improved fitness. Note that it is possible to occasionally overshoot during a crossover by drawing a crossover strength greater than one. Surprisingly, this turns out to be beneficial on many problems because it helps to expand the population into long, slender valleys by occasionally overshooting the end points of the distribution. X-type crossover is Ferret's primary search mechanism, so we normally set `par.XOver.PXOver=1` to set the crossover probability to 100%.

Ferret's "building block crossover" operator is at the heart of its linkage-learning system and has no analogy in traditional GAs. This type of crossover exchanges a building block, or a group of parameters previously identified as linked, in their entirely from one individual to another. Building block crossover efficiently propagates building blocks responsible for high-quality solutions throughout the population and mixes them with other high-performing building blocks comprised of other parameters. We normally set `par.XOverBB.PXOver=1`, which indicates a 100% chance of mixing building blocks.

Ferret makes a duplicate copy of all populations prior to mutation and crossover, effectively doubling the number of individuals. Ferret's selection operator is applied after the mutation and crossover operators, using a binary tournament scheme in which individuals are drawn randomly from the populations modified by mutation and crossover to compete against individuals drawn from the unmodified duplicate populations. Individuals that win a tournament are allowed to propagate to the next generation and the losers are destroyed. The probability of competition is normally 100%, but it is possible to reduce the selection pressure by setting `par.selection.pressure < 1`. This delays convergence, thereby allowing more time for exploration, by causing Ferret to ignore fitness values during tournament selection with probability equal to `1-par.selection.pressure`.

Sometimes a second round of competition is required when individuals tie in a tournament. This occurs commonly in multi-objective problems, when `par.selection.pressure < 1`, or when a fuzzy tolerance has been defined for a single-objective problem. For example, we map out some region of the parameter space within $\Delta\chi^2$ ($dchi2$) of the minimum value by setting `par.selection.FAbsTol=dchi2` to tell Ferret to ignore differences in fitness less than this amount. In this case, Ferret employs a niching strategy similar to that discussed by Fonseca and Fleming (1993), which prefers solutions with fewer near neighbors over solutions with a greater number of neighbors. The logic behind this preference is simple: solutions in a less populated region of parameter space are more unique, and therefore more valuable to the exploration of the space.

D.8.2 Locust Particle Swarm Optimizer Setup

Locust is a relatively simple code to configure, compared to the myriad of options allowed by Ferret.

The most important strategy parameter controlling a PSO is the number of particles in the swarm, given by `par.swarm.N`. In general, larger swarms tend to explore the parameter space more thoroughly, but may require more time to do so. Very small swarms are problematic because they may sample the space poorly and miss the global solution. There is no established rule for choosing the swarm size. One typically starts with about 100 particles and decreases the number of particles if experience shows that this decreases the run time without causing problems with reliability. Very difficult problems may require more than 100 particles, and we used `par.swarm.N=200`.

`par.swarm.cg` and `par.swarm.cp` are, respectively, the global best and personal best constants used in Equation (3.21). Both of these parameters should be of order unity, but setting *cg* slightly less than *cp* is usually helpful because this places more emphasis on exploration of the parameter space because the particles are influenced less by the global best solution. Increasing *cg* relative to *cp* places more emphasis on the exploitation of the global solution or solutions, at the expense of exploration, because all particles will be drawn to the optimal region more rapidly. We used the default values: `par.swarm.cg=0.5` and `par.swarm.cp=1`.

`par.swarm.dt` is the time step between updates to the swarm positions and velocities. Therefore, the time step *dt* affects the rate of sampling of the parameter space as particles move around on their orbits, but has no effect on the accuracy of the orbits because Locust uses an exact solution to the simple harmonic oscillator orbit equations approximated by the finite difference equation given by (3.21). We used the default value `par.swarm.dt=1`.

PSOs require damping to cause the particle swarm to settle down to a converged solution. Locust is designed such that `par.swarm.TDamp=1` corresponds to a critically damped harmonic oscillator. Generally, underdamped oscillations are required so that multiple orbits explore the parameter space before the swarm converges. We used the default value for the damping time `par.swarm.TDamp=10`.

Appendix E

Numerical Methods for Spatially Variant PSFs

To describe blurring by a spatially variant PSF we first present an efficient method using two-dimensional FFTs. We then show how to treat the problem in terms of blurring matrices and flattened image vectors. See Nagy & O'Leary (1998) for more details on the approach and Nagy et al. (2002) for a MATLAB implementation.

Consider an $N \times N$ grid of independent PSFs $\boldsymbol{P}_{ij}$ and split the unknown blurred image $\boldsymbol{Y}$ into regions $\boldsymbol{Y}_{ij}$, each of size $k \times k$:

$$\boldsymbol{Y} = \begin{array}{|c|c|c|c|} \hline \boldsymbol{Y}_{11} & \boldsymbol{Y}_{12} & \cdots & \boldsymbol{Y}_{1N} \\ \hline \boldsymbol{Y}_{21} & \boldsymbol{Y}_{22} & \cdots & \boldsymbol{Y}_{2N} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \boldsymbol{Y}_{N1} & \boldsymbol{Y}_{N2} & \cdots & \boldsymbol{Y}_{NN} \\ \hline \end{array} \tag{E.1}$$

Each of these blocks will be affected by an independent PSF. Suppose that the size of each PSF is $(r+1) \times (r+1)$ with r even, and let the unblurred $N \times N$ image be represented by $\boldsymbol{X}$.

Let us define a set of "mask" matrices $\boldsymbol{w}_{ij}$. In the case of piecewise constant PSFs, these masks are the same size as the unblurred image and are comprised of 0 entries everywhere except for the $k \times k$ block at position (i, j), where the entries of $\boldsymbol{w}_{ij}$ are set to 1.

To find the components of a given region we convolve $\boldsymbol{X}$ with the corresponding PSF P_{ij}, followed by an element-wise multiplication by the mask $\boldsymbol{w}_{ij}$. The non-zero elements of this product give $\boldsymbol{Y}_{ij}$. Proceeding in this way we build up the blurred image block by block:

$$\boldsymbol{Y}_{ij} = \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{w}_{ij} \circ (\boldsymbol{P}_{ij} * \boldsymbol{X}), \tag{E.2}$$

where the symbol "$\circ$" represents element-wise multiplication and symbol "$*$" is the convolution operation. Note that each term in the sum is determined by the convolution of the entire image $\boldsymbol{X}$ with the appropriate PSF before the mask is applied. This is crucial to ensure that "seams" will not be visible between regions in the blurred image $\boldsymbol{Y}$.

In general, it is possible to speed up this routine by calculating $\boldsymbol{Y}_{ij}$ directly. Consider splitting the unblurred image into regions $\boldsymbol{X}_{ij}^{k}$ where the superscript denotes the size of the block, in this case $k \times k$. In order to avoid artifacts and keep the correct intensity near the edges of this block after convolution, we include a number of neighboring rows and columns on each side of $\boldsymbol{X}_{ij}^{k}$. The width of this border is set by the size of the PSF, $r/2$, with regions on the image boundary padded to enforce the boundary conditions discussed in Section 5.1. These extended regions are then denoted $\boldsymbol{X}_{ij}^{(r+k)}$. The PSFs are padded to match the extended regions in size, resulting in $\boldsymbol{P}_{ij}^{(r+k)}$. The blurred extended region is found by the convolution

$$\boldsymbol{Y}_{ij}^{(r+k)} = \left(\boldsymbol{P}_{ij}^{(r+k)} * \boldsymbol{X}_{ij}^{(r+k)}\right). \tag{E.3}$$

The central $k \times k$ block of this product is clipped out and placed in the (i, j) position of $\boldsymbol{Y}$. The process is repeated until the entire blurred image is filled in. Time is saved working with extended regions and padded PSFs since we only need to calculate the convolution over the $(r+k) \times (r+k)$ block for each PSF rather than the entire image as in Equation E.2, and the construction of masks is not needed. The convolutions can be carried out efficiently with two-dimensional FFTs.

The basic procedure can also be described by an analogous matrix-vector operation. To express the sum in Equation E.2 in terms of matrix multiplication, we define the unblurred flattened image as a vector $\boldsymbol{x}$, and the flattened blurred image as $\boldsymbol{y}$. We build a set of N^2 blurring matrices to describe the effect of each PSF on $\boldsymbol{x}$, which we denote as $\boldsymbol{B}_{ij}$. The mask matrices $\boldsymbol{w}_{ij}$ are used to construct analogous weighting matrices $\boldsymbol{D}_{ij}$. These matrices are of size $N_{pix} \times N_{pix}$, where N_{pix} is the number of pixels in the image, identical to the size of the blurring matrices $\boldsymbol{B}_{ij}$. The total blurring matrix $\boldsymbol{B}$ is then written as a weighted sum of blurring matrices $\boldsymbol{B}_{11}$, $\boldsymbol{B}_{12}$,...,$\boldsymbol{B}_{NN}$.

$$\boldsymbol{B} = \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{D}_{ij} \boldsymbol{B}_{ij}. \tag{E.4}$$

The blurred image is then found by a matrix multiplication $\boldsymbol{y}$=$\boldsymbol{Bx}$. The weighting matrices $\boldsymbol{D}_{ij}$ have the mth diagonal entry equal to 1 provided that image pixel m is in region (i, j), and all other elements 0. The weighting matrices satisfy $\sum_{i=1}^{N} \sum_{j=1}^{N}$ $\boldsymbol{D}_{ij}$ =$\boldsymbol{I}$ where $\boldsymbol{I}$ is the $N_{pix} \times N_{pix}$ identity. We adopt the use of piecewise constant PSFs but in general it is possible to include higher order interpolation schemes between PSFs using the weighting matrices. The case of linear interpolation in solving systems with spatially variant blur has been studied by Nagy & O'Leary (1998), but its inclusion complicates the procedure and did not provide a significant improvement to the quality of the solution and increased computation times (Nagy et al. 2002).

Bibliography

Ackley, D. H. 1987, A connectionist machine for genetic hillclimbing, (Boston: Kluwer)

Adorf, H.M., 1994, in The Restoration of HST Images and Spectra II, ed. R. J. Hanisch & R. L. White (Baltimore, MD: Space Telescope Science Institute), 72

Alard, C. 2009, A&A, 506, 609

Alcock, C., Allsman, R. A., Axelrod, T. S., Bennett, D. P., et al. 1993, Sky Surveys: Protostars and Protogalaxies, ASP Conference Series, 43, ed. Seifer, B. T.

Alcock, C. et al, 1997b, ApJ, 491, 436

Alcock, C. 2000, Science, 287, 5450, 74

Almassi, B. 2009, Studies in Hist.and Phil. of Mod. Phys., 40, 57

Bäck, T. 1996, Evolutionary algorithms in theory and practice, Oxford University Press

Bäck, T., Fogel, D. B. & Michalewicz, Z. 1997, Handbook of Evolutionary Computation, (New York, NY: Taylor & Francis Group)

Bäck, T. & Hoffmeister, F. 1991, Extended Selection Mechanism in Genetic Algorithms, Proc. 4th Int. Conf. on GAs, Eds: Bellew R. K. & Booker, L. B., (San Mateo, Cal: Morgan Kaufmann Publishers)

Baker, J. E. 1987, Reducing Bias and Inefficiency in the Selection Algorithm, Proc. 2nd Int. Conf. on GAs and Appl. Ed: Grefenstette, J. J., (Hillsdale, New Jersey: Lawrence Erlbaum Associated)

Bandara, K., Crampton, D., & Simard, L. 2009, ApJ, 704, 2, 1135

Baran, A. 2009, M.Sc. thesis, Univ. Manitoba

Bardsley, J. M. 2006, BIT Numer. Math., 48, 4, 651

Bersini, H., Dorigo, M., Langerman, S., Geront, G., & Gambardella, L. 1996 Results of the first international contest on evolutionary optimisation (1st iceo). In Proceedings of IEEE International Conference on Evolutionary Computation, 611, 20

Binney, J. & Tremaine, S. 1988, Galactic Dynamics, second edition, Princeton University Press

Biretta, J. 1994 in The Restoration of HST Images and Spectra II, ed. R. J. Hanisch & R. L. White (Baltimore, MD: Space Telescope Science Institute), 72

Björck, Å. 1996, Numerical Methods for Least Squares Problems, (Philadelphia, PA; SIAM)

Blandford, R. D., & Kochanek, C. S., 1987, ApJ, 321, 658

Boden, A.F., Redding, D.C., Hanisch, R.J., & Mo, J., 1995, J. Opt. Soc. Am. A, 13, 1537

Bolton, A. S., Burles, S., Koopmans, L. V. E., Treu, T. and Moustakas, L. A. 2006, ApJ, 638, 2, 703

Bolton, A. S., Burles, S., Koopmans, L. V. E., Treu, T., Gavazzi, R., Moustakas, L. A., Wayth, R. & Schlegel, D. J. 2008, ApJ, 682, 946

Bonnet, H., Mellier, Y. & Fort, B. 1994, ApJL, 427, 2, L83-L86

Bonnet, H. 1995, PhD thesis, L'Université Paul Sabatier de Toulouse

Bouchy, F., Pont, F., Santos, N. C., Melo, C., Mayor, M., Queloz, S. & Udry, S. 2004, A&A, 421, L13

Bourassa, R. R. & Kantowski, R. 1975, ApJ, 195, 1, 13

Brainerd, T. G., Blandford, R. D. & Smail, I. 1996, ApJ, 466, 623

Brewer, B. J., & Lewis. G. F. 2005, PASA, 22, 128

Brewer, B. J., & Lewis. G. F. 2006, ApJ, 637, 608

Brewer, B.J., Lewis, G. F., Belokurov, V., Irwin, M. J., Bridges, T. J. & Evans, N. W. 2011, MNRAS, 412, 4, 2521

Chae, K. H. 2007, ApJL, 658, 49

Chang, K. & Refsdal, S. 1979, Nature, 282, 561

Charbonneau, P. 1995, ApJ, 101, 309:334

Chen, Z.-P., Jian, J.-H., Li, Y. & Yu, R.-Q. 1999, Chem. and Int. Lab. Sys., 45, 409

Chung, J., Nagy, J. G., & O'Leary, D. P. 2008, Elec. Trans. Numer. Anal., 28, 149

Chwolson, O. 1924, Astron. Nach. 221, 329

Clerc, M. 2006, Particle Swarm Optimization, (Newport Beach, CA:ISTE)

Clerc, M. & Kennedy, J. 2002, IEEE Trans. on Evol. Comp., 6, 1, 58

Coles, P & Lucchin, F. 2002, Cosmology: The Origin and Evolution of Cosmic Structure, 2nd Ed., (Sussex, UK: Wiley)

Contaldi, C. R., Hoekstra, H. & Lewis, A. 2003, Phys. Rev. L., 90, 22

Coppin, K. E. K., Chapman, S. C., Smail, I., Swinbank, A. M., et al. 2010, MNRAS:Letters, 407, 1, L103

Cornwell T. J. 1982, VLA Sci. Mem. 141

d'Inverno, R. 1992, Introducing Einstein's Relativity, (Oxford: Clarendon Press)

de Vaucouleurs, G. 1948, Ann. d'Astroph., 11, 247

De Jong, K. 1975, An analysis of the behaviour of a class of genetic adaptive systems, PhD thesis, University of Michigan.

Deb, K. 1989, Genetic Algorithms in Multimodal Function Optimization, Master's thesis (Tuscaloosa: University of Alabama)

Deb, K. & Goldberg, D. E. 1989, An Investigation of Niche and Species Formation in Genetic Algorithm Optimization, Proc. 3rd Int. Conf. on GAs, 42

Deng, X.-F., He, J.-Z., Jiang, P., Luo, C.-H., Wu, P. 2007, A & A, 474, 783-791

Deng, X.-F., He, J.-Z., Jiang, P., Ma, X.-S., et al. 2007, Acta Physica Polonica B, 38, 10, 3303-3317

Diego, J. M., Protopapas, P., Sandvik, H. B. & Tegmark, M. 2005, MNRAS, 360, 477

Dobke, B. M., King, L. J., Fassnacht, C. D. & Auger, M. W., MNRAS, 397, 311

Dupke, R. A., Mirabal, N., Bregman, J. N., & Evrard, A. E. 2007, ApJ, 688, 2, 781

Dye, S., & Warren, S. J. 2005, ApJ, 623, 31

Dyson, F.W. Eddington, A.S.& Davidson, C. 1919, Phil. Trans. Roy. Soc. A, 220, 291

Engl, H.W., Hanke, M., & Neubauer, A. 2000, Regularization of Inverse Problems, (Dordrecht:Kluwer)

Einstein, A. 1905, Ann. der. Phys., 17, 821

Einstein, A. 1915, Konig. Preuss. Akademie der Wiss., 844

Einstein, A. 1916, Ann. der. Phys, 49, 7, 769

Einstein, A. 1917, Konig. Preuss. Akademie der Wiss., 142

Einstein, A. 1936, Science, 84, 2188 ,506

Ellithorpe, J., Kochanek, C. S. & Hewitt, J. N. 1996, ApJ, 464, 556

Engl, H. W., Hanke, M. & Neubauer, A. 1996, Regularization of Inverse Problems, (Dordrecht, Netherlands: Kluwer)

Faisal, M., Lanterman, A.D., Snyder, D.L., & White, R.L. 1995, J. Opt. Soc. Am. A, 12, 2593

Favati, P., Lotti, G., Menchi, O. & Romani, F. 2010, Inverse Problems, 26, 8, 085013

Fiege, J. D., Johnstone, D., Redman, R. O., & Feldman, P. A. 2004, Ap. J. 616, 925

Fiege, J. D. 2005, in ASP Conf. Ser. 343, Astronomical Polarimetry: Current Status and Future Directions, ed. A. J. Adamson et al. (San Francisco, CA: ASP), 171

Fiege, J. D., 2010, Qubist Users Guide: Optimization, Data Modeling, and Visualization with the Qubist Optimization Toolbox for MATLAB, (Winnipeg, Canada:nQube Technical Computing)

Fish, D. A., Grochmalicki, J., & Pike, E. R. 1996, J. Opt. Soc. Am. A, 13, 1

Fleming, H.E. 1990, Linear Algebr. Appl., 130, 133

Fomalont, E.B. & Sramek, R.A. 1975, ApJ, 199, 1, 749

Fonseca, C.M. & Fleming, P.J. 1993, in Proc. Fifth International Conf. on Genetic Algorithms, ed. S. Forrest (Urbana-Champaign, IL:Morgan Kaufmann), 416

Friedmann, A. 1922, Z. Phys., 10, 1, 377

Friesen, J., Rogers A. & Fiege, J. D. 2011, J. R. Astron. Soc. Can. 105, 61

Gilles, L., Vogel, C. R. & Bardsley, J. M. 2002, Inverse Probl. 18, 237

Girard, D. 1989, Numer. Math., 56, 1

Goldberg, D. E. 1989, Genetic Algorithms in Search, Optimization, and Machine Learning (Reading, MA: Addison-Wesley)

Goldberg, D. E.2002, The Design of Innovation: Lessons From and for Competent Genetic Algorithms (Norwell, MA: Kluwer)

Goldberg, D. E. & Deb, K. 1991, A Comparative Analysis of Selection Schemes used in Genetic Algorithms, Foundations of Genetic Algorithms, Ed. Rawlins, G. J. E., (San Mateo, Cal: Morgan Kaufmann Publishers)

Golub, G.H., & Reinsch, C. 1970, Numer. Math., 14, 403

Golub, G.H., Heath, M., & Wahba, G. 1979, Technometrics, 21, 2, 215-223

Grillo, C., Lombardi, M. & Bertin, G. 2008, A & A, 477, 397

Hamming, R. W. 1950, Bell System Technical Journal, 29, 2, 147

Hanke, M., 1995, Conjugate Gradient Type Methods for Ill-Posed Problems (Harlow, Essex:Longman)

Hanke, M., Hansen, P. C. 1993, Surv. Math. Ind., 3, 253

Hansen, P. C. 1992, SIAM Rev., 34, 561

Hansen, P.C., & O'Leary,D.P. 1993, SIAM J. Sci. Comput., 14, 1487

Hansen, P.C. 1994, Numer. Algorithms, 6, 1

Hansen, P.C. 1997, Rank-Deficient and Discrete Ill-Posed Problems, (Philadelphia PA: SIAM)

Hansen, P.C., Nagy, J.G., & O'Leary, D.P. 2006, Deblurring Images: Matrices, Spectra and Filtering (Philadelphia PA:SIAM)

Hansen, P.C. 2010, Discrete Inverse Problems: Insight and Algorithms (Philadelphia, PA: SIAM publishers)

Harvey, G. M. 1979, The Observatory, 99, 195

Hestenes, M. R. and Stiefel, E., 1952, Journal of Research of the National Bureau of Standards, 49, 6

Hewitt, J. N., Turner, E. L., Schneider, D. P., Burke, B. F., Langston, G. I. & Lawrence, C. R. 1988, Nature, 333, 537

Hobson, M. P., Efstathiou G. & Lasenby, A. N. 2006, General Relativity: An Introduction for Physicists (New York: Cambridge University Press)

Högbom, J. A. 1974, Astron. Astrophys. Supp., 15, 417

Holland J. H. 1975, Adaptation in Natural and Artificial Systems (Ann Arbor, MI: Univ. Michigan Press)

Hopkins, P. F., Murray, N. & Thompson, T. A. 2009, MNRAS 398, 303

Horn, J., Nafpliotis, N. & Goldberg, D. E. 1993, A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proc. 1st IEEE Conf. Evol. Comp., 1, Ed. Piscataway, N. J., 82

Hubble, E. 1936, ApJ, 84, 517

Hulse, R. A. & Taylor, J. H. 1974, ApJ, 191, L59

Jarosik, N., Bennett, C. L., Dunkley, J., et al. 2011, ApJS, 192, 14

Katsaggelos, A. K., Kang, M. G., & Banham, M. R. 1994, in The Restoration of HST Images and Spectra II, ed. R. J. Hanisch & R. L. White (Baltimore, MD: Space Telescope Science Institute), 3.

Kaiser, N. & Squires, G. 1993, ApJ, 404, 441

Kaufman, L. 1993, IEEE Trans. Med. Imag., 12, 2, 200

Kassiola, A. & Kovner I. 1993, ApJ, 417, 450

Kayser, R., & Schramm, T. 1988, A&A, 191, 39

Keeton, C.R. & Kochanek, C.S. 1998, ApJ, 495, 157-169, 1998

Kennedy J., Eberhart, R.C., & Shi, Y. 2001, Swarm Intelligence, (London:Morgan Kaufmann Publishing)

Kennefick, D. 2007, eprint arXiv:0709.0685

Kennicut R.C., Jr., et al. 2003, PASP, 115, 928

Kochanek, C. S. & Narayan, R. 1992,

Kochanek, C. S., Blandford, R. D., Lawrence, C. R. & Narayan, R. 1989, MNRAS, 238, 43

Kochanek, C. S., Falco, E. E., Impey, C., Lehar, J., McLeod, B. & Rix, H.-J. 1998, www.cfa.harvard.edu/castles/

Kochanek, C. S., Schneider, P., & Wambsganss, J. 2004, Proc. 33rd Saas-Fee Adv. Course, Part 2, ed. G. Meylan, P. Jetzer, & North, P. (Berlin: Springer)

Koopmans L. V. E. & Treu, T. 2002, ApJ, 568, L5

Kochanek, C. S. 2004, ApJ, 605,58

Koopmans, L. V. E. 2005, MNRAS, 363, 1136

Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S. & Moustakas, L. A. 2006, ApJ, 649, 2, 599

Kormann, R., Schneider, P., & Bartelmann, M. 1994, A&A, 284, 285

Kneib, J. P., Ellis, R. S., Smail, I., Couch, W. J. & Sharples, R. M. 1996, ApJ, 471, 643

Kravtsov, A. 2010, Advances in Astronomy, 281913

Kress, R. 1989, Linear Integral Equations, (Berlin: Springer)

Lauer, T. 2002, Proc. SPIE, 4847, 167

Liesenborgs, J., De Rijcke, S., & Dejonghe, H. 2006, MNRAS, 367, 1209

Liesenborgs, J., De Rijcke, S., Dejonghe, H., & Bekaert, P. 2007, MNRAS, 380, 1729

Liesenborgs, J., De Rijcke, S., Dejonghe, H., & Bekaert, P. 2009, MNRAS, 397, 341

Lynds, R., & Petrosian, V. 1986, Bull. Am. Astron. Soc. 18, 1014

Link, R., & Pierce, M. J., 1987, ApJ, 502, 63

Lukas, M. A. 1993, Numer. Math., 66, 41

Mao, S. & Schneider, P. 1998, MNRAS, 295, 587

Markevitch, M., Gonzalez, A. H., Clowe, D., Vikhlinin, A., Forman, W., Jones, C., Murray, S. & Tucker, W. 2004, Astrophys. J., 606, 2, 819

Marshall, P., Treu, T., Melbourne, J., et al. 2009, ApJ, 671,2, 1192

Michalewicz, Z. 1992, Genetic Algorithms + Data Structures = Evolution Programs (New York: Springer)

Mishra, S. K. 2006, MPRA Paper 1742, http://mpra.ub.uni-muenchen.de/1742/

Misner, C. W., Thorne, K. S. & Wheeler, J. A. 1973, Gravitation, (San Francisco: W. H. Freeman)

Mitchell, M. 1996, An Introduction to Genetic Algorithms, (Cambridge, MA: MIT Press)

Moffat, A. 1969, A&A, 3, 455

Mhlenbein H., Schomisch D. & Born J. 1991, Parallel Computing, 17, 619

Nagy, J. G. & O'Leary, D. P. 1998, SIAM J. Sci. Comput. 19, 1063

Nagy, J.G., Palmer, K.M., & Perrone, L. 2002, Numer. Algorithms, 36, 73

Nagy, J. G. & Palmer, K. M. 2003, BIT Numer. Math., 43, 1003

Nagy, J. & Strakoš, Z. 2000, Math. Model., Est. and Imag., 4121, eds. Wilson, D. C. et al., 182

Narayan, R. & Bartelmann, M. 1995, Proc. 1995 Jer. Winter School, eds: Dekel, A., Ostriker, J.P., Cambridge University Press

Narayan, R. & Nityananda, R., 1986, Ann. Rev. Astron. Astrophys., 24, 127

Negrello, M., Hopwood, R., De Zotti, G., et al. 2010, Science, 330, 6005, 800

Nocedal, J. & Wright, S. J. 1999, Numerical Optimization (New York: Springer)

Odenwald, S., Newmark, J. & Smoot, G. 1998 ApJ 500, 2, 554

Oguri, M., Inada, N., Strauss, M. A. et al. 2008, AJ, 135, 512

Paczynski, B. 1986, ApJ, 304, 1

Paczynski, B. 1987, Nature, 325, 6105, 572

Patnaik, A. R., Browne, I. W. A., King, L. J. et al. 1993, MNRAS, 261, 2, 435

Peacock, J. A. 1998, Cosmological Physics, (New York: Cambridge University Press)

Peng, C. Y., Ho, L. C., Impey, C. D. & Rix, H.-W. 2010, AJ, 139, 2097

Perlmutter, S., Aldering, G., Goldhaber, G., et al. 1999, ApJ, 517, 565

Petters, A.O., Levine, H. & Wambsganns, J. 2001, Singularity Theory and Gravitational Lensing (Boston, MA: Birkhäuser)

Poli, R., Kennedy, J., & Blackwell, T. 2007, Swarm Intell., 1, 33

Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. 2007, Numerical Recipes: The Art of Scientific Computing (3rd ed.; New York:Cambridge Univ. Press)

Price, K. V., Storn, R. M. & Lampinen, J. A. 2005, Differential Evolution: A Practical Approach (Berlin: Springer)

Read, J. 2003, http://www.qgd.uzh.ch/programs/pixelens/

Refregier, A., Massey, R., Rhodes, J., et al. 2004, AJ, 127, 6, 3102

Refsdal, S. 1964, MNRAS, 128, 295

Riess, A. G., Filipenko, A. K., Challis, P. et al. 1998, AJ, 116, 3, 1009

Robertson, H. P. 1935, ApJ, 82, 284

Robertson, D.S., Carter, W.E. & Dillinger, W.H. 1991, Nature, 349, 768

Rogers, A. & Fiege, J.D. 2011, ApJ, 727, 2, 80

Rogers, A. & Fiege, J. D., 2011, ApJ, 743, 1, 68

Rubin, V., et al. 1962, AJ, 67, 8

Saad, Y., & Schultz, M.H.1986, SIAM J. Sci. Stat. Comput., 7, 856

Sackett, P. D. 1995, www.mso.anu.edu/ psackett/NVWS/microPLANET.html

Saha, P., & Williams, L. R. 1997, MNRAS, 292,148

Saha, P., Coles, J., Macció, A.V. & Williams, L. R. 2006, ApJ, 650, L17

Saha, P., Williams, L. L. R., & Ferreras, I. 2007, ApJ, 663, 29

Saranti, D. W., Petrosian, V. & Lynds, R. 1996, ApJ, 458, 57

Schaerer, D., Hempel, A., Pelló, R., Egami, E., Richard, J., Kneib, J.-P. & Wise M. Proc. IAU 2, IAU symposium 235, Eds. Coombs, F. & Palons J.

Schmidt, M. 1963, Nature, 197, 1040

Schneider, P. 1985, A&A 143, 413

Schneider, P., Ehlers, J., & Falco, E.E. 1992, Gravitational Lenses, (Berlin:Springer Verlag)

Schneider, P., Kochanek, C. S. & Wambsganss, J. 2006, Gravitational Lensing: Strong, Weak and Micro, Saas-Fee Advanced Courses, 33, (Berlin: Springer)

Schramm, T. 1990, Astron. Astrophys., 231, 19

Schramm, T., & Kayser, R. 1987, A&A, 174, 361

Sérsic, J. L. 1968, Atlas de Galaxies Australes (Cordoba: Observatorio Astronomica)

Simard, L., et al. 2002, ApJS, 142,1

Shapiro, I. I. 1964, PRL, 13, 26, 789

Shapiro, P. R., Iliev, I.T. & Raga, A. C. 1999, MNRAS, 307, 203

Shewchuk, J. R. 1994, Lecture notes, Carnegie Mellon University

Shi, Y. & Eberhart, R. 1998, Evol. Comp. Proc., 69

Skilling, J., & Bryan, R. K. 1984, MNRAS, 211, 111

Sokolov, A. & Whitley, D. 2005, Proc. 7th Gen. Evol. Comp. Conf., Washington, DC, 1131

Soldner, J. G. v., 1804, On the Deflection of a light ray from it's rectilinear motion, by the attraction of the body at which it passes by, Berliner Astronomisches Jahrbuch, 161

Spears, W. M. & De Jong, K. A. 1991, On the Virtues of Parameterized Uniform Crossover, Proc. ICGA 4, 230

Spergel, D. N., Verde, L., Peiris, H. V. et al, 2003, ApJS, 148, 1, 175

Suyu, S. H., Marshall, P. J., Hobson, M. P., & Blandford, R. D. 2006, MNRAS, 371, 983

Suyu S. H., Blandford R. D., 2006, MNRAS, 366, 39

Suyu, S. H., Marshall, P. J., Auger, M. W. et al. 2010, ApJ, 711, 1, 201

Stark, D.P, Swinbank, A. M., Ellis, R. S., et al. 2008, Nature, 455, 775

Subramanian, K. & Chitre, S. M. 1984, ApJ, 289, 37

Syswerda, G. 1989, Uniform Crossover in Genetic Algorithms, Proc. ICGA 3, 2

The Microlensing Observations in Astrophysics (MOA) Collaboration & The Optical Gravitational Lens Experiment Collaboration 2011, Nature, 473, 349

Tikhonov, A. N. 1963, Sov. Math., 4, 1035

Treu, T. & Koopmans, L. V. E. 2004, ApJ, 611, 739

Treuhaft, R.N. & Lowe, S.T. 1991, AJ, 102, 5, 1879

Trussell, H. J. & Fogel, S. 1992, IEEE Trans. Image Proc. 1, 123

Trussell, H. J. & Hunt, B. R. 1978, IEEE Trans. Acoust. Speech, Signal Processing, 26, 608

Twomey, S. 1963, J. Assoc. Comput. Mach., 10, 97

Tyson, J. A., Wenk, R. A. & Valdes, F. 1990, ApJL, 2, 349, L1

Tyson, J.A., Kochanski, G.P., & dell'Antonio, I.P. 1998, ApJ, 498, 107

Udalski, A., Szymanski, M., Kaluzny, J., Impey, C., Lehar, J., McLeod, B. & Rix, H. W. 1992, Acta Astronomica, 42, 235

van Waerbeke, L., Mellier, Y., Erben, T. et al 2000, Astron. Astrophys., 358, 30

van Waerbeke, L. & Mellier, Y. 2003 eprint, astro-ph/0305089

Vegetti, S., & Koopmans, L. V. E. 2009, MNRAS, 392, 3, 945

Vogel, C. R. 1987, Report, Dept. of Mathematical Sciences, Montana State University, Bozeman

Vogel, C.R. 1989, Solving Ill-Conditioned Linear Systems using the Conjugate Gradient Method, (Technical report, Montana State Univ.)

Vogel, C.R. 2002, Computational Methods for Inverse Problems (Frontiers in Applied Mathematics Series, 23; Philadelphia PA:SIAM)

Wahba, G. 1977, SIAM J. Numer. Anal., 14, 651

Wahba, G., Golub, G., & Heath M., 1979, Technometrics, 21, 215

Walker, A. G. 1937, Proc. London Math. Soc., 2, 42, 1, 90

Wallington, S., Kochanek, C. S., & Narayan, R. 1996, ApJ, 465, 64

Walsh, D., Carswell R.F. & Weymann R. J. 1979, Nature, 279, 381

Warren, S. J. & Dye, S. 2003, ApJ, 590, 2, 673

Wayth, R. B., & Webster, R. L. 2006, MNRAS, 372, 3, 1187

Whitley, D. 1989, The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best, Proc. 3rd Int. Conf. on GAs, Ed: Schaffer, J. D., (San Mateo, Cal: Morgan Kaufmann Publishers)

Will, C. M., 1993, Was Einstein Right?: Putting General Relativity to the Test, (New York, BasicBooks)

Williams, R. E., Blacker, B., Dickinson, M., et al. 1996, AJ, 112, 1335

Witt, H. J. 1995, ApJ, 449, 42

Wright, A. H. 1991, Genetic Algorithms for Real Parameter Optimization, Found. of Gen. Alg., Ed. Rawlings, J. E., 205

Wucknitz, O. 2004, MNRAS, 349, 1

Zitzler, E., Deb, K. & Thiele, L. 2000, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evol. Comp., 8, 2, 173

Zwicky, F. 1937, Phys. Rev., 51, 290

Zwicky, F. 1937, Phys. Rev. 51, 679