Mon. Not. R. Astron. Soc. **360,** 477–491 (2005) doi:10.1111/j.1365-2966.2005.09021.x

Non-parametric inversion of strong lensing systems

J. M. Diego,★† P. Protopapas,† H. B. Sandvik† and M. Tegmark†
University of Pennsylvania, 209S, 33rd St, Department of Physics & Astronomy, Philadelphia, PA 19104, USA

Accepted 2005 March 15. Received 2005 March 14; in original form 2004 September 1

ABSTRACT

We revisit the issue of non-parametric gravitational lens reconstruction and present a new method to obtain the cluster mass distribution using strong lensing data without using any prior information on the underlying mass. The method relies on the decomposition of the lens plane into individual cells. We show how the problem in this approximation can be expressed as a system of linear equations for which a solution can be found. Moreover, we propose to include information about the *null space*. That is, we make use of the pixels in which we know there are no arcs above the sky noise. The only prior information is an estimation of the physical size of the sources. No priors on the luminosity of the cluster or shape of the haloes are needed, making the results very robust. In order to test the accuracy and bias of the method we make use of simulated strong lensing data. We find that the method reproduces accurately both the lens mass and source positions and provides error estimates.

Key words: methods: data analysis – galaxies: clusters: general – dark matter.

1 INTRODUCTION

Analysis of strong lensing images in galaxy clusters is one of the more subjective and intuitive fields of modern astronomy (see Blandford & Narayan 1992; Schneider, Ehlers & Falco 1993; Wambsganss 1998; Narayan & Bartelmann 1999; Kneib 2002, for comprehensive reviews on gravitational lensing). The common use of parametric models means that educated choices about the cluster mass distribution have to be made, for instance that the dark matter haloes follow the luminous matter in the cluster or that galaxy profiles posses certain symmetries. Once these choices are made, one can only hope they are the right ones. The recent improvement in strong lensing data is impressive, however, and should increasingly allow us to extract information using fewer assumptions. For instance, it should allow us to relax assumptions about the mass distributions of the cluster and instead enable us to test this. The approach of testing rather than assuming the underlying physics has been used in other branches of astrophysics and cosmology (e.g. Tegmark 2002), but has so far been largely absent in strong lensing analyses.

There are two relatively distinct fields within the area of strong lensing, namely lensing by galaxies and lensing by galaxy clusters. These processes are different both in appearance and in abundance. Galaxy-cluster systems are few and far between, and only the most abnormally dense clusters have surface densities greater than the critical density for lensing. They are, however, much more spectacular than their galaxy lens counterparts. The most massive clusters are able to create multiple images of extended objects such as distant galaxies with image separations of up to $\sim$1 arcmin. Arguably the most impressive system, A1689 (Broadhurst et al. 2005), boasts more than 100 multiple images of some 30-plus sources. Galaxy lens systems are of course on a much smaller scale, with image separations of a few arcseconds. They are far more abundant, but less impressive. The lensed objects that we are able to observe are typically high-redshift quasars, on account of their high luminosity and point-like structure, although galaxy–galaxy lensing has been observed (Brainerd, Blandford & Smail 1996; Hudson et al. 1998; Guzik & Seljak 2002).

The classification of strong lensing systems into these two groups is not merely a matter of scale. The baryons in galaxies have had time to cool and form the visible galaxy, thereby giving a cuspy profile, suitable for lensing. The cooling time for galaxy clusters exceeds the Hubble time, and the density profile is therefore far less cuspy, making clusters less ideal lenses. Although cluster lens systems are scarce, the impressive number of lensed images in each system means that they still contain a lot of information, particularly regarding cluster mass profiles. The information is harder to extract because of the relatively complicated gravitational potentials, but if this challenge can be overcome they could be highly useful probes, certainly of cluster physics and potentially also for cosmology (Yamamoto & Futamase 2001; Yamamoto et al. 2001; Chiba & Takahashi 2002; Golse, Kneib & Soucail 2002; Sereno 2002; Meneghetti et al. 2004). The intention of this paper is to improve the methods for extracting this information.

Although alternative approaches have been suggested for recovering the density field in both the weak and strong lensing regimes

★Present address: MIT Center for Space Research, Building 37-582d, 70 Vassar Street, Cambridge, MA 02138, USA.
†E-mail: jdiego@space.mit.edu (JMD); pavlos@physics.upenn.edu (PP); sandvik@hep.upenn.edu (HBS); max@hep.upenn.edu (MT)

(see for instance Kaiser & Squires 1993; Broadhurst, Taylor & Peacock 1995; Kaiser 1995; Schneider 1995; Schneider & Seitz 1995; Seitz & Schneider 1995; Bartelmann et al. 1996; Taylor et al. 1998; Tyson, Kochanski & Dell'Antonio 1998; Bridle et al. 1998; Marshall et al. 2002), the standard approach to modelling strong cluster lenses is to use parametric methods. This is motivated by the fact that the data usually do not contain more than a few arcs. This is not enough to constrain the mass distribution without the help of a parametrization. Parametric methods rely heavily on assumptions or priors on the mass distribution (Kochanek & Blandford 1991; Kneib et al. 1993, 1995, 1996, 2003; Colley, Tyson & Turner 1996; Tyson et al. 1998; Broadhurst et al. 2000, 2005; Sand, Treu & Ellis 2002). A common prior is the assumption that there is a smooth dark matter component that is correlated spatially with the centroid of the luminous matter in the cluster. The mass is then usually modelled by a large smooth dark matter halo placed on top of the central galaxy or the centroid of the luminous matter, as well as smaller dark matter haloes located in the positions of the other luminous galaxies. The parameters of each halo are then adjusted to best reproduce the observations.

There is plenty of subjectivity involved in this process, particularly in the addition of the dominant dark matter component to the cluster. The assumption that the dark matter follows the luminosity is necessary but remains the Achilles heel of parametric lens modelling. For large clusters the number of parameters in the parametric lens model quickly becomes large, but there is still no guarantee that the parametric model used is in fact capable of reproducing the mass distribution well. It is not hard to envisage complications such as dark matter substructure, asymmetric galaxy profiles, interactions between individual galaxies and the cluster, and even dark matter haloes without significant luminosity, all of which would not be well represented by typical parametric methods. In these cases, where the number of parameters is large, we may want to consider alternative non-parametric methods in which the previous problems do not have any effect on any of the assumptions. When the number of parameters is comparable in both parametric and non-parametric cases, it is interesting to explore non-parametric methods since they do not rely on the same assumptions.

This paper is not an attack on parametric methods, but rather is a defence of non-parametric alternatives. Parametric methods are usually the best way to obtain information about the gravitational potential (few multiple images, simple lens) and are not affected by resolution problems, especially in the centre of the lens, which can play a key role in reproducing the radial arcs. They do, however, suffer from some potential problems, some of which have been outlined above. These problems combined with the impressive quality of recent and upcoming data lead us in this paper to explore the potential of constraining the mass distribution without imposing priors on it. In other words we want to know what strong gravitational lensing images can tell us about cluster mass profiles whilst pretending we know nothing about the luminosity. Upcoming images of strong lensing in galaxy clusters will contain of the order of 100 arcs and should make this a manageable task (Diego et al. 2004). A non-parametric approach will provide an important consistency check, since concurring results would lend strength to the parametric approach, whereas any resulting differences would need to be addressed.

Accepting this challenge, we present in this paper a new method that makes use of all the available information in super-high-quality strong lensing systems. The method has also been thoroughly tested with simulated lensing data with very good results. We thus address the crucial issue of how well we can reproduce both the cluster mass profile and the positions and shapes of the background galaxies.

We want to stress that ours is by no means the first work proposing the use of non-parametric methods in strong lensing. Among the non-parametric methods already available are the pixellated methods of Saha & Williams (1997), Saha (2000) and Abdelsalam, Saha & Williams (1998a,b), which first established many of the ideas revisited in this work, as did the multipole approach of Kochanek & Blandford (1991) and Trotter, Winn & Hewitt (2000).

Naturally, our work shares many similarities with this former work, but there are also important and interesting differences. First, in Saha & Williams (1997) (and subsequent papers), the authors divide the lens plane into a grid similar to in the method presented here, but with the important difference that their grid is fixed while our grid is dynamical. Our grid adapts to the new estimated mass at each step. This has important implications for the solution since high-density regions will be sampled more heavily. These dense regions play a key role, particularly in the positions of the radial arcs. Failure to sample these regions properly will necessarily lead to a biased mass distribution.

Another important difference from Saha & Williams (1997) is that they make use of a prior on the mass distribution, penalizing deviations from the distribution of luminous matter. They claim that this prior does not play an important role, but we find this questionable. It is hard to quantify the effect since the method apparently was not tested on simulated lensing data. In contrast, as mentioned above, our method is thoroughly tested with simulations to quantify how well the mass distribution is recovered. We do not use any prior other than a *physical* prior on the sizes of the sources. This prior is proved to be weak provided it is chosen with a minimum of wisdom.

A third important difference between the work presented in this paper and others is that we show how to speed up the algorithm significantly by adopting techniques commonly used in optimization problems.

Moreover, as an added novelty, our algorithm includes for the first time information about the null space. Rather than using only the information in the lensed arcs, we use information that has hitherto been overlooked – the areas in the sky where no arcs are observed.

The paper is laid out as follows. In the next section we present the lens inversion problem, and provide a linear approximation. We then in Section 3 present the simulated lensing data used to test the method, before we go on to present the adaptive gridding of the lens plane in Section 4. In Section 5 we describe the various inversion algorithms, and finally in Section 6 we introduce the use of the complementary (null) space.

2 THE PROBLEM FORMULATED IN ITS BASIC *LINEAR* FORM

The fundamental problem in lens modelling is the following. Given the positions of lensed images, $\boldsymbol{\theta}$, what are the positions of the corresponding background galaxies $\boldsymbol{\beta}$ and the mass distribution of the lens, $M(\boldsymbol{\theta})$? Mathematically, this entails inverting the lens equation

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta}, M(\boldsymbol{\theta})), \tag{1}$$

where $\boldsymbol{\alpha}(\boldsymbol{\theta})$ is the deflection angle created by the lens, which depends on the observed positions, $\boldsymbol{\theta}$. From now on we will omit the vector notation unless otherwise noted. The *data points* of our problem, θ, are given by the x and y positions of each of the pixels forming the observed arcs.

The deflection angle, α, at the position θ is found by integrating the contributions from the whole mass distribution:

$$\alpha(\theta) = \frac{4G}{c^2}\frac{D_{\rm ls}}{D_{\rm s}D_{\rm l}}\int M(\theta')\frac{\theta-\theta'}{|\theta-\theta'|^2}\,{\rm d}\theta', \tag{2}$$

where $D_{\rm ls}$, $D_{\rm l}$, and $D_{\rm s}$ are the angular distances from the lens to the source galaxy, the distance from the observer to the lens, and the distance from the observer to the source galaxy respectively. In equation (2) we have made the usual thin-lens approximation, so the mass $M(\theta')$ is the projected mass along the line of sight θ'. Owing to the (non-linear) dependence of the deflection angle, α, on the position in the sky, θ, this problem is usually regarded as a typical example of a non-linear problem. We will see that this is only partially true.

The next approximation we make is to split the lens plane into $N_{\rm c}$ small regions (hereafter cells) over which the projected mass is more or less constant. We can then rewrite equation (2) as

$$\alpha(\theta) = \frac{4G}{c^2}\frac{D_{ls}}{D_{\rm s}D_{\rm l}}\sum_{N_{\rm c}} m_i\frac{\theta-\theta_i}{|\theta-\theta_i|^2}. \tag{3}$$

The first point we want to make here is that the deflection angle α may be thought of as the net contribution of many small masses m_i in the positions θ_i, each one pulling the deflection in the direction of $(\theta-\theta_i)$ with a magnitude that is proportional to $m_i/(\theta-\theta_i)$. If we divide the lens plane into a grid with $N_{\rm c}$ cells, each mass m_i can be considered as the mass contained in the cell i ($i=1,\ldots,N_{\rm c}$). If the cells are sufficiently small then the above pixellization of the mass plane will give a good approximation to the real mass distribution.

Our second point is that the problem is non-linear in one direction but linear in the other. That is, given a position in the sky θ (and given a lens) there is only one β that satisfies equation (1), but, given a position of the background galaxy β, there may be more than one position in the sky (θ) satisfying equation (1); equivalently, the source galaxy may appear lensed in more than one position in the sky. The linear nature of the problem is evident when one realizes that the only non-linear variables, namely the θ positions, are fixed by the observation and that the problem depends linearly on the unknowns (β positions and masses, m_i, in the cells).

Let us now assume that we have a data set consisting of a system of radial and tangential strong lensing arcs that are spread over N_θ pixels in the image. We will also assume that we know which arcs originate from a given source. Since both the data and the mass distribution have been discretized we can rewrite equation (1) as a simple matrix equation:

$$\beta = \theta - \Upsilon M, \tag{4}$$

where θ and β are now $2N_\theta$-element vectors containing x and y values of the observed positions and the (unknown) source positions respectively. M is the mass vector containing all $N_{\rm c}$ mass cells, and Υ is the $(2N_\theta \times N_{\rm c})$ matrix casting the mass vector into a vector of displacement angles. A more technical account of the make-up of the Υ matrix can be found in the Appendix. Equation (4) clearly demonstrates the linear nature of the problem when formulated in this manner. The problem has now been reduced to a set of $2N_\theta$ linear equations with $2N_\theta + N_{\rm c}$ unknowns (lens masses and source galaxy positions). Notice that, when the problem is formulated in this form, there are more unknowns than equations, which means that it is an underdetermined system with an infinite number of solutions. In order to identify a suitable solution for such a system, we need to add extra information or impose constraints.

One way of doing this is by reducing the number of unknowns, for instance by removing the source positions from the unknown category. This can be achieved by minimizing the dispersion in the source plane; that is, by demanding that the pixels in the source plane be as concentrated as possible for each source. In this case we are left with only $N_{\rm c}$ unknowns.

Another way of constraining the system is by assuming that the $N_{\rm s}$ sources are well approximated by point sources, which reduces the number of unknowns to $N_{\rm c} + 2N_{\rm s}$. This means effectively demanding that all observed θs for arcs corresponding to a given source can be traced back through the lens to $N_{\rm s}$ single points. With this assumption we can rewrite the lens equation in the compact form

$$\theta = \Gamma X. \tag{5}$$

Γ is now a matrix of dimension $2N_\theta \times (N_{\rm c}+2N_{\rm s})$ and X is a vector of dimension $(N_{\rm c}+2N_{\rm s})$ containing all of the unknowns in our problem (see Appendix A), the mass elements and the $2N_{\rm s}$ central (x and y) coordinates of the $N_{\rm s}$ sources. Now the system is mathematically overdetermined (more data points that unknowns) and has a unique *point-source solution*. This unique solution can be found by numerical methods, as we will see later.

The linearization of the problem means that it is in principle solvable by both matrix inversion and simple linear programming routines. In practice, the problem quickly becomes ill-conditioned and too large for direct matrix inversion, and approximate numerical methods are more suitable. The main problem with the linearization is that we do not know if the obtained linearized solution creates artificial tangential or radial arcs. Checking this requires forward solving of the lens equation, which is non-linear due to the complicated dependence of the deflection angle on θ.

We suggest a novel approach to this problem, namely using all the available information in the images; that is, we use the information inherent in the *dark areas* (pixels containing no arcs) as well as the observed arcs. By tracing the dark areas back through the lens and imposing the condition that they fall outside the sources it is possible to find the true solution without overpredicting arcs. This use of the null space is to our knowledge unprecedented, and in principle allows for a complete, linear solution to a problem usually considered non-linear.

3 SIMULATIONS

Before proceeding to invert the system of linear equations (5), it is instructive to take a closer look at the simulations that are going to be used to test the inversion algorithms. Given that most methods rely on the luminous mass distribution, this is particularly important should the galaxies not trace the dark matter accurately.

Our simulations consist of three elements. The first element of the simulation is the projected mass distribution (M) in the lens plane. We simulate a generic mass distribution of a cluster with a total projected mass of $1.119\times10^{15}\,h^{-1}\,{\rm M}_\odot$ in the field of view located at redshift $z=0.18$. The field of view of our simulation is 0.1. The mass profile is built from a superposition of 20 NFW profiles with added ellipticity. The halo masses vary from 0.25×10^{15} to $2\times10^{12}\,h^{-1}\,{\rm M}_\odot$. In the context of lensing, NFW halos seem to reproduce well the shear profile of massive clusters up to several megaparsecs (Dahle, Hannestad & Sommer-Larsen 2003; Kneib et al. 2003; Broadhurst et al. 2005). The final mass distribution is shown in Fig. 1. In the same figure we also show the two critical curves. The interior one is the radial critical curve and the exterior one the tangential critical curve. Both curves have been

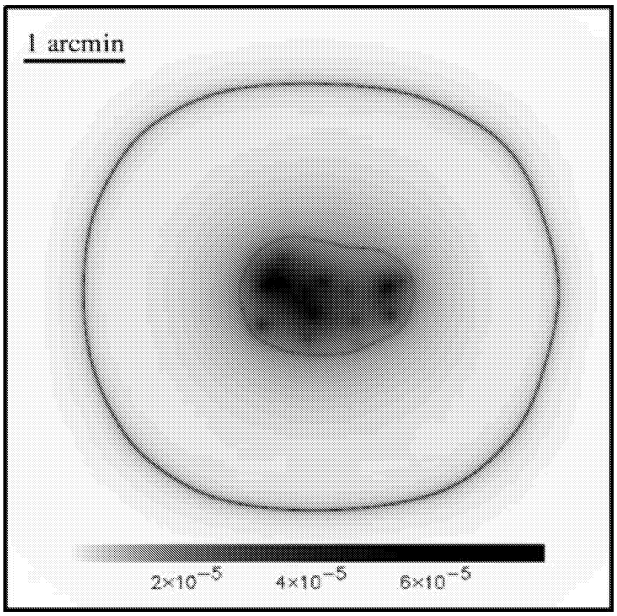

Figure 1. Original simulated mass profile. The total projected mass is $1.119 \times 10^{15}\, h^{-1}\, \mathrm{M}_{\odot}$ in the field of view (0.°1 across) and the cluster is at $z = 0.18$. The units of the grey-scale map are $10^{15}\, h^{-1}\, \mathrm{M}_{\odot}\ \mathrm{pixel}^{-1}$, where each pixel is 0.494 arcsec^2. Also shown are the radial (small) and tangential (large) critical curves for a source at redshift $z = 3$.

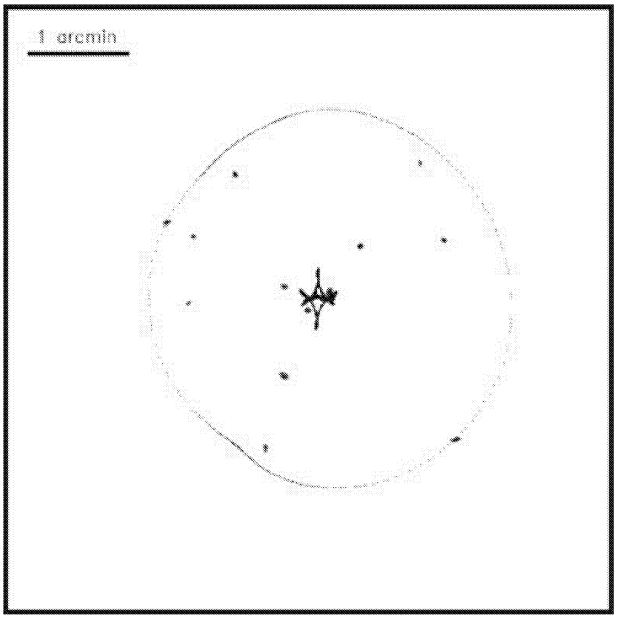

Figure 2. The 13 sources at redshift $z \in (1, 6)$. The field of view is 0.°1 across and is centred at the same point as in Fig. 1. The lines mark the position of the caustics at $z = 3$ corresponding to the radial (interior line) and tangential (exterior line) critical curves.

calculated assuming that the source is at redshift $z = 3$. The tangential critical curve is usually associated with the Einstein radius. Its large size (radius ≈ 2 arcmin) is due to the unusually high projected mass of our cluster plus its relatively low redshift ($z = 0.18$). Although the deduced Einstein radius is larger than the one observed in clusters, the simulation will serve the purpose of testing the various methods discussed here. More realistic simulations with proper mass distribution and source density will be presented in a future paper.

The second element in our simulation is the sources (β), for which we extract 13 sources from the HUDF (*Hubble Ultra Deep Field*, Beckwith et al. 2003). We assign them a redshift ($z \in [1, 6]$) and size and place them in distinct positions behind the cluster plane.

The third element is the lensed images (θ), which are calculated from the first two elements (M and β). This is achieved through a simple ray-tracing procedure. For each position θ in the image, we calculate the deflection angle, α, and then the corresponding source plane position, β, according to the lens equation (equation 1). If the calculated β coincides with one of the original sources, we assign to the lensed image the value (colour) of that source; otherwise, that point in the lensed image is left dark (value 0). We repeat the operation for all the pixels (θs) to produce a complete image. Furthermore, since the sources are small compared with our pixel size, and to avoid missing some sources, we oversample our θ pixels by subdividing them and checking each pixel at different locations ($\theta + \Delta\theta$ with $\Delta\theta < 1$ and $\theta \in [1, 512]$). The original sources are plotted in Fig. 2 and the corresponding θs in the left panel of Fig. 3.

4 GRIDDING THE MASS DISTRIBUTION

4.1 The Υ matrix

As detailed in Section 2, the basis of our non-parametric reconstruction method is the assumption that the real mass can be well approximated by a pixellized mass distribution. The Υ matrix acts as a convolution kernel, transforming the mass vector into the vector of deflection angles

$$\alpha_i = \Upsilon_{ij} M_j. \tag{6}$$

For convenience, rather than taking the mass distribution in each cell to be constant we assume that it follows a Gaussian distribution centred at the centre of the cell with some dispersion. This allows us to calculate analytically the lensing contribution from each mass cell, saving valuable computer time. We use a dispersion of $2a$, where a is the size of the cell, and we have confirmed with simulations that the lensed image generated with this choice agrees well with the true lensed image using a relatively small number of cells. The detailed structure of the Υ matrix is explained in Appendix A.

4.2 Multiresolution mass grid

Rather than taking a uniform grid, it is better to construct a *dynamical*, or *multiresolution*, grid. By sampling dense regions more heavily, it is possible to reduce drastically the number of cells needed to reproduce accurately the lensing properties of the cluster. In other words, we choose an adaptive grid that samples the dense cluster centre better than the outer regions. Since we do not actually know the density profile of the cluster, this multiresolution grid must be obtained through an iterative procedure. An example of this process can be found on the SLAP[1] (Strong Lensing Analysis Package) webpage.

Given a mass estimate (a first mass estimate can be obtained with a coarse regular grid), we split a given cell into four subcells if the mass in the cell exceeds some threshold value. The lower this threshold, the higher the number of divisions and consequently

[1] see http://darwin.cfa.harvard.edu/SLAP/

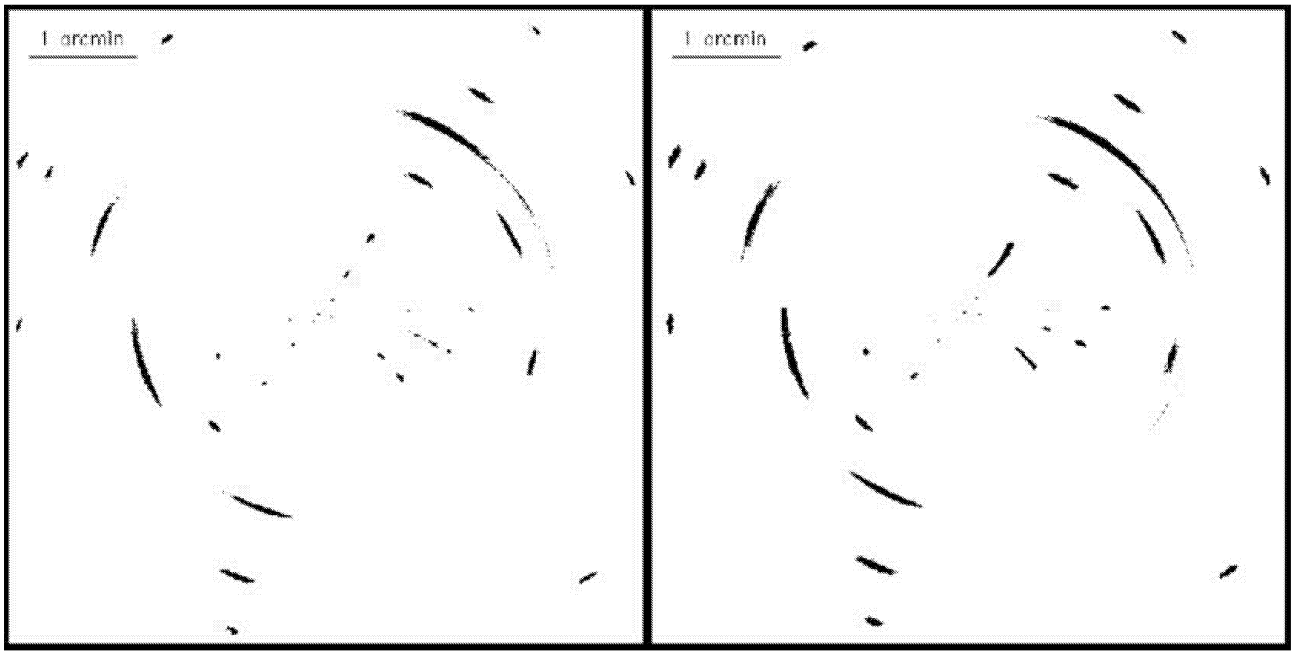

Figure 3. The sources lensed by the mass distribution of Fig. 1. The right image is the lensed image using 325 cells in the dynamical grid (see text). The left image is the exact solution using no gridding. Note the differences in the radial arcs.

the higher the final number of cells. The obtained grid can then be used for the next mass estimate, and the process can be repeated as necessary. Typically the mass estimate will improve with each iterative step, as this dynamical grid allows for the relevant regions of the cluster to become resolved.

In Fig. 4 we show an example of a gridded version of the true mass in Fig. 1 for a threshold of $M_{\rm thr} = 8.0 \times 10^{12}\, h^{-1}\, {\rm M}_{\odot}$. This grid has 325 cells. The corresponding (true) mass in the grid is shown in Fig. 5. The parameter $M_{\rm thr}$ or equivalently the number of cells, $N_{\rm c}$, can be viewed as a free parameter but also as a prior. Fixing it means that we are fixing the minimum scale or mass that we are sensitive to. In a parallel paper (Diego et al. 2004) we explored the role of $N_{\rm c}$ and found that it can have a negative effect on the results if it takes very large values. A natural upper limit for $N_{\rm c}$ is twice the number of pixels in the θ vector minus twice the number of sources (the factor of 2 accounts for the x and y positions). Our simulations show that a good election for $N_{\rm c}$ is normally 1/4 of this upper limit.

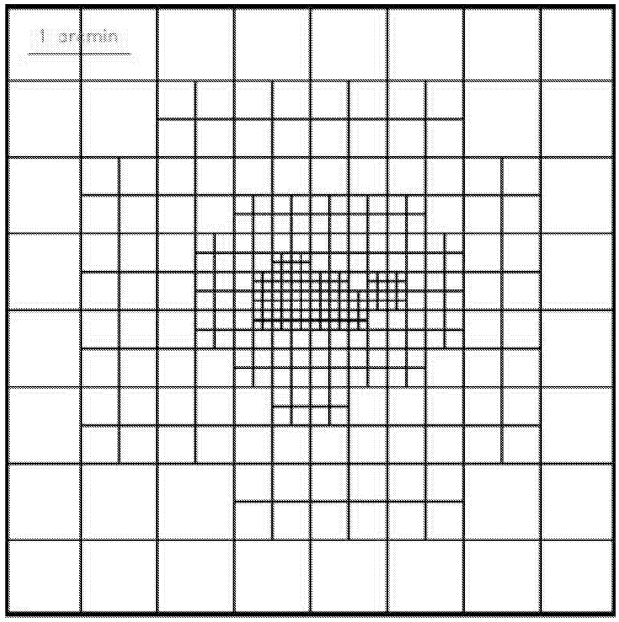

Figure 4. Dynamical grid for the mass in Fig. 1. This case corresponds to $M_{\rm thr} = 8.0 \times 10^{12}\, h^{-1}\, {\rm M}_{\odot}$.

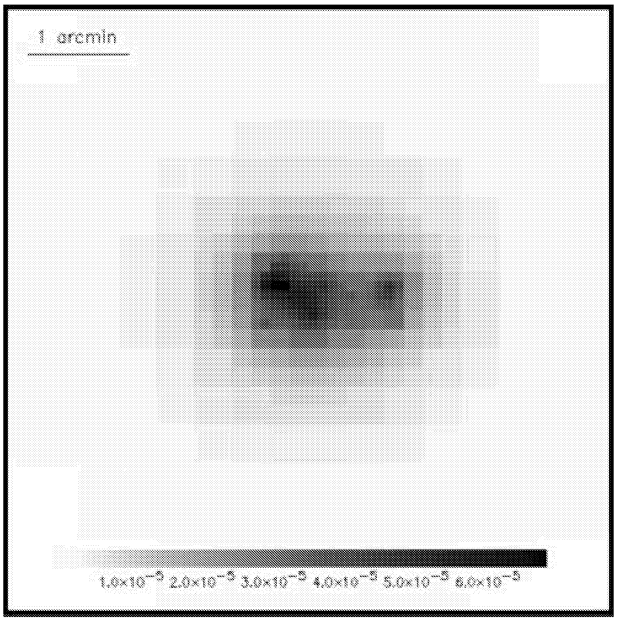

Figure 5. Masses in the cells of the dynamical grid of Fig. 4.

For the sake of clarity it is useful to give some numbers. The image on the left of Fig. 3 has 512^2 pixels from which 2156 pixels are part of one of the ≈35 arcs, so $N_\theta = 2156$. These arcs come from 13 sources ($N_s = 13$) and the lens plane is typically divided into a few hundred cells ($N_c \sim 300$).

5 INVERSION METHODS

In this section we will describe some inversion methods that can be applied to solve the problem. Most of these methods can be found in popular books such as *Numerical Recipes* (Press et al. 1997). All these algorithms are implemented in the package SLAP, which will be made available soon.

Once we have the problem formulated in its linear form with all the unknowns on one side it is tempting to try a direct inversion of equation (5). Although the Γ matrix is not square, one can find its inverse, Γ^{-1}, by decomposing Γ into orthogonal matrices. This is similar to finding the *eigenvalues* and *eigenvectors* of Γ. This approximation has its advantages as well as its drawbacks and we will explore this possibility later. We anticipate, however, that degeneracies between neighbouring pixels in the arcs as well as neighbouring cells in the lens plane (not to mention the compact nature of the sources) will result in a system of linear equations that is not well behaved. The rank of the matrix Γ will normally be smaller than its smaller dimension. Calculating the inverse in this situation is not a trivial task.

The second approach is to *rotate* our system of linear equations using a transformation, which is just Γ^{T}. This transforms Γ into a square, symmetric and positive-definite matrix of dimension $(2N_s + N_c) \times (2N_s + N_c)$, $A = \Gamma^{\mathrm{T}}\Gamma$, which is better behaved than the original Γ matrix. However, the rank of A is in general smaller than its dimension and its inverse does not exist. The hope in this case must therefore be to find an approximate rather than exact solution to the system.

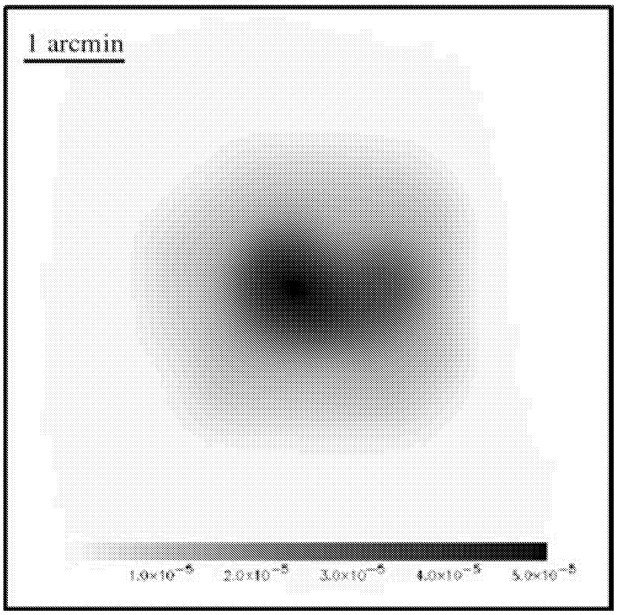

Figure 6. Smooth version of the recovered mass after minimizing the variance. The total recovered mass is $1.01 \times 10^{15}\, h^{-1}\, \mathrm{M}_\odot$. Compare this mass with the original one in Fig. 1.

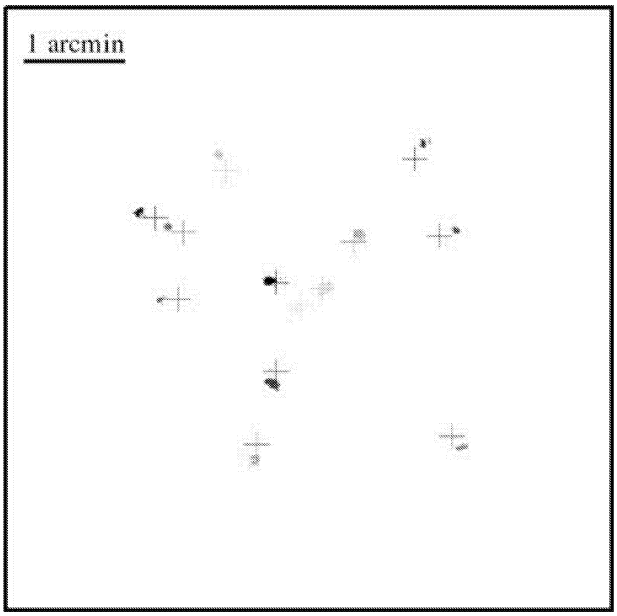

Figure 7. Recovered sources after minimizing the variance. The real positions of the sources are shown as crosses. Note that the recovered sources are farther away than the real ones. This compensates for the fact that the recovered mass is lower than the the true value.

The third approach is the simplest and will be explored first. We assume that we know nothing about the sources other than their redshift and that they are much smaller than the strong lensing arcs. This is the same as saying that the lens has to be such that it *focuses* the arcs into compact sources at the desired (known) redshift. This simple argument alone will turn out to be powerful enough to obtain a quick but good first estimate of the mass distribution in the lens plane.

We explore all these approaches below in reverse order.

5.1 A first approach: minimizing dispersion in the source plane

In this subsection we will discuss the simplest (although effective) method to obtain a fast estimation of the mass using no prior information on the lens or the sources. The problem then contains two sets of unknowns: the mass vector we want to determine, M, and the β positions of the sources. In general, for a finite number of observed arcs, there are several combinations of β and M that can reproduce the observations. The most obvious unphysical solution is the null solution, in which the mass is zero and the sources identical to the observed arcs. The easiest way to avoid such unphysical solutions is to minimize the variance of the β positions. This is equivalent to imposing that the vector M really acts as a true lens. We require big arcs with large magnifications and multiple images separated by arcseconds or even arcminutes to *focus* into a rather compact region in the source plane. This minimization process assumes that we are able to associate the multiple lensed images with a particular source. This can be achieved either with spectroscopy or with morphology and the multicolor imaging of the arcs.

To minimize the variance in the source plane it is illustrative to follow the steepest descent path, although other more effective minimization algorithms can be used (see below). Given an initial guess for the mass vector, one can calculate the derivative of the

variance in the source plane as a function of the mass and minimize in the direction of the derivative. Once a minimum is found, we calculate the derivative in the new mass position and minimize again in an iterative procedure.

The quantity to be minimized is

$$f(M) = \sum_s \sigma_s^2, \tag{7}$$

where the sum is over the number of identified sources and σ_s^2 is the variance of the source s in the source plane. That is,

$$\sigma_s^2 = \langle \beta^2 \rangle_s - \langle \beta \rangle_s^2, \tag{8}$$

where the βs are calculated from equation (4) and the average is over the βs corresponding to source s. By combining equations (4) and (8) it is easy to compute the derivative of σ_s^2 with respect to M:

$$\frac{\partial \sigma_s^2}{\partial M_j} = 2\langle \beta \rangle_s \langle \Upsilon_j \rangle_s - 2\langle \beta \Upsilon_j \rangle_s, \tag{9}$$

where Υ_j is the column j of the Υ matrix and the average is made only on the elements associated with the source s. We should note that all equations involving the vectors β, θ or $\alpha = \Upsilon M$ have two components, x and y, so there will in fact be two equations like equation (9): one for the x-component of β and the other for the y-component. Ultimately, the quantity we aim to minimize is $\sigma^2 = \sigma_x^2 + \sigma_y^2$. As already mentioned, the minimization can be done following the path of steepest descent given by equation (9). This path will end in a minimum at the new mass M^j. The process can be repeated by evaluating the new path at the new mass position until the variance is smaller than a certain ϵ. A good choice for ϵ is to take a few times the expected variance for a population of N_s galaxies at the measured N_s redshifts. Specific values for ϵ will be discussed later. In practical terms, the minimization is done through a series of iterations, gradually improving the dynamical grid as laid out in Section 4.2. For each iteration the ability of the mass distribution to focus the βs into compact sources is improved.

Equation (8) can be improved by weighting it with the amplification of the various images, assuming that the errors in the image plane are similar from image to image. This point will not be explored here. The minimization of the variance is a powerful and robust method for finding a first guess for the mass vector without making assumptions about the sources (Fig. 6). In fact, since the source positions can be estimated from equation (4), the minimization of the variance also provides us with an initial guess for these (Fig. 7). The drawback is the slow convergence of the algorithm. A typical minimization may take several hours on a 1-GHz processor. In the next subsection we will go a step further and include the β positions in the minimization, as well as speeding up the convergence by orders of magnitude.

5.2 Biconjugate gradient

The inversion of linear systems in which the matrix dimensions are of the order of 10^3 is a numerically trivial problem for today's computers, provided the matrix is well behaved. If the matrix has null or negative eigenvalues, a direct inversion is not feasible and one has to aim to solve for some approximated solution. Our system of linear equations is a good example of an ill-conditioned one. Direct inversion of the matrix is not possible because of negative eigenvalues. There is, however, another important reason why we do not want to solve exactly (or invert) the system of equations. An exact solution means that we will recover a mass distribution that puts the arcs into delta-function sources. As we will see later, this solution will be unphysical. Instead, we are interested in an approximated solution that does not solve exactly the system of equations and that has a residual. This residual will have the physical meaning of the extension of the sources or the difference between the point-like sources and the real, extended ones. The biconjugate gradient (Press et al. 1997) will be a useful way to *regularize* our problem.

The biconjugate gradient algorithm is one of the fastest and most powerful algorithms used to solve systems of linear equations. It is also extremely useful for finding approximate solutions for systems where no exact solutions exist or where the exact solution is not the one we are interested in. Our interest lies in the latter case. Given a system of linear equations

$$Ax = b, \tag{10}$$

a solution of this system can be found by minimizing the following function:

$$f(x) = c - bx + \frac{1}{2} x^{\mathrm{T}} A x, \tag{11}$$

where c is a constant. When the function $f(x)$ is minimized, its gradient is zero:

$$\nabla f(x) = Ax - b = 0. \tag{12}$$

That is, at the position of the minimum of the function $f(x)$ we find the solution of equation (10). In most cases, finding the minimum of equation (11) is much easier than finding the solution of the system in (10), especially when no exact solution exists for (10) or A does not have an inverse.

The biconjugate gradient finds the minimum of equation (11) (or equivalently, the solution of equation 10) by following an iterative process that minimizes the function $f(x)$ in a series of steps no longer than the dimension of the problem. The beauty of the algorithm is that the successive minimizations are carried out on a series of orthogonal conjugate directions, p_k, with respect to the metric A. That is,

$$p_i A p_j = 0, \quad j < i. \tag{13}$$

This condition is useful when minimizing in a multidimensional space since it guarantees that successive minimizations do not spoil the minimizations in previous steps.

Let us now turn to the system we want to solve, namely equation (5). The biconjugate gradient method assumes that the matrix Γ (matrix A in equation 10) is square. For our case this does not hold since we typically have $N_\theta \gg (N_c + N_s)$. Instead we build a new quantity, called the square of the residual, R^2:

$$R^2 = (\theta - \Gamma X)^{\mathrm{T}} (\theta - \Gamma X) \tag{14}$$

$$= 2\left(\frac{1}{2} \theta^{\mathrm{T}} \theta - \Gamma^{\mathrm{T}} \theta X + \frac{1}{2} X^{\mathrm{T}} \Gamma^{\mathrm{T}} \Gamma X \right). \tag{15}$$

By comparing equations (15) and (11) it is easy to identify the terms $c = \frac{1}{2}\theta^{\mathrm{T}}\theta$, $b = \Gamma^{\mathrm{T}}\theta$ and $A = \Gamma^{\mathrm{T}}\Gamma$. Minimizing the quantity R^2 is equivalent to solving equation (5). To see this we only have to realize that

$$b - AX = \Gamma^{\mathrm{T}}(\theta - \Gamma X) = \Gamma^{\mathrm{T}} R. \tag{16}$$

If an exact solution for equation (5) does not exist, the minimum of R^2 will give the better approximated solution to the system. The minimum can be now found easily with the biconjugate gradient (Press et al. 1997). For the case of symmetric matrices A, the algorithm constructs two sequences of vectors, r_k and p_k, and two constants, α_k and β_k:

$$\alpha_k = \frac{r_k^{\mathrm{T}} r_k}{p_k^{\mathrm{T}} A p_k}, \tag{17}$$

$$r_{k+1} = r_k - \alpha_k A p_k, \tag{18}$$

$$\beta_k = \frac{r_{k+1}^{\mathrm{T}} r_{k+1}}{r_k^{\mathrm{T}} r_k}, \tag{19}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k. \tag{20}$$

At every iteration, an improved estimate of the solution is found by

$$X_{k+1} = X_k + \alpha_k p_k. \tag{21}$$

The algorithm starts with an initial guess for the solution, X_1, and chooses the residual and search direction in the first iteration to be

$$r_1 = p_1 = b - AX_1. \tag{22}$$

Note that p_1 is nothing but ∇R^2. Thus the algorithm chooses as a first minimization direction the gradient of the function to be minimized at the position of the first guess. Then it minimizes in directions that are conjugate to the previous ones until it reaches the minimum or the square of the residual R^2 is smaller than a given value of ϵ.

The method has one potential pathological behaviour when applied to our problem. One cannot choose ϵ to be arbitrarily small. If one chooses a very small ϵ, the algorithm will try to find a solution that focuses the arcs in N_{s} sources that are delta functions. This is not surprising, as we are assuming that all the $2N_\theta$ unknown βs are reduced to just $2N_{\mathrm{s}}\beta$s, i.e. the *point-source solution*. The mass distribution that accomplishes this is usually very biased compared with the right one. It shows a lot of substructure and has large fluctuations in the lens plane. One therefore has to choose ϵ with wise criteria. Since the algorithm will stop when $R^2 < \epsilon$ we should choose ϵ to be an estimate of the expected dispersion of the sources at the specified redshifts. This is the only prior that has to be given to the method. We will see later, however, how the specific value of ϵ is not very critical as long as it is within a factor of a few of the right source dispersion (see Fig. 18 below). Instead of defining ϵ in terms of R^2 it is better to define it in terms of the residual of the conjugate gradient algorithm, r_k^2. This will speed up the minimization process significantly, since we do not need to calculate the real dispersion at each step but use the already estimated r_k. The two residuals are connected by the relation

$$r_k = \Gamma^{\mathrm{T}} R. \tag{23}$$

Imposing a prior on the sizes of the sources means that we expect the residual of the lens equation, R, to take typical values of the order of the expected dispersion of the sources at the measured redshifts. Hence we can define an R_{prior} of the form

$$R^i_{\mathrm{prior}} = \sigma_i \, RND, \tag{24}$$

where the index i runs from 1 to N_θ and σ_i is the dispersion (prior) assumed for the source associated with pixel i, and RND is a random number uniformly distributed over -1 to 1. Then, we can estimate ϵ as

$$\epsilon = r_k^{\mathrm{T}} r_k = R^{\mathrm{T}}_{\mathrm{prior}} \Gamma \Gamma^{\mathrm{T}} R_{\mathrm{prior}}. \tag{25}$$

When calculating ϵ it is better to consider only the first N_{c} columns of Γ and discard the last $2N_{\mathrm{s}}$. This is recommended to avoid the fact that the 1s in this part of the Γ matrix do not cancel out when multiplied by the random number and dominate the much smaller Γ_{ij} elements corresponding to the mass components. The last N_{c} columns of Γ should give zero contribution when multiplied by the random R^i_{prior} vector. If we choose as a prior that the sources are Gaussians with a $\sigma = 30\,h^{-1}$ kpc located at the measured redshift, this renders $\epsilon \approx 2 \times 10^{-10}$. The reader will note that the chosen σ is a few times larger than the one we would assign to a typical galaxy. We will discuss this point later. The final results are shown in Figs 8 and 9.

One has to be careful in not choosing the σ_i to be very small. In fact, they should be larger than the real dispersion of the source. Only when the number of grid points, N_{c}, is large enough can the gridded version of the true mass focus the arcs into sources that are similar in size to the real ones. If N_{c} is not large enough, the gridded version of the true mass focuses the arcs into sources that are

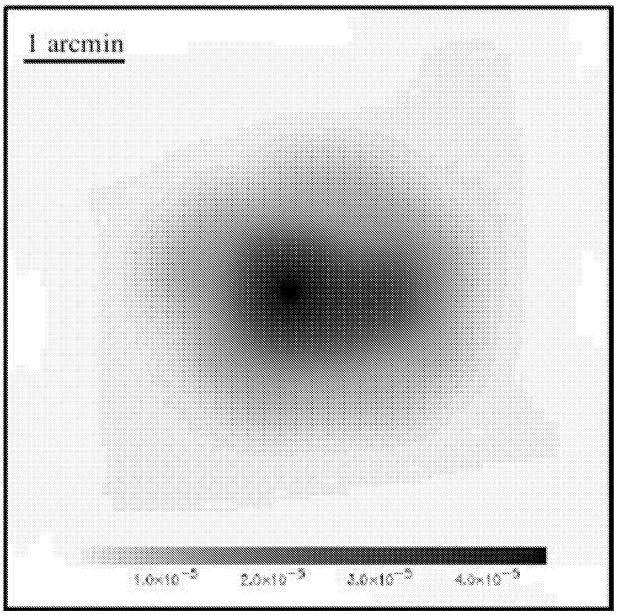

Figure 8. Recovered mass after minimizing R^2 using the biconjugate gradient algorithm. The mass has been smoothed with a Gaussian filter. The total recovered mass is $1.17 \times 10^{15}\,h^{-1}\,\mathrm{M}_\odot$.

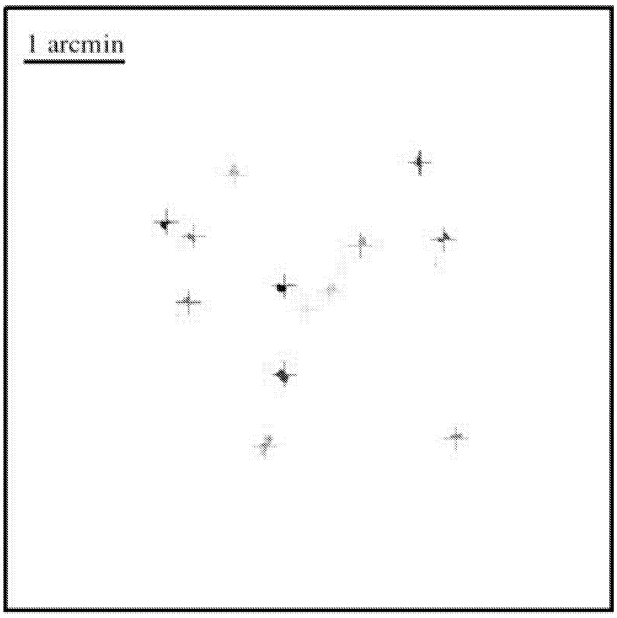

Figure 9. Recovered βs after minimizing R^2. Again, crosses represent the true positions of the sources. The recovered β_{o} falls in the middle of its corresponding cloud of β points.

larger than the real sources. We should take this into account when fixing σ_i.

It could be argued that a cleverer way of including this prior information in the algorithm would be by perturbing the β elements in equation (5) (or similarly, equation A5 in Appendix A). This is done by adding some noise to the 1s in the two **1** matrices in equation (A5) (see Appendix A). One could, for instance, add Gaussian noise with a dispersion similar to the expected dispersion of the source at redshift z. The reality, however, is that the quadratic nature of R^2 cancels out any symmetric perturbation added to the elements of Γ. Thus, the result is similar if we perturb Γ or not and we still have to include the prior and fix ϵ to be large enough so that we do not recover the *point-source solution* as shown in Figs 10 and 11. This also tells us that this method is not very promising if one wants to include parity information in the recovery of the mass and sources. In the next subsection we will give a different approach, which can include this parity information.

5.3 Singular value decomposition

The singular value decomposition (hereafter SVD) algorithm allows for decomposition of a generic $m \times n$ (with $m \geqslant n$) matrix A into the product of three matrices, two orthogonal and one diagonal (e.g. Press et al. 1997):

$$A = UWV^{\mathrm{T}}, \tag{26}$$

where U is an $m \times n$ orthogonal matrix, W is an $n \times n$ diagonal matrix whose elements in the diagonal are the singular values, and V^{T} is the transpose of an $n \times n$ orthogonal matrix. When A is symmetric, the SVD reduces to finding the *eigenvectors* and *eigenvalues* of A.

The advantage of this decomposition is that the inverse of A is given by

$$A^{-1} = VW^{-1}U^{\mathrm{T}}, \tag{27}$$

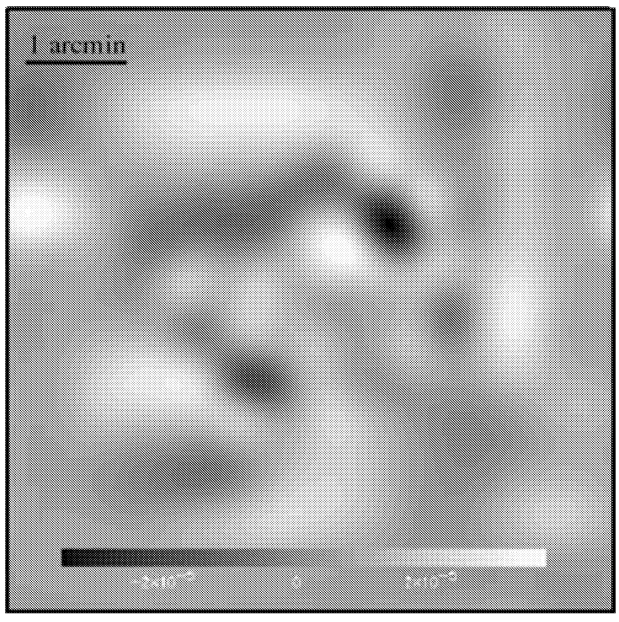

Figure 10. Recovered mass after minimizing R^2 using the biconjugate gradient algorithm and with a very small ϵ (point-source solution). The total recovered mass is $2.43 \times 10^{15}\, h^{-1}\, \mathrm{M}_{\odot}$ but there are also regions with negative masses.

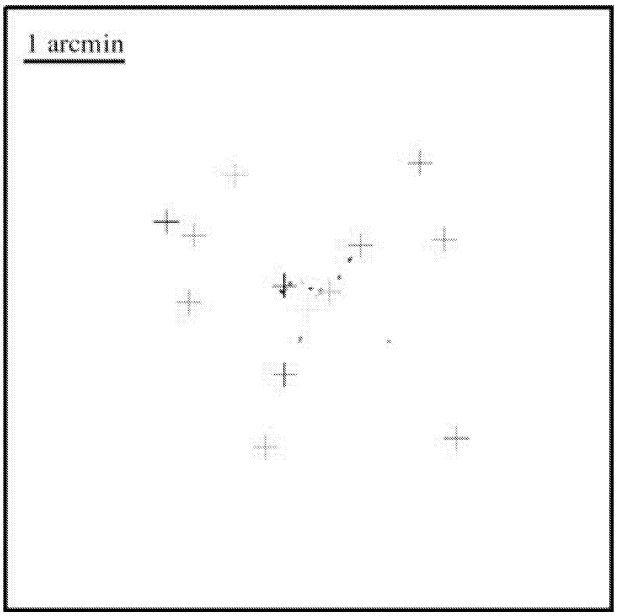

Figure 11. Recovered βs after minimizing R^2 for the point-source solution. The real source positions are shown as crosses.

where W^{-1} is another diagonal matrix whose elements are just the inverse of the elements of W; that is, $W^{-1}_{jj} = 1.0/W_{jj}$. The proof $A^{-1}A = I$ follows from the property that U and V^{T} are orthogonal ($U^{\mathrm{T}}U = VV^{\mathrm{T}} = I$).

In our case, we can use SVD to calculate the inverse of the Γ matrix and find the solution, X, directly by inverting equation (5):

$$X = \Gamma^{-1}\theta = VW^{-1}U^{\mathrm{T}}\theta. \tag{28}$$

Although the SVD allows us to invert the problem by calculating Γ^{-1}, its full power lies in its ability to solve a system approximately. The level of approximation can be controlled by setting a threshold in the matrix W^{-1}. In our problem, there will be many equations that are strongly correlated, in the sense that most of the θ positions in a single arc will come from the same source (that is, they will have almost the same β). Moreover, we have to keep in mind that we are using all the pixels in our data. This means that two equations corresponding to two neighbouring pixels will look almost exactly the same. When one computes the SVD of the matrix Γ, these two facts translate into a matrix W with elements in the diagonal that are 0 or close to 0. The inverse of W will be dominated by these small numbers and the solution will look very noisy. The good news about using SVD is that the most relevant information in the Γ matrix is packed into the first few values (the largest values in W), while the small single values in W will contain little information or the redundant information coming from neighbouring pixels. One can just approximate W by another matrix W', in which all the elements in its diagonal smaller than a certain threshold are set to 0. In the inverse of W', these elements are also set to 0. The magic of this trick is that the solution found with this approximation will contain the main trend or main components of the mass distribution.

A further advantage of using the SVD algorithm is that in this case no prior on the extension of the sources is needed. The degree of accuracy is controlled by setting the threshold in the singular values of the matrix W. Those elements in W below the threshold are set to 0, and similarly in its inverse. The threshold is usually set after looking at the singular values. The first ones will normally stay in some kind of plateau, and after then the singular values will

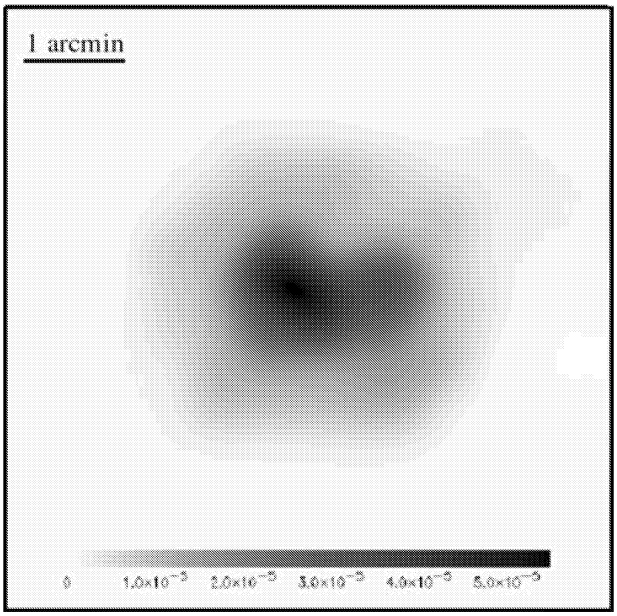

Figure 12. Recovered mass after SVD (no prior). The total recovered mass is $1.01 \times 10^{15}\, h^{-1}\, \mathrm{M}_{\odot}$.

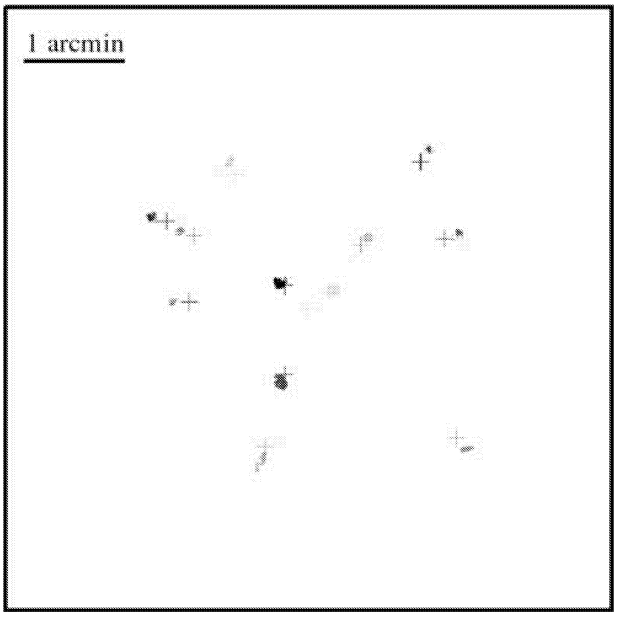

Figure 13. Recovered sources after SVD (no prior). The real positions of the sources are shown as crosses

decrease rapidly. The threshold should normally be set immediately after the plateau. In Figs 12 and 13 we show the result after decomposing Γ in its SVD decomposition and calculating its inverse with (27).

Like the two previous algorithms, the SVD has its own pathologies. The use of standard subroutines to find the SVD of Γ usually returns a *no convergence* error. This error comes from the nearly degenerate nature of Γ. One has then to increase the number of maximum iterations on these subroutines or use a coarse version of Γ in which only a small fraction of the θ positions (or, equivalently, Γ rows) are considered. Another solution is to invert the preconditioned system of equations where we previously multiplied by Γ^{T}. This allows to use all the θ pixels and find the SVD of $\Gamma^{\mathrm{T}}\Gamma$ in a small number of iterations. However, using the SVD of Γ instead of $\Gamma^{\mathrm{T}}\Gamma$ has a very interesting feature: it allows us to introduce parity information in an effective way.

In contrast to the quadratic cases of minimization of the variance or the square of the residual R^2, SVD allows us to include parity information in the Γ matrix that will not disappear when we look for the best solution. Since no $\Gamma^{\mathrm{T}}\theta$ or $\Gamma^{\mathrm{T}}\Gamma$ operations are involved, parity information will not cancel out. SVD could be an interesting way of fine tuning the solution by including the extra information coming from the parity of the arcs.

6 INCORPORATING THE NULL SPACE

So far we have made use of the information contained in the observed strong lensing arcs. This gives us a solution that explains the data in the sense that it predicts arcs in the right positions, but it is possible that the solution could overpredict arcs.

To avoid this we propose for the first time to include information from the null space, $\tilde{\theta}$; that is, from the part of the image that does not contain any arc. This space tell us where an arc should not appear, but it does not tell us where the hypothetical $\tilde{\beta}$ should be. The only fair thing we can do is to impose the condition that none of the pixels in the null space should fall into the estimated β of our solution. By achieving this we will have a solution that predicts the right arcs while not overpredicting arcs. The solution will then be fully consistent with the observed data.

The null space is connected to the solution X by

$$\tilde{\beta} = \tilde{\theta} - \tilde{\Upsilon} M. \tag{29}$$

It is evident that we want the new solution X (M and β_{o}) to be such that the new $\tilde{\beta}$ do not fall within a circle of radius $p(k)$ centred at each one of the β_{o}, where $p(k)$ is the prior with information on how extended the sources are. The null space will perturb the solution X in such a way that the new solution, $X' = X + \Delta\tilde{X}$, is an approximated solution of equation (5) and satisfies all the constraints of the form $\tilde{\beta} \ni \beta$.

The way we incorporate the constraints is by adopting an approach commonly used in penalized quadratic programming. We will minimize the new function

$$\phi(X) = R^2 + \lambda \sum_k g(f_k), \tag{30}$$

where λ is a constant that guarantees that, in the first iterations, the second term in equation (30) does not dominate the first, and g_k is a function that will penalize those models predicting $\tilde{\beta}$ falling near the β positions that minimize R^2. As a penalizing function we will choose an asymptotically divergent Gaussian,

$$g(f_k) = \frac{1}{\exp(f_k) - \mu}, \tag{31}$$

with

$$f_k = \frac{\left[\tilde{\beta}(k)_x - \beta_x^{\mathrm{o}}\right]^2 + \left[\tilde{\beta}(k)_y - \beta_y^{\mathrm{o}}\right]^2}{\sigma_k^2}, \tag{32}$$

where $\tilde{\beta}(k)_x$ is the x-component of $\tilde{\beta}$ for the pixel k in the null space. β_x^{o} is our estimated value of β (x-component), and similarly for the y-component. There are as many constraints of the form f_k as there are pixels, $\tilde{\theta}$, in the null space.

The parameter μ controls the degree of divergence of the Gaussian function. When $\mu = 0$ we recover the classical Gaussian, but as μ approaches 1 the Gaussian becomes progressively sharper without

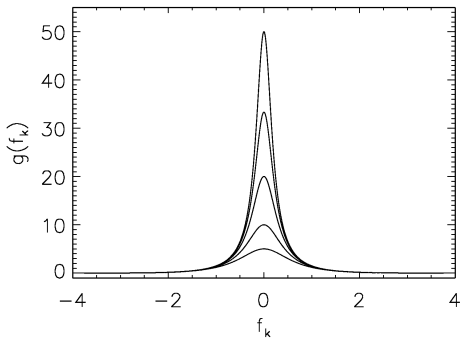

Figure 14. Penalizing function showing values of $\mu = 0.8, 0.9, 0.95, 0.97,$ and 0.98 and $\sigma = 1$. Note that the width of the curve does not change with increasing μ.

increasing its dispersion. For $\mu = 1$ the Gaussian is infinite at $f_k = 0$ (see Fig. 14). By minimizing the function $\phi(X)$ with increasing values of μ we will find that in the limit $\mu \to 1$ the solution will push away those $\tilde{\beta}$ that originally fell in the region defined by the set of β.

Ideally we want to include all the *dark* or empty pixels in the $\tilde{\theta}$ space, but this is, in general, found to be a waste of memory and computing time. The fact is that only the $\tilde{\theta}$ pixels that are hitting (or close to hitting) one of the N_s estimated β positions of the sources will have an impact on the solution. Most of the $\tilde{\theta}$ pixels in the null space already satisfy all the constraints for the actual solution X. For this reason we will include only those $\tilde{\theta}$ for which the solution X predicts that their corresponding $\tilde{\beta}$ are close to hitting (or actually hitting) a source. The $\tilde{\theta}$ space will include the observed θs as a subspace. We have to exclude this subspace from $\tilde{\beta}$ before minimizing equation (30). This is, again, just an optimization process. In practice, one can include all the pixels in the image (excluding those containing part of an arc) in the null space.

After the minimization process is finished, the new solution will have the arcs falling in compact regions around β_o, while the extra arcs produced by the previous solution will fall in regions outside areas around the β_o.

From our simulation we have seen that the addition of the null space induces small changes in the mass plane and tends to stabilize the solution in the sense that it makes the recovered mass profile independent of the threshold ϵ. This is in fact an interesting bonus which comes with the addition of the null space. The new function to be minimized, ϕ, can be minimized until the true minimum is found. In this case, there is no equivalent of a point-source solution. Since some of the $\tilde{\theta}$ are in fact very close to the observed θ, the solution that minimizes ϕ will *focus* those $\tilde{\theta}$ and θ in neighbouring regions in the source plane. When minimizing ϕ there will be two competing effects. One will tend to increase the mass minimize R^2 (point-source solution). The other will tend to reduce the mass so that the $\tilde{\beta}$ will be pushed away from the β positions. The outcome will be a balanced situation between the β trying to collapse into compact sources and the $\tilde{\beta}$ trying to escape the wells in the β positions.

To quantify the effect accurately, more simulations are needed. This will be done in a future paper, but as an illustration we show in Fig. 15 the recovered mass after imposing the constraints in the $\tilde{\theta}$ space. The total mass is now only 1.2 per cent lower than the true mass. The new mass also contains more structure and starts showing the internal distribution of the main components of the cluster.

In Fig. 16 we show the predicted position for the arcs after combining the best mass together with the best estimate for the positions of the sources. To compute the arcs we have assumed that each source is a circle with a radius of $15\,h^{-1}$ kpc centred on the estimated best source positions and at the measured redshifts. By comparing Figs 16 and 3 (left) we see that the predicted arcs match very well with the observed (simulated) data, with the exception of some of the arcs near the centre of the image.

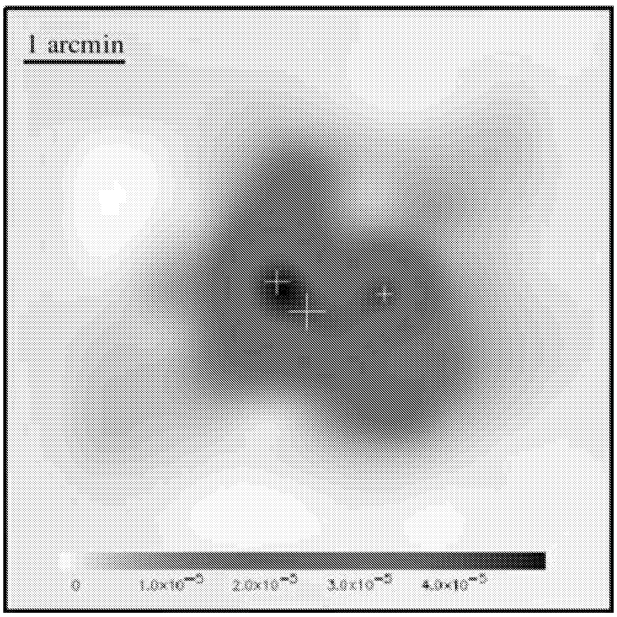

Figure 15. Recovered mass (smoothed) after minimizing ϕ. The crosses show the positions of the three main haloes in the cluster. The size of the cross is proportional to the mass of the halo. The total recovered mass is $1.1055 \times 10^{15}\,h^{-1}\,M_\odot$.

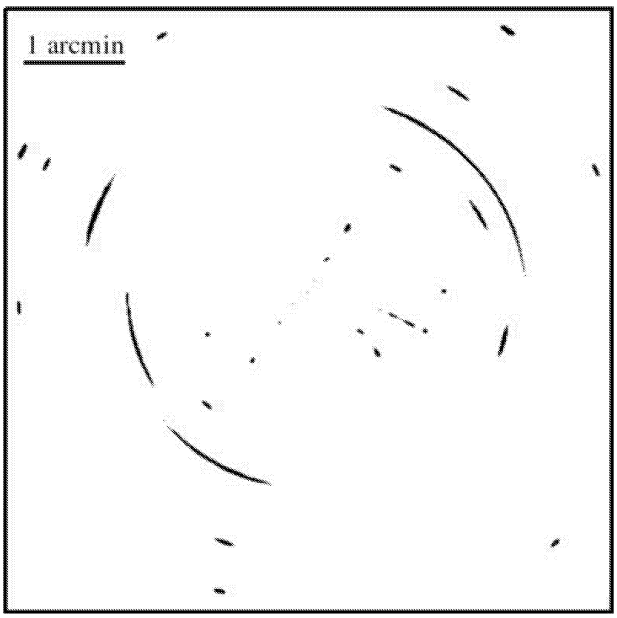

Figure 16. The plot shows the predicted arcs according to the best mass and source solution. We have assumed that the sources are circles with radius $15\,h^{-1}$ kpc centred on the source positions and at the measured redshifts. This result should be compared with the true arcs seen in Fig. 3. The match is almost perfect.

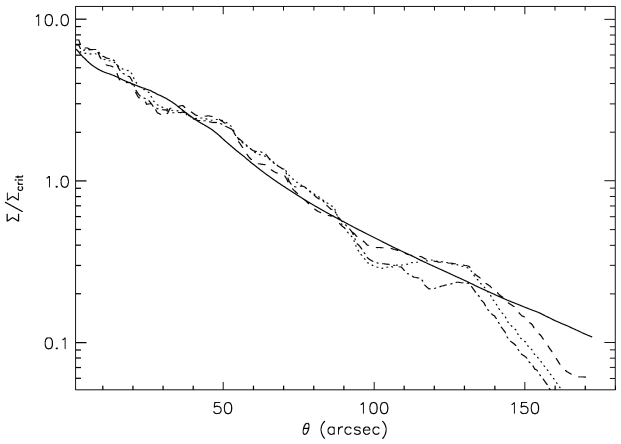

Figure 17. Original profile (solid line) compared with the recovered ones. The dashed line is the biconjugate gradient case, the dot-dashed line the SVD case, and the dotted line represents the case when the null space is included. The minimum of the variance is similar but with less mass in the tails. All profiles have been normalized by the critical surface density for a source at redshift $z_s = 3$.

7 DISCUSSION

In this paper we have presented several approaches to non-parametric lens modelling. Using no information about luminosity distributions whatsoever, these methods perform remarkably well on simulated strong lensing data. One of the main conclusions of this work is simply that it works: it is possible to recover information about the mass distribution without using prior information on the same if one has a sufficiently large number of arcs. This can be seen in Fig. 17, in which we show the recovered mass profiles compared with the original one. The recovered mass traces the real mass distribution up to the furthest data point in the simulated image. Beyond this point, the recovered profile is insensitive to the data. When the minimization stops, the cells in the outer regions stay with a mass close to the one they had in the first step of the minimization. This point is discussed in more detail in Diego et al. (2004).

We are optimistic about the performance of these methods when used on future data, but we would like to emphasize the potential pathologies. We will now discuss the major issues. As we saw in Section 2, the inversion algorithms rely on a linearization of the lens equation. We achieve this by decomposing the lens plane into small mass cells and assuming that the gridded mass is a fair representation of the true underlying mass. This is true when the number of cells in the grid is large enough, but for a uniform grid this may mean using several thousand mass cells, making the problem ill-conditioned and underdetermined. By inverting the lens equation in a series of iterations we introduce an adaptive grid that optimizes the number of cells by sampling dense regions more heavily and using larger cells for underdense regions. This allows for good sampling of the lens without a huge number of cells. However, the gridded mass plane is still an approximation to the true mass plane so we expect the solution to be an approximation to the true solution as well. Since a solution comprises not only the masses but also the positions and extents of the sources, this means that we should expect these also to be approximations to the true ones. This has already been identified as one of the potential pathologies of the algorithm: namely, if we try to focus the sources into very compact regions, with sizes comparable to the typical galaxy sizes at the relevant redshifts, the obtained mass is in general different from the true mass. In fact, the best results are obtained when the mass plane focuses the arcs into regions that are a few times larger than the extent of the true sources. As we pointed out before, this reconstructed mass corresponds to a *short-sighted* version of the lens. This problem can be overcome by requiring that the minimization algorithms stop once the recovered sources are a few times larger than the true sources. The extent of the true sources can be guessed from the redshift.

The introduction of such a prior causes us to ask how sensitive the solution is to the specific guess of prior. The answer is that it depends on exactly 'how bad' that choice is. As long as we assume a source size significantly larger than the true sizes, the actual solution does not change much and resembles the true mass distribution well. However, when trying to approach the true source size, the solutions change rapidly away from the realistic model. The situation is shown in Fig. 18. As shown in Section 5.2, a given physical size corresponds to a given threshold, ϵ, and as long as this threshold is sufficiently large, $\geqslant 10^{-10}$ (corresponding to a physical size of $\geqslant 30\,h^{-1}$ kpc) the total mass is well behaved. If we instead demand that the physical size of the reconstructed sources be more realistic, say $\sim 12\,h^{-1}$ kpc, we get a threshold of $\sim 10^{-11}$, for which the mass distribution is already starting to diverge away from the true mass. We therefore want the recovered sources to be larger than the true ones. In other words we want to recover a *short-sighted* cluster! In previous sections we have demonstrated non-parametric lens modelling using several different algorithms with promising results. However we have not yet addressed the question of uniqueness. Is there a unique solution, and, if not, how different are the possible solutions?

The good news is that minimizing quadratic functions such as the variance or the square of the residual guarantees that there is only one absolute minimum. This absolute minimum corresponds to the *point-source solution*. The bad news is that this is not the solution we are looking for. As shown above, trying to focus the sources too much introduces artefacts in the mass distribution, so we need to stop the minimization at some step before the absolute minimum. In two dimensions it is easy to visualize that the quadratic function will have the shape of a valley, and stopping the minimization at some point before the actual minimum means choosing an ellipse on which all solutions are equally good. In many dimensions this is harder to visualize, but we expect our obtainable solutions to lie on an N-dimensional ellipsoid around the minimum.

To get a quantitative grasp of how much these solutions differ in mass and source positions, we solve the equations for a range of random initial conditions. This is a manageable task since our minimization algorithm is extremely fast, taking only about ~ 1 s on a 1-GHz processor. Moreover, for speed purposes, we fix the grid

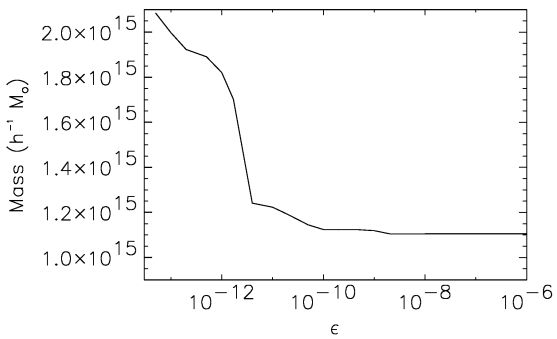

Figure 18. Total mass as a function of the threshold ϵ used in the biconjugate gradient minimization. The true mass is $1.119 \times 10^{15}\,h^{-1}\,M_\odot$.

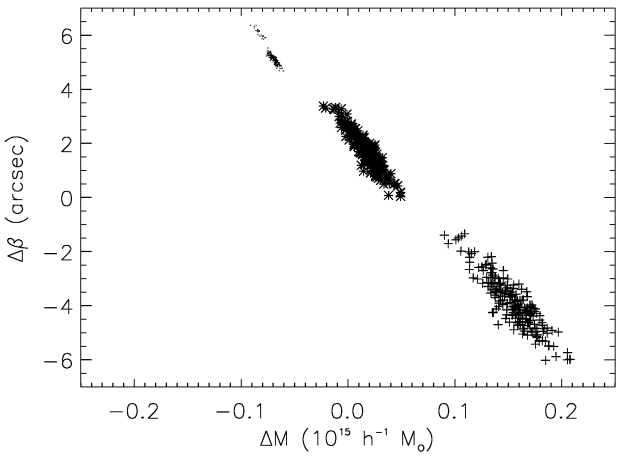

Figure 19. Dispersion in the solutions for three sets of initial conditions. Upper left (dots), starting with a random realization of small masses. Centre (asterisks), starting with a random realization but with a total mass of $M \approx 1.06$. Bottom right (crosses), starting with high random masses. In all cases, the threshold was set to $\epsilon = 2 \times 10^{-10}$ and the starting β positions were chosen as random in a box of 100×100 centred on the centre of the image.

to the one corresponding to the solution shown in Fig. 8. Fixing the grid speeds up the process significantly, since the $\Gamma^{\mathrm{T}}\Gamma$ matrix and the $\Gamma^{\mathrm{T}}\theta$ vector do not need to be recalculated in each minimization. In Diego et al. (2004) a similar process was followed carrying multiple minimizations, but this time the grid was changed dynamically based on the solution found in the previous step. The authors found in that paper that the dispersion of the solutions increases due to the extra variability of the grid.

The results are shown in Fig. 19. The minimization process described in Section 5.2 will stop at a different point in the N-dimensional ellipsoid for each set of initial conditions. If the total mass of the initial mass distribution is very low, the minimization will stop at solutions with mass below the true mass and β positions further away from the centre of the cluster than the real ones (dots). If instead we start with a total mass much larger than the true value, the minimization process returns higher masses and β positions closer to the centre of the potential (crosses). The situation improves when we impose the condition that the total mass of the initial distribution have a reasonable value. In this case, the minimization stops in a region close to both the right mass and the right βs. This argument motivates an iterative minimization in which successive estimations for the mass are obtained at each step and used in the next.

Among the three algorithms presented in this work, the minimization of the variance is the least powerful because of its low convergence. It is, however, interesting from a pedagogical point of view since it shows the degeneracies existing between the total mass of the cluster and the positions of the sources. The biconjugate gradient is orders of magnitude faster and is capable of finding the point-source solution in a few seconds. The point-source solution is, however, unphysical and a prior associated with the size of the sources is needed in order to stop the minimization process at the proper place. The good news is that the algorithm shows a weak sensitivity to this prior, provided it is chosen with a minimum of wisdom. Since the point at which the minimization stops depends on where the minimization starts, this also allows us to study the range of possible models consistent with the data by minimizing many times while changing the initial conditions. This feature makes the biconjugate gradient very attractive for studying the space of solutions. This space can be reduced by including the extra information contained in the null space, that is, the areas in the sky with no observed arcs. We have seen how this information can be naturally included in the minimization process by introducing a penalty function.

Finally, the SVD has the interesting feature that it allows extra information to be included regarding the parity or resolved features in the arcs. This possibility has not been studied in detail in this paper, but it is definitely worthwhile to explore it, since it would allow smaller details in the mass distribution to be recovered. The main drawback of the SVD is that the decomposition fails to converge when all the information is used and one has to use a *coarse* version of the data instead. The second drawback is that some of the singular values in the matrix W are very small. These values will dominate the inverse if they are not *masked* by a threshold in the matrix W. Choosing the value for this threshold is not very critical as long as its amplitude is large enough to *mask* the small singular values in W.

The accuracy of the methods presented in this paper allow for high-precision non-parametric mass reconstructions and direct mapping of the dark matter distribution of clusters. Previous works have suggested some discrepancy between mass estimates derived from different data (Wu & Fang 1997). Accuracy, combined with the speed of the algorithm, opens the door to cosmological studies. Strong lensing analyses were predicted to yield interesting cosmological constraints, but as a result of the uncertainties in our understanding of galaxy clusters they have yet to live up to these predictions. A fast, per cent level determination of cluster masses from lensing observations could allow for sufficient statistical sampling to provide information about cosmological parameters. An interesting follow up to this work would be to establish what number of strong lensing systems and what quality of data are necessary in order to make interesting constraints using our methods.

The methods presented here should not, however, be applied indiscriminately. For instance, if the reconstructed mass is systematically biased towards recovering less cuspy central regions, this reduces the possibility of drawing conclusions about mechanisms for dark matter annihilation (Spergel & Steinhardt 2000; Wyithe, Turner & Spergel 2001; Boehm et al. 2004). If one is interested in using our methods to discuss this topic, that bias must be quantified. Owing to lens resolution issues we anticipate that our reconstruction algorithms indeed have a bias in this direction and it is likely that other algorithms may suffer from similar problems. These and other potential systematic errors will be investigated in future works.

7.1 Future work

This paper is not intended to be an exhaustive exploration of the accuracy of non-parametric mass reconstruction methods but rather to help to reignite debate and competition in the field, and to demonstrate the feasibility and power of such methods in the face of new data. Specific issues such as magnitudes of the mass profile bias, source morphologies, surface brightness of the arcs, parity, projection effects and other relevant issues discussed above should be explored in order to improve the results. Special emphasis should be paid to the effects that the gridding of the lens plane has on the results. An attempt to study this issue has been made in a second paper (Diego et al. 2004), in which the authors have shown that the grid may play an important role. In this paper we have presented the results using a very specific simulation. Although the algorithm

has been tested with different simulations with positive results, it would be desirable to make a more detailed study of how the result is affected by issues such as the structure of the lens, its symmetry, and the number of sources and accuracy of their redshifts. Such a study would be even more interesting if a comparative study between parametric and non-parametric methods were done using the same simulations. It would also be interesting to explore the effects of adding constraints in the mass of the type $M_i > 0$, which may help to stabilize the solution even in the absence of a prior on the source sizes. This can be accomplished by adopting the techniques used in quadratic programming. All theses issues, although very interesting, are beyond the scope of this paper and will be studied in subsequent papers.

Although not discussed in this paper, a very interesting piece of work would be to investigate the potentiality of an accurate strong lensing analysis as a cosmological tool (e.g. Link & Pierce 1998; Yamamoto et al. 2001; Golse, Kneib & Soucail 2002; Soucail, Kneib & Golse 2004). Previous works (Yamamoto et al. 2001; Chiba & Takahashi 2002; Sereno 2002; Dalal, Holder & Hennawi 2004) suggest that the lensing observables are primarily dependent on the lens model, while the dependence in the cosmological parameters is minor. Accurately constraining the lens model could open the door to do cosmology with strong lensing images. It is easy to see that a single image of a cluster with dozens of arcs coming from sources at different redshifts will constrain the lens model accurately, which, given the large number of distances involved in the analysis, will tell us something about the cosmology. All these issues are of great interest and will be studied in subsequent papers.

ACKNOWLEDGMENTS

This work was supported by NSF CAREER grant AST-0134999, NASA grant NAG5-11099, the David and Lucile Packard Foundation and the Cottrell Foundation. The work was also partially supported by B. Jain through lousy poker playing. We thank D. Rusin and G. Bernstein for helpful discussions. We would like also to thank J. P. Kneib for helpful suggestions.

REFERENCES

Abdelsalam H. M., Saha P., Williams L. L. R., 1998a, MNRAS, 294, 734
Abdelsalam H. M., Saha P., Williams L. L. R., 1998b, AJ, 116, 1541
Bartelmann M., Narayan R., Seitz S., Schneider P., 1996, ApJ, 464, L115
Beckwith S. V. W. et al., 2003, AAS, 202, 1705
Blandford R. D., Narayan R., 1992, ARA&A, 30, 311
Boehm C., Hooper D., Silk J., Casse M., Paul J., 2004, Phys. Rev. Lett., 92, 1301
Brainerd T. G., Blandford R. D., Smail I., 1996, ApJ, 466, 623
Bridle S. L., Hobson M. P., Lasenby A. N., Saunders R., 1998, MNRAS, 299, 895
Broadhurst T. J., Taylor A. N., Peacock J. A., 1995, ApJ, 438, 49
Broadhurst T. J., Huang X., Frye B., Ellis R., 2000, ApJ, 534, L15
Broadhurst T. J. et al., 2005, ApJ, 621, 53
Chiba T., Takahashi R., 2002, Prog. Theor. Phys., 107, 625
Colley W. N., Tyson J. A., Turner E. L., 1996, ApJ, 461, L83
Dahle H., Hannestad S., Sommer-Larsen J., 2003, ApJ, 588, 73
Dalal N., Holder G., Hennawi J. F., 2004, ApJ, 609, 50
Diego J. M., Sandvik H. B., Protopapas P., Tegmark M., Benitez N., Broadhurst T., 2004, MNRAS, submitted, preprint astro-ph/0412191
Golse G., Kneib J.-P., Soucail G., 2002, A&A, 387, 788
Guzik J., Seljak U., 2002, MNRAS, 335, 311
Hudson M. J., Gwyn S. D. J., Dahle H., Kaiser N., 1998, ApJ, 503, 531
Kaiser N., 1995, ApJ, 439, L1
Kaiser N., Squires, 1993, ApJ, 404, 441
Kneib J.-P., 2002, in Natarajan P., ed., Proc. Yale Cosmology Workshop, The Shapes of Galaxies and their Dark Matter Halos. World Scientific, Singapore, p. 50
Kneib J.-P., Mellier Y., Fort B., Mathez G., 1993, A&A, 273, 367
Kneib J.-P., Mellier Y., Pello R., Miralda-Escudé J., Le Borgne J.-F., Boehringer H., Picat J.-P., 1995, A&A, 303, 27
Kneib J.-P., Ellis R. S., Smail I. R., Couch W., Sharples R., 1996, ApJ, 471, 643
Kneib J.-P. et al., 2003, ApJ, 598, 804
Kochanek C. S., Blandford R. D., 1991, ApJ, 375, 492
Link R., Pierce M. J., 1998, ApJ, 502, 63
Marshall P. J., Hobson M. P., Gull S. F., Bridle S. L., 2002, MNRAS, 335, 1037
Meneghetti M., Jain B., Bartelmann M., Dolag K., 2004, astro-ph/0409030
Narayan R., Bartelmann M., 1999, in Dekel A., Ostriker J. P., eds, Proc. 1995 Jerusalem Winter School, Formation of Structure in the Universe. Cambridge Univ. Press, Cambridge, p. 360
Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1997, Numerical Recipes in Fortran 77. Cambridge Univ. Press, Cambridge
Saha P., 2000, AJ, 120, 1654
Saha P., Williams L. L. R., 1997, MNRAS, 292, 148
Sand D. J., Treu T., Ellis R. S., 2002, ApJ, 574, 129
Schneider P., 1995, A&A, 302, 639S
Schneider P., Seitz C., 1995, A&A, 294, 411
Schneider P., Ehlers J., Falco E. E., 1993, Book Review: Gravitational Lenses. Springer-Verlag, Berlin
Seitz C., Schneider P., 1995, A&A, 297, 287
Sereno M., 2002, A&A, 393, 757
Soucail G., Kneib J.-P., Golse G., 2004, A&A, 417, L33
Spergel D. N., Steinhardt P. J., 2000, Phys. Rev. Lett., 84, 3760
Taylor A. N., Dye S., Broadhurst T. J., Benitez N., van Kampen E., 1998, ApJ, 501, 539
Tegmark M., 2002, Phys. Rev. D, 66, 3507
Trotter C. S., Winn J. N., Hewitt J. N., 2000, ApJ, 535, 671
Tyson J. A., Kochanski G. P., Dell'Antonio I., 1998, ApJ, 498, L107
Wambsganss J., 1998, Living Rev. in Relativity, 1, 12
Wyithe J. S. B., Turner E. L., Spergel D. N., 2001, ApJ, 555, 504
Wu X.-P., Fang L.-Z., 1997, ApJ, 483, 62
Yamamoto K., Futamase T., 2001, Prog. Theor. Phys., 105, 5
Yamamoto K., Kadoya Y., Murata T., Futamase T., 2001, Prog. Theor. Phys., 106, 917

APPENDIX A: THE Υ AND Γ MATRICES

The Υ matrix contains the information on how each mass element j affects the ith deflection angle. That is,

$$\alpha_i = \Upsilon_{ij} M_j. \tag{A1}$$

Precisely how the matrix is constructed is a matter of convenience. Our specific choice is to put both x and y components of all the arc pixels into vectors with $2N_\theta$ elements. The resulting Υ is then a $2N_\theta \times N_c$ matrix.

In fact, the structure of the Υ matrix and the vectors is irrelevant as long as they combine to correctly represent the lens equation as

$$\beta_i = \theta_i - \Upsilon_{ij} M_j. \tag{A2}$$

Each element in the Υ matrix is computed as follows:

$$\Upsilon_x(i, j) = \lambda[1 - \exp(-\delta/2\sigma^2)]\frac{\delta_x}{\delta^2}, \tag{A3}$$

where

$$\lambda = 10^{15}\,\mathrm{M}_\odot \frac{4G}{c^2}\frac{D_{ls}}{D_l D_s}. \tag{A4}$$

The index i runs over the N_θ observed θ pixels, and the index j runs over the N_c elements in the mass vector, M. The factor δ_x in equation

(35) is just the difference (in radians) between the x position in the arc (or x of pixel θ_i) and the x position of the cell j in the mass grid $[\delta_x = \theta_x(i) - \theta'_x(j)]$. Similarly, we can define $\delta_y = \theta_y(i) - \theta'_y(j)$ and $\delta = \sqrt{\delta_x^2 + \delta_y^2}$. Moreover, for Υ_y we only have to change δ_x to δ_y. Since we include the factor $10^{15}\,\mathrm{M}_\odot$ in λ (see equation 36), the mass vector M in equation (4) will be given in units of $10^{15}\,h^{-1}\,\mathrm{M}_\odot$. The h^{-1} dependence comes from the fact that in λ we have the ratio $D_{\rm ls}/(D_{\rm l}D_{\rm s})$, which goes as h. We calculate Υ_x and Υ_y separately, because the final Υ matrix entering in equation (4) contains both Υ_x and Υ_y (the same holds for the vectors β and θ). One can rearrange the x and y components in any order.

The structure of the Γ matrix is identical to the matrix Υ but with the difference that it has $2N_{\rm s}$ additional columns (the location of the extra columns is irrelevant as long as it is consistent with the location of the $2N_{\rm s}$ $\beta_{\rm o}$ unknowns in the X vector). Is easy to see that each one of these extra columns (with dimension $2N_\theta$) corresponds to one of the $N_{\rm s}$ sources. Since the Γ matrix has to contain both the x and y components, the first/second half of each one of the extra columns will be all 0s, depending on whether it corresponds to the y/x component of $\beta_{\rm o}$. The other half will be full of 0s and 1s, the 1s being in the positions associated with that particular source, the 0s elsewhere. In other words, the lens equation can be written explicitly as ($\mathbf{a}$ denotes a matrix and $\boldsymbol{a}$ denotes a vector)

$$\begin{pmatrix} \boldsymbol{\theta_x} \\ \boldsymbol{\theta_y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Upsilon_x} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{\Upsilon_y} & \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \boldsymbol{M} \\ \boldsymbol{\beta_o^x} \\ \boldsymbol{\beta_o^y} \end{pmatrix}, \tag{A5}$$

where again $\boldsymbol{\theta_x}$ and $\boldsymbol{\theta_y}$ are two N_θ-dimensional vectors containing the x and y positions, respectively, of the pixels in the observed arcs. The two ($N_\theta \times N_{\rm c}$) matrices $\boldsymbol{\Upsilon}_x$ and $\boldsymbol{\Upsilon}_y$ contain the x and y lensing effect of the cell j on the θ pixel i. The $N_\theta \times N_{\rm s}$ dimensional matrices $\mathbf{1}$ and $\mathbf{0}$ are full of 0s (the $\mathbf{0}$ matrix) or contain 1s (the $\mathbf{1}$ matrix) in the i positions ($i \in [1, N_\theta]$), where the ith θ pixel comes from the jth source ($j \in [1, N_{\rm s}]$), and 0 elsewhere. The vector $\boldsymbol{M}$ contains the $N_{\rm c}$ gridded masses we want to estimate. $\boldsymbol{\beta_o^x}$ contains the central x positions of the $N_{\rm s}$ sources. Similarly, $\boldsymbol{\beta_o^y}$ will contain the central y positions.

This paper has been typeset from a TEX/LATEX file prepared by the author.