Faculty of Physics and Astronomy
University of Heidelberg

Combining Weak and Strong Gravitational Lensing for Galaxy-Cluster Mass Reconstructions

Diploma thesis
in Physics

submitted by
Julian Merten
born in Hammelburg, Germany
2008

This diploma thesis has been carried out by

Julian Merten

at the

Institute for Theoretical Astrophysics Heidelberg

under the supervision of

Prof. Matthias Bartelmann

Über die Kombination von schwachem und starkem Gravitationslinseneffekt zur Rekonstruktion der Massenverteilung in Galaxienhaufen

Zusammenfassung

In der vorliegenden Arbeit präsentieren wir eine neue Methode zur Rekonstruktion der Massenverteilung von Galaxienhaufen mit Hilfe einer Kombination des schwachen und des starken Gravitationslinseneffektes. Für die Entwicklung und zur Bestätigung kosmologischer Modelle ist ein profundes Wissen über die Häufigkeitsverteilung, Gesamtmasse und Massenverteilung von Galaxienhaufen nötig.

Der starke Gravitationslinseneffekt ist entscheidend für eine gute Rekonstruktion des Galaxienhaufenkerns, während der schwache Gravitationslinseneffekt eine Rekonstruktion des Galaxienhaufens innerhalb des gesamten Beobachtungsfeldes erlaubt.

Nachdem wir die Grundgedanken unserer Rekonstruktionsmethode vorgestellt haben, beschreiben wir die numerische Implementierung. Abschließend widmen wir uns konkreten Anwendungen und stellen dabei einen neu entwickelten Gravitationslinsensimulator vor, welcher realistische Simulationen des Gravitationslinseneffektes erzeugt und daher eine umfassende Kalibrierung unserer Rekonstruktionsmethode erlaubt. Alle durchgeführten Tests bestätigen die Genauigkeit und Zuverlässigkeit unserer Methode, was uns abschließend die Rekonstruktion der Massenverteilungs des bekannten Galaxienhaufens MS 2137 erlaubt.

Combining Weak and Strong Gravitational Lensing for Galaxy-Cluster Mass Reconstructions

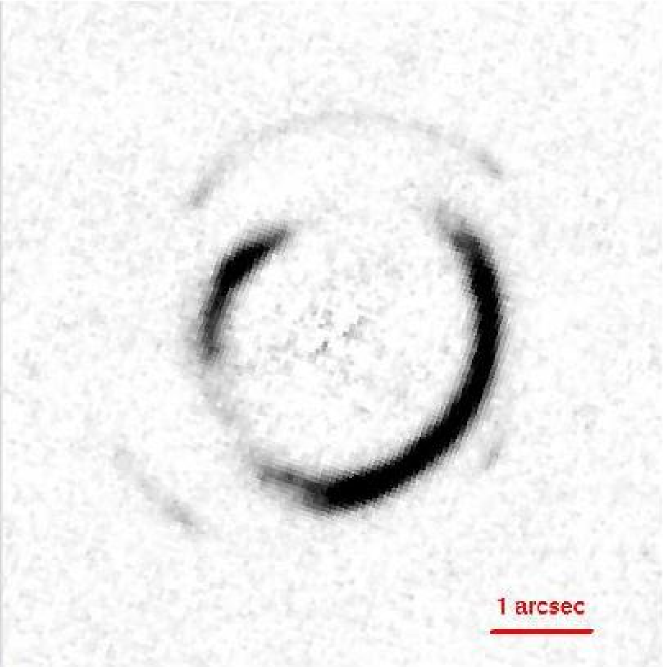
Abstract

In this thesis; we present a novel method to combine weak and strong gravitational lensing for reconstructing the mass distribution in galaxy clusters, which are important objects for cosmology. Their abundance, total mass and mass distribution has to be known very accurately to develop and to support cosmological models.

While strong-lensing effects are important for a good reconstruction of a galaxy cluster near its core, weak-gravitational lensing allows a reconstruction in the complete observed field.

After describing the main principles of our method, we show how it is implemented numerically. The last part of the thesis is dedicated to applications and introduces a lately developed lensing simulator which creates realistic lensing simulations and allows therefore comprehensive calibrations of any reconstruction method, based on gravitational lensing. All tests confirm the accuracy and reliability of our method and allow for the mass-distribution reconstruction of the well-known galaxy cluster MS 2137 in the end of this work.

> This is the kind of conversation that ends in a gunshot!
>
> Dr. Stephen Franklin
> Head of Medical Staff, Babylon 5



The discovery of a double Einstein ring in the SDSS data using HST/ACS. (from Gavazzi et al., 2008)

Contents

List of Figures

1 Introduction

Cosmology describes the evolution of the universe as a whole and has become one of the most interesting and exciting fields in astrophysics during the last decades. Based on Einstein's General Theory of Relativity, cosmology now also includes parts of the particle physics branch like the quantum theory of fields and its applications. Starting point for cosmological models are the assumptions of homogeneity and isotropy of the universe, which might sound strange in the first place. Inserting those assumptions into Einstein's field equations leads to the Friedmann equations which describe the scale evolution of the universe.

Today most cosmologists have great confidence in the so called cosmological standard model which predicts a Big-Bang as the starting point of the universe, which is defined as that moment in time when the scale factor of the universe was identical or close to zero. Shortly after the Big-Bang the universe entered a phase of inflationary expansion, driven by a scalar quantum field called inflaton. During that phase the universe expanded exponentially and the first structures in the universe were seeded by quantum fluctuations of the inflaton field. After inflation stopped, the universe was dominated by radiation, followed by a matter dominated phase, and apparently entered a phase of exponential expansion again. Today the energy content of the universe is dominated by the cosmological constant Λ or something similar which is sometimes also called Dark Energy. We live in a universe where Λ contributes $\sim 75\%$ to the energy content of the universe. The remaining $\sim 25\%$ consist of matter, of which only 20% consist of matter contained in the standard model of particle physics, like leptons or baryons[1]. The remaining matter is called Dark Matter and its nature is not completely understood by now. The most prominent candidate for the dark-matter particle arises from the simplest supersymmetric extension of the standard model of particle physics and is a linear combination of the zino, the photino and the higgsino[2]. Because of the notation of the cosmological constant as Λ and the short notation "DM" for dark matter, the cosmological model described above is simply called ΛCDM model. The "C" stands for cold and means that the dark particles are treated as non-relativistic.

Another very important step for comology was the discovery and the study of the Cosmic Microwave Background (CMB). Imprinted in the slight temperature fluctuations of the CMB is information on the universe at a much earlier time, namely from the time of recombination $\sim$ 360000 years after the Big-Bang. The CMB opened a window for a better understanding of structure formation in the universe.

Among the most important structures in the observed universe are galaxy clusters, which are the largest gravitationally bound objects in the universe. Multiple techniques to inverstigate their mass distribution are known and gravitational lensing is one of them.

In this thesis, we present a method to combine weak and strong gravitational-lensing effects. While strong-lensing effects are observed near the cores of massive galaxy clusters, including spectacular features like giant, luminous arcs or Einstein rings, weak-lensing effects are a lot more difficult to observe, but cover the complete observed field of a galaxy cluster. Therefore, a combination of the two effects allows a reconstruction of the galaxy-cluster mass distribution on all cluster scales.

[1] In cosmology all "ordinary" matter as we know it is called baryonic matter which is wrong in the context of particle physics

[2] Supersymmetric partners of the Z-boson, the photon and the Higgs-boson

The outline of this thesis is as follows. After discussing the basic theory of galaxy clusters and gravitational lensing in chapters two and three, we give the theoretical framework of our joint reconstruction method in chapter four. Chapters five and six are dedicated to the concrete numerical implementation of the reconstruction method using state-of-the-art computer and algorithms. Starting from the analysis of observational data, we show step-by-step how the mass distribution of a galaxy cluster is reconstructed. Chapter seven is then focused on applications. After testing and calibrating the method with simulated cluster data, we also perform the reconstruction of the real galaxy cluster MS 2137.

2 Galaxy Clusters

Galaxy Clusters (GCs) are the largest gravitationally bound objects in the universe (for reviews see Bartelmann, 2007; Schneider, 2006). At first they were identified as regions with a significant overdensity of galaxies. For example in 1958, Abell compiled a catalog of GCs with the criterion that within an angular radius of $\theta_A = \frac{1'.7}{z}$, he found more than 50 galaxies in a magnitude interval of $m_3 \leqslant m \leqslant m_3 + 2$, with m_3 being the apparent magnitude of the third brightest galaxy in the cluster. GCs normally fill volumes with linear dimensions of some Mpc and their mass lies in a range of $10^{14} - 10^{15} M_\odot$. They consist of mainly three different kinds of mass components. First there are the visible stars and galaxies. Furthermore we have the *intracluster medium* (ICM), which is composed of thermal baryonic gas, hot enough to emit in the X-Ray band, making GCs the brightest extragalactic X-Ray sources besides AGNs. The missing mass is located in a dark matter halo, which contributes by far the largest fraction to the total mass of the cluster.

2.1 General properties

2.1.1 Observations and Virial Theorem

The most important thing to know about a GC is its total mass and the distribution of mass within the cluster, thus its density profile. First observational hints on this issue were obtained by looking at the velocity distribution of the contained galaxies, which is centered around the bulk velocity of the whole cluster. One can define a velocity dispersion by

$$\sigma_v^2 = \langle v_\parallel^2 \rangle - \langle v_\parallel \rangle^2, \tag{2.1}$$

where $v_\parallel$ is the velocity component along the line of side. Measurements yield a typical velocity dispersion of galaxies in galaxy clusters of $\lesssim 1000 \text{km s}^{-1}$.

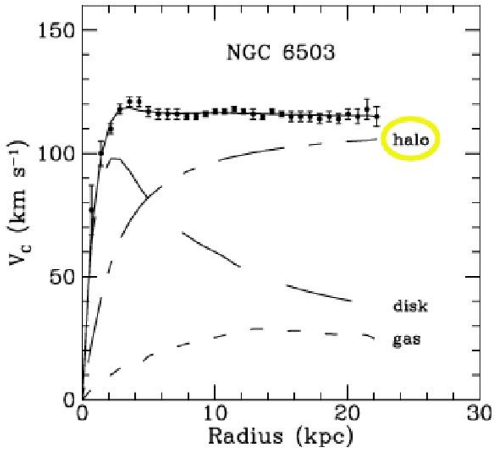


Figure 2.1: A typical rotation curve.

By applying the virial theorem

$$2\langle T \rangle = -\langle V \rangle \tag{2.2}$$

$$3m\sigma_m^2 = \frac{GMm}{R} \tag{2.3}$$

for a galaxy of mass m enclosed in a cluster of mass M at Radius R, we derive our first mass estimate

$$M \sim \frac{3R\sigma_v^2}{G} = 10^{15} h^{-1} Mpc \left(\frac{R}{1.5h^{-1}Mpc} \right) \left(\frac{\sigma_v}{1000 \text{km s}^{-1}} \right). \tag{2.4}$$

Vera Rubin first pointed out that there is a strong need for additional mass, because of the shape of *rotation curves*, which plot velocity dispersion against radius. The flatness of these curves for high r could not be explained by only the visible amount of mass in stars and galaxies. This was another hint for the existence dark matter in galaxy clusters, which was proposed for the first time by Zwicky (1933).

Combining the virial theorem with observations requires

$$\sigma_v^2 = \frac{GM}{3R} \overset{\text{large r}}{\Longrightarrow} \quad \text{const.} \tag{2.5}$$

which means that every suggested density profile, which should match the observations, has to fulfill the following radial dependence

$$\rho(r) \propto \frac{1}{r^2}, \tag{2.6}$$

due to the obvious relation between density and mass

$$M(r) = 4\pi \int_0^r r^{'2} dr^{'} \rho(r^{'}). \tag{2.7}$$

2.1.2 Density Profiles

Let us assume a GC to be a virialised self-gravitating system, consisting only of gravitationally interacting dark matter particles. By assuming hydrostatic equilibrium Euler's equation reads as following:

$$\frac{dP}{dr} = -\rho \frac{GM(r)}{r^2}, \tag{2.8}$$

which states that the pressure gradient is equal to the acceleration caused by the self-gravity of the system. The derivative with respect to r yields

$$\frac{d}{dr}\left(\frac{r^2}{\rho}\frac{dP}{dr}\right) + 4\pi G r^2 \rho = 0. \tag{2.9}$$

By further assuming thermal equilibrium, we treat the cluster material as an ideal gas with temperature T and use the relation between pressure and density $Pm = \rho kT$, as well as the relation between temperature and velocity dispersion $3kT = m\langle v^2 \rangle$, with $\langle v^2 \rangle$ being the mean squared velocity of a particle, we arrive at

$$\frac{dP}{dr} = \frac{kT}{m}\frac{d\rho}{dr} = \frac{\langle v^2 \rangle}{3}\frac{d\rho}{dr} = \sigma_v^2 \frac{d\rho}{dr}. \tag{2.10}$$

Here we assumed that T is independent of r and the fact that $\langle v^2 \rangle = \sigma_x^2 + \sigma_y^2 + \sigma_z^2$, giving the line-of-sight velocity dispersion

$$\sigma_v^2 = \frac{\langle v^2 \rangle}{3}, \tag{2.11}$$

which can be measured by the redshifts of the cluster galaxies.

Using all together we obtain

$$\frac{d}{dr}\left(\frac{\sigma_v^2 r^2}{\rho}\frac{d\rho}{dr}\right) + 4\pi G r^2 \rho = 0. \tag{2.12}$$

As is shown easily, one solution of this differential equation is

$$\rho(r) = \frac{\sigma_v^2}{2\pi G r^2}. \tag{2.13}$$

This density distribution is called *Singular Isothermal Sphere* and satisfies obviously the condition for a flat rotation curve. Unfortunately the distribution is singular at $r = 0$ and the total mass goes to infinity for $r \rightarrow \infty$.

To avoid these problems, one can introduce more elaborate self-gravitating models with an upper cut-off in the velocity distribution. These models are called *King models* (King, 1966, 1972, 1981) and have no analytical expression. An approximation for the central region of these profiles is given by

$$\rho(r) = \rho_0 \left[1 + \left(\frac{r}{r_c}\right)^2\right]^{-\frac{3}{2}}, \quad \text{with} \quad \rho_0 = \frac{9\sigma_v^2}{4\pi G r_c^2}. \tag{2.14}$$

A generalisation of the King profile is the so called *β profile* (Cavaliere & Fusco-Femiano, 1976, 1978), which introduces an additional fitting parameter β.

$$\rho(r) = \rho_0 \left[1 + \left(\frac{r}{r_c}\right)^2\right]^{-\frac{3\beta}{2}}. \tag{2.15}$$

We can repeat these calculations for the motion of galaxies within the external potential wells created by the dark-matter dominated total mass of the cluster. Therefor the distribution of galaxies in a galaxy cluster also follows a King profile which is confirmed by obseravtions.

2.1.3 Intra-Cluster Medium and X-Ray Emission

Having started our discussion mainly on dark matter particles and their density profiles, we now focus on the ICM. This is directly related to the discovery that GCs are strong X-Ray emitters; in fact they are one of the brightest X-Ray sources in the sky. Looking at the spectra of the X-Ray emission, it was shown that the emission is thermal and spatially distributed over the whole cluster, exceeding the area of light emitting galaxies.

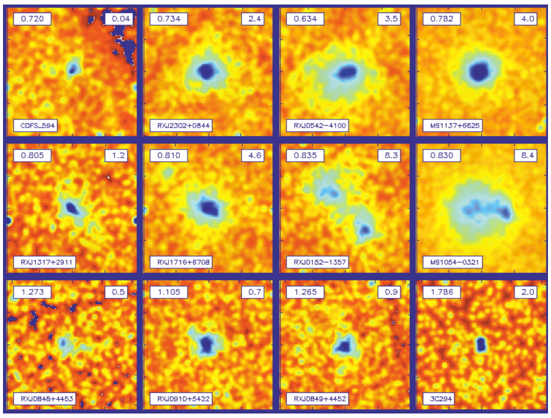


Figure 2.2: Chandra archival images of twelve distant clusters at $0.7 < z < 1.3$.

It was concluded that the ICM consists of a hot, thermal plasma, where the electrons scatter from the ions and emit thermal *bremsstrahlung* (so called free-free emission).

The theory of radiation processes states that the X-Ray *emissivity* $j_v(\boldsymbol{x})$ is given by

$$j_\nu(\boldsymbol{x}) = C \frac{\rho^2}{\sqrt{T}} e^{-h\nu/kT}, \tag{2.16}$$

with the photon frequency ν and the plasma temperature T. One should notice the squared dependence on the density which is due to the fact that the X-Ray emission is a two body process.

By also assuming thermal and hydrostatic equilibrium for the ICM particles, and because of the comparison with the density distribution of galaxies which see the same potential wells created by the overall mass of the GC, the ICM density should follow a β-profile:

$$\rho_{\text{gas}} = \rho_0 \left(1 + \frac{r^2}{r_c^2}\right)^{-3\beta/2}. \tag{2.17}$$

Using the ρ^2-dependence for the X-Ray-emissivity we obtain

$$j_\nu(r) \propto \left(1 + \frac{r^2}{r_c^2}\right)^{3\beta}, \tag{2.18}$$

and after projecting in the observation plane, we arrive at the X-Ray-flux per unit solid angle

$$S_X = S_{XO}\left(1+\frac{\theta}{\theta_c}\right)^{-3\beta+1/2}. \tag{2.19}$$

Fitting the density profile to the observed surface brightness of Galaxy Clusters in X-Ray bands, gives $r_c \sim 200h^{-1}$kpc and $\beta \sim 2/3$.

Furthermore, Eq. 2.10 yields the following total mass of the system in hydrostatic equilibrium:

$$M(r) = -\frac{rkT}{Gm}\left(\frac{d\ln\rho}{d\ln r}+\frac{d\ln T}{d\ln r}\right). \tag{2.20}$$

So with the β-profile for the ICM particles fitted by X-Ray observations, we get the X-Ray mass estimate:

$$M(r) = \frac{3\beta rkT}{Gm}\frac{r^2/r_c^2}{1+r^2/r_c^2}. \tag{2.21}$$

Please note that with the advancement to high resolution X-Ray space telescopes, like Chandra or XMM-Newton, multiple fits of the observed X-Ray-flux with differing temperature are possible.

Unfortunately one has to be very careful with these mass estimates, because they rely on quite strong assumptions regarding the dynamical and thermal state of the cluster. Especially cluster mergers can contaminate the result significantly. This makes it necessary to compare with other methods like gravitational lensing, which do not rely on the same assumptions.

2.2 Galaxy Clusters and Cosmology

2.2.1 Cluster Abundance in the Universe

In this chapter we want to point out why GCs are important objects for cosmology. The first thing is that cosmological structure formation in a ΛCDM universe predicts a bottom-up model, which means that large structures form last. So GCs reflect the late evolution of cosmic structures. Second, GCs are very large objects, so we are talking about scales which are not so far in the nonlinear regime as typical scales for galaxies for example; so the abundance of galaxy clusters and their evolution can be a profound test for cosmological structure formation.

The pioneering work regarding the abundance of GCs in the universe by cosmological considerations was done by Press & Schechter (1974). Without going into the details of the formalism, we present the main results.

The fraction of the overall volume occupied by collapsed structures depending on the smoothing radius of the density field, where the smoothing radius is set by the mass of the objects one is interested in, is given by

$$f_{\text{coll}}(M(r),z) = \frac{2}{\sqrt{2\pi}\sigma(R,z)}\int_{\delta_c}^{\infty} d\delta \;\, e^{-\delta^2/2\sigma^2(R,z)}, \tag{2.22}$$

under the assumption that the density field is well described by a Gaussian random field.

By differentiating with respect to M and multiplying with the mean number density of these objects one arrives at a more convenient form of Press-Schechter mass funtion, which gives the number density of collapsed objetcs in a mass interval between M and M + dM.

$$\frac{dn(M,z)}{dM} = \sqrt{\frac{2}{\pi}}\frac{\rho_m\delta_c}{3M^2\sigma}e^{-\delta_c^2/2\sigma^2}\left[-\frac{R}{\sigma}\frac{d\sigma}{dR}\right], \tag{2.23}$$

where z is cosmological redshift, ρ_m the mean density of the objects with mass M and δ_c the critical overdensity, which is needed for an object to collapse.

This formula is of particular importance because it computes the expected abundance of GCs in a given cosmological model. This can then be verified by observations. For doing so, one needs to identifiy GCs and determine their mass, which is the focus of this thesis.

2.2.2 Cosmological Simulations

Since the development of powerful computation facilities, simulations have become more and more important for studying cosmological structure formation. The main concept of these simulations is quite simple. One takes N particles of a given mass and puts them into a box of a given size. After setting the initial conditions for the particle distribution in the box, one adds physics like gravity and hydrondynamics. By dividing the time evolution of the simulation into finite time steps, positions and velcocities for each particle are updated at each time step. As long as you want to have as many particles as possible, this can become very expensive in computation time.

A state-of-the-art simulation is the *Millenium Simulation* (Springel et al., 2005) which contains 2160^3 particles in a box of 500 h^{-1}Mpc. The simulation was performed using the SPH tree code[1] Gadget (Springel et al., 2001; Springel, 2005). This kind of simulation provides a very powerful tool to study the evolution of the universe, because one can observe its behaviour for different kinds of cosmologies and physics.
Especially it allows one to look at the formation and evolution of Dark Matter halos and their density profiles, and to compare them with observations.
Navarro, Frenk and White (see Navarro et al., 1996) determined the density distributions of dark matter Halos in numerical simulations and found a universal fitting formula which is the *NFW-profile*

$$\rho_{NFW}(r) = \frac{\rho_s}{\frac{r}{r_s}\left(1+\frac{r}{r_s}\right)^2}. \tag{2.24}$$

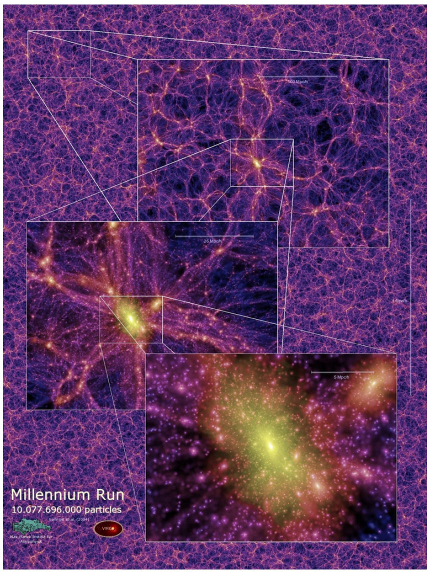


Figure 2.3: The filament structure of the Millenium run on different scales.

This density profile has two parameters, the characteristic density ρ_s and the scaling radius r_s. This profile in different varations is now largely used to model the structure of GCs, even if it is not known why the structure of dark matter halos should have this specific shape.

[1] SPH stands for smoothed particle hydrodynamics, which treats the particles not as pointlike but as expanded quantities with a certain cut-off kernel. A tree code computes not all interaction equations for a particles by dividing the simulation into subgrids and averaging, with increasing resolution for high density areas

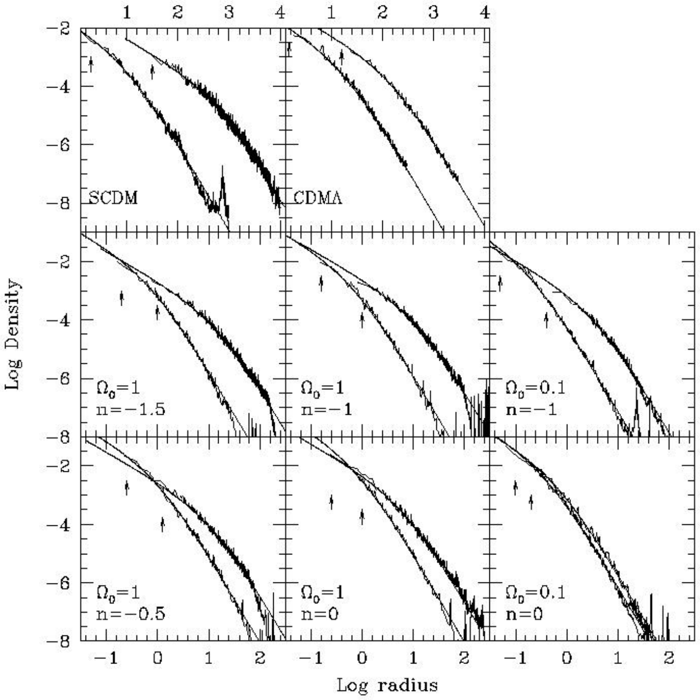


Figure 2.4: Density profiles of dark matter halos in the simulations of Navarro, Frenk and White.

3 Gravitational Lensing

3.1 Basic Theory of Lensing

3.1.1 Light Deflection

Now we focus on *Gravitational Lensing*, which is the the main theory this thesis is based on (see Meneghetti, 2006; Bartelmann & Schneider, 2001; Wambsganss, 1998, for reviews on lensing).

Einsteins's Theory of General Relativity predicts that light follows the null geodesics of a given space-time metric. That means nothing else than the deflection of light rays in the presence of massive objects like GCs. Because clusters are such massive objects, concentrated on a relatively small volume, compared to the distances of sources and observer where one wants to measure the light deflection, one can describe the theory of gravitational lensing with a geometrical-optics approach. (For a strict treatment in the context of General Relativity refer to Bartelmann & Schneider (2001)). As sketched in Fig.3.1, there are three main ingredients to a lensing scenario. The source, placed on the source plane, the lens itself, localised on the lens plane and the observer. The corresponding angular-diameter distances are D_d for the observer-lens distance, D_s for the observer-source distance and D_{ds} for the lens-source distance. Let $\boldsymbol{\theta}$ be the angular position of an observed image, $\boldsymbol{\beta}$ the position of this image in the sky if it was not lensed, and $\hat{\boldsymbol{\alpha}}$ the corresponding deflection angle. If we consider in a first step the deflecting mass as a point mass, General Relativity predicts a deflection angle

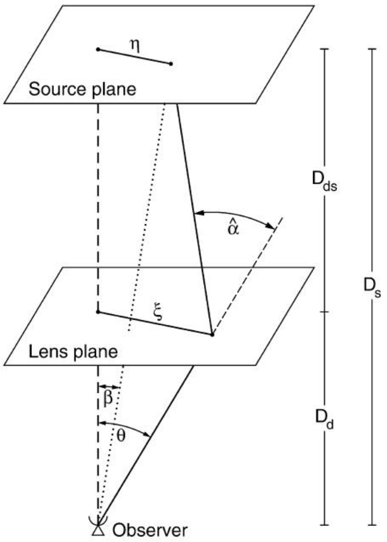


Figure 3.1: Sketch of a typical gravitational lens system (Figure from Bartelmann & Schneider, 2001)

$$\hat{\alpha} = \frac{4GM}{c^2 \xi}, \tag{3.1}$$

where ξ is the impact parameter in the lens plane. (A complete calculation can be seen in Meneghetti (2006)). For a general mass distribution with a mass element dm at $\boldsymbol{r} = (\xi_1', \xi_2', r_3')$, the impact parameter becomes a vector $\boldsymbol{\xi} - \boldsymbol{\xi}'$. This holds if r_3 is chosen such that the light rays propagate along this coordinate and you assume the light rays to be straight lines while passing the mass. (Similar to the Born approximation in quantum physics). These assumptions make sense if the deflection of the light rays is weak, compared to the scales where the deflecting mass distribution changes significantly. This is true for lensing by galaxies or clusters of galaxies. We finally obtain

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \frac{4G}{c^2} \int d^2\xi' \int dr_3' \rho(\xi_1', \xi_2', r_3') \frac{\boldsymbol{\xi} - \boldsymbol{\xi}'}{|\boldsymbol{\xi} - \boldsymbol{\xi}'|^2}, \tag{3.2}$$

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \frac{4G}{c^2} \int d^2\xi' \Sigma(\boldsymbol{\xi}') \frac{\boldsymbol{\xi} - \boldsymbol{\xi}'}{|\boldsymbol{\xi} - \boldsymbol{\xi}'|^2}. \tag{3.3}$$

In the last step we introduced the *surface mass density*,

$$\Sigma(\boldsymbol{\xi}') \equiv \int dr_3 \rho(\xi_1', \xi_2', r_3'). \tag{3.4}$$

3.1.2 The Lens Equation

As one can see from Fig 3.1

$$\boldsymbol{\eta} = \frac{D_s}{D_d}\boldsymbol{\xi} - D_{ds}\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}); \tag{3.5}$$

and by using $\boldsymbol{\eta} = D_s\boldsymbol{\beta}$ and $\boldsymbol{\xi} = D_d\boldsymbol{\theta}$, we find the *lens equation*

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \frac{D_{ds}}{D_s}\hat{\boldsymbol{\alpha}}(D_d\boldsymbol{\theta}) \equiv \boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta}), \tag{3.6}$$

with the *scaled deflection angle* $\boldsymbol{\alpha}(\boldsymbol{\theta})$. In addition we define another crucial lensing quantity, namely the *convergence* κ

$$\kappa(\boldsymbol{\theta}) = \frac{\Sigma(D_d\boldsymbol{\theta})}{\Sigma_{cr}} \quad \text{with} \quad \Sigma_{cr} = \frac{c^2}{4\pi G}\frac{D_s}{D_d D_{ds}}. \tag{3.7}$$

Now the deflection angle reads

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \frac{1}{\pi}\int_{\mathbb{R}_2} d^2\theta' \kappa(\boldsymbol{\theta}')\frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{|\boldsymbol{\theta} - \boldsymbol{\theta}'|^2}. \tag{3.8}$$

The important thing about the convergence is that it decides if a lens system is called critical. Because if $\kappa \geq 1$ somewhere in the mass distribution, the lens equation has more than one solution and multiple images can be formed from the same source. This is a sufficient but not necessary condition.

3.1.3 The Lensing Potential

As Eq. 3.8 implies, the deflection angle can be written as the gradient of a potential, which we call from now on *lensing potential*

$$\psi(\boldsymbol{\theta}) = \frac{1}{\pi}\int_{\mathbb{R}_2} d^2\theta' \kappa(\boldsymbol{\theta}')\ln|\boldsymbol{\theta} - \boldsymbol{\theta}'|. \tag{3.9}$$

The important lensing quantities become now derivatives of this potential with respect to angular coordinates

$$\boldsymbol{\alpha} = \nabla\psi \tag{3.10}$$

$$\kappa = \frac{1}{2}\nabla^2\psi = \frac{1}{2}\left(\frac{\partial^2}{\partial\theta_1^2} + \frac{\partial^2}{\partial\theta_2^2}\right)\psi = \frac{1}{2}(\psi_{,11} + \psi_{,22}) \tag{3.11}$$

$$\gamma_1 = \frac{1}{2}\left(\frac{\partial^2}{\partial\theta_1^2} - \frac{\partial^2}{\partial\theta_2^2}\right)\psi = \frac{1}{2}(\psi_{,11} - \psi_{,22}) \tag{3.12}$$

$$\gamma_2 = \frac{\partial^2}{\partial\theta_1\theta_2}\psi = \psi_{,12}. \tag{3.13}$$

In the last two lines, we defined the components of the *shear*

$$\gamma = \gamma_1 + i\gamma_2 = |\gamma|e^{2i\phi}. \tag{3.14}$$

3.1.4 Magnification and Distortion

We now come to the effect of gravitational lensing on image shapes, compared to the shapes of the sources. With a given surface-brightness distribution $I^{(s)}(\boldsymbol{\beta})$ in the source plane and an observed surface-brightness distribution $I(\boldsymbol{\theta})$, the statement of Liouville's theorem is

$$I(\boldsymbol{\theta}) = I^{(s)}[\boldsymbol{\beta}(\boldsymbol{\theta})], \tag{3.15}$$

because no photon creation or destruction is involved in gravitational lensing.

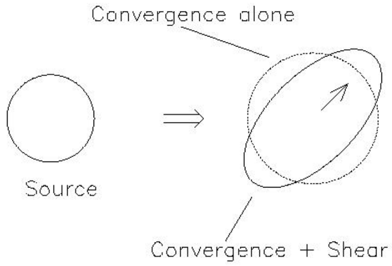


Figure 3.2: Illustration of the effects caused by shear and convergence (From Narayan & Bartelmann, 1996).

The relation between $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ is given by the lens equation and if we assume that the size of the source is negligible, compared to the scales on which the properties of the lens change, we can locally linearize the lens equation. Linearising Eq. 3.6 around a point $\boldsymbol{\theta_0}$ leads to

$$I(\boldsymbol{\theta}) = I^{(s)}[\boldsymbol{\beta_0} + \mathcal{A}(\boldsymbol{\beta_0}) \cdot (\boldsymbol{\theta} - \boldsymbol{\theta}_0)], \tag{3.16}$$

with $\boldsymbol{\beta_0} = \boldsymbol{\beta}(\boldsymbol{\theta_0})$ and the *Jacobian Matrix*

$$\begin{aligned} \mathcal{A}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\theta}} &= \left(\delta_{ij} - \frac{\partial^2 \psi(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\right) \\ &= \begin{pmatrix} 1-\kappa-\gamma_1 & -\gamma_2 \\ -\gamma_2 & 1-\kappa+\gamma_2 \end{pmatrix} \\ &= (1-\kappa)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \gamma \begin{pmatrix} \cos 2\phi & \sin 2\phi \\ \sin 2\phi & -\cos 2\phi \end{pmatrix}. \end{aligned} \tag{3.17}$$

The last form of the Jacobian is convenient, because one can nicely see that the convergence is responsible for a change in size of the images while the shear causes also a change in the shape of the images, see Fig 3.2.

The *magnification* μ is the ratio between the fluxes of the observed image and the unlensed source. It is given by

$$\mu = \frac{1}{\det\mathcal{A}} = \frac{1}{(1-\kappa)^2 - |\gamma|^2}. \tag{3.18}$$

A set of points in the source plane where the Jacobian becomes singular is called *critical curve* and the corresponding image curves in the source plane are called *caustics*.

3.1.5 Lensing by an Isothermal Sphere

As a very nice example we want to present lensing by a singular isothermal sphere. With the results from Chapter 2.1.2 we obtain

$$\Sigma(\xi) = \frac{\sigma_v^2}{2G\xi}, \quad \kappa(\theta) = \frac{\theta_E}{2\theta}, \quad \text{with} \quad \theta_E = 4\pi \left(\frac{\sigma_v}{c}\right)^2 \frac{D_{ds}}{D_s}, \tag{3.19}$$

where θ_E is called the *Einstein deflection angle*, which sets an important lensing scale (e.g. the angular separation of multiple images). In our example, also the other lensing quantities look very simple

$$|\boldsymbol{\alpha}| = \theta_E, \quad \psi = \theta_E|\boldsymbol{\theta}|, \quad \mu(\boldsymbol{\theta}) = \frac{|\boldsymbol{\theta}|}{|\boldsymbol{\theta}| - \theta_E}. \tag{3.20}$$

We see that the Einstein angle defines a critical curve, which is called *Einstein circle*.

Figure 3.3: Lens system SDSSJ0737+3216, left panel: reduced image; right panel: reconstructed critical curve and caustic (From Marshall et al., 2007).

3.2 Lensing by Galaxy Clusters

3.2.1 Weak Lensing

We will now focus more concretely on lensing by galaxies and especially galaxy clusters. A lens is called critical if it develops extended critical curves. This usually happens if $\kappa \approx 1$. This also distinguishes between two very characteristic regimes, weak lensing and strong lensing. Let us first focus on weak lensing, which means $\kappa \ll 1$.

With a given set of background galaxies behind our lens, the images of these galaxies are slightly distorted due to lensing. While talking about images, we always talk about the surface brightness $I(\boldsymbol{\theta})$ that we can observe. Once we have measured the surface brightness, we can define the centre of the image and the tensor of its second brightness moments

$$\bar{\boldsymbol{\theta}} \equiv \frac{\int d^2\theta q_I[I(\boldsymbol{\theta})]\boldsymbol{\theta}}{\int d^2\theta q_I[I(\boldsymbol{\theta})]}, \quad Q_{ij} \equiv \frac{\int d^2\theta q_I[I(\boldsymbol{\theta})](\theta_i - \bar{\theta}_i)(\theta_j - \bar{\theta}_j)}{\int d^2\theta q_I[I(\boldsymbol{\theta})]}, \quad i, j \in \{1, 2\} \tag{3.21}$$

with a suitable weight function q_I, which cuts the image off in the observed field (e.g. a Heaviside step function).

With the knowledge of the brightness tensor, we can define the *complex ellipticity* (please note that other definitions of ellipticity are possible):

$$\varepsilon \equiv \frac{Q_{11} - Q_{22} + 2iQ_{12}}{Q_{11} + Q_{22} + 2(Q_{11}Q_{22} - Q_{12}^2)^{1/2}}. \tag{3.22}$$

In the same way we define the surface brightness tensor for the sources $Q^{(s)}$. Then the lensing transformation is given by

$$Q^{(s)} = \mathcal{A}Q\mathcal{A}^T = \mathcal{A}Q\mathcal{A}. \tag{3.23}$$

In terms of ellipticities we get

$$\varepsilon^{(s)} = \begin{cases} \dfrac{\varepsilon - g}{1 - g^*\varepsilon} & \text{for } |g| \leq 1 \\ \dfrac{1 - g\varepsilon^*}{\varepsilon^* - g^*} & \text{for } |g| > 1 \end{cases} \tag{3.24}$$

with the *reduced shear*

$$g(\boldsymbol{\theta}) \equiv \frac{\gamma(\boldsymbol{\theta})}{1 - \kappa(\boldsymbol{\theta})}. \tag{3.25}$$

The inverse transformation is given by

$$\varepsilon = \begin{cases} \dfrac{\varepsilon^{(s)} + g}{1 + g^*\varepsilon^{(s)}} & \text{for } |g| \leq 1 \\ \dfrac{1 + g\varepsilon^{(s)*}}{\varepsilon^{(s)*} + g^*} & \text{for } |g| > 1 \end{cases} . \tag{3.26}$$

As long as the lensing signal is weak and has to be treated statistically, we want to average over a certain number of galaxies. By doing so, we assume that the intrinsic average source ellipticity will vanish and we get the following simple expression for the expectation value of the measured ellipticity

$$\langle \varepsilon \rangle = \begin{cases} g & \text{for } |g| \leq 1 \\ \dfrac{1}{g^*} & \text{for } |g| > 1 \end{cases} . \tag{3.27}$$

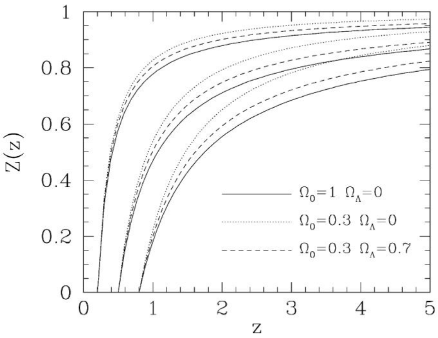


Figure 3.4: The cosmological weight funtion for different cosmologies and lens redshifts (from Bartelmann & Schneider, 2001).

This relation is true if we assume that all our sources are located at the same redshift, but as we saw in Subsection 3.2.1 the critical surface mass density depends on redshift through the angular diameter distances. To take this into account we define the *cosmological weight function*

$$Z(z) \equiv \frac{D_\infty D_{ds}}{D_{d\infty} D_s} H(z - z_d), \tag{3.28}$$

where z_d is the redshift of the lens and $D_\infty, D_{d\infty}$ are the angular diameter distances between observer and infinity and between lens and infinity respectively. Our expectation for the ellipticity then becomes

$$\langle \varepsilon \rangle = \begin{cases} \dfrac{Z(z)\gamma}{1 - Z(z)\kappa} & \text{for } |g| \leq 1 \\ \dfrac{1 - Z(z)\kappa}{Z(z)\gamma^*} & \text{for } |g| > 1 \end{cases} . \tag{3.29}$$

With this knowledge we are now in the position to perform a mass reconstruction, based on weak lensing. We focus here on so called direct inversion methods, because we will discuss different approaches later in more detail. So let us go back to Eq. 3.9 and rewrite it as

$$\gamma(\boldsymbol{\theta}) = \frac{1}{\pi} \int_{\mathbb{R}^2} d^2\theta' \mathcal{D}(\boldsymbol{\theta} - \boldsymbol{\theta}') \kappa(\boldsymbol{\theta}'), \tag{3.30}$$

with the geometric kernel

$$\mathcal{D}(\boldsymbol{\theta}) \equiv \frac{\theta_2^2 - \theta_1^2 - 2i\theta_1\theta_2}{|\boldsymbol{\theta}|^4} = \frac{-1}{(\theta_1 - i\theta_2)^2}. \tag{3.31}$$

Convergence and shear are both related to the lensing potential. By transforming ψ, κ and γ to Fourier-Space $(\hat{\psi}, \hat{\kappa}, \hat{\gamma})$ with the wavevector $\boldsymbol{k}$ corresponding to $\boldsymbol{\theta}$ we find the following relation by eliminating the potential:

$$\hat{\kappa} = k^{-2} \left[(k_1^2 - k_2^2)\hat{\gamma}_1 + 2k_1k_2\hat{\gamma}_2 \right]. \tag{3.32}$$

By applying the convolution theorem and going back to real space we obtain the result

$$\kappa(\boldsymbol{\theta}) - \kappa_0 = \frac{1}{\pi} \int_{\mathbb{R}^2} d^2\theta' \mathcal{D}^*(\boldsymbol{\theta} - \boldsymbol{\theta}')\gamma(\boldsymbol{\theta}') = \frac{1}{\pi} \int_{\mathbb{R}^2} d^2\theta' \Re[\mathcal{D}^*(\boldsymbol{\theta} - \boldsymbol{\theta}')\gamma(\boldsymbol{\theta}')], \tag{3.33}$$

which can be used with observations, in the case of pure weak lensing ($\kappa \ll 1, |\gamma| \ll 1$), where $< \varepsilon >\sim \gamma$. If we have n ellipticity measurements ε_i, we find

$$\kappa(\boldsymbol{\theta}) = \frac{1}{n\pi} \sum_i \Re[\mathcal{D}^*(\boldsymbol{\theta} - \boldsymbol{\theta}_i)\varepsilon_i]. \tag{3.34}$$

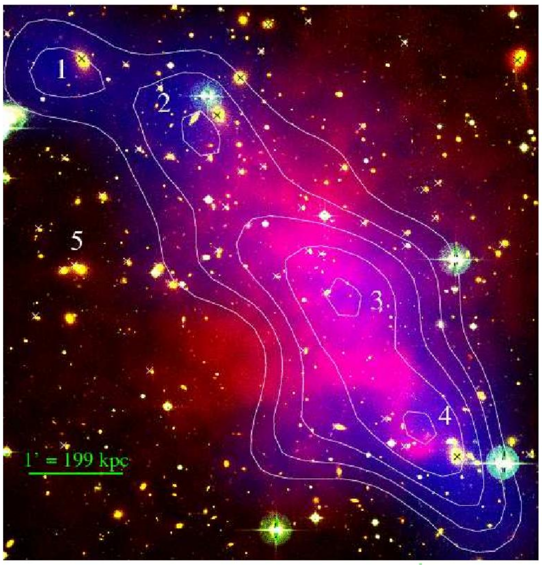


Figure 3.5: The "Cosmic Train Wreck" A520 with Chandra X-Ray Map (red) and weak lensing contours (blue). (From Mahdavi et al, 2007).

A generalisation of this method also takes into account the reduced shear, while using an iterative approach

$$\kappa(\boldsymbol{\theta}) - \kappa_0 = \frac{1}{\pi}\int_{\mathbb{R}^2} d^2\theta^{'}[1-\kappa(\boldsymbol{\theta}^{'})]\Re[\mathcal{D}^*(\boldsymbol{\theta}-\boldsymbol{\theta}^{'})\gamma(\boldsymbol{\theta}^{'})], \tag{3.35}$$

where one starts from a convergence $\kappa \equiv 0$ and reinserts the updated value at each iteration step. This process converges quickly towards a unique solution.
Several weak lensing reconstructions with these direct methods have been carried out in the last years. For a reconstruction of the Bullet Cluster (1ES 0657-558) see Clowe et al. (2004). A recent reconstruction of Abell 520 was performed by Mahdavi et al. (2007) and a compilation of weak lensing reconstruction is given in Dahle (2007). Although the mass reconstruction method described above is fast and easy to implement, it suffers from several weaknesses:

- It is quite sensitive to noise, so one has to introduce for example a Gaussian smoothing kernel;
- In reality one is dealing with limited observation fields. But the integral in Eq. 3.33 extends over the whole $\mathbb{R}^2$, which means that you have to set $\gamma = 0$ outside the observed field. This leads to unwanted boundary artefacts;
- The method is not very flexible, in the sense that it is not possible to add additional constraints from other observations, besides ellipticity measurements;
- As one can check easily, the reduced shear is invariant under the transformation $\kappa \rightarrow \kappa^{'} = (1-\lambda) + \lambda\kappa, \;\; \gamma \rightarrow \gamma^{'} = \lambda\gamma$, which makes it impossible to find a unique result, without additional considerations. The invariance under this transformation is called *mass sheet degeneracy*.

3.2.2 Strong Lensing

While weak lensing has slightly distorted background galaxies as observables, which can only be treated statistically, strong lensing is based on more spectacular effects. Another difference to weak lensing is that strong lensing only takes place near the cluster core where the lens becomes critical. The main observations are the following:

- Multiple images of background sources, where all the different images carry the spectral information of the source. Due to this fact mutiple images can be identified through a spectral or colour analysis;

- Highly distorted images of background sources like gravitational arcs or arclets, which were first identified by Fort et al. (1988). Here the images lie close to the critical curve of the cluster.

To do a mass reconstruction based on strong lensing constraints, one has to take into account several things. First a cluster can either have a more or less uniform shape, or it can contain a lot of substructure. This can be treated with a multipole expansion

$$\kappa(\boldsymbol{\theta}) = \kappa_0(\boldsymbol{\theta}) + \sum_{m=1}^{\infty} \kappa_m(\boldsymbol{\theta}) e^{im\phi}, \tag{3.36}$$

where it depends on the level of accuracy, to which order one can expand. The monopole term describes an axially symmetric lens. The quadrupole term, for example, describes the ellipticity of the lens, which plays a very important role for a good reconstruction. Furthermore, one should also think of the so called external shear, i.e. perturbations coming from the outer regions of the cluster. Here strong lensing is normally not important anymore, but it can affect very well the lensing properties in the strong regime and act as an additional lensing potential

$$\psi_\gamma(\boldsymbol{\theta}) = \frac{\gamma_e}{2}(\theta_1^2 - \theta_2^2). \tag{3.37}$$

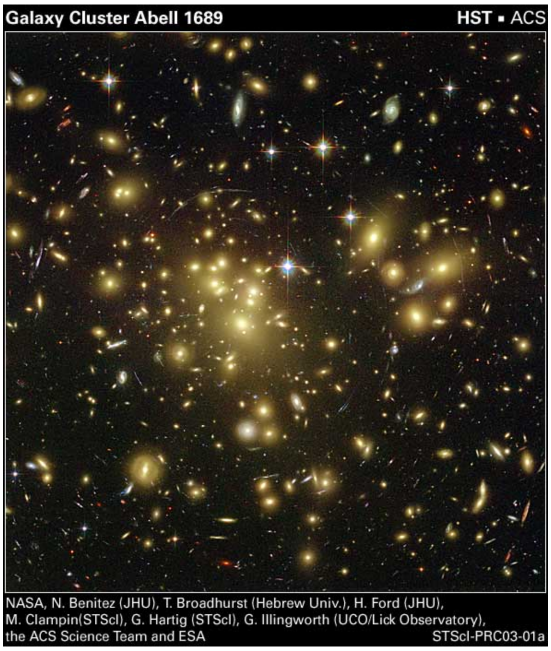

Galaxy Cluster Abell 1689
HST • ACS
NASA, N. Benitez (JHU), T. Broadhurst (Hebrew Univ.), H. Ford (JHU), M. Clampin(STScI), G. Hartig (STScI), G. Illingworth (UCO/Lick Observatory), the ACS Science Team and ESA
STScI-PRC03-01a

Figure 3.6: The very massive Galaxy Cluster A1689

Once one has collected all available constraints, there are two possibilities for a mass reconstruction. First, there is a parametric approach, which is by now more common and successful. Hence we have a set of parameters $\boldsymbol{p}$ describing our lens, and N observations. We now use the lens equation,

$$\boldsymbol{\beta}_i = \boldsymbol{\theta}_i - \boldsymbol{\alpha}(\boldsymbol{\theta}_i, \boldsymbol{p}) \quad 1 \leq i \leq N, \tag{3.38}$$

to define a χ^2-Function

$$\chi^2_{src} = \sum_{i=1}^{N} \left(\frac{\boldsymbol{\beta} - \boldsymbol{\beta}_i}{\sigma_i} \right)^2, \tag{3.39}$$

where the source position $\boldsymbol{\beta}$ is of course also not known and is thus a model parameter. Analogously we can define for the image positions

$$\chi^2_{img} = \sum_{i} \left(\frac{\boldsymbol{\theta}_i(\boldsymbol{\beta}) - \boldsymbol{\theta}_i}{\sigma_i} \right)^2. \tag{3.40}$$

Finally the overall χ^2 is minimised with respect to the parameters and the result is obtained with an accuracy that depends on the parametrisation and on the quality of the data.
Another possibility is a completely non-parametric approach, where the surface mass density

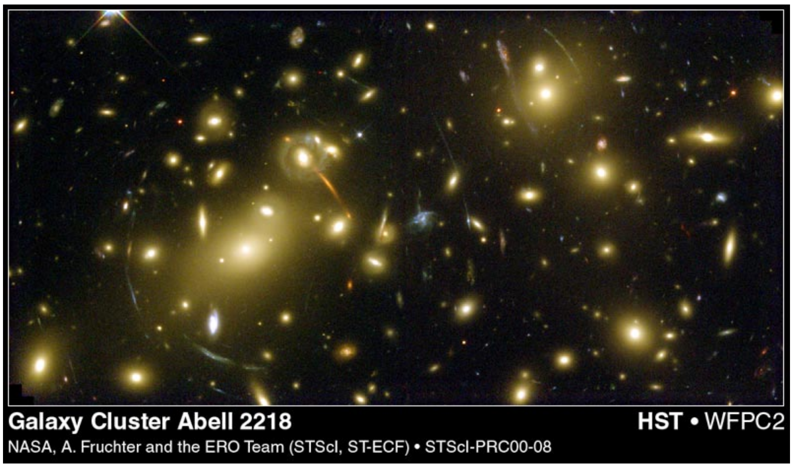

Galaxy Cluster Abell 2218
HST • WFPC2
NASA, A. Fruchter and the ERO Team (STScI, ST-ECF) • STScI-PRC00-08

Figure 3.7: The Galaxy Cluster A2218 which shows a lot of strong lensing features.

of the lens is decomposed into an adequate set of density functionals. These funtionals are multiplied with unknown decomposition coefficients and create a set of parameters $\boldsymbol{k}$. Now the lens equation for every image i looks as follows:

$$\boldsymbol{\beta}_i = \boldsymbol{\theta}_i - A_i \boldsymbol{k}, \tag{3.41}$$

with a matrix A, which gives the deflection angles of the images in terms of the unknown decomposition coefficients $\boldsymbol{k}$. The problem with such an approach is that the resulting systems of equations are underdetermined so in the end one has to make additional assumptions again.

For recent strong lensing reconstructions of A1689 see Broadhurst et al. (2005), Halkola et al. (2006) or Tu et al. (2007). A reconstruction of A2218 was done by Elíasdóttir et al. (2007). Comerford et al. (2006) analysed a sample of 10 GCs, observed with the HST and Rzepecki et al. (2007) reconstructed RCS0224-0002. To see examples of non-paramteric approaches, refer to Diego et al. (2007) or Liesenborgs et al. (2007).

Another important aspect of strong lensing that we have not mentioned so far, is that the abundance of gravitational arcs can be used to draw conclusions on cosmology (see Bartelmann et al., 1998; Wambsganss et al., 2004; Fedeli et al., 2006).

3.3 Lensing to 2nd Order: Flexion

3.3.1 2nd Order Lens Equation

At the end of our introduction to gravitational lensing we turn our focus again to the lens equation (Eq. 3.6). In the weak lensing approach, we expanded the lens equation to first order, to find the source-image relations. Due to this fact, weak gravitional lensing loses information, especially in richly structured parts of the lens. We try to alleviate this by expanding to second order.

Starting from Eq. 3.15 we expand the components of $\boldsymbol{\beta}$ around the origin

$$\beta_i = \frac{\partial \beta_i}{\partial \theta_j}\theta_j + \frac{1}{2}\frac{\partial^2 \beta_i}{\partial \theta_j \theta_k}\theta_j\theta_k + O(\boldsymbol{\theta}^3). \tag{3.42}$$

We already expressed the first term with the help of the Jacobian (Eq.3.17), and by defining

$$D_{ijk} = \frac{\partial A_{ij}}{\partial \theta_k}, \tag{3.43}$$

we express the second order term through derivatives of the Jacobian.
The gradient of the convergence,

$$\nabla\kappa = \begin{pmatrix} \gamma_{1,1} + \gamma_{2,2} \\ \gamma_{2,1} - \gamma_{1,2} \end{pmatrix}, \tag{3.44}$$

gives us the coefficients of D_{ijk}

$$D_{ij1} = \begin{pmatrix} -2\gamma_{1,1} - \gamma_{2,2} & -\gamma_{2,1} \\ -\gamma_{2,1} & -\gamma_{2,2} \end{pmatrix}, \quad D_{ij2} = \begin{pmatrix} -\gamma_{2,1} & -\gamma_{2,2} \\ -\gamma_{2,2} & 2\gamma_{1,2} - \gamma_{2,1} \end{pmatrix}. \tag{3.45}$$

By Taylor expanding again Eq. 3.15 and inserting our results, we arrive at the 2nd-order lens equation

$$I(\boldsymbol{\theta}) \simeq \left\{ 1 + \left[(A - I)_{ij}\theta_j + \frac{1}{2} D_{ijk}\theta_j\theta_k \right] \frac{\partial}{\partial\theta_i} \right\} I^{(s)}(\boldsymbol{\theta}). \tag{3.46}$$

3.3.2 Flexion Formalism

We now present a formalism developed by Goldberg & Bacon (2005), which describes the lensing quantites in a very elegant and convenient way.
In our case we deal with 2-dimensional vector fields so it is reasonable to map these fields into the complex plane:

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \rightarrow v_1 + iv_2. \tag{3.47}$$

By doing so, it is also necessary to define derivative operators, which obey the condition that they reproduce our results obtained so far in the classical formalism. This is done by the following complex gradient operators,

$$\partial \equiv \partial_1 + i\partial_2 \qquad \partial^\dagger \equiv \partial_1 - i\partial_2. \tag{3.48}$$

Applying this to the scalar lensing potential reproduces the deflection angle, the Laplacian and the convergence

$$\partial\psi = \alpha \tag{3.49}$$
$$\partial\partial^\dagger = \partial^\dagger\partial = \nabla^2 \tag{3.50}$$
$$\partial^\dagger\alpha = \partial^\dagger\partial\psi = 2\kappa. \tag{3.51}$$

We should note here that the potential is a scalar field and the deflection angle is a spin 1 vector field. So we raised the spin by one, by applying ∂. On the other hand, we obtained the scalar convergence field by applying $\partial^\dagger$ to α. This shows that the complex gradient operators can be understood as spin-raising and spin-lowering operators.
This also shows that the shear is a spin-2 field

$$\gamma = \frac{1}{2}\partial\partial\psi, \tag{3.52}$$

and the relation between convergence and shear is now given by

$$\kappa = \partial^{-1}\partial^\dagger\gamma. \tag{3.53}$$

So far we just reproduced the classical lensing quantities, but as long as every quantity which appears in the expansion of the lens equation will be a derivative of the lensing potential, we developed a tool to describe higher-order lensing in a very intuitve way. Going to second order

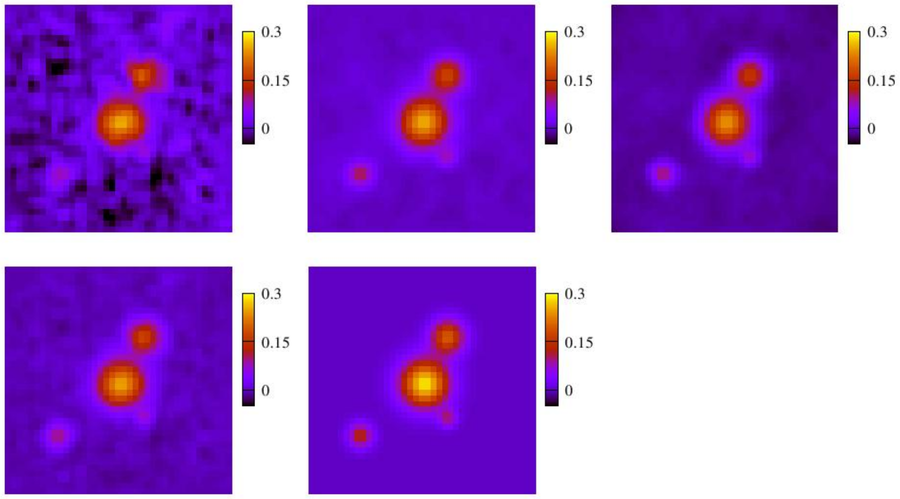


Figure 3.8: Estimated convergence maps from the shear (top left panel), the first flexion (top centre panel) and the second flexion (top right panel) with 40 x 40 lens cells. In the bottom center panel these maps are combined with minimal noise weighting. The comparison with the true convergence map rebinned at this resolution (bottom right panel) shows very little deviation.

introduces two new quantities called first flexion F and second flexion G, which are the third derivative combinations of the potential:

$$F = \frac{1}{2}\partial\partial^{\dagger}\partial\psi = \partial\kappa\partial^{\dagger}\gamma, \tag{3.54}$$

$$G = \frac{1}{2}\partial^{3}\psi = \partial\gamma. \tag{3.55}$$

Wrting this out gives us quantities similar to Eq. 3.45

$$F = [\gamma_{1,1} + \gamma_{2,2}] + i[\gamma_{2,1} - \gamma_{1,2}] = \frac{1}{2}\left[[\psi_{,111} + \psi_{,122}] + i[\psi_{,112} + \psi_{,222}]\right], \tag{3.56}$$

$$G = [\gamma_{1,1} + \gamma_{2,2}] + i[\gamma_{2,1} - \gamma_{1,2}] = \frac{1}{2}\left[[\psi_{,111} - 3\psi_{,122}] + i[3\psi_{,112} - \psi_{,222}]\right]. \tag{3.57}$$

3.3.3 Applications

In principle, every weak lensing approach can only benefit from the additional flexion information, but the implementation of flexion in current lensing reconstructions suffers from difficulties. A big problem is to extract the flexion signal from observations, which is not trivial at all, since also the correct extraction of the shear alone is still under discussion. Also it is not completely clear how to treat this signal in the right way, because also flexion is affected by a mass-sheet-degeneracy problem (see Schneider & Er, 2007). For a recommended review on these issues refer to Melchior (2006). Included there is also a combined shear and flexion analysis for a simulated galaxy cluster lensing scenario, which we present in Fig 3.8. As one can see the results look quite promising and justify the effort to implement flexion in future lensing reconstructions, based on real data.

4 Combining Weak and Strong Lensing for Potential Reconstruction

In this chapter we will give a summary of the basic concepts of our reconstruction method, which combines weak and strong gravitational lensing methods. To justify a newly developed method, one should first look at the current status of observations. While we want to apply our method to current data, what do we gain if it relies on a data quality that has not been reached so far?

4.1 Status of Observations

4.1.1 Ground Based Telescopes

We start with a compilation of state-of-the-art ground based telescopes, which are able to produce high quality data, which is essential for modern astronomy. Of course there are a lot more telescopes around the world, but we want to give a brief overview of the most important ones:

- The **Canada-France-Hawaii-Telescope** (CFHT) is based on Mauna Kea (4200 m), Hawaii. It is operated by the National Research Council Canada, the Centre National de la Recherche Scientifique of France and the University of Hawaii. The optics mainly consist of a 3.6 m main mirror and it is equipped with several scientific instruments. The most important is the MegaPrime/MegaCam optical camera, which covers a huge area of one deg^2. This results in a resolution of $0.187''$ per pixel. The second important instrument is the WIRCam for near infrared, which covers a field of (20 x 20) arcmins^2, with an angular resolution of $0.3''$ per pixel. The telescope became famous because of the CFHT Legacy Survey (CFHTLS), providing a huge observational database, from wide to deep exposures.

- The **Very Large Telescope** (VLT) is located on Cerro Paranal (2600 m), Atacama Desert, Chile. The operator is the European Southern Observatory (ESO). This very impressive telescope consists of four 8.2m main mirrors, in separated enclosures, accompanied by four 1.8 m auxiliary telecopes and two additional survey telescopes. The four main telescopes can be combined via interferometry and thus have the effective area of a 16 m class

Figure 4.1: CFHT, with Keck and Subaru shadows.

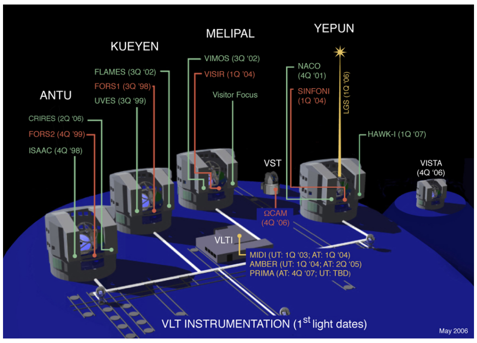


Figure 4.2: Schematic Drawing of VLT and its instruments.

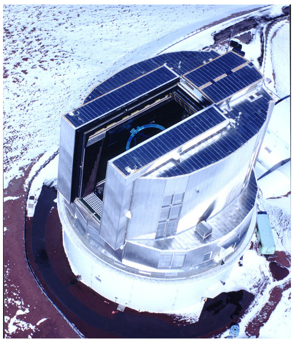


Figure 4.3: The Japanese Subaru telescope.

Figure 4.4: The two Kecks from above. Notice the huge main mirrors.

telescope. There is a huge amount of attached instruments, which we are not willing to list here. One of the most important is FORS1 for optical imaging, which covers an area of (6.8 x 6.8) arcmins^2, with a resolution of 0.25$''$ per pixel.

- Also on Mauna Kea, Hawaii, we find **Subaru**, run by the National Astronomical Observatory of Japan. Equipped with a 8.2 m main mirror, it carries instruments in the optical and infrared band. The Suprime Cam optical camera covers a large field size of (32 x 27) arcmins^2, with a resolution of 0.26$''$ per pixel. The Multiple Object Infrared camera and spectrograph (MOIRCS) covers a (4 x 7) arcmins^2 field with a 0.117$''$ resolution.

- The last major telescope on Mauna Kea is **Keck**, which is the largest optical telescope in the world. It is composed of two 9.8 m main mirrors, in separated enclosures. Attached are multiple instruments for observations in the optical and near infrared band, in combination with an advanced adaptive optics system. Also combined exposures via interferometry are possible.

- We finish our overview of ground based telscopes with the **Large Binocular Telescope** (LBT), located on Mount Graham (3300 m), Arizona, USA. It is operated by a consortium of Institutes from the USA, Italy and Germany [1]. The main optics are two 8.4 m mirrors, mounted in a single enclosure. The most important instrument is the Large Binocular camera, for optical and infrared exposures. It covers a field of (23 x 23) arcmins^2, with an angular resolution of 0.23$''$.

4.1.2 Space Based Telescopes

Besides the classical ground based telescopes, there is also the possibility to use space based telescopes. Because of their very special location, they provide major advantages. First of all, they can also observe in the ultraviolet regime, which cannot be done with ground telescopes, because the atmosphere does not give an observation window in that wavelength regime. Two other advantages related to the absence of the atmosphere, are the far higher angular resolution and the ability to resolve also very faint objects with relatively small mirrors. A more economical advantage is the much higher time of possible usage, because daylight and weather do not play a role in space.

But this also brings us to the main disadvantage of space telescopes, which is the sheer cost of such a mission, which execeeds ground based observatories by several factors.

In the following, we list two space based missions, where one of them is well known and in space for some time and the other is a future project, with the focus on cosmological applications:

[1] MPIA and LSW Heidelberg are highly involved in LBT instrumentation

Figure 4.5: Face on shot of LBT.

Figure 4.6: Picture of the Hubble Space Telescope taken from space.

- The famous **Hubble Space Telescope** (HST) is in a 570 km orbit around Earth and operated by NASA. Installed inside the satellite is a 2.4 m mirror with several attached optical and near infrared instruments. Installed since 2002 is the Advanced Camera for Surveys (ACS), which can be used in different modes. In wide mode it covers a (202 x 202) arcsecs^2 field, with an amazing resolution of $0.049''$ per pixel and in high resolution mode it covers a field of (29.1 x 26.1) arcsecs^2, with an astonishing resolution of $0.026''$ per pixel. Unfortunately not operational anymore is the Wide Field and Planetary Camera 2 (WFPC2), which is responsible for most of the nice pictures in the press.

- To give a preview on future space missions, we also add here the **Dark Universe Explorer** (DUNE), which was object of a pre-study phase by the French space agency (CNEF) and is now propsed to the Cosmic Vision program of the European Space Agency (ESA). The proposal contains many contributions from Europe and the US. It is designed to contain a 1.2 m mirror with a 0.5 deg^2 camera, specialised on the needs of large sky coverage and weak lensing surveys. It will make it possible to obtain tight constraints on cosmological models. Unfortunately the lift-off of this mission is scheduled not before 2015 if it will be approved by ESA.

4.1.3 Lensing Observables

Now we know how to obtain high quality CCD images of a certain object in the sky, but this is not exactly what we want. What we will need for our lensing analysis are weak and strong lensing observables, which are measured background galaxy ellipticities and arc or multiple image positions. The whole process of CCD image reduction and analysis is a highly non-trivial and challenging task and fills many diploma and Ph. D. theses, but we want to give at least a short outline of the methods applied. So assuming that we have a raw CCD image from a telescope camera, the following steps lead to the desired data:

1. Image Detection,
2. Shape Determination,
3. PSF Corrections.

The first step is more or less straightforward, including de-biasing and flat-fielding, which means that one removes the background and thermal dark-current noise of your CCD. Also the removal of cosmic rays and bad pixels is done in this step, which requires to have several frames of the same image, slightly shifted in position. For a final image, these frames are remapped and stacked. After one has obtained the reduced image, one can use well established software packgages, like FOCAS (Jarvis & Tyson, 1981) or SExtractor (Bertin & Arnouts, 1996) to extract and distinguish the objects in the image. This is not easy because objects can overlap and have to be distinguished and separated from stars. Also, weak lensing needs a very high density of background galaxies, which includes very faint objects, where the S/N ratio becomes small.

Having identified the objects, one needs to determine their shapes, which is probably the most difficult part in the analysis-pipeline. Measuring shapes means finding the second moments of the brightness tensor, where the important quadrupole moment is given in Eq. 3.21. The main problem in this measurement is that the shape of the image is distorted by the PSF[2], due to atmospheric seeing or inaccuracies of the telescope. The physical meaning of the PSF is that pointlike sources are smeared out and the mathematical meaning is that the surface brightness before observation $I(\boldsymbol{\theta})$, is convolved with the PSF, $P(\boldsymbol{\theta})$:

$$I^{\text{obs}}(\boldsymbol{\theta}) = \int d^2\vartheta I(\boldsymbol{\vartheta})P(\boldsymbol{\theta} - \boldsymbol{\vartheta}). \tag{4.1}$$

The PSF is usually obtained by looking at reference stars which are assumed to be point sources, and measuring their shapes

The deconvolution of Eq. 4.1 is rather complicated and has been established in several ways. One is the so called KSB-method (Kaiser, Squires, & Broadhurst, 1995) and its variants. Another possibility is the image decomposition into shapelets (Refregier, 2003; Massey & Refregier, 2005; Melchior et al., 2007). We want to point out again that this step in the analysis is crucial, because weak lensing relies on slight distortions which have to be measured with high precision. Compared to ellipticity measurements, the detection of arcs is rather easy, as long as one possibly has high resolution Hubble ACS images. It is usually done by eye, but efficient algorithms are also available (Seidel & Bartelmann, 2007). More complicated is the detection of multiple images, where additional spectroscopic or photometric redshift information is necessary.

4.2 Maximum Likelihood Reconstruction

We present now the main concept of our mass reconstruction method for GCs, which is a so called maximum likelihood reconstruction. We have seen maximum likelihood reconstructions before, namely in Eq. 3.38, and we will use them to combine weak and strong lensing. This approach is quite different from the direct inversion methods in 3.2.1 and works as follows:

We have N measurements y_i which we assume to be Gaussian distributed, with variance σ_i and parameter values x_i. This gives us triples (x_i, y_i, σ_i). For these, we try to find a function which approximates the measured values as well as possible. Such a function $g(x_i)$ is called a *fitting function*. A convenient definiton of a good fit is the minimum of the mean square deviations χ^2, weighted with the corresponding datapoint variances:

$$\chi^2 \equiv \sum_{i=1}^{N} \frac{(y_i - g(x_i))^2}{\sigma_i^2}. \tag{4.2}$$

This assumes that datapoints are statistically independent. The straightforward generalisation of this case, with statistically dependent datapoints, is the following definition:

$$\chi^2 \equiv \sum_{i,j}^{N} (y_i - g(x_i)) \mathcal{C}_{ij}^{-1} (y_j - g(x_j)), \tag{4.3}$$

with the *covariance matrix* $\mathcal{C}_{ij}$. Now the difficulty is to find a reasonable representation of the fitting funtion $g(x_i)$, which was done in 3.2.2 by a set of parameters. In our reconstruction, we will do this in a purely nonparametric way.

[2] Point-Spread-Function, crucial quantity for every telescope, which has to be known as good as posssible

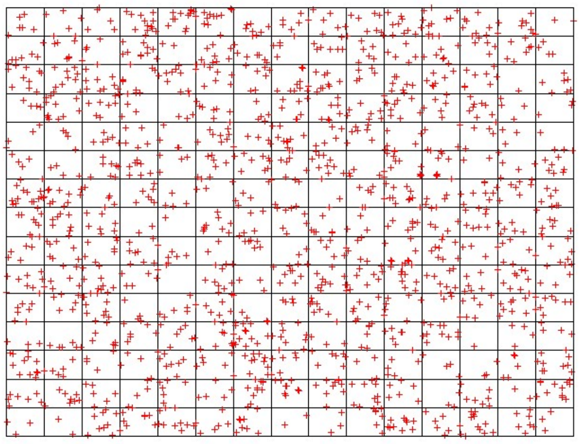


Figure 4.7: Dividing the observed galaxy-field into a grid. The red crosses are possible galaxy positions with ellipticity measurement. The grid resolution shown here is 15x15 pixels

4.2.1 Dividing the Field into a Grid

As a starting point, we take the output from a galaxy-shape and strong-lensing analysis. This will be a list of the two ellipticity values and the strongly lensed images, at certain angular positions.

As we pointed out before, a single galaxy ellipticity measurement is useless for a weak lensing signal, due to the intrinsic ellipticity of the sources. We avoid this by averaging over a certain number of galaxies to get one datapoint. Afterwards, we divide the observed galaxy cluster field into a grid of N subcells and assign to each cell an ellipticity value due to the averaging process, and obtain N datapoints for our χ^2-minimisation. Because of that, the following reconstruction will highly depend on the resolution of this grid.

Furthermore, if a number of M gridcells, which we will call from now on pixels, is covered by strongly lensed images, we will gain an additional number of M constraints for the reconstruction. What we have to do now, is to describe our fitting function g in such a way, that it is defined on the grid described above.

4.2.2 Resolution Problems

Before we can start to define the χ^2-functions for our joint lensing reconstruction, we have to address two issues, concerning the resolution of our reconstruction grid.

The first aspect is related to the weak lensing regime. So let us do a little back-of-the-envelope calculation.

- We want to average over ≈ 10 galaxies per pixel.
- Ground based observations today can deliver $\rho_{\mathrm{gal}} \sim 35$ background galaxies per arcmin^2.
- The typical angular field size of a galaxy cluster is $A_{\mathrm{field}} = 40\ \ \mathrm{arcmin}^2$.
- The field is assumed to be quadratic and we divide it into N_{pix} equal quadratic pixels.

$$\begin{aligned} A_{\mathrm{pix}} \cdot \rho_{\mathrm{gal}} &\overset{!}{=} 10 \\ A_{\mathrm{field}} &= N_{\mathrm{pix}} \cdot A_{\mathrm{pix}} \\ \Rightarrow \quad N_{\mathrm{pix}} &= \frac{1}{10} \cdot A_{\mathrm{field}} \cdot \rho_{\mathrm{gal}} = 126. \end{aligned}$$

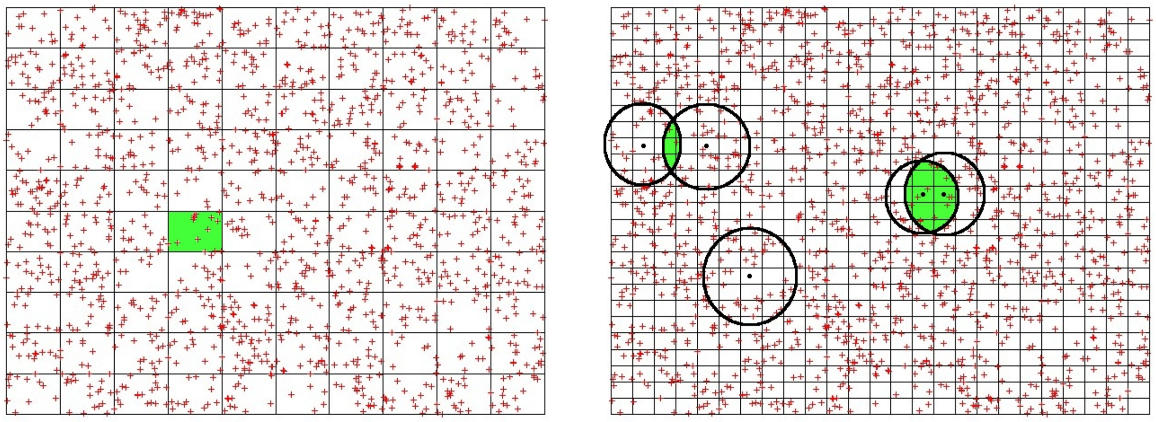


Figure 4.8: Left panel: Very coarse 10x10 pixel grid. In greenan example for a pixel with 8 included galaxies is shown. One can also see the irregular distribution of galaxies with several voids. This problem is fixed in the right panel, where a 20x20 pixel grid is presented. The circles show the adaptive averaging for each individuel pixel, which causes overlap, illustrated in green.

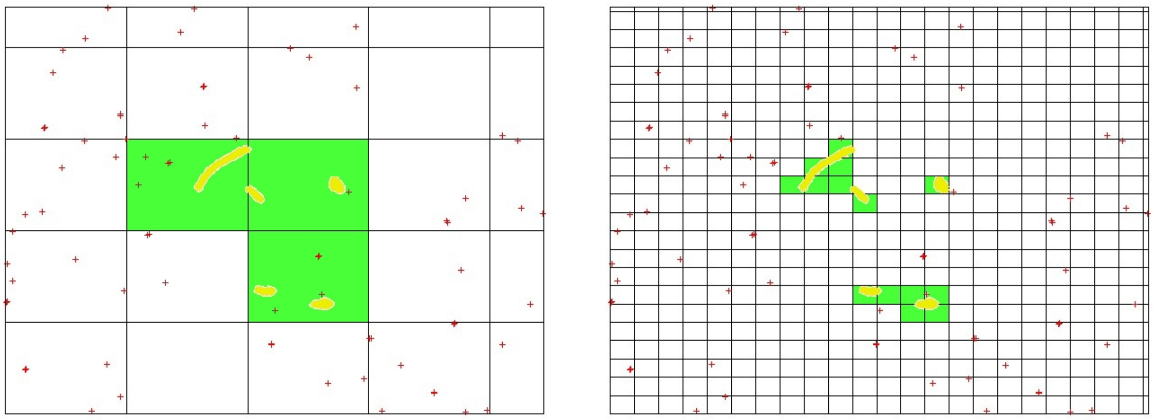


Figure 4.9: Zoom on the inner part of the cluster. As one can see in the left panel, a coarse 20x20 original resolution for the whole field, is not able to resolve arc positions. On the right panel a 100x100 pixel resolution with respect to the whole field is able to follow the shape of the arcs.

That gives us the possibility to divide the field into a 11x11 grid, which is of course not a convenient resolution if one wants to see substructures in the cluster. Another problem arises from the fact that in a realistic field, the galaxies are not homogeneously distributed, so by applying a homogeneous grid, it is likely to have pixels with less than 10 or even without any galaxy contained. We solve both problems with an *adaptive averaging procedure*, which means that we average within circles around the pixel centre. The radius of these circles is stepwise increased until the circle contains the wanted number of galaxies. Different pixels will need different radii, depending on the galaxy density in that area of the field. If one does this on a high resolution grid, one has to take into account that ellipticity datapoints can be correlated, because they share galaxies during the averaging process (see Fig.4.8).

The second aspect affects the strong lensing regime, especially arc positions. Because strong lensing is confined to much smaller scales, a lot of position information is lost, if one uses strong lensing constraints at the same resolution as weak lensing. This requires a refined grid at the cluster center which is capable of resolving the exact shape of the arcs (see Fig. 4.9).

4.2.3 Defining χ^2-Functions

We now address the most important ingredient of our mass reconstruction, the definition of the χ^2-functions. As we mentioned before, we want to perform a non-parametric reconstruction, which means that we do not want to have any model-dependent parameters in the χ^2-minimisation. The key to do so is the lensing potential. By relating a datapoint on the grid directly to the lensing potential ψ, evaluated at this specific grid point, the values of ψ at every grid position become the quantities with respect to which we minimise the χ^2-function. In other words, we are searching for a discrete representation of the lensing potential, which is most likely to cause the observed effects.

Because the weak and strong lensing constraints are independent of each other but reflect the same underlying potential, the overall χ^2 becomes the sum of two contributions,

$$\chi^2(\psi) = \chi^2_w(\psi) + \chi^2_s(\psi). \tag{4.4}$$

The reason why we were choosing the lensing potential as a reconstruction quantity is that it is a much smoother quantity than e.g. the convergence, which makes it much more resistant against noise.

The reconstruction result is then obtained by minimising the χ^2 with respect to the potential on every grid position l, as minimisation parameters

$$\frac{\partial \chi^2(\psi)}{\partial \psi_l} = \frac{\partial \chi^2_w(\psi)}{\partial \psi_l} + \frac{\partial \chi^2_s(\psi)}{\partial \psi_l} \overset{!}{=} 0 \tag{4.5}$$

Let us first have a look at the weak lensing term. We saw in Eq. 3.29 what the expectation value of the ellipticity looks like, so we define

$$\chi^2_w = \sum_{i,j} (\varepsilon_k - g(\psi))_i \mathcal{C}^{-1}_{ij} (\varepsilon_k - g(\psi))_j, \tag{4.6}$$

and in the case of $|g| \leq 1$, we get:

$$\chi^2_w(\psi) = \left(\varepsilon - \frac{Z(z)\gamma(\psi)}{1 - Z(z)\kappa(\psi)}\right)_i \mathcal{C}^{-1}_{ij} \left(\varepsilon - \frac{Z(z)\gamma(\psi)}{1 - Z(z)\kappa(\psi)}\right)_j, \tag{4.7}$$

where we used Einstein's sum convention as we will do from now on without explicitely mentioning it again. The case $|g| > 1$ is not interresting in our reconstruction since it would only affect very few pixels in the reconstruction grid.

A quantity that we have not discussed yet is the covariance matrix $\mathcal{C}_{ij}$. The standard deviation σ_i, for each weak lensing pixel, is obtained during the averaging as the standard deviation of the mean. This standard deviation is composed of three different parts

$$\sigma = \sigma_{\text{int}} + \sigma_{\text{sys}} + \sigma_l, \tag{4.8}$$

with the noise due to intrinsic ellipticty σ_{int}, systematic error (measurement noise) σ_{sys} and lensing noise σ_l. Under lensing noise we understand the noise which arises from the fact that the galaxies over which we average, are spatially separated, so the properties of the lens change inside a pixel. Here one can also see, that the radii of the averaging circles should not become too large, otherwise the correlation in the lensing signal tends to be lost.

We assume that for a sample of sufficient size, the intrinsic ellipticity noise tends to zero, which leaves the lensing and the systematic noise. Starting from the definition of the covariance matrix:

$$\mathcal{C}_{ij} = \langle (x_i - \langle x_i \rangle) \, (x_j - \langle x_j \rangle) \rangle, \tag{4.9}$$

and by using the fact that the correlation between two pixels due to lensing is proportional to the overlap area shown in Fig. 4.8, we arrive at

$$\mathcal{C}_{ij} = w_{ij}\sigma_i\sigma_j. \tag{4.10}$$

We approximate the overlap area by the number of galaxies shared by two pixels which gives the weighting factor

$$w_{ij} = \frac{N_{ij}}{1/2(N_i + N_j)}. \tag{4.11}$$

N_i and N_j are the sizes of each pixel-averaging sample and N_{ij} is the number of galaxies that belong to both samples. This factor has exactly the wanted properties, i.e. it is unity on the diagonal and vanishes for completely separated and uncorrelated pixels.

The strong lensing term looks much simpler. We have seen in Chapter 3 that the critical curve of a lens is defined as the line where the determinant of the Jacobian vanishes. So given that we know at which pixels the critical curve is located, we define

$$\chi_s^2(\psi) = \frac{|\mathrm{det}A(\psi)|_i^2}{\sigma_{is}^2} = \frac{|(1 - Z(z)\kappa(\psi))^2 - |Z(z)\gamma(\psi)|^2|_i^2}{\sigma_{is}^2}, \tag{4.12}$$

where the error estimate σ_s is mainly related to the pixelization of our grid, which determines the inaccuracy of the position of the critical curve. We approximate this uncertainty to first order with the help of the Einstein angle (see Cacciato et al., 2006).

$$\sigma_s \approx \left.\frac{\partial \mathrm{det}A}{\partial \theta}\right|_{\theta_c} \delta\theta \approx \frac{\delta\theta}{\theta_E}, \tag{4.13}$$

with the angular inaccuracy due to pixelization $\delta\theta$.

Strictly speaking, the formula above only holds for an isothermal sphere, but nevertheless it gives us a good approximation for the noise in the critical curve position.

To evaluate Eq. 4.5 we have to connect convergence and shear to the grid values of the potential, which we will do in Chapter 6.

4.2.4 Advantages of our Method

At the end of this section we want to emphasize that the main advantage of the maximum likelihood approach is its enourmous flexibility. In principle, one can use every observable constraint and connect it in some way to the lensing potential. Of course this is not limited to lensing. One simply has to define separated χ^2-functions which are statistically independent, and minimise their sum with respect to the potential. Possible other constraints than those used in our work are:

- Multiple image systems (see Bradač et al., 2005),
- Flexion (see Leonard et al., 2007),
- Magnification,
- Galaxy dynamics (see Sand et al., 2007),
- X-Ray surface brightness (see Lemze et al., 2007).

Future improvements of our method should include as many of these constraints as possible.

4.3 The Result: Lensing Potential

As we pointed out before, we are reconstructing the lensing potential ψ, because it is more stable against noise and we can directly relate it to the lensing observables in an easy way (see Eq. 3.13). The potential itself is not a direct observable, but linear combinations of its derivatives, so we always can add a constant, a linear funtion in $\boldsymbol{\theta}$, or a harmonic function. Also the lensing potential is affected by the mass-sheet-degeneracy, because the quantities, from which we obtain it are. Recall, that not only the reduced shear, but also the critical curve is invariant under the mass-sheet transformation. In the final analysis, this allows us the following transformation of the potential (Bradač et al., 2004):

$$\psi(\boldsymbol{\theta}, z) \rightarrow \psi^{'}(\boldsymbol{\theta}, z) = \frac{1-\lambda}{2}\boldsymbol{\theta}^2 + \lambda\psi(\boldsymbol{\theta}, z), \tag{4.14}$$

where λ is an arbitrary constant.
This is the reason why the resulting potential on every grid position ψ_j might look shifted or distorted. In fact, this is not a problem at all because we only need its curvature.

4.3.1 Convergence Maps

We obtain a physically meaningful quantity like the convergence by simply applying the Laplacian to the potential. To obtain the shear, we have to apply the other combinations of second derivatives to the potential. The convergence is a reasonable quantity, because it shows us in an intuitive way the mass distribution of the cluster through the surface mass density.
Unfortunately, the convergence is again affected by the mass-sheet-degeneracy. It can be transformed like

$$\kappa(\boldsymbol{\theta}, z) \rightarrow \kappa^{'}(\boldsymbol{\theta}, z) = (1-\lambda) + \lambda\kappa(\boldsymbol{\theta}, z). \tag{4.15}$$

So how to fix the convergence? First of all, it has the meaning of a density, so it should not be negative. Our reconstruction method does not take this into account, so it is possible that we will find results with negative convergence values. This can be transformed, using Eq. 4.15. Furthermore, if our observed field is sufficiently large, we would assume that $\kappa \rightarrow 0$ for the outskirts of our field, so we can use Eq. 4.15 again, for normalisation. A more elaborate method is the use of a quantity in the reconstruction, which is not invariant under the mass sheet transformation and determines potential and convergence. One possibility would be the use of inverse magnification

$$R = \frac{1}{\mu} = (1-\kappa)^2 - \gamma^2 \approx 1 - 2\kappa, \tag{4.16}$$

embedded in the maximum likelihood approach

$$\chi^2_{\mathrm{mag}}(\psi) = \frac{(R - R(\psi))^2_i}{\sigma^2_{Ri}}. \tag{4.17}$$

Of course in order to use this method, one has to provide a good idea how to measure a magnification signal:

$$\mu = \frac{d^2\Omega_{\mathrm{lensed}}}{d^2\Omega_{\mathrm{unlensed}}}. \tag{4.18}$$

Another approach to solve the transformation problem was proposed by Bradač et al. (2004, 2005), who used the knowledge of the redshift distribution of the sources.

4.3.2 Mass estimates

Once we have the convergence possibly transformed with the methods shown above, it is quite easy to get a mass estimate for our lens. Convergence and surface mass density are connected by

$$\Sigma(D_d\boldsymbol{\theta}) = \frac{c^2}{4\pi G}\frac{D_s}{D_d D_{ds}}\kappa(\boldsymbol{\theta}). \tag{4.19}$$

If we have the redshift of the lens, we also know the area of one pixel and if we have at least the mean redshift of the sources, we can calculate the surface mass density, which gives us the total mass of the cluster after summing over the whole grid.

5 Implementation I: General Considerations and Input Data Handling

After the theoretical outline of the basic ideas and concepts of the joint weak and strong lensing reconstruction method, we now focus on the concrete numerical implementation. In this and the following chapter, we will show that the method is implemented in a self-contained package[1]. This allows us to perform a complete reconstruction, from input data to final the result, without "by hand" adjustments.

The complete galaxy cluster reconstruction can be split into two parts. The first part, which is addressed in this chapter, is the preparation of the weak and strong lensing input data. This means obtaining the grid data points, described in 4.2.3, which can be used as input of the second part, the reconstruction itself. This will be addressed in Chapter 6.

5.1 General Implementation Framework

5.1.1 Code Language

We decided to write the code in C++ for several reasons. C++ has become a quasi-standard in the scientific community, more and more replacing Fortran. Of course it is not a good reason to write one's own code in a certain language just because everybody else is writing in that language. So we will point out the main advantages of our choice.

First of all C++ is object oriented. This makes it flexible and allows to extend existing code in an easy and convenient way. Also, the usage of classes and objects disentangles the code and does not produce long and confusing "Spaghetti-Code", like in procedural languages. As a result it is much easier for an external user to understand, apply and develop the code.

Another issue is the availability of external packages and libraries. One wants to be able, to use well known standards and algorithms, for which reliable routines have already been implemented. This is the case for C++, because of its broad distribution. For example, this can be seen very well, if one thinks of graphical user interfaces, where powerful interfaces exist for a long time. One of the most important advantages of C++ over Fortran, besides its object-orientation, is that C++ is also very common outside the physics community, where one will almost not find Fortran at all. This makes it also possible to interact with other branches, like applied mathematics, for example.

Now that we talked about the advantages, we should also mention the disadavantages of a C++ implementation. Unfortunately a lot of mathematical concepts are not implemented from the beginning, like vectors, matrices and linear algebra operations. To use them, one has to rely on external packages. Another difficulty is the visualisation of data and results, where one often has to use commerical, external packages like IDL or MatLab, to get a satisfactory result. A fast and easy to use OpenSource alternative is Python[2] and the corresponding plotting libraries, like Matplotlib[3].

[1] The source code is available in our repository: https://www.ita.uni-heidelberg.de/svn/sawlens

[2] http://www.python.org

[3] http://matplotlib.sourceforge.net/

5.1.2 External Packages

As we pointed out, there exists a huge variety of external, freely available libraries besides the comprehensive C Standard Library. In our implementation, we make use of several external packages with different functions, that we want to list and describe here:

- The *GNU Scientific Library*[4] (GSL), is a very substantial compilation of mathematical routines and tools, for scientific applications. It covers a whole bunch of different areas, like differential equations, polynomials, minimization methods, numerical integration and Fourier transformation. It is very well documented and easy to use. In our code we use GSL for vector and matrix handling, random number generation, statistics and for solving linear systems of equations.
- *Numerical Recipes*[5] (NR) is a series of books providing numerical solutions for mathematical problems. It is available for several programming languages, including Pascal, C, C++ and Fortran. The advantage of NR is its availability in printed form, which makes it worth reading if one is interested in numerics. The provided routines seem to be outperformed by equivalent routines from GSL. In our code we use NR for two dimensional, bicubic spline interpolation.
- The *Linear Algebra Package*[6] (LAPACK) was originally written in Fortran 77, but is also available in C and C++. It is highly optimized for linear algebra operations and in this sector still the best package available. In our code we use the LAPACK routines, contained in the Automatically Tuned Linear Algebra Software[7] (ATLAS), for high-dimension matrix inversion and linear system solution.
- The common standard in Astronomy for data transport is the *Flexible Image Transport System* (FITS). We also make wide use of this data format to store information on disk during the reconstruction process. The library which is needed by C++ to read and write files in the FITS-format is *CFITSIO*[8] and the C++ wrapper CCfits[9].
- To make the execution of our code as fast as possible, we also implemented a parallel version, which means that the code is not running on a single processor, but on several processors in parallel. The communication between the processors is done by the *Message Passing Interface*[10] (MPI), which is the standard interface for parallel implementation and available in Fortran and C/C++. Also the N-Body-Code Gadget that we mentioned before uses MPI.

These are the packages our code uses during the reconstruction. We want to point out that all these packages are in the public domain and freely available. One exception is Numerical Recipes, which could be easily replaced by GSL routines. For visualisation we used different solutions like fitsview, DS9, gnuplot, or the Python module Matplotlib.

5.1.3 Runtime and Requirements

Having talked about the software requirements, we should also discuss the requirements on hardware, especially memory and CPU speed.
The largest objects our code has to handle are matrices of dimension $\sim$ 1000x1000. Normally, no

[4] http://www.gnu.org/software/gsl
[5] http://www.nr.com
[6] http://www.netlib.org/lapack
[7] http://www.netlib.org/atlas
[8] http://heasarc.gsfc.nasa.gov/fitsio
[9] http://heasarc.gsfc.nasa.gov/fitsio/CCfits/
[10] http://www.mcs.anl.gov/mpi

more than 10 of these matrices are needed simultaneously in the main memory. While we are calculating with double precision, this means a memory need of less than 100MB, which is not a problem on current machines. A much bigger issue is computation time. We will see later that taking into account all pixel correlations in the reconstruction process, which is necessary for input fields with low galaxy density, becomes extremely expensive with respect to computation time. While calculating with adequate accuracy and resolution, this can take several days on a normal desktop machine. This can be avoided, by ignoring pixel correlations or working at lower resolution. Both mean loosing accuracy, but the runtime drops down to $\sim$ 1h for a single galaxy cluster. The more appropriate way to avoid long runtime is using the parallel version of the code. On a 16 processor machine, the runtime for a complete reconstruction of a single galaxy cluster at maximum resolution is again at the level of $\sim$ 1h.

5.2 Preparing Ellipticity Field Data

5.2.1 Input Data

We will now focus on the analysis of the weak lensing data. This data consists only of a plain-text file, which lists the position and ellipticity measurement for every distorted background galaxy. A cutout of such a file can be seen in List. 5.1. One should mention here that the coordinates are arbitrary as long as they are consistent, which means that all constraints, used as input for the code, are related to the same coordinate system. In this work we will give the coordinates in arcseconds relative to the brightest cluster galaxy (BCG).

```
  rec      dec     ellip1    ellip2
 −75.7    170.4   −0.1893    0.3443
 −65.3    199.1    0.0559    0.0728
 −79.8    197.2    0.3716    0.1883
 −27.0   −153.7    0.0726    0.0158
 −46.6    −75.4    0.0917   −0.4271
 −31.6    −64.8   −0.0202   −0.2538
 −23.6     −6.0    0.2162    0.4960
 −46.3     −0.9   −0.3442    0.1080
 −27.7     17.5    0.2529    0.0060
 −18.1    105.6   −0.1647    0.4169
```

Listing 5.1: An example for input data, for a sample of 10 galaxies

The code reads and saves internally the positions and ellipticities for all galaxies. While constructing the corresponding C++ class, the user has to give the resolution in terms of the number of pixels that the reconstruction grid should have. Afterwards, all the important field information, like minimum and maximum coordinate values, field size, galaxy density and the pixel size are calculated in the chosen resolution. The final grid is then build up, by calculating the coordinates of each pixel center, in the coordinates, given by the input file.

5.2.2 Adaptive Averaging

With the information of the pixel-coordinates, the user can start the adaptive averaging process by giving the number of galaxies that an averaging-sample for a single pixel should contain. For all pixels, the code checks how many galaxies lie in a certain radius around the pixel center. The initial radius is given by the pixel size and is then increased stepwise until the given galaxy number is reached. The step size of the increment can be defined by the user and controls accuracy and speed of the process. Once the final number of galaxies is reached, the average and standard deviation values of the ellipticity sample are calculated and stored. Of course this is done for both ellipticity components. An example of the result in different grid resolutions can be seen in Fig. 5.1.

5.2.3 The Covariance-Matrix

To derive the covariance matrix as described in 4.2.3, the code remembers for each individual pixel which galaxies were used in the averaged sample. As a result it can be determined easily how many galaxies two pixels share and visualised in form of a matrix (see Fig. 5.2). For a 22x22 grid, this will be a 484x484 matrix, where each row-column combination represents the connection between two pixels. Of course this matrix is symmetric, and on the diagonal we can see how many galaxies where used during averaging for a specific pixel. On the number of secondary diagonals one can see the grade of correlation. In the example of Fig. 5.2, each pixel is correlated with its two nearest neighbours. We obtain the final covariance matrix with Eq. 4.9, because we can now calculate the weighting factor w_{ij} (see Eq. 4.11). This procedure has to be done for both ellipticity components and the resulting covariance matrices have to be inverted. At this point we use LAPACK, because inverting a matrix of dimension $\sim$ 1000x1000 is not a simple task. LAPACK does this fast and reliably. The resulting inverse of the covariance matrix, can be seen in Fig. 5.2. As an additional example, we also add the case of a much coarser grid of resolution 12x12. The corresponding matrices have dimension 144x144 and one can clearly see that the correlations become much weaker. In fact only direct neighbours are correlated. This is easy to understand, because the pixel size is larger and the overlap of the averaging circles is much smaller. Of course this is only true, if the number of galaxies in a sample considered sufficient is kept constant.

5.2.4 Output Data

As a result of the ellipticity field preparation, the code writes the complete results into a FITS file, which contains:

- Average ellipticity for both components and every pixel in the form of a matrix;
- Modulus of complex ellipticity;
- Standard deviation for both components and all pixels;
- The galaxies assigned to a pixel in form of a ”galaxy number” x ”pixel number” punch-card matrix;
- The matrix which indicates the number of galaxies shared by two pixels;
- Both inverses of the covariance matrices;

Furthermore, all relevant field information, like the field size, the pixel size and the number of averaged galaxies per pixel, are stored separately.

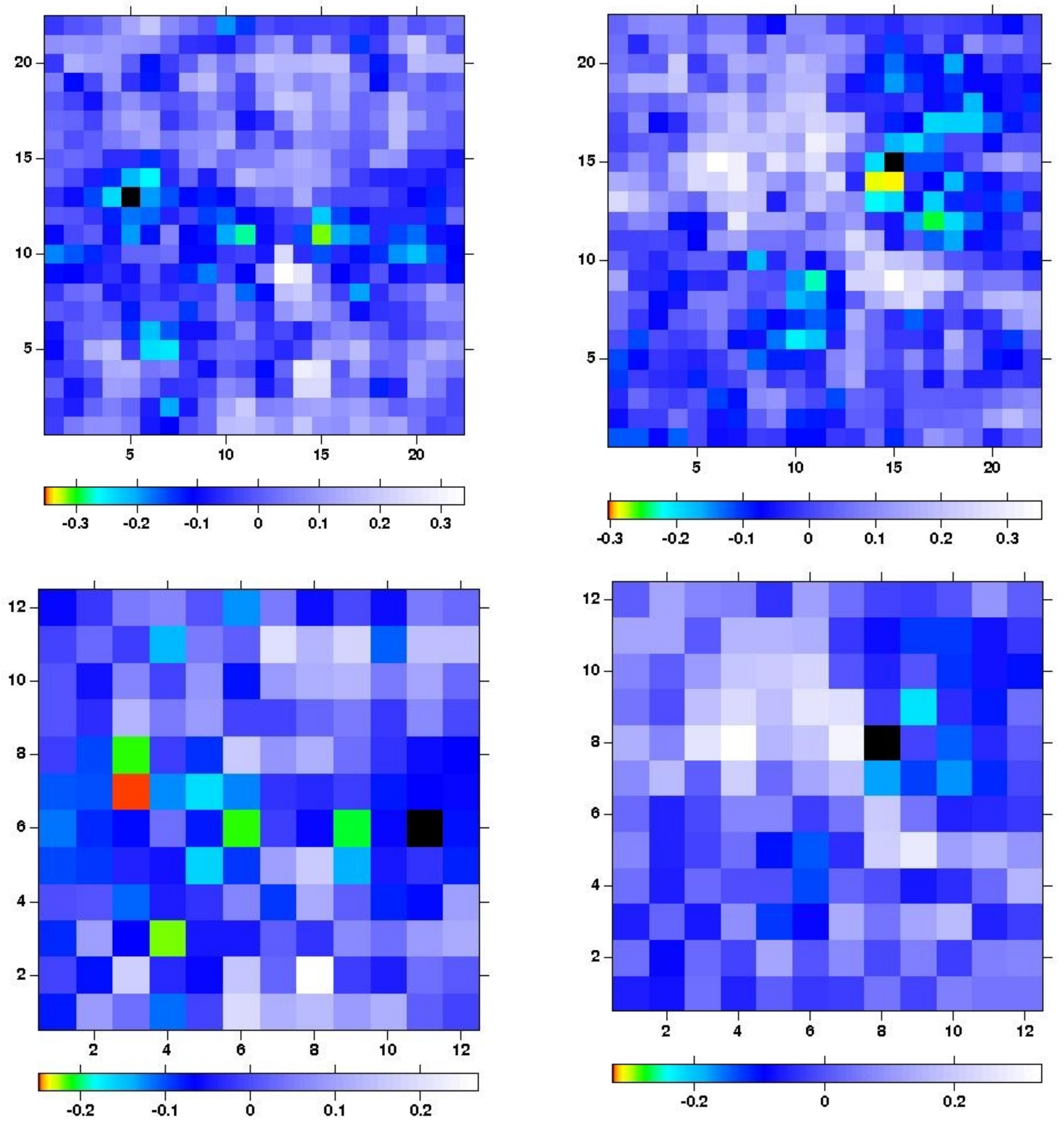


Figure 5.1: Result of the adaptive averaging procedure. Top panels show a grid resolution of 22x22, bottom panels a resolution of 12x12. Left panels show the first ellipticity component , right panels show the second component.

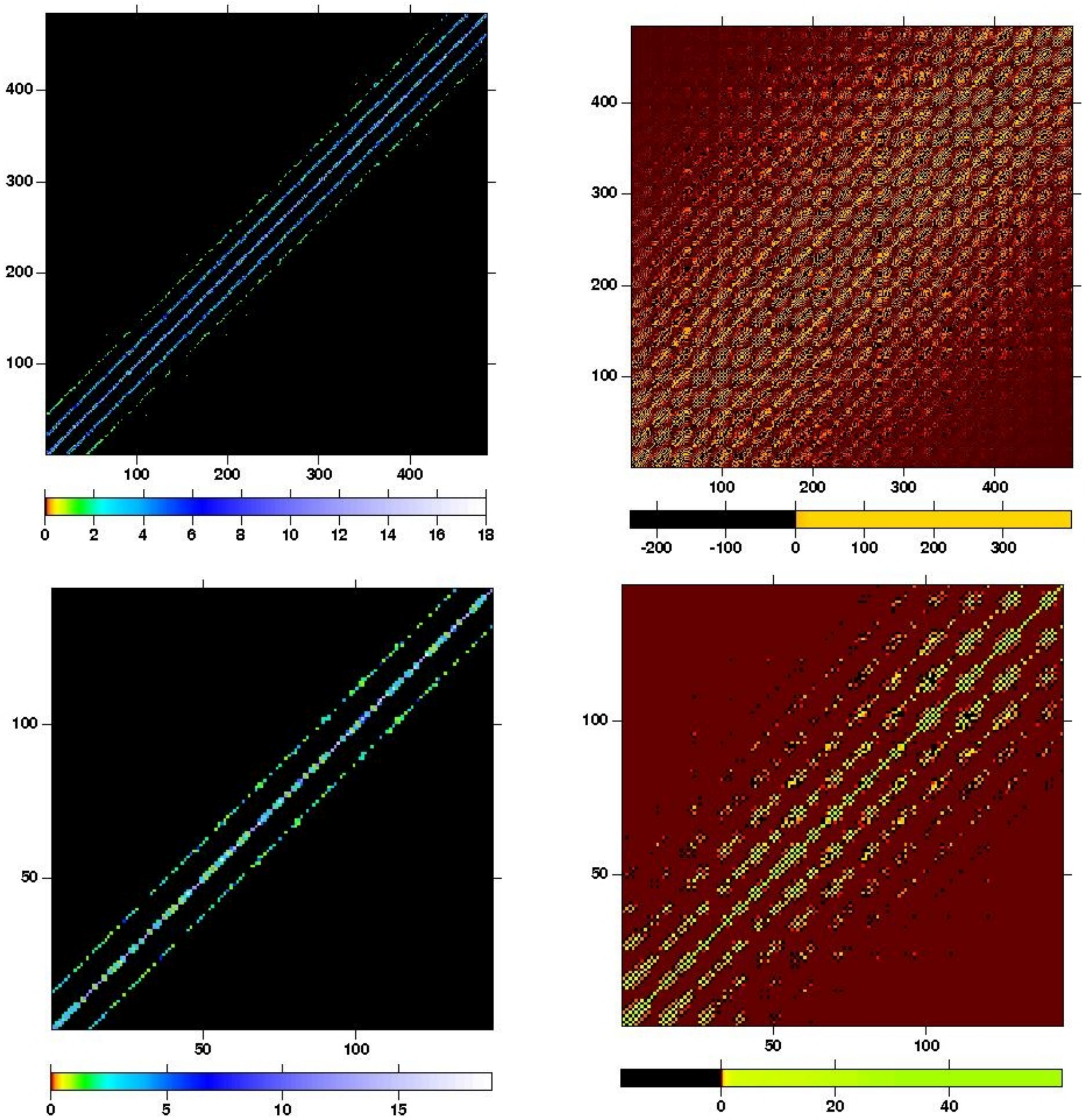


Figure 5.2: The figures on the right panels show the number of galaxies that two pixel have in common. One can clearly see the decrease when leaving the diagonal area. One can also see the decrease in the correlation when one goes from 22x22 (top) to 12x12 (bottom) resolution. The figures on the right show the corresponding inverse of the covariance matrix on high (top) and low (bottom) resolution.

5.3 Preparing Critical Curve Data

5.3.1 Approximation of Critical Curve positions

Before we show how to handle the strong lensing data, we have to deal with the fact that we cannot observe the critical curves directly. What we need is a good approximation for the position of the critical curve, which is given by arc positions that we can observe very well. We show in Fig. 5.3 that arcs follow the position of the critical curves, as long as the resolution of the grid is not extremely high. But even on high resolution, the difference in pixel-position between arc and critical curve is at most two pixels, and Cacciato et al. (2006) showed that deviations of this size do not affect the reconstruction significantly. A more severe problem is that not everywhere around a critical curve, arcs are forming, so we will not be able to trace the complete curve. It will be the task of future work to deal with this issue more accurately, including high resolution Hubble ACS data, which shows the strong lensing features much more clearly.

Another possibility would be to rely on critical-curve reconstructions from a parametric strong lensing analysis and feed that critical curve into the code. The drawback of this approach is that one gives up the completely non-parametric reconstruction by using profile assumptions in the critical curve determination.

Once decided how to treat the critical curves, the input file will be again just a plain text file, with a series of position coordinates of arcs or the critical curve estimation. Of course, one has to be careful that the coordinates match with the coordinates used in the weak lensing data.

5.3.2 Arcs on Different Resolution

Once the critical curve positions are read by the code, it depends again on the chosen resolution how this is translated onto the grid. In the case of the critical curves, this translation is much easier than in the weak lensing case because we only have to decide if a pixel of the grid is crossed by a critical curve or not. This is done by a matrix which represents the grid and has two possible values which declare a pixel as critical curve member or not. Two examples on different resolutions are given in Fig. 5.4.

5.3.3 Output Data

In addition to the results from the ellipticity field analysis, the output FITS file is now extended by a matrix which contains the information if a pixel is part of the critical curve. Also, not significant pixels are removed, if to few of the input data points fall into that pixel. This threshold can be set (e.g. 1% of the overall input coordinates).

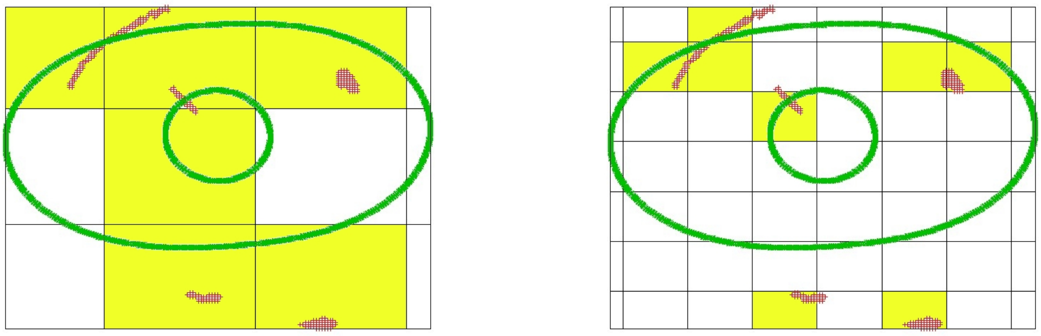


Figure 5.3: Arc position and estimated position of the critical curves. Left panel shows 32x32 grid resolution where one can see no deviation. Right panel shows 75x75 where one sees that the deviation in position is still small. The critical curves were estimated by a parametric strong lensing reconstruction from Comerford et al. (2006).

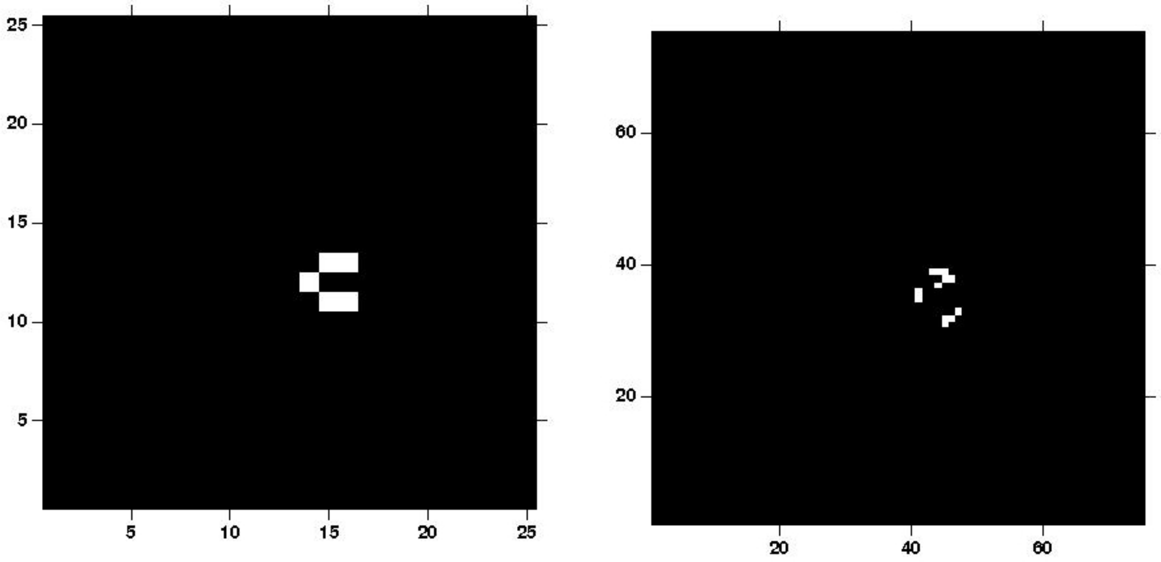


Figure 5.4: The output matrix of the critical curve data preparation. Left panel shows a realistic weak lensing resolution of 26x26 pixels. Right panel shows a realistic strong lensing resolution of 75x75 pixels, which can follow the arc positions.

6 Implementation II: Reconstruction Routines

In this chapter we will explain how the reconstruction itself works, that is minimising the χ^2-Function (Eq. 4.5) and obtaining the lensing potential. This can be done in a non-parametric way with our grid approach, also allowing the use of numerical techniques keeping the computation time at an agreeable level.

6.1 Grid Methods

6.1.1 The Missing Link

Let us briefly summarise what we have achieved so far. We gave the theoretical framework of the reconstruction in section 4.2 and obtained the datapoints for the reconstruction in chapter 5. What we have to do now is to combine both aspects and implement the method numerically. We defined the χ^2-functions in section 4.2.3 and we already pointed out that we want to use the values of the lensing potential values on every grid point as minimisation parameters. The fitting function in the case of weak lensing is the reduced shear:

$$g(z, \psi) = \frac{Z(z)\gamma(\psi)}{1 - Z(z)\kappa(\psi)}, \tag{6.1}$$

and in the case of strong lensing the Determinant of the Jacobian:

$$\det\mathcal{A}(z, \psi) = (1 - Z(z)\kappa(\psi))^2 - |Z(z)\gamma(\psi)|^2. \tag{6.2}$$

In order to minimise the resulting χ^2-function just with respect to the lensing potential, we have to write the convergence and the shear in terms of the lensing potential. The relation between these quantitities is

$$\kappa(\psi) = \frac{1}{2}(\psi_{,11} + \psi_{,22}), \tag{6.3}$$

$$\gamma_1(\psi) = \frac{1}{2}(\psi_{,11} - \psi_{,22}), \tag{6.4}$$

$$\gamma_2(\psi) = \psi_{,12}. \tag{6.5}$$

Using a well known approach in numerical mathematics to solve the Laplace-equation (see Rannacher, 2006), we approximate the second derivatives of the potential with *finite differencing techniques*.

6.1.2 Finite Differences

Hence, chosen a certain grid resolution, we want to reconstruct the potential for every grid position (x, y), $\;x, y \in \mathbb{N}$. First, we enumerate these grid positions, line by line. (See Fig. 6.1)
Afterwards, we use the central difference quotients, to find a discrete representation of functions, which approximate the second derivatives at a certain grid position. This is called finite

1 (1,1)	**2** (1,2)	**3** (1,3)	**4** (1,4)	**5** (1,5)
6 (2,1)	**7** (2,2)	**8** (2,3)	**9** (2,4)	**10** (2,5)
11 (3,1)	**12** (3,2)	**13** (3,3)	**14** (3,4)	**15** (3,5)
16 (4,1)	**17** (4,2)	**18** (4,3)	**19** (4,4)	**20** (4,5)
21 (5,1)	**22** (5,2)	**23** (5,3)	**24** (5,4)	**25** (5,5)

Figure 6.1: Possible grid positions (x,y) on a 5x5 grid, enumerated line by line.

differencing. The approximation functions are given by:

$$\begin{aligned}\kappa(x,y) \approx h^{-2}\Big\{ & -\frac{2}{3}\psi(x,y) - \frac{1}{6}\psi(x-1,y) - \frac{1}{6}\psi(x+1,y) \\ & -\frac{1}{6}\psi(x,y-1) + \frac{1}{3}\psi(x-1,y-1) + \frac{1}{3}\psi(x+1,y-1) \\ & -\frac{1}{6}\psi(x,y+1) + \frac{1}{3}\psi(x-1,y+1) + \frac{1}{3}\psi(x+1,y+1)\Big\},\end{aligned} \tag{6.6}$$

$$\gamma_1(x,y) \approx h^{-2}\left\{\frac{1}{2}\psi(x-1,y) + \frac{1}{2}\psi(x+1,y) - \frac{1}{2}\psi(x,y-1) - \frac{1}{2}\psi(x,y+1)\right\}, \tag{6.7}$$

$$\begin{aligned}\gamma_2(x,y) \approx h^{-2}\Big\{ & \psi(x,y) - \frac{1}{2}\psi(x-1,y) - \frac{1}{2}\psi(x+1,y) \\ & -\frac{1}{2}\psi(x,y-1) - \frac{1}{2}\psi(x,y+1) + \frac{1}{2}\psi(x-1,y-1) + \frac{1}{2}\psi(x+1,y+1\Big\},\end{aligned} \tag{6.8}$$

where h is the sidelength of a pixel. Please note that one has to use different finite difference schemes at the edges and borders of the grid, because not every needed pixel still lies within the grid. The full schemes for convergence and shear are shown in Fig. 6.2 and 6.3.

With the enumeration shown in Fig. 6.1, we can write the discrete potential ψ on a (n x m) grid, $n, m \in \mathbb{N}$, as a vector $\psi \in \mathbb{R}^{nm}$. What becomes slightly more complicated now is addressing the correct positions in the vector. Of course, direct left and right grid-point neighbours are just separated by -1 and +1 positions in the vector, respectively. But top and bottom neighbours are now separated by -n and +n positions, respectively. The advantage of this enumeration is that the finite differencing becomes a simple matrix multiplication:

$$\kappa_i = \mathcal{K}_{ij}\psi_j \tag{6.9}$$

$$\gamma_i^1 = \mathcal{G}_{ij}^1\psi_j \tag{6.10}$$

$$\gamma_i^2 = \mathcal{G}_{ij}^2\psi_j \tag{6.11}$$

Here $\mathcal{K}_{ij}$, G_{ij}^1, $\mathcal{G}_{ij}^2$ are sparse band matrices, which carry the finite differencing scheme information. This means that every line has only nine entries for the convergence, four entries for the first shear component and seven for the second shear component, the rest are zero. An example for such a band matrix is given in Fig. 6.2.

6.1.3 Adaptive Grids

During our reconstruction process it may become necessary to calculate on grids which are not necessarily rectangular anymore. This is the case if we want to follow particular critical-curve

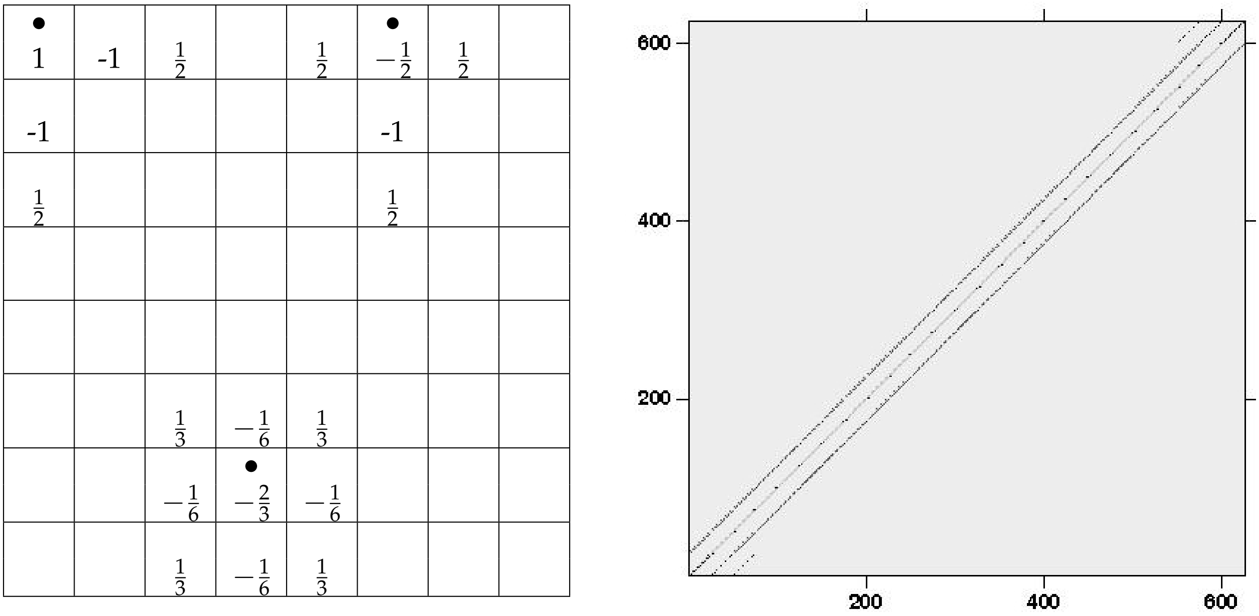


Figure 6.2: Left panel: The finite differencing scheme for the convergence approximation. Written in the cells are the coefficients in the difference quotients. Right panel: The corresponding band matrix representation. One can see the change in the scheme near top and bottom. The distance of the secondary diagonals to the main diagonal gives exactly the x-dimension of the grid, because it is the separation between top/bottom grid neighbours.

shapes, as we shall see in one of the following sections. Also on these adaptive grids, we want to use our approach to represent the whole grid by a vector $\psi \in \mathbb{R}^N$, where N is the number of gridpoints, and do the finite differences by matrix multiplications. This is still possible if one takes into account two difficulties. First, the grid points which are sitting at the edges and borders have to be identified and flagged in order to use the right schemes at the right positions. The other problem is that the distance between top and bottom grid-point neighbours in the vector is not trivial anymore and has to be calculated.
Both is done automatically by the code. Only the shape of the grid has to be given, then the code builds up a border structure and defines each pixel as border, edge or normal pixel. Afterwards, the non-trivial distances in the vector are calculated. An example is given in Fig. 6.4 and the corresponding pixel information is given in List. 6.1

```
#       type    xpos    ypos    top     2top    bottom  2bottom
1       11      5       3       4       10      0       0
2       15      6       3       4       10      0       0
3       12      7       3       4       10      0       0
4       11      4       4       6       14      0       0
5       15      5       4       6       14      4       0
6       10      6       4       6       14      4       0
```

Listing 6.1: The grid information on the first 6 pixels in Fig. 6.4. First column gives just the pixel number within the adaptive grid, second column declares pixels as edge, border or normal. Columns three and four give the positions in the shape grid for book-keeping and the last columns count the distances to the top/bottom neighbours. One should compare this list with certain pixels in Fig. 6.4 for a better understanding.

●	-1	$\frac{1}{2}$		$\frac{1}{2}$	● $-\frac{3}{2}$	$\frac{1}{2}$	
1					1		
$-\frac{1}{2}$					$-\frac{1}{2}$		
			$-\frac{1}{2}$				
		$\frac{1}{2}$	●	$\frac{1}{2}$			
			$-\frac{1}{2}$				

● 1	-1			$\frac{1}{2}$	●	$-\frac{1}{2}$	
-1	1			$-\frac{1}{2}$		$\frac{1}{2}$	
		$\frac{1}{2}$	$-\frac{1}{2}$				
		$-\frac{1}{2}$	● 1	$-\frac{1}{2}$			
			$-\frac{1}{2}$	$\frac{1}{2}$			

Figure 6.3: Left panel: Finite differencing scheme for the first shear component. Right panel: Second shear component.

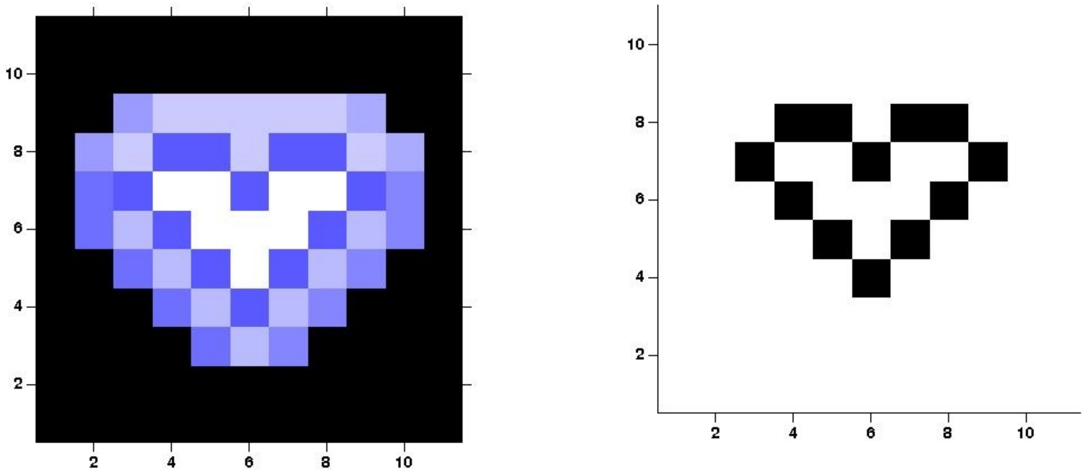


Figure 6.4: An example for an irregulary shaped grid. In the left panel one sees the adaptive grid with a border structure, introducing the needed edges and border points which are distinguished by color. The right panel shows the original input shape which is meant to be followed by the adaptive grid.

6.2 Minimising the χ^2-Function

6.2.1 Linearisation in Grid Space

The matrix representation of the finite differences (Eqs. 6.10, 6.11 and 6.11) allows us finally to formulate the χ^2-minimisation, which reads for the weak lensing term as follows:

$$\chi^2_w = \chi^2_1 + \chi^2_2, \tag{6.12}$$

where we splitted the weak lensing χ^2 into two parts for the two ellipticity components.
Of course, we have to perform the minimisation for both components , but for simplicity we show the calculation just for one component, and one has to substitute $\varepsilon, \gamma, \mathcal{C}, \mathcal{G}$ with $\varepsilon^1, \gamma^1, \mathcal{C}^1, \mathcal{G}^1$ and $\varepsilon^2, \gamma^2, \mathcal{C}^2, \mathcal{G}^2$, respectively.

$$\begin{aligned}
\chi^2 &= \left(\varepsilon_i - \frac{Z_i\gamma_i}{1-Z_i\kappa_i}\right) \mathcal{C}^{-1}_{ij} \left(\varepsilon_j - \frac{Z_j\gamma_j}{1-Z_j\kappa_j}\right) \\
&= \underbrace{\frac{\mathcal{C}^{-1}_{ij}}{(1-Z_i\kappa_i)(1-Z_j\kappa_j)}}_{\mathcal{F}_{ij}} \left(\varepsilon_i(1-Z_i\kappa_i) - Z_i\gamma_i\right)\left(\varepsilon_j(1-Z_j\kappa_j) - Z_j\gamma_j\right) \\
&= \mathcal{F}_{ij}\left[(\varepsilon_i - \varepsilon_i Z_i\kappa_i - Z_i\gamma_i)(\varepsilon_j - \varepsilon_j Z_j\kappa_j - Z_j\gamma_j)\right] \\
&= \mathcal{F}_{ij}\left[\varepsilon_i\varepsilon_j - \varepsilon_i\varepsilon_j Z_j\kappa_j - \varepsilon_i Z_j\gamma_j - \varepsilon_i\varepsilon_j Z_i\kappa_i + \varepsilon_i\varepsilon_j Z_i Z_j\kappa_i\kappa_j \right. \\
&\qquad \left. + \varepsilon_i Z_i Z_j\kappa_i\gamma_j - \varepsilon_j Z_i\gamma_i + \varepsilon_j Z_i Z_j\kappa_j\gamma_i + Z_i Z_j\gamma_i\gamma_j\right],
\end{aligned} \tag{6.13}$$

where we put the nonlinearities $(1 - Z\kappa)$ together in the prefactor-matrix $\mathcal{F}_{ij}$. We will deal with this term later and simply keep it constant for now.
Now we minimise this equation for every potential parameter ψ_l,

$$\frac{\partial\chi^2(\psi)}{\partial\psi_l} \overset{!}{=} 0 \tag{6.14}$$

$$\begin{aligned}
\frac{\partial\chi^2(\psi)}{\partial\psi_l} = \mathcal{F}_{ij}\Bigg[& -\varepsilon_i\varepsilon_j Z_j \frac{\partial}{\partial\psi_l}\kappa_j(\psi) - \varepsilon_i Z_j \frac{\partial}{\partial\psi_l}\gamma_j(\psi) - \varepsilon_i\varepsilon_j Z_i \frac{\partial}{\partial\psi_l}\kappa_i(\psi) \\
& + \varepsilon_i\varepsilon_j Z_i Z_j\kappa_i \frac{\partial}{\partial\psi_l}\kappa_j(\psi) + \varepsilon_i\varepsilon_j Z_i Z_j\kappa_j \frac{\partial}{\partial\psi_l}\kappa_i(\psi) \\
& + \varepsilon_i Z_i Z_j\kappa_i \frac{\partial}{\partial\psi_l}\gamma_j(\psi) + \varepsilon_i Z_i Z_j\gamma_j \frac{\partial}{\partial\psi_l}\kappa_i(\psi) \\
& - \varepsilon_j Z_i Z_j \frac{\partial}{\partial\psi_l}\gamma_i(\psi) + \varepsilon_j Z_i Z_j\kappa_j \frac{\partial}{\partial\psi_l}\gamma_i(\psi) \\
& + \varepsilon_j Z_i Z_j\gamma_i \frac{\partial}{\partial\psi_l}\kappa_j(\psi) + Z_i Z_j\gamma_i \frac{\partial}{\partial\psi_l}\gamma_j(\psi) + Z_i Z_j\gamma_j \frac{\partial}{\partial\psi_l}\gamma_i(\psi)\Bigg].
\end{aligned} \tag{6.15}$$

Using $\gamma_i = \mathcal{G}_{ik}\psi_k$, $\kappa_i = \mathcal{K}_{ik}\psi_k$ and $\frac{\partial}{\partial\psi_l}\mathcal{K}_{ik}\psi_k = \mathcal{K}_{ik}\delta_{kl}$ allows us to take the derivatives:

$$\begin{aligned}
\frac{\partial\chi^2(\psi_k)}{\partial\psi_l} = \mathcal{F}_{ij}\big[& -\varepsilon_i\varepsilon_j Z_j\mathcal{K}_{jk}\delta_{kl} - \varepsilon_i Z_j\mathcal{G}_{jk}\delta_{kl} - \varepsilon_i\varepsilon_j Z_i\mathcal{K}_{ik}\delta_{kl} \\
& + \varepsilon_i\varepsilon_j Z_i Z_j\mathcal{K}_{ik}\psi_k\mathcal{K}_{jk}\delta_{kl} + +\varepsilon_i\varepsilon_j Z_i Z_j\mathcal{K}_{jk}\psi_k\mathcal{K}_{ik}\delta_{kl} \\
& + \varepsilon_i Z_i Z_j\mathcal{K}_{ik}\psi_k\mathcal{G}_{jk}\delta_{kl} + \varepsilon_i Z_i Z_j\mathcal{G}_{jk}\psi_k\mathcal{K}_{ik}\delta_{kl} \\
& - \varepsilon_j Z_i Z_j\mathcal{G}_{ik}\delta_{kl} + \varepsilon_j Z_i Z_j\mathcal{K}_{jk}\psi_k\mathcal{G}_{ik}\delta_{kl} \\
& + \varepsilon_j Z_i Z_j\mathcal{G}_{ik}\psi_k\mathcal{K}_{jk}\delta_{kl} + Z_i Z_j\mathcal{G}_{ik}\psi_k\mathcal{G}_{jk}\delta_{kl} + Z_i Z_j\mathcal{G}_{jk}\psi_k\mathcal{G}_{ik}\delta_{kl}\big].
\end{aligned} \tag{6.16}$$

From the last equation one can see that we can now write Eq. 6.14 as a linear system of equations

$$\mathcal{B}_{lk}\psi_k = \mathcal{V}_l, \tag{6.17}$$

with the coefficient matrix

$$\begin{aligned}\mathcal{B}_{lk} = \mathcal{F}_{ij}\big[&\varepsilon_i\varepsilon_j Z_i Z_j \mathcal{K}_{ik}\mathcal{K}_{jl} + \varepsilon_i\varepsilon_j Z_i Z_j \mathcal{K}_{jk}\mathcal{K}_{il} + \varepsilon_i Z_i Z_j \mathcal{K}_{ik}\mathcal{G}_{jl}\\ &+ \varepsilon_i Z_i Z_j \mathcal{G}_{jk}\mathcal{K}_{il} + \varepsilon_j Z_i Z_j \mathcal{K}_{jk}\mathcal{G}_{il} + \varepsilon_i Z_i Z_j \mathcal{G}_{ik}\mathcal{K}_{jl}\\ &+ Z_i Z_j \mathcal{G}_{ik}\mathcal{G}_{jl} + Z_i Z_j \mathcal{G}_{jk}\mathcal{G}_{il}\big],\end{aligned} \tag{6.18}$$

and the result vector

$$\mathcal{V}_l = \mathcal{F}_{ij}\left[\varepsilon_i\varepsilon_j\mathcal{K}_{jl} + \varepsilon_i Z_j \mathcal{G}_{jl} + \varepsilon_i\varepsilon_j Z_i \mathcal{K}_{il} + \varepsilon_j Z_i \mathcal{G}_{il}\right]. \tag{6.19}$$

At this point, one should not forget that we still have non-linear terms in the $\mathcal{F}_{ij}$-matrix, which we just assume as constant right now. We will take care of them in the next section.
We now repeat the exercise for the strong lensing term:

$$\chi_s^2 = \frac{(\det\mathcal{A})_i^2}{\sigma_i^2} = \frac{((1 - Z_i\kappa_i)^2 - |Z_i\gamma_i|^2)^2}{\sigma_i^2}, \tag{6.20}$$

where the non-linear terms are isolated and taken as a constant for now:

$$\begin{aligned}\frac{\partial\chi_s^2(\psi_k)}{\partial\psi_l} &= \frac{2(\det\mathcal{A})_i}{\sigma_i^2}\frac{\partial}{\partial\psi_l}(\det\mathcal{A}(\psi_k))_i\\ &= \frac{2(\det\mathcal{A})_i}{\sigma_i^2}\left[\frac{\partial}{\partial\psi_l}(1 - Z_i\kappa_i(\psi_k)^2 - \frac{\partial}{\psi_l}|Z_i\gamma_i(\psi_k)|^2\right]\\ &= \frac{2(\det\mathcal{A})_i}{\sigma_i^2}\left[2(1 - Z_i\kappa_i(\psi_k))\left(-Z_i\frac{\partial}{\partial\psi_l}\kappa_i(\psi_k)\right) - \frac{\partial}{\partial\psi_l}\left(Z_i^2\gamma_{1i}^2(\psi_k) + Z_i^2\gamma_{2i}^2(\psi_k)\right)\right]\\ &= \frac{2(\det\mathcal{A})_i}{\sigma_i^2}\left[2(1 - Z_i\kappa_i(\psi_k))(-Z_i\mathcal{K}_{il}) - 2Z_i\gamma_{1i}(\psi_k)Z_i\mathcal{G}_{il}^1 - 2Z_i\gamma_{2i}(\psi_k)Z_i\mathcal{G}_{il}^2\right]\\ &= \frac{4(\det\mathcal{A})_i}{\sigma_i^2}\left[Z_i^2\mathcal{K}_{ik}\mathcal{K}_{il}\psi_k - Z_i\mathcal{K}_{il} - Z_i^2\mathcal{G}_{ik}^1\mathcal{G}_{il}^1\psi_k - Z_i\mathcal{K}_{il} - Z_i^2\mathcal{G}_{ik}^2\mathcal{G}_{il}^2\psi_k\right].\end{aligned} \tag{6.21}$$

This gives another linear sytem:

$$\mathcal{B}_{lk}^*\psi_k = \mathcal{V}_l^*, \tag{6.22}$$

with a strong lensing coefficient matrix:

$$\mathcal{B}_{lk}^* = \frac{4(\det\mathcal{A})_i}{\sigma_i^2}Z_i^2(\mathcal{K}_{ik}\mathcal{K}_{il} - \mathcal{G}_{ik}^1\mathcal{G}_{il}^1 - \mathcal{G}_{ik}^2\mathcal{G}_{il}^2), \tag{6.23}$$

and the result vector:

$$\mathcal{V}_l^* = \frac{4(\det\mathcal{A})_i}{\sigma_i^2}Z_i\mathcal{K}_{il}. \tag{6.24}$$

Finally, we collect the results to obtain the solution for Eq. 4.5, by solving a linear system with the following coefficient matrix and result vector:

$$
\begin{aligned}
\mathcal{B}_{lk} =& \sum_{i,j} \mathcal{F}^1_{ij} Z_i Z_j \\
&\cdot \left[\varepsilon^1_i \varepsilon^1_j \mathcal{K}_{ik}\mathcal{K}_{jl} + \varepsilon^1_i \varepsilon^1_j \mathcal{K}_{jk}\mathcal{K}_{il} + \varepsilon^1_i \mathcal{K}_{ik}\mathcal{G}^1_{jl} + \varepsilon^1_i \mathcal{G}^1_{jk}\mathcal{K}_{il} + \varepsilon^1_j \mathcal{K}_{jk}\mathcal{G}^1_{il} + \varepsilon^1_j \mathcal{G}^1_{ik}\mathcal{K}_{jl} + \mathcal{G}^1_{ik}\mathcal{G}^1_{jl} + \mathcal{G}^1_{jk}\mathcal{G}^1_{il} \right] \\
&+ \sum_{i,j} \mathcal{F}^2_{ij} Z_i Z_j \\
&\cdot \left[\varepsilon^2_i \varepsilon^2_j \mathcal{K}_{ik}\mathcal{K}_{jl} + \varepsilon^2_i \varepsilon^2_j \mathcal{K}_{jk}\mathcal{K}_{il} + \varepsilon^2_i \mathcal{K}_{ik}\mathcal{G}^2_{jl} + \varepsilon^2_i \mathcal{G}^2_{jk}\mathcal{K}_{il} + \varepsilon^2_j \mathcal{K}_{jk}\mathcal{G}^2_{il} + \varepsilon^2_j \mathcal{G}^2_{ik}\mathcal{K}_{jl} + \mathcal{G}^2_{ik}\mathcal{G}^2_{jl} + \mathcal{G}^2_{jk}\mathcal{G}^2_{il} \right] \\
&+ \sum_m \frac{4(\det\mathcal{A})_m}{\sigma^2_m} Z^2_m \\
&\cdot \left[\mathcal{K}_{mk}\mathcal{K}_{ml} - \mathcal{G}^1_{mk}\mathcal{G}^1_{ml} - \mathcal{G}^2_{mk}\mathcal{G}^2_{ml} \right] \\
\mathcal{V}_l =& \sum_{i,j} \mathcal{F}^1_{ij} \left[\varepsilon^1_i \varepsilon^1_j \mathcal{K}_{jl} + \varepsilon^1_i Z_j \mathcal{G}^1_{jl} + \varepsilon^1_i \varepsilon^1_j Z_i \mathcal{K}_{il} + \varepsilon^1_j Z_i \mathcal{G}^1_{il} \right] \\
&+ \sum_{i,j} \mathcal{F}^2_{ij} \left[\varepsilon^2_i \varepsilon^2_j \mathcal{K}_{jl} + \varepsilon^2_i Z_j \mathcal{G}^2_{jl} + \varepsilon^2_i \varepsilon^2_j Z_i \mathcal{K}_{il} + \varepsilon^2_j Z_i \mathcal{G}^2_{il} \right] \\
&+ \sum_m \frac{4(\det\mathcal{A})_m}{\sigma^2_m} Z_m \mathcal{K}_{ml},
\end{aligned}
\tag{6.25}
$$

where i,j indicate the summation over the complete grid and m over those pixels which are assumed to be part of a critical curve.

6.2.2 Numerics and MPI Implementation

The main problem with respect to computation time is now to build up the linear system, because of the following reason. Hence, performing a reconstruction of a cluster at a resolution of (32x32) pixels:

- 1024 potential values ψ_k.
- The dimension of the coefficient matrix $\mathcal{B}_{lk}$ is (1024 x 1024)
  $\Rightarrow$ 1048576 matrix elements
- The length of the result vector $\mathcal{V}_l$ is 1024
  $\Rightarrow$ 1024 vector elements
- One matrix element is built up by 16 terms, each of which is a summation over i,j and three terms, each of which is a summation over m.
  One result vector element consists of 8 terms, which are summations over i,j and one summation term over m.
- Each term in one of these sums is a product of seven numbers at maximum.
- Both indices i and j run from 1 to 1024
- The index m runs over the number of critical curve pixels and is therefor negligible.

$\Rightarrow$ $1048576 \cdot 16 \cdot 1024 \cdot 1024 \cdot 7 + 1024 \cdot 8 \cdot 1024 \cdot 7 \approx 1.2 \cdot 10^{14}$ arithmetic operations.

By assuming the standard performance of an ordinary desktop machine to be $\sim$ 3 GFLOPS, we expect a runtime for a single reconstruction of $\sim$ 1 day. If we take into account all the iterations we need and which we shall discuss in the next section, we end up with 20 of these

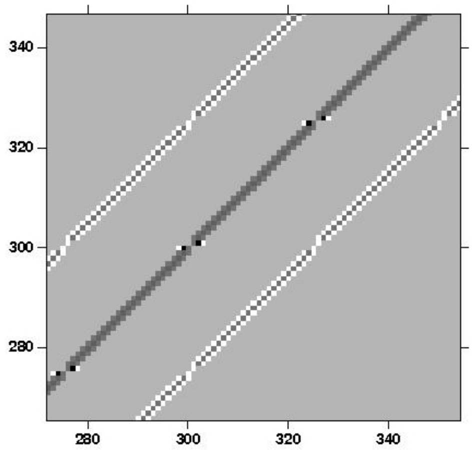


Figure 6.5: Zoom on the area around the diagonal of the $\mathcal{K}_{ij}$ matrix. One can see all points in a column which are not zero.

reconstructions. The result is a runtime of two to three weeks. We will now show how to reduce this to ~ 1 hour.

1. The first thing one should recognise is that some of the terms in the coefficient matrix and the result vector (Eq. 6.25) look quite similar. In fact, pairs of them become identical if the covariance matrix is symmetric. In our case this is true, so we reduced the number of terms by a factor of two.

2. Now we look directly at the terms which build up the linear system. Each term contains at least one of the finite differencing matrices $\mathcal{K}_{ij}$, $\mathcal{G}^1_{ij}$ or $\mathcal{G}^2_{ij}$. From these matrices we know that they are sparse band matrices with at most nine entries per column, so it would make absolutely no sense to sum over a complete column, like in Eq. 6.25. (To see a closeup of the diagonal area, see Fig 6.5). In our code we sum only over these entries near the main diagonal, which are non-vanishing and save a factor of (matrix-dimension)$^2/9$ for each index summation. This is a huge achievement and this alone makes it possible to run the code also on high grid dimensions.

3. Another aspect is directly connected to the one above. In earlier versions, we used higher order finite differences schemes, which included up to 25 surrounding points for a single derivative. By doing so, one gains accuracy in the derivative approximation, but the drawback is an increase in computation time, because the bandwidth of the corresponding matrices becomes bigger. This corresponds to significantly larger sums in the coefficient matrix terms.

4. Even with the huge decrease of calculations mentioned above, calculating all coeffcient matrix and result vector terms is still quite a task for a single processor. But since all these terms are completely independent of each other, they can be easily partitioned to different processors. Once all single terms are calculated on different nodes, the headnode sums the different results. This part is implemented in our code in such a way that it is checked how many nodes are available and the summing process is divided into the given number of processors, using MPI. We also note here that only the different terms are given to the different processors, not the matrix multiplication itself, so one would not gain anything by using more processors then terms available. One task for future work will be to also partition the matrix multiplications, by splitting the summation interval.

By using all the techniques mentioned above, we decreased the calculation time by a factor of order ~ 500. The numerical analysis of the problem is not completed yet and will allow more computation time decrement, which will be necessary for future high resolution reconstructions.

6.2.3 Solving Linear Systems

After the full linear system of equations has been obtained, the lensing potential is derived by solving the linear system. In our case, this is done by GSL routines, using advanced Gauss-Jordan algorithms (see Rannacher, 2006). The computation time for solving a linear system of this dimension is negligible in comparison to the time needed to build up the linear system. This changes when going to higher dimensions, which might make it necessary to migrate to LAPACK-routines.

6.3 Regularisation and 2-Step-Iteration

6.3.1 Regularisation

By now the χ^2-function looks as follows:

$$\chi^2(\psi) = \chi^2_w(\psi) + \chi^2_s(\psi) \tag{6.26}$$

with the weak and strong lensing contributions. We now introduce a third term, which is a so-called regularisation term R,

$$\chi^2(\psi) = \chi^2_w(\psi) + \chi^2_s(\psi) + R(\psi). \tag{6.27}$$

The regularisation term is dependent on the potential and disfavors certain reconstruction solutions. R is defined in such a way that unwanted solutions make the regularisation term large. By doing so one can avoid that the reconstruction is following intrinsic noise patterns, which do not reflect the correct solution. In this work we follow the regularisation scheme of Bradač et al. (2005), where the regularisation term is just a comparison with a convergence result obtained before:

$$R = \eta_i \left(\kappa_i^{\text{before}} - \kappa_i(\psi)\right)^2 \tag{6.28}$$

The prefactor η_i controls the strength of the regularisation and is crucial for the reconstruction. It should be chosen in such a way that the overall χ^2 is of order unity and of course low enough that changes in the reconstruction still can take effect. Minimising Eq. 6.28 leads again to a linear system:

$$\begin{aligned}\frac{\partial R(\psi_k)}{\partial \psi_l} &= \eta_i \frac{\partial}{\partial \psi_l}\left(\kappa_i^b - \kappa_i(\psi_k)\right)^2 \\ &= 2\eta_i(\kappa_i^b - \kappa_i)\left(-\frac{\partial}{\partial \psi_l}\mathcal{K}_{ik}\psi_k\right) \\ &= 2\eta_i\left(\kappa_i^b - \mathcal{K}_{ik}\psi_k\right)(-\mathcal{K}_{il}) \\ &= 2\eta_i\left(-\kappa_i^b\mathcal{K}_{il} + \mathcal{K}_{ik}\mathcal{K}_{il}\psi_k\right),\end{aligned} \tag{6.29}$$

which contributes one additional term to the coefficient matrix and the result vector.

$$\mathcal{B}_{lk}^{\text{reg}} = \sum_i \eta_i \mathcal{K}_{ik}\mathcal{K}_{il} \tag{6.30}$$

$$\mathcal{V}_l^{\text{reg}} = \sum_i \eta_i \kappa_i^b \mathcal{K}_{il} \tag{6.31}$$

Additional regularisation terms for the shear are also possible.

6.3.2 Inner-Level Iteration

As we mentioned several times before, we still have a problem with the non-linear terms in the coefficient matrix and result vector, which we simply held constant until now. We solve this problem in the same way as in the direct inversion technique in section 3.2.1. We start from a first guess for the convergence and insert the corresponding non-linear terms. After that, we perform the reconstruction and obtain a result for the potential. From this result we calculate convergence and shear. We take this as a new guess for the non-linear terms and perform another reconstruction. This gives us a converging, iterative process. We control the convergence behaviour of the iteration by comparing the convergence results from the actual and the former iteration steps. If the change is below a given threshold, we stop the iteration. The drawback of this method is that we now need several iterations (3-5 in practice) for a complete reconstruction, which takes some time at high resolution. Bradač et al. (2005) showed that the initial guess of the convergence value does not affect the reconstruction, but at most the number of iterations. Our experience confirms their result, which means that the initial guess of a convergence equal to zero is fair enough.

6.3.3 Outer-Level Iteration

One reason why we introduced the regularisation terms before is that we also do a second type of iteration, which we call outer-level iteration. The galaxy density of today's observations would allow only a resolution of ($\sim$10 x 10) pixels, with uncorrelated datapoints. Of course one wants to go to higher resolution, which implies that data points become correlated and the reconstruction will become affected by that. Another problem at higher resolution is, that the noise patterns in the data get more and more pronounced and the reconstruction tends to overfit these noise patterns. And a last problem at high resolution is that the convergence values of some individual pixels will increase, which makes the initial guess of a vanishing convergence less and less accurate. The result are many inner-level iterations.
A very elegant way to avoid all these listed problems is to introduce another iteration which works as follows:

- One starts at the smallest possible resolution, where the pixels are not or almost not correlated.
- This resolution will be too coarse for fine-structure noise patterns to appear.
- Starting from the initial convergence guess zero, one performs the inner-level iteration until the reconstruction converges.
- The obtained result is then interpolated to a slightly higher resolution.
- This interpolation is taken as comparison function in the regularisation and as new initial guess in the inner-level iteration at the higher resolution.
- This process is repeated until the final resolution is reached.

This two-iteration concept delivered by far the best reconstruction results, but there are two drawbacks. First you need several iterations, which increases the reconstruction time. Second, it introduces a new parameter, which is the regularisation weighting η, which has to be well calibrated in order not to underfit the data in later steps. The transition from a low to high resolution can be seen in Fig. 6.6 .

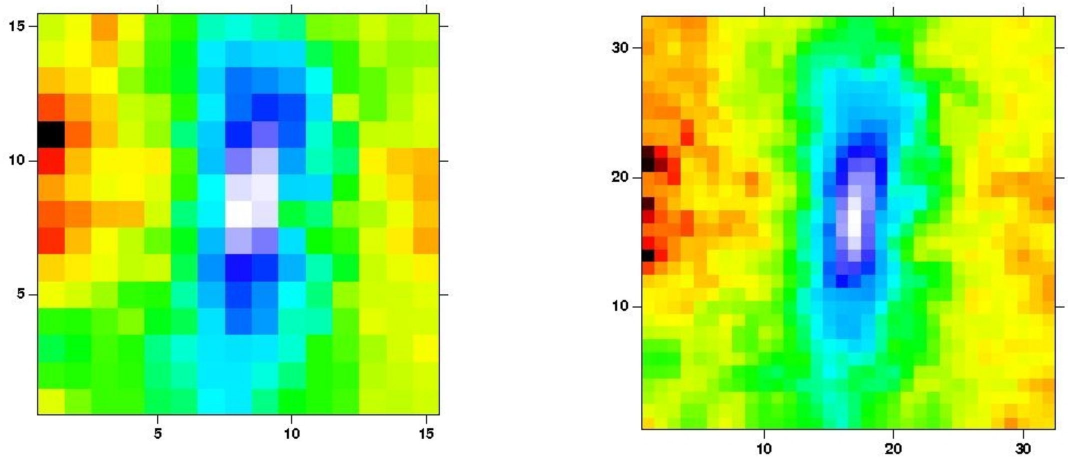


Figure 6.6: Left panel: Resulting Convergence Map on a starting resolution of (15x15). The resolution is so coarse that no noise patterns appear. Right panel: Final resolution of (32x32), after several iterations.

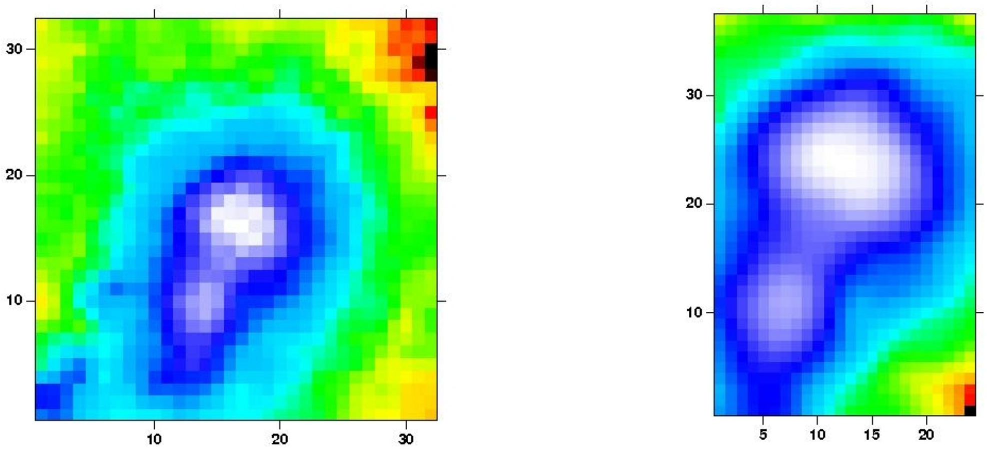


Figure 6.7: Left panel: Complete combined weak and strong lensing reconstruction at low resolution. Right panel: Zoom into the interpolated cluster core of the low-resolution reconstruction.

6.4 Strong Lensing at High Resolution

6.4.1 Interpolation

Hence, we have finished a complete reconstruction as described in the previous sections. The problem that we also mentioned in section 4.2.2 is now that even on the last steps of the outer level iteration, the grid is still not fine enough to follow the shape of the critical curves. We deal with the problem by interpolating the lensing potential from the final reconstruction so far and calculating convergence and shear at that resolution. Then, we zoom into the cluster core where the strong-lensing information is available. The interpolation is done by a bicubic spline interpolation routine. The interpolated result serves as template for the following high-resolution reconstruction.

6.4.2 Modified χ^2-Function

We start from a different χ^2-function:

$$\chi^2(\psi) = \chi^2_s(\psi) + R(\psi) \tag{6.32}$$

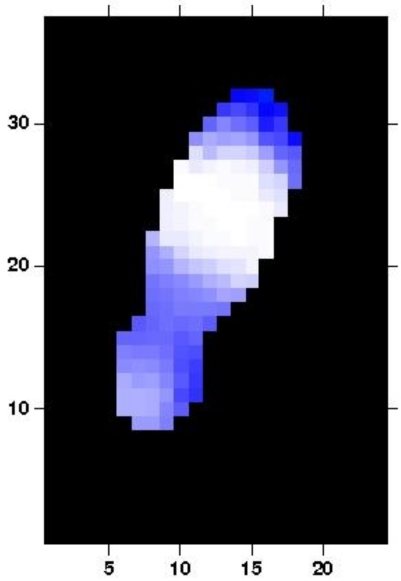


Figure 6.8: The convergence map of a high resolution reconstruction, using adaptive grids.

The reason why no weak-lensing χ^2-function appears anymore is that at this high resolution, we do not have weak lensing datapoints anymore, since we do not want to interpolate the weak lensing data. The strong-lensing datapoints are available at this high resolution, since one can follow the critical curves very accurately. This does not mean that we do not use the weak lensing data in this high resolution reconstruction anymore, because the interpolated result enters in the regularisation function. So the result will be significantly based on the weak lensing constraints. We finish the reconstruction by inserting the high-resolution cluster-core result into the result obtained at a coarser resolution, which consists of the complete cluster field.

6.4.3 Adaptive Grid Usage

Since the cluster-core reconstruction described above is based on interpolated results, we want to use as few interpolated pixels as possible. Because of this we make use of the adaptive grids, described in section 6.1.3, which are able to follow the exact shape of the critical-curve estimate and use much less datapoints for regularisation. So the reconstruction will mainly be based on the strong-lensing data alone. But still we need the constraints from the interpolated grid points, because otherwise the linear system would be underdetermined.

6.5 Running the Code

After these longer chapters on the implementation of the reconstruction method, we want to summarize how the reconstruction is practically done. This should not be a code-listing but a listing of the different steps which are necessary to achieve the final result.
The whole reconstruction process can be divided into three steps.

6.5.1 Data Preparation

The first thing the code does is to prepare the input data in such a way that it can be used by the reconstruction routines. The main ingredient here is the adaptive averaging routine.

1. Starting point is an ellipticity catalog and an arc-position catalog, both as plain text files. Those files are read by the code.

2. For each resolution between the initial resolution and the final resolution, the ellipticities are averaged, the covariance matrices are calculated and the arc-positions on the grid are saved. Typical values for the initial resolution would be a (10x10) grid and a (30x30) grid, for the final resolution.

3. Arc positions are also marked at a much higher resolution (e.g. (75x75)), where they are used later for a final high-resolution reconstruction as described in section 6.4.

4. All relevant results are saved in separate FITS files for each resolution, or are directly passed to the reconstruction routines.

6.5.2 Low Resolution Reconstruction

After the needed data for the reconstruction routines is now available, the combined weak and strong-lensing reconstruction starts. The main ingredient in these routines is building up the linear systems whose solutions are the wanted results.

1. The reconstruction starts at the initial resolution.

2. The starting convergence is assumed to be zero and a first reconstruction is performed. From the result we obtain a new guess for the convergence.

3. This procedure is repeated until we achieve convergence in the result.

4. The potential is interpolated to the next higher resolution and taken as a comparison function for the reconstructions at this resolution. Again the inner-level iteration is done until we have a convergent result.

5. This outer-level iteration is repeated until we have a result at the final resolution. The stepsize of the outer-level iteration can be set by the user.

6.5.3 High Resolution Reconstruction

The last step in the reconstruction is a reconstruction at a resolution which can follow the critical-curve estimate. It uses adaptive grids.

1. The result of the low-resolution reconstruction is interpolated to the new resolution.

2. The cluster core, where the strong lensing features appear, is cut out.

3. The adaptive grid on the cluster core, which is defined by the concrete shape of the critical curve estimation, is built up.

4. The reconstruction described in section 6.4 is performed on the adaptive grid. Comparison function is the interpolated low resolution result.

5. The high-resolution result is inserted into the low-resolution result and the reconstruction is finished.

7 Applications and Results

We will now focus on the application of our reconstruction code. After testing the numerical efficiency of the code we will perform reconstruction tests with synthetic galaxy-cluster data to prove that our method is working correctly and reliably. Afterwards we perform cluster reconstructions with more realistic lensing scenarios, which also include a complete image analysis. We conclude the applications with a real galaxy-cluster reconstruction, namely with the well known strong-lensing cluster MS 2137.

7.1 Runtime Comparison

As we pointed out in Chapter 6, we use several techniques to keep the runtime of the code as short as possible. In two small tests we show that these techniques shorten the runtime significantly, while not compromising the correct result.

7.1.1 Effective Multiplication Schemes

In the first test we want to show that the code is not wasting time by performing unnecessary arithmetic operations. For doing so we look at the first term of the coefficient matrix $\mathcal{B}_{lk}$, which we need to build up to perform a reconstruction by solving a linear system of equations

$$\mathcal{B}^1_{lk} = \sum_{i,j} \mathcal{F}^1_{ij} Z_i Z_j \varepsilon^1_i \varepsilon^1_j \mathcal{K}_{ik} \mathcal{K}_{jl}. \tag{7.1}$$

A runtime test with this term is set up by computing its contribution to the total coefficient matrix. We choose reconstruction resolutions of (14x14), (20x20) and (25x25) pixels, which means that we end up with finite differences and coefficient matrices of dimensions (196x196),(400x400) and (625x625) respectively. For the values of $\mathcal{F}_{ij}$, Z_i and ε_i, standard values as needed in a typical reconstruction are used, of course the exact values are unimportant in a runtime test. First we compute the complete sum over i and j, without taking into account that most of the elements in the finite differences matrix $\mathcal{K}$ are vanishing. After that, we calculate the term again by just summing over the necessary elements.

In Fig. 7.1 one can see the residuals between the two methods, which just show numerical noise, so the enhanced method which is not taking the complete sum over i and j delivers the correct result. This is not very surprising at all, but the difference in runtime (see Table 7.1) looks very convincing and proves the efficiency of our method.

Resolution	Method1: all indices [s]	Method2: non-vanishing indices [s]
14x14	282	2
20x20	4715	9
25x25	28507	20

Table 7.1: Runtime comparison of the different methods on different resolutions. 14x14 resolution was performed on an Intel Pentium 4, 3.2 GHz Dual Core (1 core used), 2MB Shared Cache, 1GB main memory. 20x20 and 25x25 resolution was performed on an AMD Opteron 250, 2.4GHz, 1MB Cache, 8GB main memory; which has a higher numerical performance.

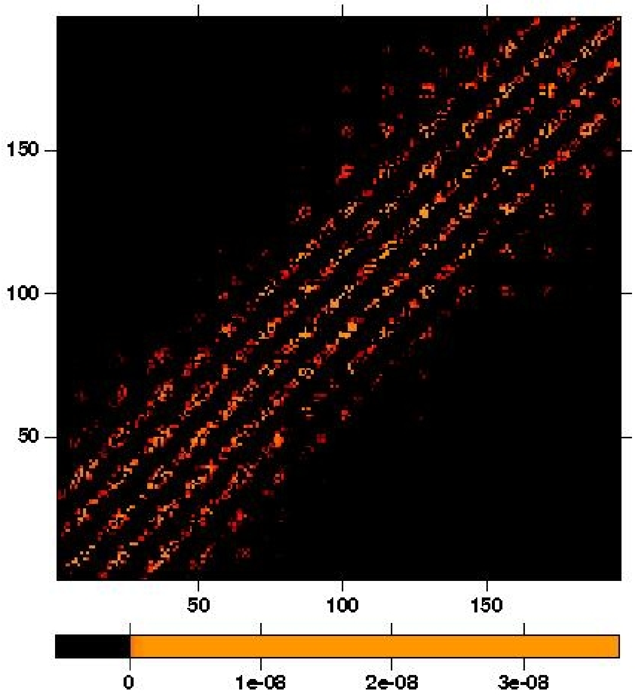


Figure 7.1: The residuals between the complete summation method and the speed-optimised term calculation. The largest difference is $3.73 \cdot 10^{-8}$ and is mostly due to the accuracy of the visualisation library.

Final Resolution	single [s]	MPI [s]
20x20	849	244
24x24	2112	591
30x30	6215	1867

Table 7.2: Runtime comparison of the single processor version and the MPI Implementation. Single processor machine: Intel Pentium 4, 3.2 GHz Dual Core (1 core used), 2MB Shared Cache, 1GB main memory. MPI machine: 8 node, 16 processor, AMD Opteron 250, 2.4GHz, 1MB Cache, 2GB main memory (per node) Linux PC cluster, InfiniBand-Network.

7.1.2 Single Processor and MPI Implementation

In a second test we show the overall runtime of our method and the fact that the MPI-implementation performs better than the single processor version. In Table 7.2 we show the runtime for different reconstruction scenarios. In fact we show the reconstruction time for the simulated cluster reconstruction, which will be shown in detail in the next chapter. The starting resolution for all three cases was a (10x10) pixel reconstruction grid, but the final resolution is different. The outer-level-iteration stepsize was two pixels and for each step, five inner-level iterations were performed.

7.2 Synthetic Data

We will now test our reconstruction method with simulated data. This data consists of several galaxy clusters obtained from N-body simulations described in Bartelmann et al. (1998). Earlier tests of similar combined weak and strong lensing maximum-likelihood reconstruction methods are described in Cacciato (2005) and Cacciato et al. (2006).

7.2.1 Reconstruction Data and Outline of Testing Method

Since our reconstruction routines are constructed to be applied to real observational data, we use the simulated cluster data to create a semi-realistic lensing scenario. The simulated cluster data is available in form of the convergence and shear maps of the cluster on a resolution of (512x512) pixels. This makes it possible to calculate the reduced shear of the cluster, which is the expected weak lensing observable, for 262144 data points. The exact field size is given by the parameters of the simulation, so by the box-size and the redshift of the simulation. To define a coordinate system the (256,256) pixel is taken as coordinate origin and the positions of all other pixels are calculated, depending on the pixel side length with respect to this origin. From all available data points we choose now a small number with the help of a random number generator. This number should be comparable with the number of lensed galaxies in a realistic field. In this way we obtain a realistic reduced shear catalog for our cluster which can be used to start the field preparation process implemented in the code. As strong-lensing observable we use the original critical curve of the cluster. It is calculated by searching for sign changes in the Jacobian determinant of the simulated cluster and is translated into the coordinates defined above. With these constraints the code should be able to reproduce the original convergence map of the cluster with a resolution that is dependent on the number of data points that we use for the reduced shear catalog.

7.2.2 An Undercritical Cluster Example

The first simulated cluster reconstructed by the code is an undercritical example (see Fig. 7.2), which means that it does not produce an extended critical curve. In this case the reconstruction depends completely on the weak lensing observables.

The reconstruction was performed with two different choices for the number of data points. In a first run 3000 data points were chosen by the random number generator as an example with a relatively high number of data points, and in a second run this number was drastically reduced to 750 data points. The two reduced shear fields are plotted in Fig. 7.3.

For both samples the data preparation routines of the code were applied with 10 averaged data points per reconstruction pixel. This made it possible to reach a reconstruction resolution of (30x30) pixels for the 3000 data-point sample and a resolution of (20x20) for the 750 data-point sample. The data-point correlation matrices for both final resolutions are shown in Fig. 7.4.

The resulting κ-maps of the reconstructions are shown in Fig. 7.5 and Fig. 7.6, together with the appropriately rebinned, original κ-map of the cluster for comparison. As one can see, the results look very promising. The (30x30) pixel reconstruction resolves all main features of the cluster almost perfectly and even the (20x20) pixel reconstruction with only 750 input data points produces a useable result. A more quantitative comparison is done in Fig. 7.7 by the κ-profile along the increasing main diagonal of the field. The straight line is the original cluster profile and the crosses are obtained by the two reconstructions. Also in the profile plot the results look very well, especially because the original κ-profile (red line) was taken from the full-resolution κ-map. The underestimation of the central peak is typical for a pure weak-lensing reconstruction and is due to the fact that a smoothing process over a number of galaxies was applied during the data-point preparation. Of course this step would not be necessary since one has perfect data in this simulated lensing scenario, but for the application to real data this step is inevitable and keeps this test closer to real applications. Another reason why the reconstruction points tend to underestimate the real profile is due to mass-sheet degeneracy. Again we want to stay close to reality, so we normalised the field simply to be non-negative, without any further knowledge of the cluster, which means that the point with the lowest convergence is set to zero. This is not exactly true for the real κ-map.

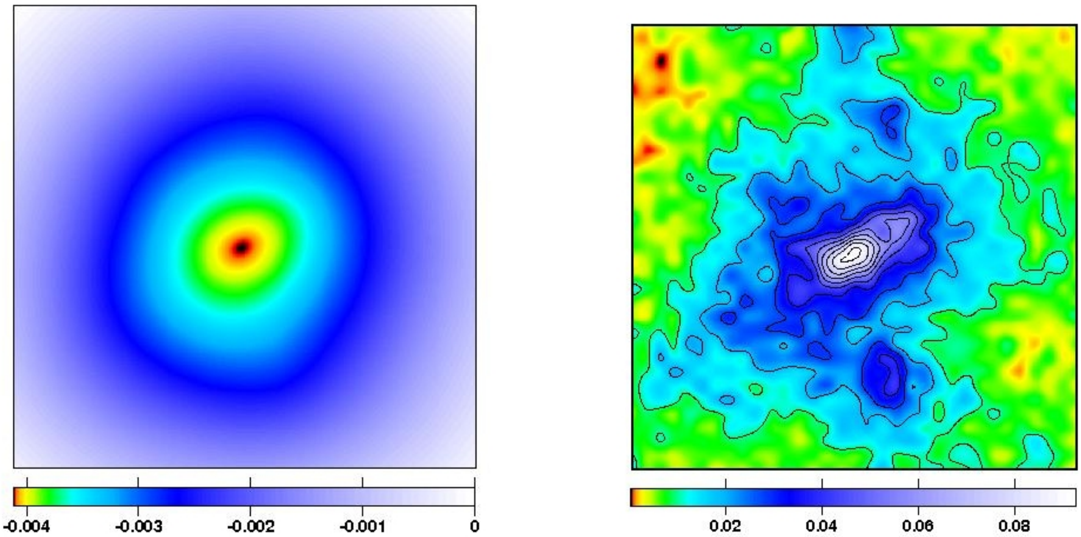


Figure 7.2: Lensing potential (left) and corresponding convergence map (right) of an undercritical galaxy cluster simulated in a ΛCDM scenario. The cluster redshift is $z_c = 0.93$ and the cube-side length of the simulation is 5 (comoving) Mpc which corresponds to 1.96 arcmins. The total mass of the cluster is $5.5 \times 10^{14} M_{\odot}$. The contours in the right panel start at a convergence value of $\kappa = 0.01$ and end at $\kappa = 0.09$ with linear scaling.

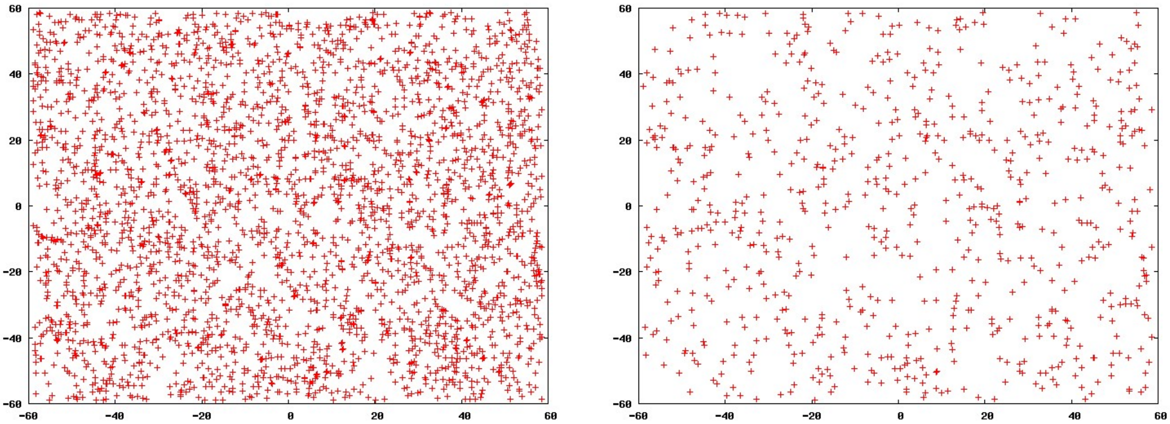


Figure 7.3: The chosen reduced-shear data points in the coordinates relative to the central pixel of the simulated cluster in arcsec. Left panel shows 3000 data points, right panel 750 data points.

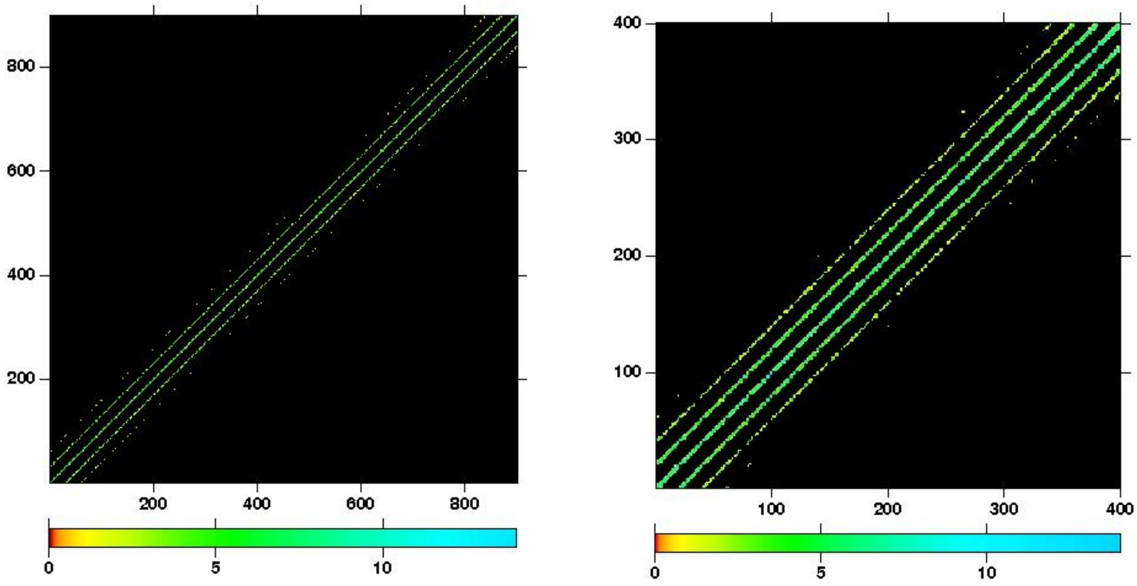


Figure 7.4: Left panel: Number of shared reduced-shear data points for each reconstruction pixel for the 3000 data-point sample. The reconstruction resolution is (30x30), so one sees a (900x900) correlation matrix. Right panel: Same matrix for the 750 data point sample. Even at the lower reconstruction resolution (20x20) one sees more correlation.

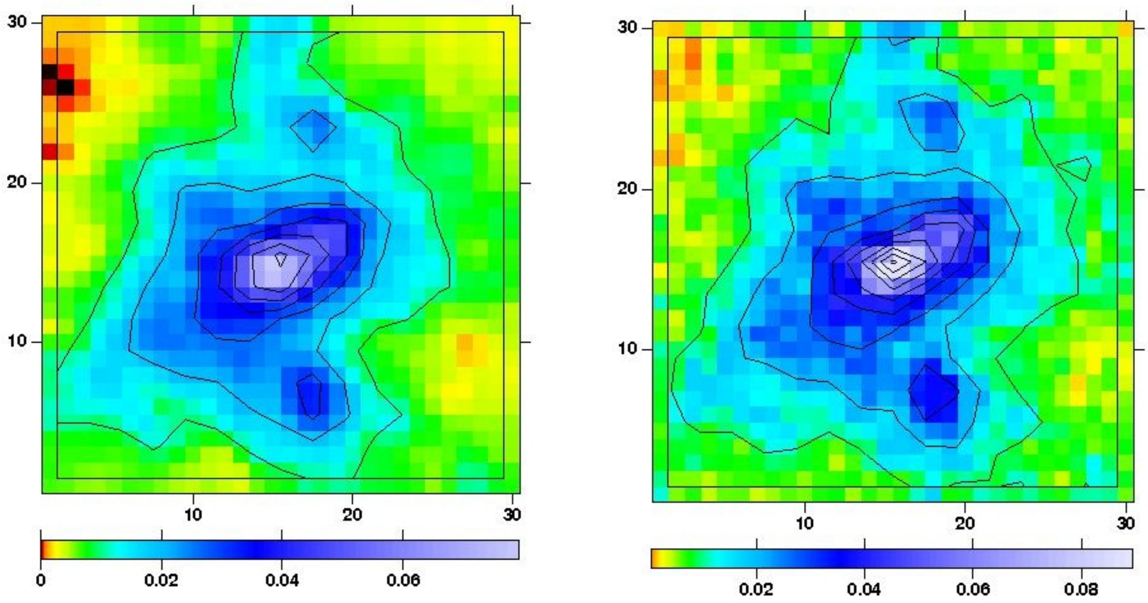


Figure 7.5: Left panel: Convergence map of the 3000 data-point reconstruction. The resolution is (30x30) pixels and the color scale is logarithmic. The contours start at $\kappa = 0.01$ and the spacing is $\Delta\kappa = 0.008$. Right panel: The convergence map of the original cluster rebinned to the reconstruction resolution. The parameters for color and contours are identical to the reconstruction.

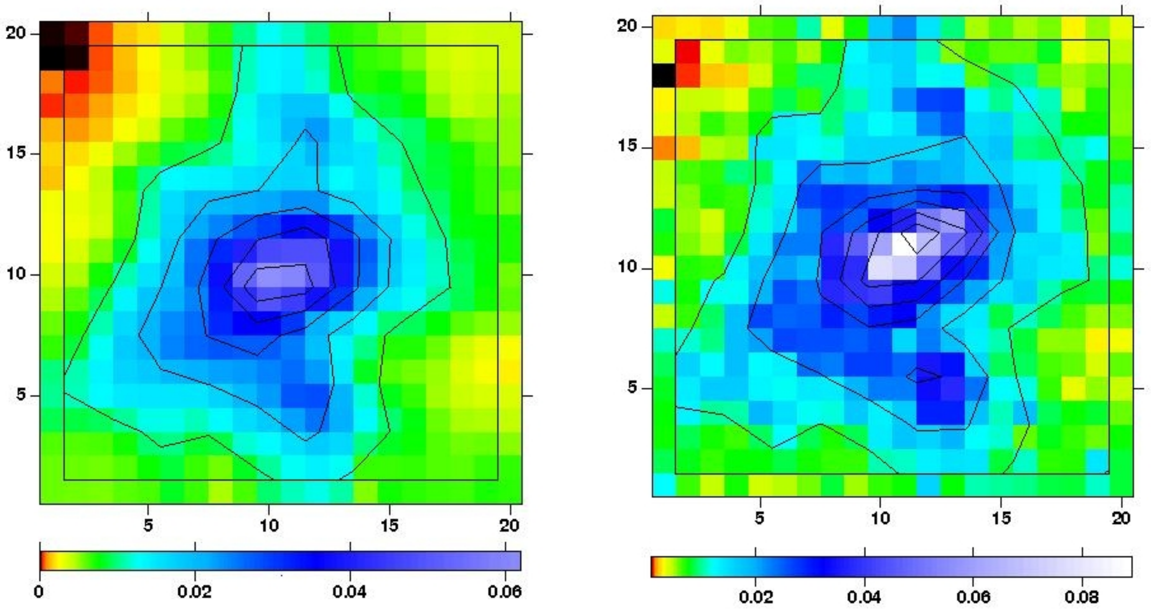


Figure 7.6: Left panel: Convergence map of the 750 data-point reconstruction. The resolution is (20x20) pixels and the color scale is logarithmic. The contours start at $\kappa = 0.01$ and the spacing is $\Delta\kappa = 0.008$. Right panel: The convergence map of the original cluster rebinned to the reconstruction resolution. The parameters for color and contours are identical to the reconstruction.

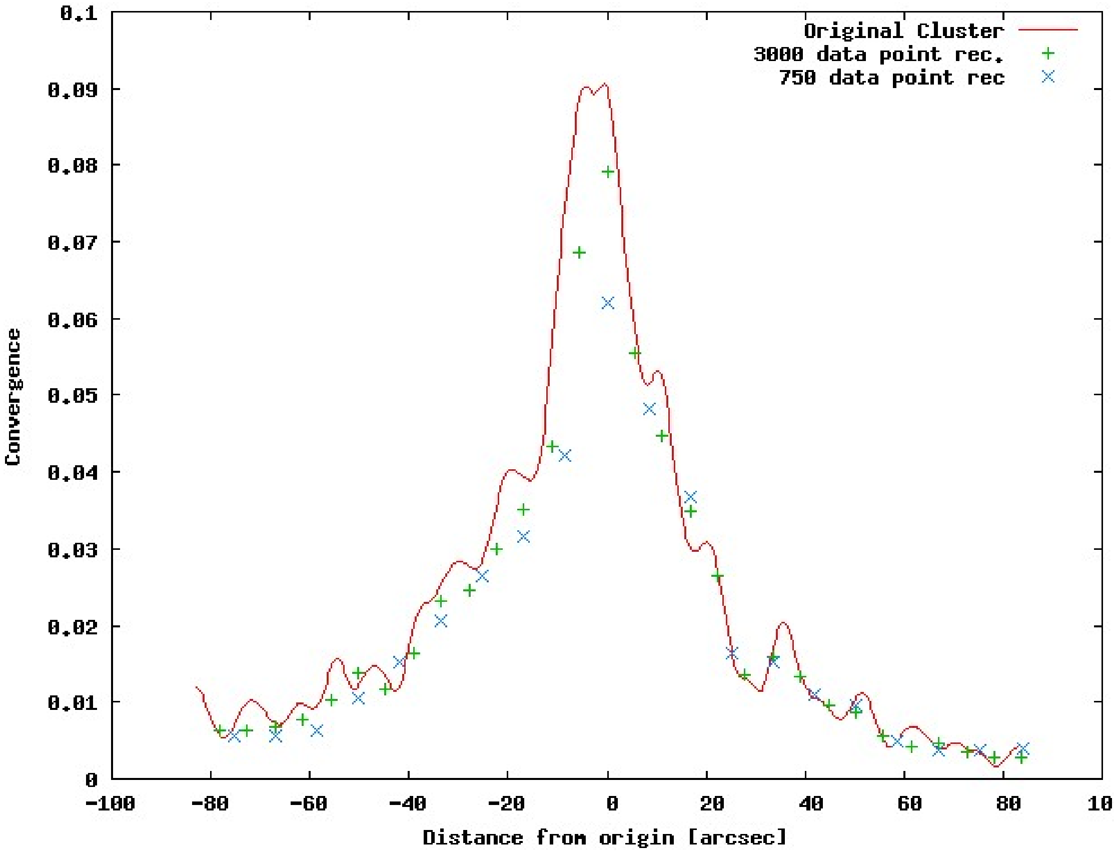


Figure 7.7: The κ-profile along the increasing main diagonal of the field.

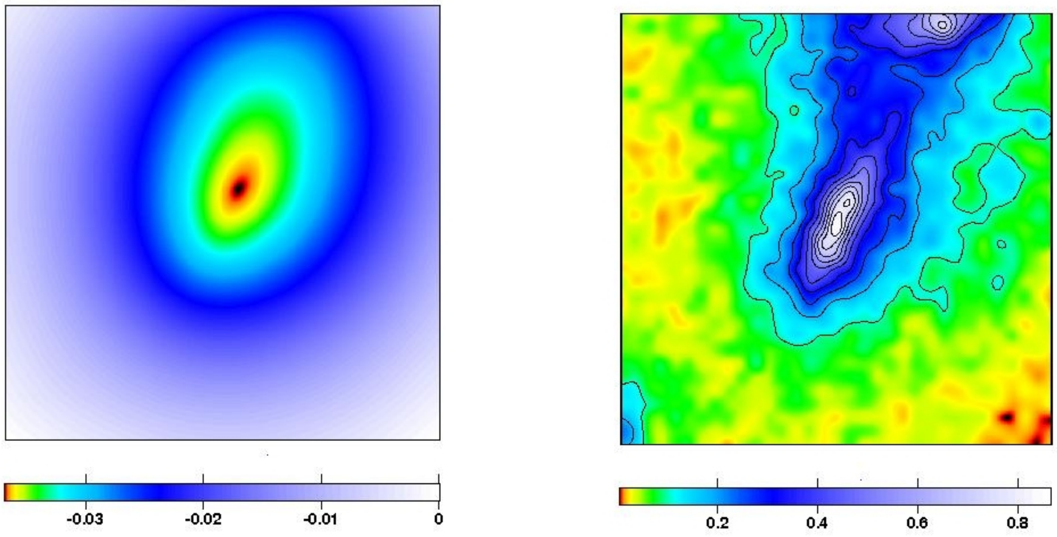


Figure 7.8: Lensing potential (left) and corresponding convergence map (right) of a critical galaxy cluster simulated in a ΛCDM scenario. The cluster redshift is $z_c = 0.52$ and the cube-side length of the simulation is 5 (comoving) Mpc, which corresponds to 3.13 arcmins. The total mass of the cluster is $1.1 \times 10^{15} M_{\odot}$. The contours in the right panel start at a convergence value of $\kappa = 0.1$ and end at $\kappa = 0.9$ with linear scaling.

7.2.3 A Critical Cluster Example

The second cluster used to test the reconstruction method is critical and develops an extended critical curve (see Fig. 7.8). With a cluster of this kind the joint reconstruction method can use its full potential. The weak lensing data were simulated just like before, this time with 2000 data-points. In addition, the critical curve of the simulated cluster was used during the reconstruction. The reduced shear catalog, together with the position of the critical curve, is shown in Fig. 7.9. The possible reconstruction resolution with this number of data points is (26x26) pixels, while averaging over 10 data points per reconstruction pixel. Because of the extended critical curve it was possible to add an additional high-resolution reconstruction step as described in Chapter 6. The resolution for this additional reconstruction step was (75x75) pixels zoomed into the cluster core.

The result of the joint reconstruction without the additional high-resolution reconstruction step can be seen in Fig. 7.10, again compared to the rebinned original cluster. The high resolution reconstruction result was inserted into the low resolution κ-map in Fig. 7.11. One should notice the increased resolution in the cluster core. The original map at (75x75) pixel resolution shows of course more detail in the outer parts of the cluster because in these areas the reconstruction cannot take advantage of the increased resolution. Again it is obvious that the reconstruction is in good agreement with the original cluster, even with the smoothed input catalog.

In Fig. 7.12 the κ-profile along a horizontal line through the cluster core is shown. The red line is the real cluster profile, taken from the full resolution κ-map. This time three types of reconstruction points are added, one showing the result from a pure weak-lensing reconstruction, one showing the result from a low-resolution combined weak and strong-lensing reconstruction and the third one showing the results from a high-resolution combined reconstruction. One can clearly see that we are now even able to reach the κ-peak of the simulated cluster. This is due to the addition of strong-lensing constraints. Even at low resolution, one sees a significant increment in the profile, and at high resolution the reconstruction matches the original cluster almost perfectly.

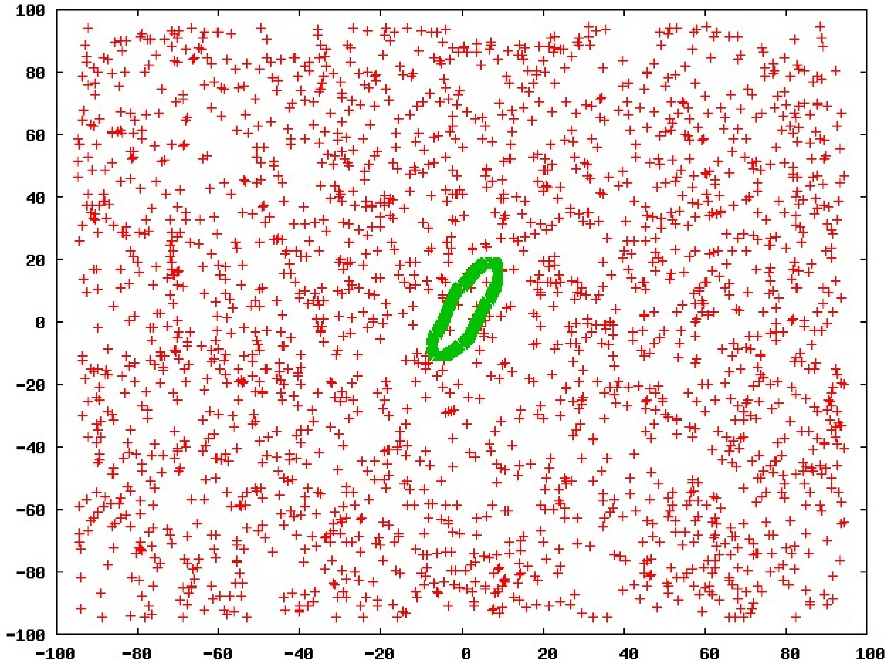


Figure 7.9: Reduced-shear catalog and critical curve of the critical cluster.

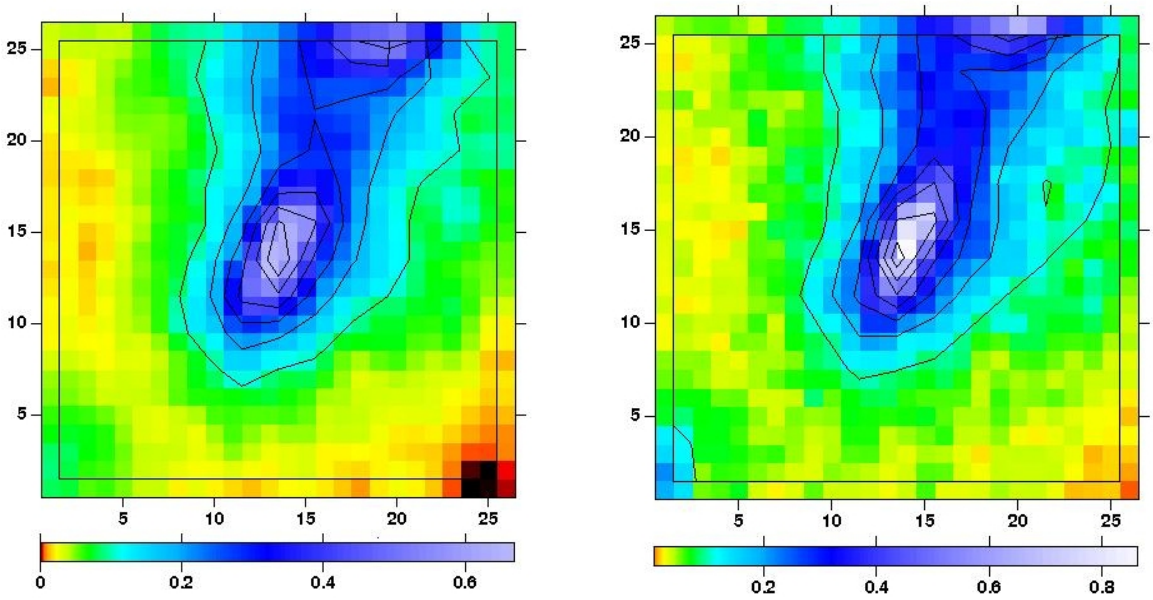


Figure 7.10: Left panel: Convergence map of the low-resolution reconstruction. The resolution is (26x26) pixels and the color scale is logarithmic. The contours start at $\kappa = 0.1$ and the spacing is $\Delta\kappa = 0.08$. Right panel: The convergence map of the original cluster rebinned to the reconstruction resolution. The parameters for color and contours are identical to the reconstruction.

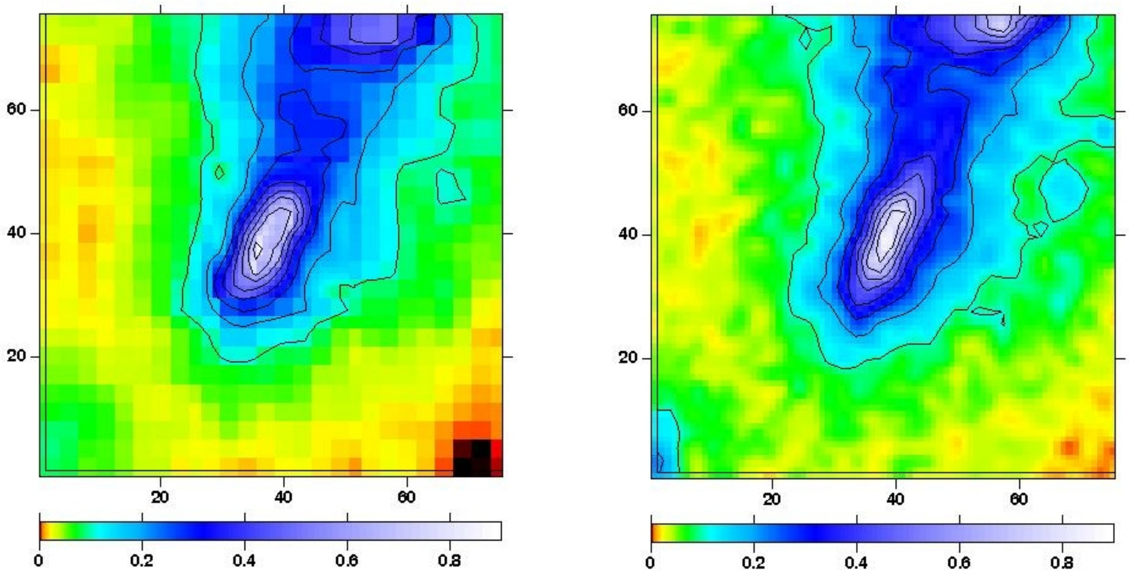


Figure 7.11: Left panel: Convergence map of the high-resolution reconstruction. The resolution has (75x75) pixels in the inner part of the map and the color scale is logarithmic. The contours start at $\kappa = 0.1$ and the spacing is $\Delta\kappa = 0.08$. Right panel: The convergence map of the original cluster rebinned to the reconstruction resolution. The parameters for color and contours are identical to the reconstruction.

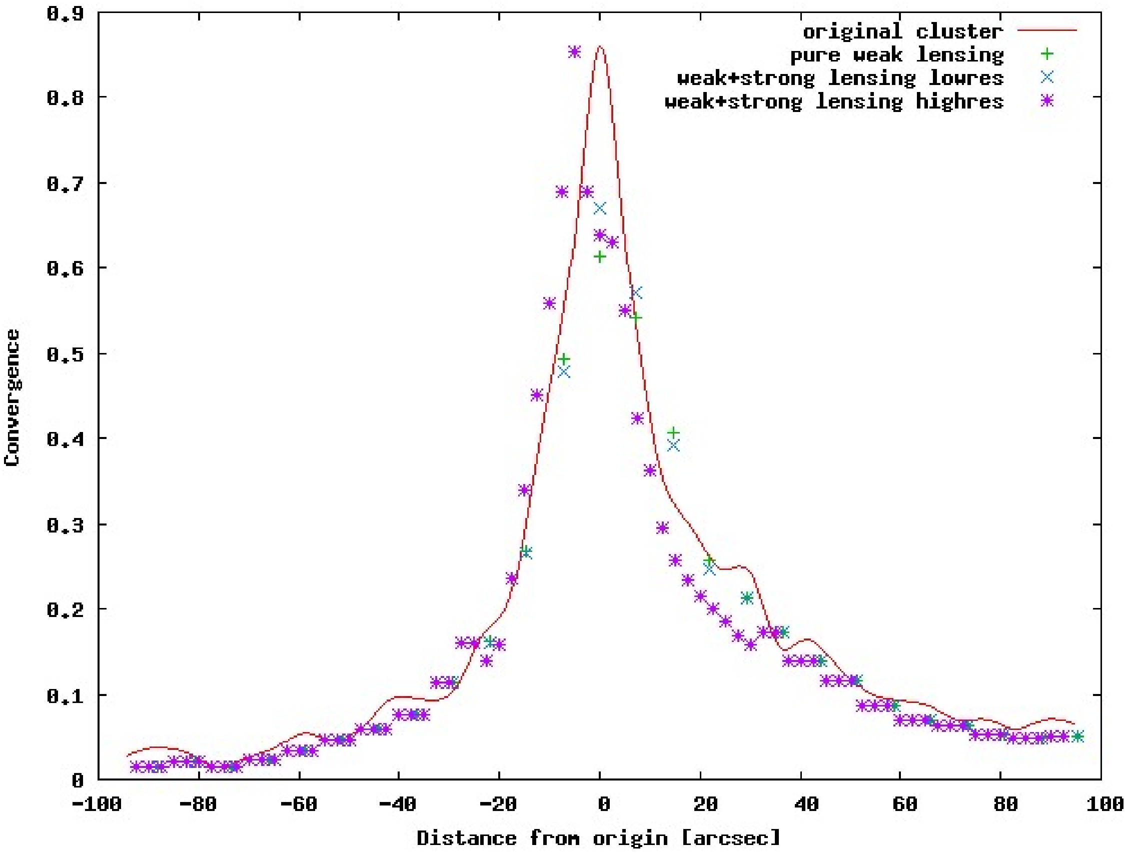


Figure 7.12: The radial κ-profile along the main diagonal of the critical cluster field. The slight shift to the left of the high resolution reconstruction is due to a one pixel inaccuracy of the routine which inserts the refined result into the low-resolution map.

7.3 Realistic Lensing Simulation

Even if the synthetic cluster data gave very good results and the tests were constructed to be close to reality, they can just serve as a proof of concept for several reasons. The main drawback of the performed reconstructions was the perfect input data. The lensing observables, reduced shear and critical curves were calculated from the simulation and for this reason idealised. A more realistic input for the reconstruction would be produced by a raytracing simulation between a number of individual sources and the observer with a given mass distribution in-between. This takes also into account source properties like intrinsic ellipticity which contaminate the weak lensing signal. In fact, intrinsic ellipticity is the reason why it is averaged over a number of background galaxies. It is also just an approximation that the expectation value of the image ellipticity is equal to the reduced shear of the lens, based on a locally linearised lens equation. In a raytracing simulation based on the full lens equation and the knowledge of the mass distribution of the lens the correct image distortion is calculated. This also leads to the fact that highly distorted images like arcs were not simulated yet, which should be the case, because the reconstruction method is based on arc positions, as long as the exact positions of the critical curves cannot be observed.

The second important point which was so far different to a real lensing analysis is that no real image frames of a galaxy cluster field were analysed in order to obtain the lensing signal. As was already pointed out, this step is crucial and highly non-trivial. Measuring the ellipticity of an image is difficult because one has to take into account image properties like e.g. size or surface brightness. Even separating different images from each other can be difficult. Furthermore, one has to correct for instrument errors and seeing conditions which means that one has to have good knowledge of the PSF model. Also the density of distorted background galaxies will decrease in the center of a galaxy cluster because of the foreground light of the BCG galaxy. So not only the reconstruction routines have to be tested in a real lensing scenario with a known cluster as deflector, but also the methods to obtain the lensing observables. This point was ignored so far.

7.3.1 Outline of the Method

To set up a realistic galaxy-cluster lensing situation, we used a lensing simulator developed by M. Meneghetti (INAF-Osservatorio di Bologna) which includes all the points which were listed above. With this simulator, we are in the position to test our reconstruction method under real conditions, while we still have full knowledge of the deflecting mass distribution.

We just want to give a short outline of how the simulator works; for a full review on the topic see Meneghetti et al. (2007b). As a starting point a set of sources is generated in a user-defined field-of-view at random positions and with random orientation. The distances of the sources, their morphological types and intrinsic magnitudes are chosen such that the luminosity function of the VIMOS VLT Deep Survey (Le Fèvre et al., 2005) is reproduced. The two-dimensional surface-brightness distribution of the source galaxies is modelled by a shapelet-approach, which means that the surface-brightness distribution is decomposed into a set of shapelet functions (see Refregier, 2003; Melchior et al., 2007) and described by the corresponding shapelet coefficients (see Fig. 7.13). In the simulator, these coefficients are again chosen in such a way that they adopt observed data, in this case the GOODS HST/ACS data (Giavalisco et al., 2004). After the sources are created, their images are mapped through the deflecting mass distribution onto the CCD of a specific instrument, using a ray-tracing code. The lens is arbitrary and can be obtained by N-body simulations or analytical models. Several ray-tracing implementations are possible, including tree algorithms, Fast-Fourier-transform methods and multiple-plane ray-tracing. Before the images are created, the code also simulates typical observational effects like the PSF and seeing in form of a convolution of the surface brightness. The form of the PSF-model can be chosen, which makes it possible to mimick a variety of different

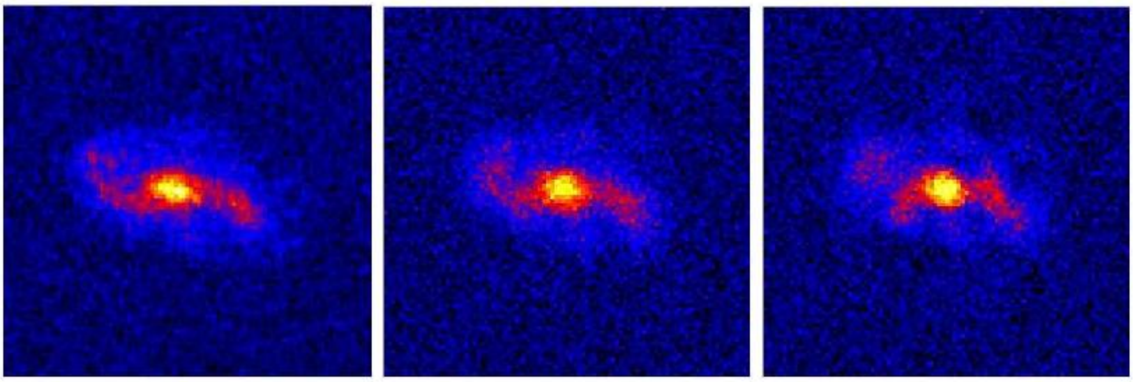


Figure 7.13: Example of generation of synthetic galaxies with shapelets. In the left panel the original image of a GOODS galaxy is shown. The galaxy is decomposed into 9 x 9 shapelets and reconstructed in the central panel. A second galaxy is then generated from the same shapelet decomposition by slightly modifying the coeffcients and displayed in the right panel. The same color scale is used for all three images. To reproduce the background and the noise properties of the image in the left panel, we have simulated an observation with HST/ACS (lter F850LP, texp = 10600 sec). Each frame is 2.9″ on a side. (from Meneghetti et al., 2007b)

existing and future instruments. Seeing is included by a convolution with a Gaussian kernel. Finally, other possible sources of noise, like the sky background or photon noise, are added and the final image is created.

7.3.2 The Input Data

With the method described above, a simulated (2501x2501) pixel CCD-image was created (see Fig. 7.14). The field size was 400″x 400″ and the PSF-model was taken from the Subaru telescope. The galaxy cluster which serves as a lens was taken from a N-body simulation and is described in detail by Dolag et al. (2005) and Meneghetti et al. (2007a) (see Fig. 7.14).
The weak lensing analysis of the field was done by M. Radovich (INAF-Osservatorio di Napoli) using the KSB method. It delivered an ellipticity catalog of 1826 background galaxies. Their distribution in the observation field can be seen in Fig. 7.16. The arc-positions were obtained by eye. A zoom of the original image on the area where the strong-lensing features are visible is shown in Fig. 7.15. We used five arcs in the reconstruction which are also marked in Fig. 7.16.

7.3.3 The Reconstruction

Two different reconstruction runs were performed at low resolution, both using weak and strong lensing constraints. The first run used ten background galaxies for one reconstruction pixel. The result looked quite noisy, due to the fact that simulated but realistic observational data was used. For this reason we decided to increase the number of background galaxies per reconstruction pixel to 15 for the second run. The quality of the result was increased significantly. Both κ-maps are shown in Fig. 7.17. For the second run also a high-resolution reconstruction on a refined grid was performed and the result is shown in Fig. 7.18. As one can see, it looks not as good as the results of the synthetic tests. Especially the outskirts of the cluster are not resolved at all, which is due to the fact that the measured weak-lensing signal is not reliable enough in those low-density areas of the cluster. Please note the wide field size of 400″x400″. Additional smoothing is necessary since noise effects clearly dominate the lensing signal. Promising is the good result for the central part of the cluster, where the main shape is reproduced well and also substructures are recovered. The size of the central κ-bulge is still underestimated but this should be fixed easily with improvements in the strong-lensing

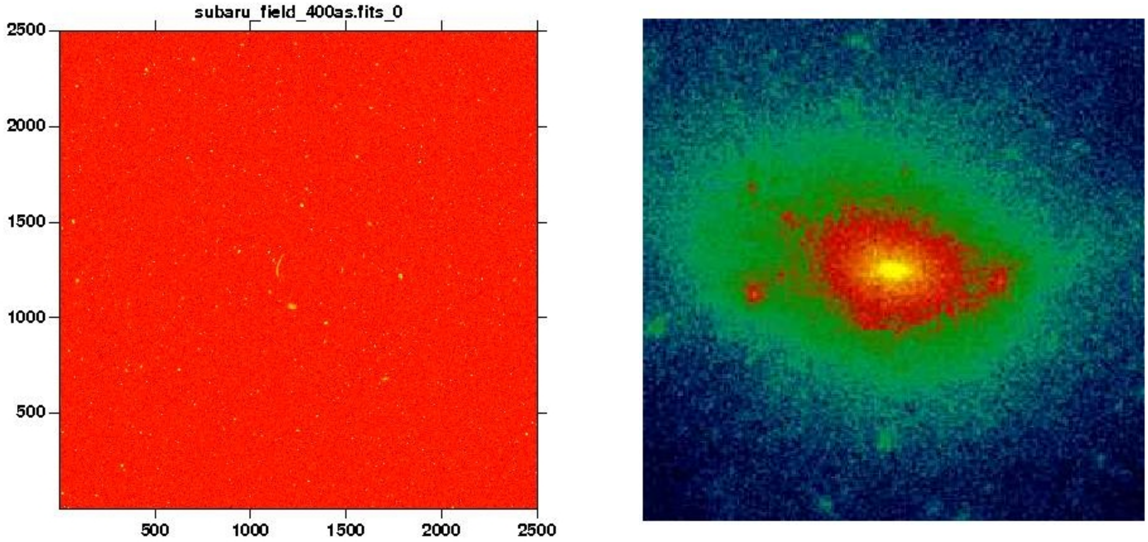


Figure 7.14: Left panel: The simulated CCD-image using the characteristics of the ground based Subaru telescope. The field size is 400″x400″. Right panel: The deflecting galaxy cluster. The side-length corresponds to 372″.

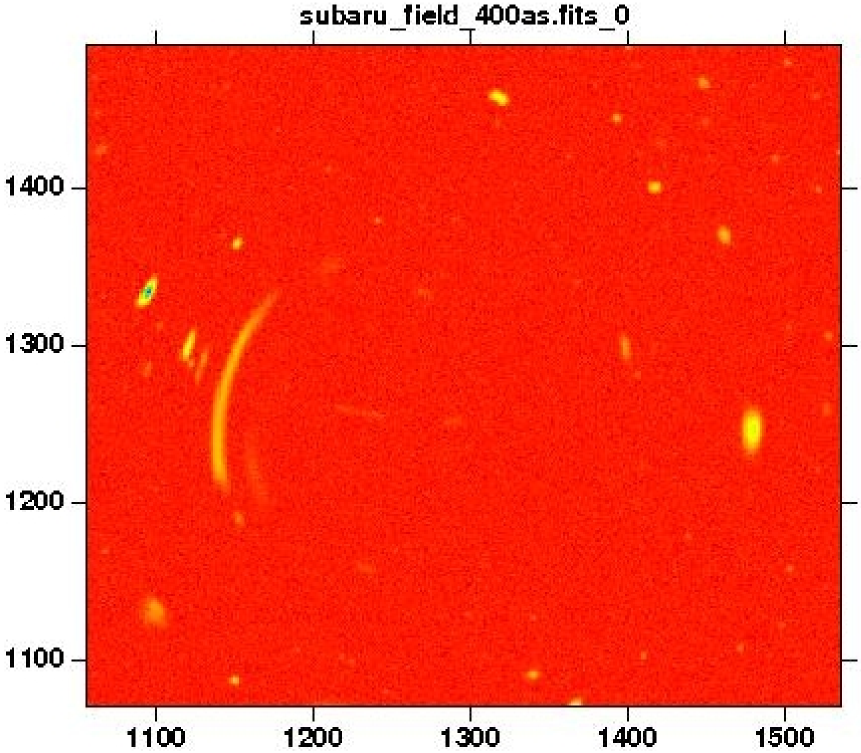


Figure 7.15: Zoom on the center of the field where the arcs are clearly visible. The size of the image is ∼ 50″x50″.

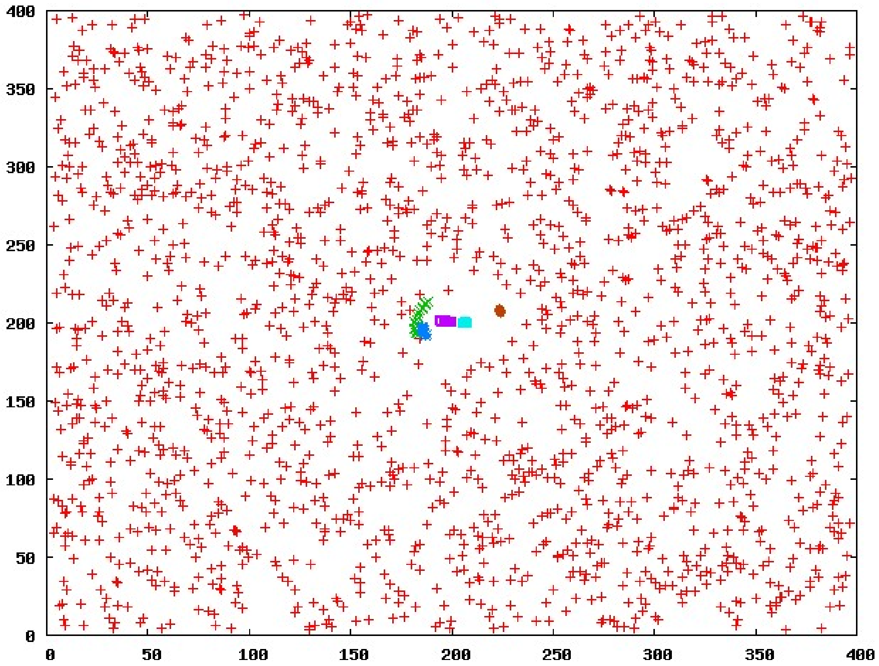


Figure 7.16: The distribution of the background galaxies and the position of the five arcs used during the reconstruction. The coordinates are given in arcsec relative to the lower right corner of the field.

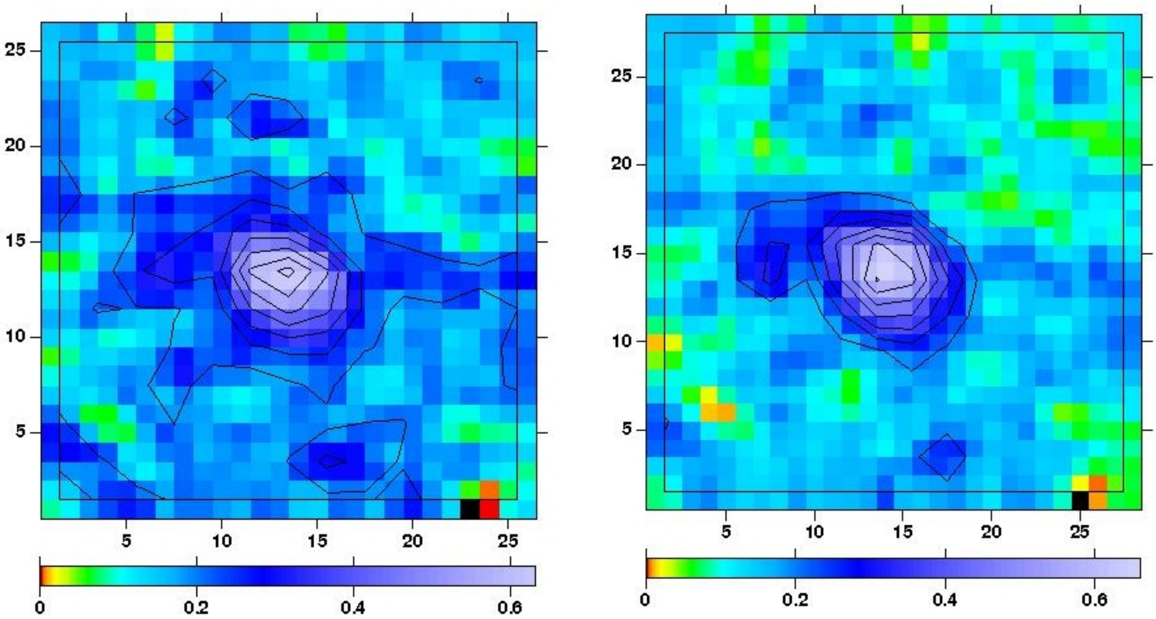


Figure 7.17: Left panel: Reconstructed κ-map using 10 galaxies per pixel. The resolution is (26x26) pixels. Right panel: κ-map using 15 galaxies on a resolution of (28x28) pixels. Both figures use logarithmic color scale. The contours start from $\kappa = 0.2$ with a spacing of $\Delta\kappa = 0.045$

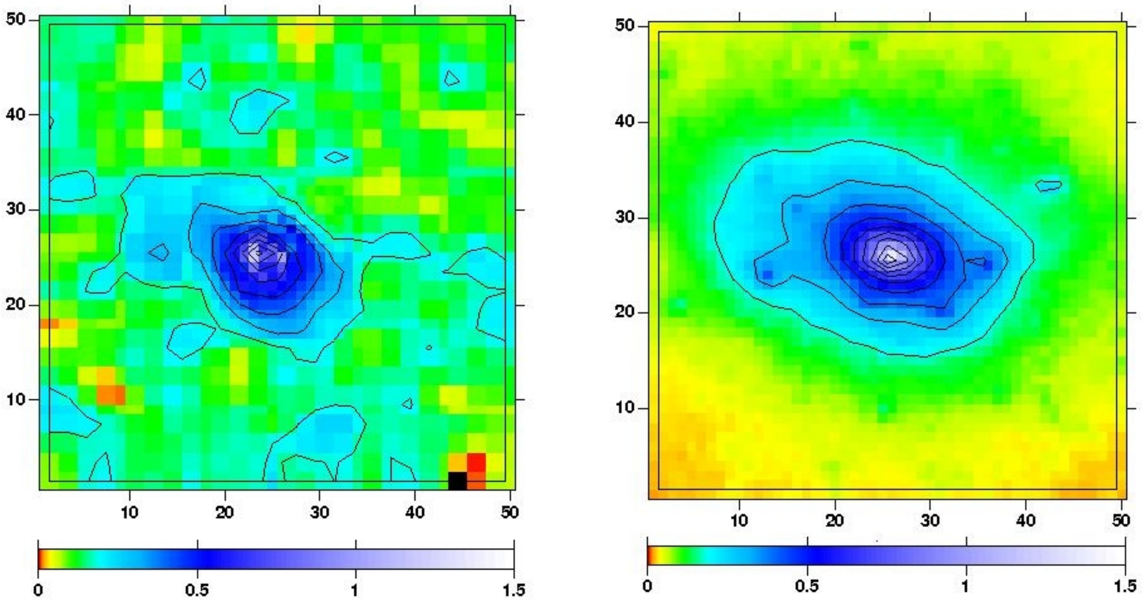


Figure 7.18: Left panel: Convergence-map of the high-resolution reconstruction. The color scale is logarithmic and the contours start from $\kappa = 0.18$ with a spacing of $\Delta\kappa = 0.092$. Right panel: Original cluster rebinned to the reconstruction resolution, color scale and contour properties are identical.

section of the reconstruction method. The pure focus on arc positions is a fair, but relatively coarse approach and will be improved in future work. Nevertheless the high-resolution reconstruction pushed the convergence in the center by a factor of two and improved the result tremendously, especially because in this realistic simulation a huge void dominates the field of distorted background galaxies. This example emphasizes how important the inclusion of strong-lensing constraints is in a reconstruction of a massive cluster. Unclear is the bad result in the area of pixel coordinate (45,35) where the original cluster shows substructure and the reconstruction a huge void.

We conclude that our reconstruction method works very well in areas where we can use both weak and strong-lensing constraints and where the weak-lensing signal is still high. Unfortunately, it is still significantly limited by the quality of the lensing signal in a real image analysis, which makes it necessary to calibrate the reconstruction method very carefully with the help of the lensing simulator. This will be one of the main tasks of future work.

7.4 MS 2137

At the end of this thesis the reconstruction routines are also applied to the real galaxy cluster MS2137. It is a rich cluster at redshift $z_c = 0.313$, dominated by a bright single cD-galaxy. It shows a giant, tangential arc and a radial arc. It was the first radial arc which was discovered (see Fort et al., 1992). Also observed are three additional arclets. The photometric redshift of all arcs was determined to be $z_{\rm arc} = 1.6 \pm 0.1$. The tangential arc and two of the arclets belong to the same source. Also the radial arc produces one counter-image. The cluster was studied in several works, see Gavazzi et al. (2003); Gavazzi (2005); Comerford et al. (2006) and Sand et al. (2007). All these studies used parametric reconstruction routines which differ from our method. A detailed comparison with their results will be the subject of future work but is out of the scope of this thesis for the following reasons. As mentioned, parametric approaches were used, which rely on a certain mass distribution model, which makes a direct comparison difficult since we produce a density map. Another problem is that we believe that our methods needs more calibration with the simulator developed by Meneghetti et al. to make a legitimate

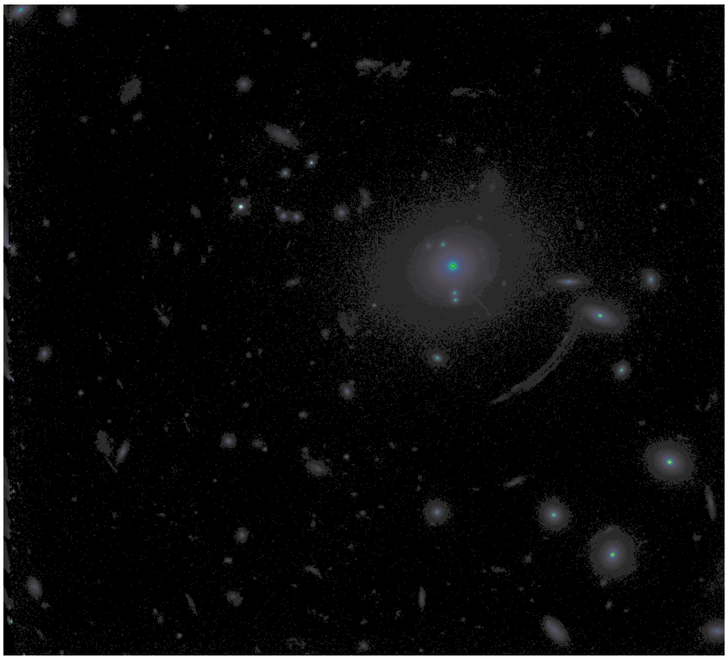


Figure 7.19: The field of MS2137 shown in a HST/WFPC2 image. The huge tangential and the radial arc are clearly visible.

statement about the mass distribution of the cluster. Furthermore a comprehensive error analysis has to be implemented which is not trivial in a non-parametric likelihood reconstruction. All these points will be adressed in the continuation of this work.

7.4.1 The Input Data

The weak-lensing analysis is based on an VLT/FORS observation during summer 2001 and were kindly provided by R. Gavazzi (IAP, Paris). The field size was 405″x405″. The ellipticity catalog was obtained with a KSB-method and provided 1500 background-galaxy ellipticity measurements. The arc positions were obtained by a HST/WFPC2 exposure using the F702W filter. It is shown in Fig. 7.19. The field of background galaxies, together with the arc positions in coordinates relative to the cD-galaxy, are shown in Fig. 7.20.

7.4.2 The Reconstruction

During the reconstruction the correct arc redshifts of $z_{\mathrm{arc}} = 1.6$, which are fortunately very well known in the case of MS2137, were used. Based on the experience from the simulated tests, we averaged over 15 galaxies per reconstruction pixel. The low-resolution reconstruction was performed on a (25x25) pixel grid (see Fig. 7.21), which was refined afterwards to (75x75) pixels for a reconstruction which resolves the arcs better (see Fig. 7.22). Unfortunately there is large void in the observational background galaxy data in the upper middle part of the field. In this area also the reconstruction is not able to resolve any structures while this is the case in the other parts of the field. But still our results agree with former reconstructions which show that MS2137 is a quite radially symmetric and relaxed cluster. Furthermore the smooth κ-map supports the application of our method to real data.

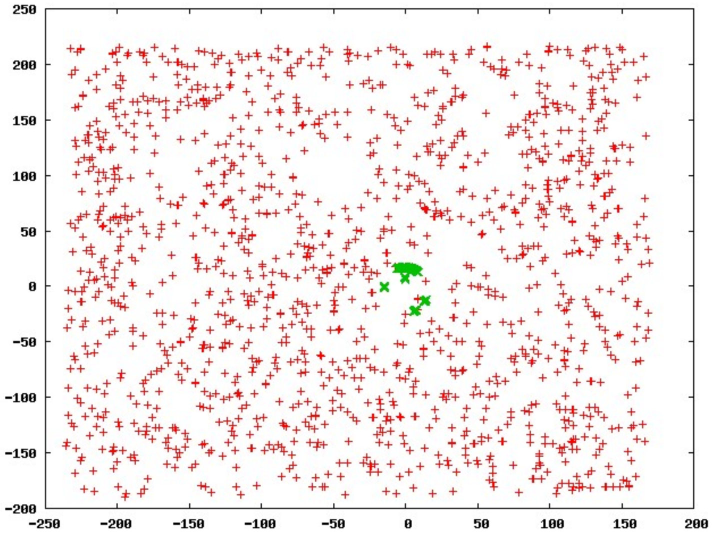


Figure 7.20: The field of background galaxies and the positions of the five arcs in arcsec relative to the cD-galaxy of the cluster.

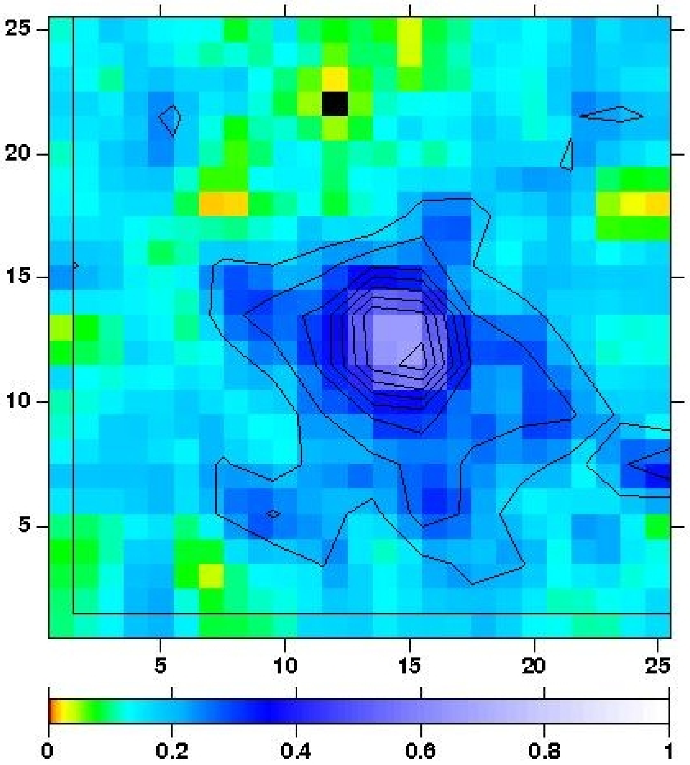


Figure 7.21: Low resolution reconstruction (25x25 pixel) κ-map. The side-length corresponds to 1.8 Mpc. The color scale is logarithmic and the contours start at $\kappa = 0.02$ with a spacing of $\Delta\kappa = 0.045$.

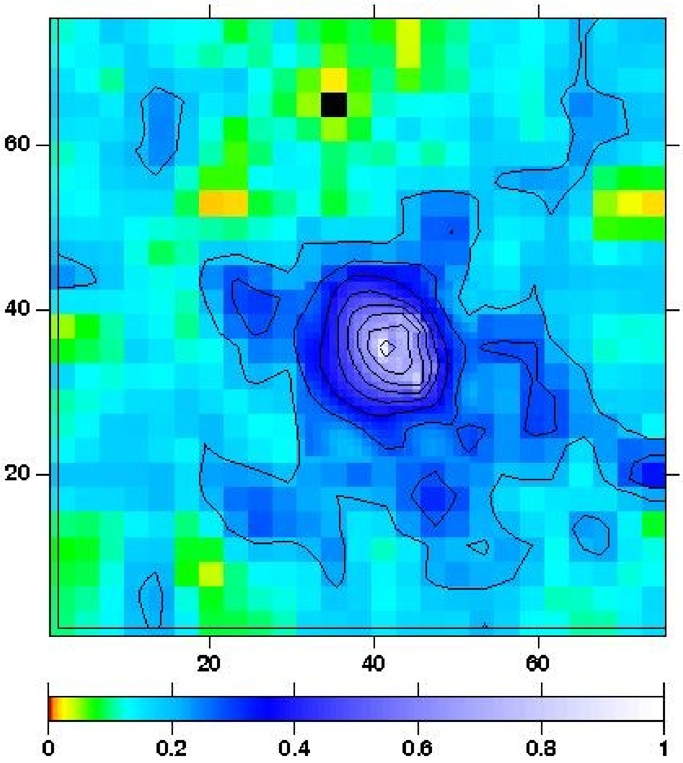


Figure 7.22: High resolution reconstruction on a refined grid (75x75 pixel). The contours start at $\kappa = 0.2$ with a spacing of $\Delta\kappa = 0.07$

8 Conclusions and Outlook

In this work we presented in great detail a newly developed method for galaxy cluster-mass reconstruction, based on a combination of weak and strong gravitational lensing. One of the biggest advantages of our method ist the fact that it is fully non-parametric, and allows a reconstruction of the mass distribution of a galaxy cluster without any a priori assumptions on the density profile. Our reconstruction is purely based on observational data which gives it an advantage over classical parametric approaches. Of course, especially weak-lensing reconstruction methods also use non-parametric approaches, but they cannot incorporate additional strong lensing constraints, which results in an extreme underestimation of the mass distribution in the central regions of the cluster. We also want to point out that our method is easily extendable, due to its maximum-likelihood nature.

The numerical implementation is in principle just a technical detail, but we showed that the χ^2-minimisation can be performed by simply solving a linear system of equations. Linear systems are very well known in numerical mathematics, which means that reliable and fast algorithms are already implemented. Furthermore we put considerable effort into developing routines which build up these linear systems as fast as possible and thus allow a complicated two-level-iteration scheme which accounts for the non-linearities in the reduced shear and for avoiding an overfitting of intrinsic noise patterns in the input data.

The application of the method to data showed several things. First of all the method gives almost perfect results when applied to ideal input data. Also in the application to synthetic data, it became obvious how important the inclusion of strong lensing is in the reconstruction of critical clusters, in order not to underestimate the central peaks of the mass distribution. Very valuable was the test with the lensing simulator which gave very good results for the reconstruction of areas with high or intermediate mass, but performed worse in the outskirts of the galaxy cluster where the lensing signal is weak. Also our method would need additional smoothing here. The application to the real galaxy cluster MS 2137 showed similar features like in the κ-map of the cluster in the scenario created by the lensing simulator which make us assume that the results in the more central regions of the cluster are reliable.

Even if our method already gives remarkable results, there is still much room for improvements. Future work should be structured as follows. First, we want to implement an additional strong-lensing constraint provided by multiple image systems. Their addition was already shown by Bradač et al. (2005) and can be included easily in our linear system approach. The drawback is that multiple image systems are much harder to identify than simple arc positions, but automated algorithms to find these systems are already announced in Bradač et al. (2007). What we also want to improve is the usage of arc positions. With our method right now we are only able to improve the reconstruction of a few points on the reconstruction grid. The use of additional information will improve the reconstruction of the complete cluster center. An exciting issue for the future might be gravitational flexion, since more and more methods arise which are possibly able to measure a flexion signal in the observed data. Finally we should include a more careful treatment of the mass-sheet-degeneracy for example by using a magnification signal and implement a rather simple Kaiser-Squires routine to obtain an initial guess as starting condition for the maximum-likelihood-reconstruction routines.

Extensive usage of the lensing simulator will be very important in the future especially for the calibration of our method. This tool is very useful with respect to the application to real data because we are able to simulate the characteristics of any instrument, and we can specifically calibrate our code before applying it to real data delivered by this instrument. After calibration, we should be able to draw reliable conclusions ON the mass-distribution of real galaxy clusters. Since we want to use our method for real data, a close collaboration with groups involved in image analysis will become necessary since our code is highly dependent on the quality of the lensing signal. This field is also very active right now and the development of reliable algorithms for measuring the shear signals is by far not completed yet.

Acknoledgments

With the last sentences of this work I want to mention those persons who made my work a lot easier and more convenient. First of all, many and special thanks go to Matthias Bartelmann who was a continuous source of information and advice, not only in questions which are directly related to this thesis. I also want to thank Matthias because he gave me the opportunity not only to write my thesis in his research group, but also to continue my work at ITA Heidelberg in the future. In addition I am glad that Joachim Wambsganß agreed to be the second corrector of this thesis. I'm deeply grateful towards Marcello Cacciato and Massimo Meneghetti who helped me to understand and develop the principles of our reconstruction method. Of course I want to thank the complete cosmology-group at ITA for a lot of fun with and without an astrophysical background. So, thank you Matteo, Ana (good luck in Munich), Jean-Claude, Irina, Cosimo (good luck in Bologna and for Milan), Christian, Emanuel, Ewald (good luck in Munich), René, Peter M., Peter E. (good luck in Walldorf), Mischa, Martin (good luck in St. Andrews) and Madeleine. Particular thanks go to Claudia and Ciccio for finding an awful amount of typos and comma-mistakes. Some greek guys did some calculations for me, thanks a lot and I hope they appreciate. Finally I want to thank all my friends and especially my family for their support when I was fed up with astrophysics.

A Appendix: C++ Implementation

By intention we did not show concrete source code. The complete package is ∼ 10000 lines long so it would be useless to describe code details. What we want to do is showing the most important classes of the code and a part of the MPI-version of the driver routine.

A.1 The Inputfield-Class

```
//FINAL MPI
//inputfield.h
//defines the inputfield class which deals with data preparations wrt
//to ellipticity data and ccurve positions
//methods in inputfield.cpp


#ifndef          INPUTFIELD_H_
# define         INPUTFIELD_H_

#include <iostream>
#include <fstream>
#include <string>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_statistics.h>
#include <gsl/gsl_linalg.h>
#include <cmath>
#include "rw_fits.h"
#include "util.h"
#include "soph_math.h"

using namespace std;

class AdaptiveInputField
{
 private:

  string name;
  //name of cluster
  const char *filename;
  //filename of ellip-data ascii-file
  const char *ccurvefilename;
  //same for ccurve-position ascii-file
  int numgal;
  //number of available galaxies
  int suffnumgal;
  //number of galaxies to average over per pixel
```

```
int numxpixels;
//number of wanted pixels in x-direction, y defined
//by fieldsize
int numccurvepts;
//num of available ccurve positions
double fieldx;
//fieldsize in x-direction in whatever units
double fieldy;
double fieldsize;
double minrec;
//smallest available rec-position of a galaxy in field
double mindec;
double maxrec;
double maxdec;
int x_dim;
//number of x-pixels in grid-map
int y_dim;
double pixelsize;
//sidelength of a pixel in whatever units
double galdensity;
//galaxy density of the field
double completemeanellip1;
//average value of overall ellip1
double completemeanellip2;
double ellip1variance;
//variance of overall ellip1 sample
double ellip2variance;
bool covariancecheck1;
//flag if inverse of covaraincematrix is useable
bool covariancecheck2;
gsl_vector *rec;
//position catalogue in rec
gsl_vector *dec;
gsl_vector *ellip1;
//ellip1 catalogue
gsl_vector *ellip2;
gsl_matrix *meanellip1;
//ellip1 for every pixel
gsl_matrix *meanellip2;
gsl_matrix *sigmaellip1;
//variance of pixelellip1
gsl_matrix *sigmaellip2;
gsl_matrix *ellip;
//modulus of ellip per pixel
gsl_matrix *sigmaellip;
//variance of ellipmodulus
gsl_matrix_int *galaxyowners;
//what galaxy uses a pixel
gsl_matrix_int *galaxyshare;
//which pixels share galaxies and how many
gsl_matrix *covariance;
```

```
  //forgot for what this is, but afraid to delete it
  gsl_matrix *ellip1covariance;
  //covariance matrix for ellip1
  gsl_matrix *ellip2covariance;
  gsl_vector *ccurverec;
  //ccurvepositions in rec
  gsl_vector *ccurvedec;
  gsl_matrix_int *ccurve;
  //ccurve in pixels on given resolution
  gsl_matrix *ccurveerror;
  //error on ccurve


 public:

  AdaptiveInputField(string,const char*,int,int);
  //standard constructor, needs name, ellipfile
  //and numpixels and suffnumgal
  AdaptiveInputField(string,const char*,int,int,double,double,double,double);
  //same as above, but a cutout-window is given by the 4 corners
  ~AdaptiveInputField();
  //destructor, frees memory
  void average(double);
  //adaptive averaging, radius increment has to be given
  void docovariance();
  //calculates covariance matrices
  void ccurvebuild(const char*);
  //marks the critical curve
  //ccurve-catalog-file has to be given
  void writetofits(string,int);
  //writes field-fits-file, selections are 0 for pure ellip info,
  //1 for pure ccurve and else for both
  void showinfo(int);
  //gives field info to console, same slection as above
  void showinfo(int,const char*);
  //prints to given ascii file again selections possible
};

#endif      /* !INPUTFIELD_H_ */
```

A.2 The GalaxyCluster-Class

```
//FINAL MPI
//galaxycluster.h
//Defines the GalaxyCluster class used to save all
//the important informations of a cluster wrt lensing
//The objects of this class are updated every iteration step
//methods in galaxycluster.cpp

#ifndef        GALAXYCLUSTER_H
# define       GALAXYCLUSTER_H

#include <gsl/gsl_matrix.h>
#include <string>
#include "fin_dif.h"
#include "util.h"
#include "rw_fits.h"


using namespace std;

class GalaxyCluster
{

 private:

  int x_dim;
  //size in x-dimension of the grid which represents cluster
  int y_dim;
  double x_frct;
  //fraction of grid size wrt total field size
  gsl_matrix *pot;
  //grid which represents cluster potential
  gsl_matrix *shear1;
  //first component of shear
  gsl_matrix *shear2;
  gsl_matrix *conv;
  //convergence
  gsl_matrix *jacdet;
  //jacobian determinant
  gsl_matrix *reredshear;
  //first component of reduced shear
  gsl_matrix *imredshear;
  gsl_matrix_int *ccurve;
  // seems to have become useless

  public:

  GalaxyCluster(int, int, double);
  //standard constructor, needs x_dim ,y_dim and x_frct
  ~GalaxyCluster();
  //destructor, frees memory
  void buildfrompot();
  //builds all other properties out of potential
  void buildwithoutpot();
  //doing the same with given conv and shear
  void masssheetnormalise();
  // doing mass-sheet-trafo by setting most negative conv to 0
  void buildborders();
  //migrated to fin_dif
  void readfrom(string, gsl_matrix*);
  //reads from external gsl,selections are pot,
  //shear1/2,conv,jacdet,reredahear, imredshear
  void readfrom(string, gsl_matrix_int*);
  //reads the int gsls
  void readfrom(string, gsl_vector*);
  //same in vector format
  void readfrom(string, gsl_vector_int*);
  void writeto(string, gsl_matrix*);
  //writes to external gsl
  void writeto(string,gsl_matrix_int*);
  void writeto(string, gsl_vector*);
  void writeto(string,gsl_vector_int*);
  void edit(string, int, int, double);
  //edits point of selection at position i,j
  void edit(string,int,int,int);
  void readfromfits(string, string);
  //reads selection from external fits pHDU, needs filename
  void readfromfits(string,string,string);
  //reads selection from external fits extension,
  //needs filename and extension
  double show(string,int);
  double show(string,int,int);
  //prints value of selection at i,j
  int showx();
  //prints x_dim
  int showy();
  double showxfrct();
  int showdim();
  void writetofits(string, string);
  //writes selection to fits pHDU, selection all also possible,
  //which writes all quantities

  void writetofits(string,string,string);
  //writes to fits extension
};

#endif      /* !GALAXYCLUSTER_H */
```

A.3 The ClusterCore-Class

```
//FINAL MPI
//clustercore.h
//Defining the stronglensing relevant class which describes
//the core of the Galaxy Cluster
//final analysis uses adaptive grids

#ifndef        CLUSTERCORE_H_
# define       CLUSTERCORE_H_

#include <iostream>
#include <fstream>
#include <gsl/gsl_matrix.h>
#include <string>
#include "galaxycluster.h"
#include "adaptivefindif.h"

using namespace std;

class ClusterCore
{
 private:

  int x_dim;
  //size of the x-dimension of the cluster spanning grid
  int y_dim;
  double x_frct;
  //fraction of size wrt coarse grid in x
  int adaptivepts;
  //# of points interesting for adaptive grid
  int maxgap;
  //maximum separation between points of the adaptive grid
  gsl_matrix_int *type_map;
  //matrix which declares pixel properties
  gsl_matrix_int *adaptive_info;
  //essential core of the adaptive grid,
  // contains all necessary information esp. position info
  gsl_matrix *pot;
  //potential of core grid
  gsl_matrix *conv;
  //same in convergence
  gsl_matrix *shear1;
  gsl_matrix *shear2;
  gsl_matrix *jacdet; s
  gsl_matrix *redshear1;
  gsl_matrix *redshear2;
  gsl_matrix *ellip1;
  //more or less open slot eg. for ellip infos on clustercore
  gsl_matrix *ellip2;
  gsl_matrix *elliperror;
 public:

  ClusterCore(int,int,double);
  //standard constructor, needs x_dim, y_dim, x_frct
  ~ClusterCore();
  //destructor, clears memory
  int showx();
  int showy();
  double showxfrct();
  int showdim();
  int numadaptive();
  //prints adaptivepts
  int showmaxgap();
  //shows the biggest separation
  void readfrom(string, gsl_matrix*);
  void readfrom(string, gsl_matrix_int*);
  void readfrom(string, gsl_vector*);
  void readfrom(string, gsl_vector_int*);
  void writeto(string, gsl_matrix*);
  void writeto(string, gsl_matrix_int*);
  void writeto(string, gsl_vector*);
  void writeto(string, gsl_vector_int*);
  void readfromfits(string,string);
  void readfromfits(string,string,string);
  void writetofits(string, string);
  void writetofits(string,string,string);
  void buildfrompot();
  void buildwithoutpot();
  void addccurve(gsl_matrix_int*);
  //adds the critical courve to type_map,
  // makes possible to build up adaptive grid
  void buildborders();
  //builds borders in type_map, without ccurve,
  //makes it possible to use adapt fin dif grid
  //on clustercore withhout ccurve
  void buildadaptive();
  //crucial routine which builds necessary
  //borders and adaptive_info, also defines adptivepts
  void writeadatable(string);
  //writes adaptive_info to dat
  void giveadaptivecomponents(gsl_vector*, gsl_vector*,
  gsl_vector*,gsl_vector*,gsl_vector*,
  gsl_vector*,gsl_vector*);
  //gives the reconstruction relevant values like conv and shear
  //in adaptive form with length adaptivepts
  void insertadaptiveresults(gsl_matrix*,
  gsl_matrix*,gsl_matrix*);
  //the other way round, inserts conv and shear in cluster
};
```

A.4 The Finite-Differences-Class

```
//FINAL MPI
//fin_dif.h
//declares the very importan fin_dif class which describes a finite
//differencing matrix to approximate second derivatives


#ifndef         FIN_DIF_H_
# define        FIN_DIF_H_

#include <string>
#include <cmath>
#include <gsl/gsl_matrix.h>
#include "util.h"

using namespace std;

class Fin_dif_grid
{

 private:

    int x_dim;
    //size in x of the described grid
    int y_dim;
    double x_frct;
    gsl_matrix_int *type;
    //auxilliary map to declare corners and borders
    gsl_vector_int *type2;
    //just a conversion to vector for convenience

 public:
    Fin_dif_grid(int, int, double);
    //standard constructor needs x_dim ,y_dim ,x_frct
    ~Fin_dif_grid();
    //destructor frees memory
    int showx();
    int showy();
    int dim();
    double s1value(int, int);  /
    //gives value of shear1 fin_dif-matrix
    //at position i,j
    double s2value(int, int);
    //same in shear2
    double cvalue(int, int);
    //same in convergence
    void s1write(gsl_matrix*);
    //writes shear1 matrix to external gsl
    void s2write(gsl_matrix*);
    void cwrite(gsl_matrix*);
    void s1multvec(gsl_vector*, gsl_vector*);
    //gives result-vector of s1matrix*external gsl
    void s2multvec(gsl_vector*, gsl_vector*);
    void cmultvec(gsl_vector*, gsl_vector*);
    double s1multvecswitched(int, gsl_vector*);
    //gives value of
    //vector external gsl * s1mattrix at position i
    double s2multvecswitched(int, gsl_vector*);
    double cmultvecswitched(int, gsl_vector*);
    double ccproduct(int,int,gsl_vector*);
    //gives value of external
    //external gsl *cmatrix * transposed cmatrix at i,j
    double s1s1product(int,int,gsl_vector*);
    double s2s2product(int,int,gsl_vector*);
    double Blkterm(int,int,gsl_vector_int*,gsl_vector*,
    gsl_vector*,gsl_matrix*,gsl_matrix*,gsl_vector*,gsl_vector *,
    gsl_vector *,gsl_vector *,int);
    //gives the coefficient matrix terms, at pos i,j
    //selections are
    //0-15 for weak lensing terms
    //50-52 for regularisation terms
    //100-102 for strong lensing terms
    double Vlterm(int,gsl_vector_int*,gsl_vector*,gsl_vector*,
    gsl_matrix*,gsl_matrix*,gsl_vector*,gsl_vector*,
    gsl_vector *,gsl_vector*,gsl_vector*,gsl_vector*,gsl_vector *,int);
    //gives the result vertor terms at pos i,
    //selections are
    //0-7 for weak lensing
    //50-52 for regularisation
    //100 for strong lensing
};

#endif      /* !FIN_DIF_H_ */
```

A.5 Some of the MPI-Driver-Routine

```
//sawmpi_lowres.cpp
//main driver of the mpi version
//shown is the initial reconstruction,
//one complete outer-level iteration
//and the following interpolation

#include <iostream>
#include <cmath>
#include <gsl/gsl_matrix.h>
#include "rw_fits.h"
#include "mpi.h"
#include "mpi_coreroutines.h"
#include "mpi_comm.h"
#include "galaxycluster.h"
#include "reconstruction_kernel.h"
#include "nrinterpol.h"
#include "util.h"

using namespace std;

int main(int argc, char** argv)
{
  int my_rank;
  int p;


  int startdim = 10;
  int finaldim = 26;
  double reg = 50.0;
  double convthresh = 0.001;

  //object declarations

  gsl_vector *ellip1 = gsl_vector_calloc(startdim*startdim);
  gsl_vector *ellip2 = gsl_vector_calloc(startdim*startdim);
  gsl_vector_int *ccurve = gsl_vector_int_calloc(startdim*startdim);
  gsl_vector *dummy = gsl_vector_calloc(startdim*startdim);
  gsl_matrix *Cij1 = gsl_matrix_calloc(startdim*startdim,startdim*startdim);
  gsl_matrix *Cij2 = gsl_matrix_calloc(startdim*startdim,startdim*startdim);
  double strongsigma;
  gsl_vector *eta = gsl_vector_calloc(startdim*startdim);
  gsl_vector *wredshift = gsl_vector_calloc(startdim*startdim);
  gsl_vector *sredshift = gsl_vector_calloc(startdim*startdim);

  gsl_vector *recpot = gsl_vector_calloc(startdim*startdim);
  gsl_vector *recconv = gsl_vector_calloc(startdim*startdim);
  gsl_vector *recshear1 = gsl_vector_calloc(startdim*startdim);
  gsl_vector *recshear2 = gsl_vector_calloc(startdim*startdim);
  gsl_vector *refconv = gsl_vector_calloc(startdim*startdim);
gsl_vector *refshear1 = gsl_vector_calloc(startdim*startdim);
gsl_vector *refshear2 = gsl_vector_calloc(startdim*startdim);
gsl_vector *control1 = gsl_vector_calloc(startdim*startdim);
gsl_vector *control2 = gsl_vector_calloc(startdim*startdim);

gsl_matrix *proccoeff = gsl_matrix_calloc(startdim*startdim,startdim*startdim)
gsl_vector *procdata = gsl_vector_calloc(startdim*startdim);


gsl_matrix *Fij1 = gsl_matrix_calloc(startdim*startdim,startdim*startdim);
gsl_matrix *Fij2 = gsl_matrix_calloc(startdim*startdim,startdim*startdim);
gsl_vector *strongfactor = gsl_vector_calloc(startdim*startdim);
double control;

//MPI start

MPI_Init(&argc,&argv);
MPI_Comm_rank(MPI_COMM_WORLD,&my_rank);
MPI_Comm_size(MPI_COMM_WORLD,&p);

//initial reconstruction assuming conv=0

if(my_rank == 0)
  {
    GalaxyCluster cluster1(startdim,startdim,1.0);

    read_pimg("field10.fits",ellip1);
    read_imge("field10.fits","av_ellip2",ellip2);
    read_imge("field10.fits","ellip_covariance1",Cij1);
    read_imge("field10.fits","ellip_covariance2",Cij2);
    read_imgeint("field10.fits","CCurve-Pos",ccurve);
    read_imge("field10.fits","CCurve-Error",dummy);
    strongsigma = gsl_vector_get(dummy,0);
    gsl_vector_set_all(wredshift,1.0);
    gsl_vector_set_all(sredshift,1.0);
    gsl_vector_set_all(eta,reg);

    weak_init(startdim,startdim,1.0,ccurve,ellip1,ellip2,Cij1,
    Cij2,refconv,refshear1,refshear2,eta,strongfactor,
    wredshift,sredshift,recpot);
    cluster1.readfrom("pot",recpot);
    cluster1.buildfrompot();
    cluster1.masssheetnormalise();
    cluster1.writeto("conv",recconv);
    cluster1.writeto("shear1",recshear1);
    cluster1.writeto("shear2",recshear2);
    cluster1.writetofits("all","resultmpi10_init.fits");
  }
//sending the result

send_gsl_toworld(recconv,startdim*startdim,p,my_rank);
```

```
gsl_vector_memcpy(refconv,recconv);
send_gsl_toworld(recshear1,startdim*startdim,p,my_rank);
gsl_vector_memcpy(refshear1,recshear1);
send_gsl_toworld(recshear2,startdim*startdim,p,my_rank);
gsl_vector_memcpy(refshear2,recshear2);
MPI_Barrier(MPI_Comm MPI_COMM_WORLD);

read_pimg("field10.fits",ellip1);
read_imge("field10.fits","av_ellip2",ellip2);
read_imge("field10.fits","ellip_covariance1",Cij1);
read_imge("field10.fits","ellip_covariance2",Cij2);
//read_imgeint("field10.fits","CCurve-Pos",ccurve);
read_imge("field10.fits","CCurve-Error",dummy);
strongsigma = gsl_vector_get(dummy,0);
gsl_vector_set_all(wredshift,1.0);
gsl_vector_set_all(sredshift,1.0);
gsl_vector_set_all(eta,reg);

//start inner level iteration

for(int i = 0; control > convthresh ; i++)
  {
    gsl_vector_memcpy(control1,recconv);
    Weakfactor(startdim*startdim,startdim*startdim,recconv,
    wredshift,Fij1,Fij2,Cij1,Cij2);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);
    Strongfactor(startdim,startdim,recconv,recshear1,recshear2,
    sredshift,strongsigma,strongfactor);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);
    procdata = gsl_vector_calloc(startdim*startdim);
    proccoeff = gsl_matrix_calloc(startdim*startdim,startdim*startdim);
    Data_vector(startdim,startdim,1.0,ccurve,ellip1,
    ellip2,Fij1,Fij2,refconv,refshear1,refshear2,eta,
    strongfactor,wredshift,sredshift,procdata,my_rank,p);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);
    Coeff_matrix(startdim,startdim,1.0,ccurve,ellip1,
    ellip2,Fij1,Fij2,refconv,refshear1,refshear2,eta,
    strongfactor,wredshift,sredshift,proccoeff,my_rank,p);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);

    recv_gsl_fromworld(procdata,startdim*startdim,p,my_rank);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);
    recv_gsl_fromworld(proccoeff,startdim*startdim,
    startdim*startdim,p,my_rank);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);

    //evaluating result

    if(my_rank == 0)
      {
        Solve_system(proccoeff,procdata,recpot,startdim*startdim);
        GalaxyCluster cluster1(startdim,startdim,1.0);
        cluster1.readfrom("pot",recpot);
        cluster1.buildfrompot();
        cluster1.masssheetnormalise();
        cluster1.writeto("conv",recconv);
        cluster1.writeto("shear1",recshear1);
        cluster1.writeto("shear2",recshear2);
        gsl_vector_memcpy(control2,recconv);
        control = change(control1,control2,cluster1.showdim());
        cout <<"Maximum␣change␣in␣convergence:␣" <<control <<endl;
      }

    //telling the MPI-World

    send_gsl_toworld(recconv,startdim*startdim,p,my_rank);
    send_gsl_toworld(recshear1,startdim*startdim,p,my_rank);
    send_gsl_toworld(recshear2,startdim*startdim,p,my_rank);
    MPI_Barrier(MPI_Comm MPI_COMM_WORLD);
  }

//writing result

if(my_rank == 0)
  {
    GalaxyCluster cluster1(startdim,startdim,1.0);
    cluster1.readfrom("pot",recpot);
    cluster1.buildfrompot();
    cluster1.masssheetnormalise();
    cluster1.writeto("conv",recconv);
    cluster1.writeto("shear1",recshear1);
    cluster1.writeto("shear2",recshear2);
    cluster1.writetofits("all","resultmpi10_iter.fits");

  }
MPI_Barrier(MPI_Comm MPI_COMM_WORLD);

//-----------------------------------------------------------------

//start outer level-iteration

startdim = startdim + 2;
gsl_vector *nofuture1 = gsl_vector_calloc(startdim*startdim);

//building comparison function

if(my_rank == 0)
  {
    smoothinterpol(recpot,startdim-2,startdim-2,2,nofuture1);
  }

// from here the whole process atrts again
```

```

 .
 .
 .
 .
 .
 .
 .
 .

  MPI_Finalize();
  return 0;
}
```

Bibliography

Bartelmann, M. 2007, Cosmology, Lecture Notes (University of Heidelberg)

Bartelmann, M., Huss, A., Colberg, J. M., Jenkins, A., & Pearce, F. R. 1998, A&A, 330, 1

Bartelmann, M. & Schneider, P. 2001, Phys. Rep., 340, 291

Bertin, E. & Arnouts, S. 1996, A&AS, 117, 393

Bradač, M., Lombardi, M., & Schneider, P. 2004, A&A, 424, 13

Bradač, M., Schneider, P., Lombardi, M., & Erben, T. 2005, A&A, 437, 39

Bradač, M., Schrabback, T., Erben, T., et al. 2007, ArXiv e-prints, 0711.4850v1

Broadhurst, T., Benítez, N., Coe, D., et al. 2005, ApJ, 621, 53

Cacciato, M. 2005, Master's thesis, Universita degli Studi di Bologna

Cacciato, M., Bartelmann, M., Meneghetti, M., & Moscardini, L. 2006, A&A, 458, 349

Cavaliere, A. & Fusco-Femiano, R. 1976, A&A, 49, 137

Cavaliere, A. & Fusco-Femiano, R. 1978, A&A, 70, 677

Clowe, D., Gonzalez, A., & Markevitch, M. 2004, ApJ, 604, 596

Comerford, J. M., Meneghetti, M., Bartelmann, M., & Schirmer, M. 2006, ApJ, 642, 39

Dahle, H. 2007, ArXiv Astrophysics e-prints, 0701598v1

Diego, J. M., Tegmark, M., Protopapas, P., & Sandvik, H. B. 2007, MNRAS, 375, 958

Dolag, K., Vazza, F., Brunetti, G., & Tormen, G. 2005, MNRAS, 364, 753

Elíasdóttir, Á., Limousin, M., Richard, J., et al. 2007, ArXiv e-prints, 0710.5636v1

Fedeli, C., Meneghetti, M., Bartelmann, M., Dolag, K., & Moscardini, L. 2006, A&A, 447, 419

Fort, B., Le Fevre, O., Hammer, F., & Cailloux, M. 1992, ApJ, 399, L125

Fort, B., Prieur, J. L., Mathez, G., Mellier, Y., & Soucail, G. 1988, A&A, 200, L17

Gavazzi, R. 2005, A&A, 443, 793

Gavazzi, R., Fort, B., Mellier, Y., Pelló, R., & Dantel-Fort, M. 2003, A&A, 403, 11

Gavazzi, R., Treu, T., Koopmans, L. V. E., et al. 2008, ArXiv e-prints, 0801.1555v1

Giavalisco, M., Ferguson, H. C., Koekemoer, A. M., et al. 2004, ApJ, 600, L93

Goldberg, D. M. & Bacon, D. J. 2005, ApJ, 619, 741

Halkola, A., Seitz, S., & Pannella, M. 2006, MNRAS, 372, 1425

Jarvis, J. F. & Tyson, J. A. 1981, AJ, 86, 476

Kaiser, N., Squires, G., & Broadhurst, T. 1995, ApJ, 449, 460

King, I. R. 1966, AJ, 71, 64

King, I. R. 1972, ApJ, 174, L123+

King, I. R. 1981, QJRAS, 22, 227

Le Fèvre, O., Vettolani, G., Garilli, B., et al. 2005, A&A, 439, 845

Lemze, D., Barkana, R., Broadhurst, T. J., & Rephaeli, Y. 2007, ArXiv e-prints, 0711.3908v1

Leonard, A., Goldberg, D. M., Haaga, J. L., & Massey, R. 2007, ApJ, 666, 51

Liesenborgs, J., de Rijcke, S., Dejonghe, H., & Bekaert, P. 2007, MNRAS, 380, 1729

Mahdavi, A., Hoekstra, H., Babul, A., Balam, D. D., & Capak, P. L. 2007, ApJ, 668, 806

Marshall, P. J., Treu, T., Melbourne, J., et al. 2007, ArXiv e-prints, 0710.0637v1

Massey, R. & Refregier, A. 2005, MNRAS, 363, 197

Melchior, P. 2006, Master's thesis, Universität Heidelberg

Melchior, P., Meneghetti, M., & Bartelmann, M. 2007, A&A, 463, 1215

Meneghetti, M. 2006, Introduction to Gravitational Lensing, Lecture Notes (University of Heidelberg)

Meneghetti, M., Argazzi, R., Pace, F., et al. 2007a, A&A, 461, 25

Meneghetti, M., Melchior, P., Grazian, A., et al. 2007b, ArXiv e-prints, 0711.3418v1

Narayan, R. & Bartelmann, M. 1996, ArXiv Astrophysics e-prints, 9606001

Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563

Press, W. H. & Schechter, P. 1974, ApJ, 187, 425

Rannacher, R. 2006, Einfuehrung in die numerische Mathematik, Lecture Notes (University of Heidelberg)

Refregier, A. 2003, MNRAS, 338, 35

Rzepecki, J., Lombardi, M., Rosati, P., Bignamini, A., & Tozzi, P. 2007, A&A, 471, 743

Sand, D. J., Treu, T., Ellis, R. S., Smith, G. P., & Kneib, J. 2007, ArXiv e-prints, 0710.1069v1

Schneider, P. 2006, Extragalactic Astronomy and Cosmology (Springer)

Schneider, P. & Er, X. 2007, ArXiv e-prints, 0709.1003v1

Seidel, G. & Bartelmann, M. 2007, A&A, 472, 341

Springel, V. 2005, MNRAS, 364, 1105

Springel, V., White, S. D. M., Jenkins, A., et al. 2005, Nature, 435, 629

Springel, V., Yoshida, N., & White, S. D. M. 2001, New Astronomy, 6, 79

Tu, H., Limousin, M., Fort, B., et al. 2007, ArXiv e-prints, 0710.2246v1

Wambsganss, J. 1998, Living Reviews in Relativity, 1, 12

Wambsganss, J., Bode, P., & Ostriker, J. P. 2004, ApJ, 606, L93

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den ______________________________ Unterschrift:______________________________