Mesh-free free-form lensing I: Methodology and application to mass reconstruction

Julian Merten \(^1,2\)\star

\(^1\) Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA  
\(^2\) California Institute of Technology, MC 249-17, Pasadena, CA 91125, USA

© 2014 California Institute of Technology. Government sponsorship acknowledged. All rights reserved. Submitted to the Monthly Notices of the Astronomical Society.

ABSTRACT

Many applications and algorithms in the field of gravitational lensing make use of meshes with a finite number of nodes to analyze and manipulate data. Specific examples in lensing are astronomical CCD images in general, the reconstruction of density distributions from lensing data, lens–source plane mapping or the characterization and interpolation of a point-spread-function. We present a numerical framework to interpolate and differentiate in the mesh-free domain, defined by nodes with coordinates that follow no regular pattern. The framework is based on radial basis functions (RBFs) to smoothly represent data around the nodes. We demonstrate the performance of Gaussian RBF-based, mesh-free interpolation and differentiation, which reaches the sub-percent level in both cases.

We use our newly developed framework to translate ideas of free-form mass reconstruction from lensing onto the mesh-free domain. The method uses weak-lensing and strong-lensing constraints and ideally follows the distribution of input data. By reconstructing a simulated mock lens we find that strong-lensing only reconstructions achieve \(< 10\%\) accuracy in the areas where these constraints are available but provide poorer results when departing from these regions. Weak-lensing only reconstructions give \(< 10\%\) accuracy outside the strong-lensing regime, but cannot resolve the inner core structure of the lens. Once both regimes are combined, accurate reconstructions can be achieved over the full field of view and with a spatial resolution which is optimally adapted to the input data.

Key words: Gravitational lensing: strong – gravitational lensing: weak – methods: numerical – galaxies: clusters: general – dark matter – large-scale structure of Universe.

1 INTRODUCTION

Many techniques of astrophysical data analysis, although they in principle work on smooth and continuous data, are confined to a discrete numerical domain, which evaluates the input data using a finite number of analysis nodes. This numerical domain is usually referred to as a mesh and the coordinates of its nodes can have different dimensionality, depending on the application. In many cases the structure of these node coordinates follows a regular pattern, so a constant separation of nodes in each dimension. This regular pattern is convenient because it reduces the numerical complexity of the problem and simplifies many numerical algorithms since the spatial structure of the data is highly symmetric and easy to implement. However, many applications need a more sophisticated description of the spatial distribution of input data. A more general mesh layout is provided by adaptive mesh refinement (AMR) which increases the resolution of the mesh wherever this is necessary and affordable in terms of CPU-time. The distances between nodes in each dimension is now adaptive but still follows a regular pattern. However, real astrophysical data, e.g. the distribution of galaxies or stars in a certain patch of the sky can be distributed in a very irregular fashion with density fluctuations, clusters and voids. When translating the input data onto a regular, structured mesh, this can lead to highly oversampled and partly unconstrained meshes or interpolation is needed, introducing the associated interpolation errors. The other extreme are undersampled meshes where averaging techniques compress the data onto a regular
mesh. This comes at the price of smoothing out information and thus not making use of the full potential of the data.

In this work we present a framework which can deal with a mesh-free structure in the input data distribution. This means that the coordinates of the nodes follow no regular pattern and can have any values within the numerical domain. Two important types of data manipulation on such structures are interpolation and differentiation and we will introduce efficient algorithms which can achieve both of these tasks in any spatial dimension. The key to such a framework are radial basis functions (RBFs), functions which only depend on the distance of their evaluation points to certain reference points.

This work, which focuses on mass reconstruction from gravitational lensing, is only the first in a series, which will exploit our mesh-free numerical techniques. Two regimes are typically distinguished in lensing mass reconstruction. Strong lensing is usually confined to the inner-most core of the gravitational lens and produces spectacular observational constraints such as multiple images of the same source, gravitational arcs or even rings. The domain of weak lensing is further away from the center of the lens but spans large areas and manifests itself by the weak distortion in the shape of background galaxies behind the lens. Reconstruction techniques are divided into two classes, although this distinction is by no means unique or even consistent in some cases. Parametric techniques (e.g. Kneib et al. 1996; Broadhurst et al. 2005; Smith et al. 2005; Halkola et al. 2006; Jullo et al. 2007; Zitrin et al. 2009; Oguri 2010; Newman et al. 2013; Jullo et al. 2014; Monna et al. 2014; Johnson et al. 2014, for some recent examples) assume a parametric form of the underlying mass density distribution for the lens and typically make the assumption that light traces mass in the positioning of these parametric forms. On the other hand, free-form\(^1\) methods (see e.g. Broadhurst et al. 1995; Bartelmann et al. 1996; Abdelsalam et al. 1998; Bridle et al. 1998; Seitz et al. 1998; Bradaˇc et al. 2005; Cacciato et al. 2006; Liesenborgs et al. 2006; Diego et al. 2007; Jee et al. 2007; Coe et al. 2008; Bradaˇc et al. 2009; Merten et al. 2009; Williams \\& Saha 2011; Merten et al. 2011, 2014; Diego et al. 2014, for some recent examples) usually do not make this assumption and purely rely on the input data either based on weak lensing, strong lensing or a combination of the two. This is possible while using a reconstruction mesh and directly inverting the underlying equations describing lensing on this mesh. In the following, we introduce a free-form method combining weak and strong lensing, which uses our new mesh-free numerical framework. This method translates original ideas by Bartelmann et al. (1996), Seitz et al. (1998), Bradaˇc et al. (2005), Cacciato et al. (2006) and Merten et al. (2009) into the flexible and efficient mesh-free numerical domain.

This work is structured as follows: In Sec. 2 we introduce RBFs and how they can be used to numerically interpolate and differentiate. In Sec. 3 we use the developed techniques to implement a free-form reconstruction algorithm that can be used in the mesh-free domain and consistently combines the regimes of weak- and strong-gravitational lensing. We test our implementation with numerically simulated data in Sec. 4 and we conclude in Sec. 5. In App. A we provide more details on the performance of interpolation with RBFs to the interested reader, as we do in App. B for mesh-free numerical differentiation. App. C gives some missing but not crucial details on the concrete implementation of the reconstruction algorithm. In many graphical illustrations in this paper we have to visualize mesh-free data. We do so by using the Voronoi tessellation of the evaluation points and by assigning a function value to each Voronoi cell, which refers to the function value at the respective coordinate.

\section{Radial Basis Functions}

In this general methodology section we will deal with functions which are defined on a finite number of evaluation points. Based on this set of points, we will interpolate functions in their numerical domain and calculate their derivatives. The general idea which enables us to do so is based on RBFs, where we expand the discretely defined functions into a set of radially dependent functions around the evaluation points. For a thorough discussion of the concept of radial basis functions see Fausthauer (2007) and references therein.

\subsection{Unstructured meshes and mesh-free data}

We define a mesh \( \mathcal{M} \) as a finite collection of \( N \) support points \( \vec{x} \)

\[ \mathcal{M} = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_N]. \quad (1) \]

The dimensionality \( D \) of the mesh is given by the number of coordinates needed to define each support point \( \vec{x} = (x_1, x_2, ..., x_D) \). In most cases we will focus on the case \( D = 2 \) with \( \vec{x} = (x_1, x_2) \), but the methodology presented in this section generalizes to \( D = 1 \), \( D = 3 \) or to higher dimensions. For simplicity, we will shorthand the cases \( D = 1 \), \( D = 2 \), \( D = 3 \), ..., \( D = N \) etc., with 1D, 2D, 3D, ..., ND.

This definition of a mesh \( \mathcal{M} \) is general in the sense that it includes cases with regularly distributed nodes or randomly distributed nodes. In this work we will show examples for both kinds of meshes, but generally focus on unstructured, meaning irregularly shaped mesh configurations. The range of coordinates depends on the numerical domain and the mesh is defined on but for simplicity we will mostly restrict ourselves to coordinates \( x_i \in [-1, ..., 1] \) with \( i \in [1, ..., D] \). Unstructured meshes can be still restricted when it comes to the distribution of their nodes. In the field of finite elements, for example, specific elements need to be formed with restrictions on the aspect ratios of their edges. For our purposes we do not have such restrictions, which is why we distinguish the notion of unstructured meshes from our, more general, case where the distribution of node coordinates follows no restrictions and which we call mesh-free.

\subsection{Interpolation with radial basis functions}

We analyze a function \( f \) defined at \( n \) nodes \( \vec{x}_1, ..., \vec{x}_n \) and denote the values of the function at these points \( f(\vec{x}_1), ..., f(\vec{x}_n) \) with \( f_1, ..., f_n \). We use RBFs \( \phi(\vec{x}) = \phi(\|\vec{x} - \vec{x}_i\|) \) to interpolate \( f \) to any position \( \vec{x} \) in the numerical domain. Throughout this work, radial distances are defined as the Euclidean
norm \( L_2 \) with the short-hand notation \( \| \vec{x} - \vec{x}_0 \| = r \) for a given reference point \( \vec{x}_0 \). Typical choices for RBFs are listed in Tab. 1, but in this work we will restrict ourselves to Gaussian RBF’s. In this case, the only free parameter, \( \epsilon \), is called the shape parameter and has to be chosen carefully as will be discussed in great detail in Sec. 2.4 and App. A. For a more thorough discussion of the underlying mathematical concepts we refer to Fornberg et al. (2011), Larsson et al. (2013), Fornberg et al. (2013), Flyer et al. (2014) and references therein.

We write the interpolant \( \tilde{f} \) of the function \( f \) defined at nodes \( \vec{x}_1, \ldots, \vec{x}_n \) as a weighted sum over RBF’s

\[
\tilde{f}(\vec{x}) = \sum_{i=1}^{n} \lambda_i \phi(||\vec{x} - \vec{x}_i||) \tag{2}
\]

with weighting coefficients \( \lambda_i \). Because of \( \tilde{f}(\vec{x}_i) = f_i \), we can calculate the coefficients \( \lambda \) by solving the linear system of equations

\[
\begin{pmatrix}
\phi(||\vec{x}_1 - \vec{x}_1||) & \ldots & \phi(||\vec{x}_1 - \vec{x}_n||) \\
\vdots & \ddots & \vdots \\
\phi(||\vec{x}_n - \vec{x}_1||) & \ldots & \phi(||\vec{x}_n - \vec{x}_n||)
\end{pmatrix}
\begin{pmatrix}
\lambda_1 \\
\vdots \\
\lambda_n
\end{pmatrix}
= \begin{pmatrix}
f_1 \\
\vdots \\
f_n
\end{pmatrix}
\tag{3}
\]

It is important to note that for distinct nodes, this linear system cannot become singular, no matter how the nodes are scattered in any number of dimensions. Fig. 1 shows a graphical illustration of the expansion of a function into Gaussian RBF’s around evaluation points. The accuracy of the interpolation is certainly dependent on the shape parameter and on the choice and number of nodes. We discuss the shape parameter further in Sec. 2.4 and present a detailed performance analysis of the RBF-based interpolation scheme in App. A.

### 2.3 Mesh-free numerical derivatives

Numerical derivatives are usually calculated by means of finite differencing (FD)

\[
Df(\vec{x}) \approx \sum_{i=1}^{n} w_i f(\vec{x}_i) \tag{4}
\]

for a linear differential operator \( D \) and with evaluation points \( \vec{x}_i \), defining a FD stencil. There are several ways of finding the weights \( w \) for the FD on regular meshes, ranging from the Lagrange interpolation polynomial, Taylor expansion, monomial test functions to the elegant Padé-algorithm. For a thorough description of all different techniques and the original references, see Fornberg (1998). Here we focus on monomial test functions since it will set the stage for mesh-free RBF-based FD later.

The motivation for the use of monomial test functions is to enforce that Eq. 4 holds exactly when \( f \) is a polynomial of degree \( n-1 \). In the 1D case, the weights in Eq. 4 are then given by the solution of the linear system of equations

\[
\begin{pmatrix}
1 & 1 & \ldots & 1 \\
x_1 & x_2 & \ldots & x_n \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1}
\end{pmatrix}
\begin{pmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{pmatrix}
= \begin{pmatrix}
D1(\vec{x}) \\
Dx(\vec{x}) \\
\vdots \\
Dx^{n-1}(\vec{x})
\end{pmatrix}
\tag{5}
\]

where \( x_1, \ldots, x_n \) are the evaluation points of the finite-differencing stencil and the differential operator \( D \) is applied to the monomial test functions \( x^{n-1} \) at the point of interest \( \vec{x} \). This approach easily generalizes to higher dimensions by inserting also the other coordinate components of \( \vec{x} \) and mixed-component monomial test functions. The case of \( D = \Delta \) and \( n = 5 \) recovers the well known finite-differencing stencil on a regular mesh with node separation \( h \), evaluated at \( \vec{x} = \vec{x}_0 \)

\[
h^{-2} \begin{pmatrix} 1 & 1 & -4 & 1 & 1 \end{pmatrix}
\begin{pmatrix}
f((x_0 + h, y_0)) \\
f((x_0 - h, y_0)) \\
f(x_0) \\
f((x_0, y_0 + h)) \\
f((x_0, y_0 - h))
\end{pmatrix}
\approx \Delta f(\vec{x}_0).
\tag{6}
\]

Inspired by this approach, we substitute the monomial test functions with RBFs and advance to a mesh-free formulation by centering the RBFs on the evaluation points of the function. In complete analogy to Eq. 5, this Ansatz leads to the following linear system of equations to find the
finite differencing weights

\[
\mathcal{F} \begin{pmatrix}
 w_1 \\
 \vdots \\
 w_n 
\end{pmatrix} = \begin{pmatrix}
 D\phi(||\vec{x} - \vec{x}_1||) \\
 \vdots \\
 D\phi(||\vec{x} - \vec{x}_n||) 
\end{pmatrix}.
\] (7)

Monomial terms can be included again in order to increase the accuracy of the numerical derivatives. In the 2D case with monomial terms up to second order the FD weights are given by the solution to

\[
\begin{pmatrix}
 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 1 & x_n & y_n & x_n^2 & y_n^2 & x_n y_n \\
 1 & \ldots & 1 \\
 x_1 \ldots x_n \\
 y_1 \ldots y_n \\
 x_1^2 \ldots x_n^2 \\
 y_1^2 \ldots y_n^2 \\
 (x_1 y_1) \ldots (y_1 y_n)
\end{pmatrix}
\begin{pmatrix}
 w_1 \\
 \vdots \\
 w_n \\
 w_{n+1} \\
 w_{n+2} \\
 w_{n+3} \\
 w_{n+4} \\
 w_{n+5} \\
 w_{n+6}
\end{pmatrix} =
\begin{pmatrix}
 D\phi(||\vec{x} - \vec{x}_1||) \\
 \vdots \\
 D\phi(||\vec{x} - \vec{x}_n||) \\
 D1(\vec{x}) \\
 Dx(\vec{x}) \\
 Dy(\vec{x}) \\
 D(x^2)(\vec{x}) \\
 D(y^2)(\vec{x}) \\
 D(xy)(\vec{x})
\end{pmatrix}
\] (8)

Only \(w_1, \ldots, w_n\) are used as weights in Eq. 4. The remaining values \(w_{n+1}, \ldots, w_{n+6}\) have no obvious meaning. This 2D RBF FD scheme with up to second order monomial terms applies to most of our applications in this paper and is the scheme that we use in this work from now on, if not otherwise stated. The performance of RBF-derived FD stencils is discussed in the next section and in great detail in App. B.

### 2.4 Discussion of the shape parameter

The shape parameter controls the width of the Gaussian RBF and crucially controls the accuracy of the RBF application no matter if one uses RBFs for interpolation or FD. For a complete discussion of the shape parameter we refer the interested reader to the work by Fornberg et al. (2011), Fornberg et al. (2013) and Larsson et al. (2013). In the following we will heuristically analyze the effect of the shape parameter by introducing the test function

\[
f(x, y) = 1 + \sin(4x) + \cos(3x) + \sin(2y). \tag{9}\]

Based on 180 randomly chosen evaluation points, we interpolate the test function to 900 random locations (nodes) on the unit disk. For a more complete description of the test setup we refer the interested reader to App. A. In Fig. 2 we show the shape parameter dependence of the average and maximum relative error of the interpolation and also include the dependence on the number of nearest neighbor evaluation points used to carry out the interpolation for each interpolant evaluation point. As one can clearly see, there is an optimal choice for the shape parameter in order to achieve maximum accuracy. This choice depends on the actual node coordinates, and the number of evaluation points. Once the shape parameter is optimized (Fornberg et al. 2013), very high accuracies in the interpolation can be achieved, reaching an average relative interpolation error \(\sim 10^{-5}\). We discuss this in more detail in App. A. As a final remark on the theory of RBF interpolation we point out, that the approach can be further optimized by choosing a spatially varying shape parameter. Since the choice of a variable shape parameter, depending on the distribution of evaluation points, is not trivial (Fornberg & Zuev 2007) we focus for now on the special case of a single \(\epsilon\) value. We will present a more general method, that adaptively varies the shape parameter, in the course of the development of our mesh-free methods.

For RBF-based FD stencils, it has been shown that the respective FD weights only give accurate results if the condition number \(C^2\) of the coefficient matrix in Eq. 7 is close to critical\(^2\), and thus depends on the machine precision of the implementation (Fornberg et al. 2013). One has to carefully monitor this condition number, which can be directly controlled by the shape parameter. In Fig. 3 we show the accuracy of the numerical first \(x\) derivative as a function of this condition number and the number of nearest neighbors in the stencil. The accuracy is rapidly increasing when approaching the ill-condition point, where \(C\) is approaching the critical value, of the system. Once the system is ill-conditioned, or close to that, the accuracy is decreasing again and shows irregular behavior. This is discussed in much more detail in Fornberg et al. (2013) and we give a more detailed heuristic assessment in App. B. In the following we mostly use 16 to 32 nearest neighbors to define the FD stencil in each evaluation point and therefore we

\(2\) The condition number of the coefficient matrix is defined as the ratio between the largest and smallest singular value of the SVD decomposition of the coefficient matrix.

\(3\) This means that \(\log(C) \geq \epsilon\), where \(\epsilon\) is the machine precision of your numerical implementation.
adjust the shape parameter to keep the condition number of \( F \) in the \( 10^{16} \) - \( 10^{17} \) region. In order to avoid the dependence on shape parameter altogether, the use of polyharmonic spline (PHS) type RBFs (compare Tab. 1) is a sensible approach. The performance of PHS-RBFs in RBF-FD applications is currently under investigation and shows promising results (Bengt Fornberg, private communication). We will implement these RBFs in the numerical framework that we present in the next section and will perform a series of tests to analyze the accuracy and feasibility of PHS-type RBFs for our purposes.

2.5 Numerical implementation

We briefly describe our own numerical implementation of mesh-free interpolation and differentiation with RBFs. This implementation will also be made publicly available\(^4\).

The library is written in C++ and mainly provides, among several helper routines, two classes. The central class describes a collection of \( n \) nodes with arbitrary coordinates in either 1D, 2D or 3D and is initially defined by an input vector \( X = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n] \). For both, interpolation and differentiation in this mesh-free domain, it is important to find the nearest neighbors of each node \( \vec{x}_i \). The mesh-free nodes class provides this functionality by means of a kd-tree based ordering algorithm. In our implementation we use the publicly available library FLANN\(^5\) (Muja & Lowe 2009). The mesh-free nodes class then provides useful features like node coordinate queries and returns node index vectors for the nearest neighbors of each node.

The second central class implements radial basis functions. The shape parameter, the origin of the coordinate system and the dimensionality of the problem (1D, 2D or 3D) can be set. The user can evaluate the RBF and its derivatives at different coordinates. Currently, only Gaussian RBFs are implemented but the class uses inheritance from virtual functions to generally represent an RBF and allows the user to implement more general cases by providing the functional form of the RBF and its derivatives.

The combination of the mesh-free nodes and the RBF class enables the functionality, which was described in the course of this section. Once the functional form of an RBF is provided, mesh-free interpolation to arbitrary evaluation points is enabled and the finite differencing weights for each evaluation point can be calculated and returned as a matrix \( W_k \). This allows the user to differentiate a function \( \Psi_k = [\Psi(\vec{x}_1), \Psi(\vec{x}_2), \ldots, \Psi(\vec{x}_n)] \) by a simple matrix multiplication

\[
D\Psi_l \approx W_{lk}\Psi_k. 
\]

Currently all differential operators \( D \) up to third order are implemented in our classes.

3 LENSING MASS RECONSTRUCTION

We apply the RBF framework to a concrete astrophysical application, mass reconstruction from gravitational lensing. After a short lensing primer we show how different constraints from gravitational lensing can be combined in a free-form way by using a mesh-free approach, where the nodes can follow a completely irregular layout. Hence, the reconstruction can ideally follow the spatial distribution of the lensing constraints. The lensing constraints are ellipticity measurements of weakly lensed galaxies, the position of strong-lensing multiple-image systems and estimators for the positions of the strong-lensing critical curve of the lens.

3.1 Lensing primer

Einstein’s theory of general relativity predicts the deflection of light rays due to gravitational potentials (see e.g. Bartelmann 2010, for a complete derivation). By introducing the thin-lens approximation, which assumes that the distances between objects in the lensing scenario are much larger than the spatial extent of these objects, the lens mapping can be described by a lens equation

\[
\vec{\beta} = \vec{\theta} - \vec{\alpha}(\vec{\theta}). \tag{11}
\]

This central equation describes how the 2D angular position in the source plane \( \vec{\theta} = (\theta_1, \theta_2) \) is mapped by a deflection angle \( \vec{\alpha} = (\alpha_1, \alpha_2) \) onto the angular coordinates \( \vec{\beta} = (\beta_1, \beta_2) \) in the lens plane. The deflection angle can be related to a lensing potential

\[
\psi(\vec{\theta}) := \frac{1}{\pi} \int |D\vec{\beta}| \Sigma(\vec{\theta}) \frac{\ln|\vec{\theta} - \vec{\beta}|}{|\vec{\theta} - \vec{\beta}|}, \tag{12}
\]

that inherits the surface-mass density of the lens \( \Sigma(D_\theta \vec{\theta}) \). The cosmological background model enters this equation through the critical surface mass density for lensing given by

\[
\Sigma_{cr} = \frac{c^2}{4\pi G} \frac{D_k}{D_l D_h}, \tag{13}
\]

where \( c \) is the speed of light and \( G \) is Newton’s constant. The angular diameter distance between observer and lens

---

\(^4\) Upon request initially and at https://bitbucket.org/jmerten82/libmfree after publication of the peer-reviewed manuscript.

\(^5\) http://www.cs.ubc.ca/research/flann/
$D_l$, between observer and source $D_s$, and between lens and source $D_b$ set the geometry of the lensing scenario.

When introducing the edth operators (Newman & Penrose 1962) $\partial := (\frac{\partial}{\partial \psi_l} + i\frac{\partial}{\partial \psi_i})$ and $\partial^* := (\frac{\partial}{\partial \psi_i} - i\frac{\partial}{\partial \psi_l})$, lensing quantities are easily related to the lensing potential (see e.g. Bacon et al. 2006; Schneider & Er 2008)

\[
\begin{align*}
\alpha &:= \partial \psi \quad s = 1 \\
2\gamma &:= \partial \partial \psi \quad s = 2 \\
2\kappa &:= \partial \partial^* \psi \quad s = 0
\end{align*}
\]  

where $\alpha$ is the deflection angle, $\gamma$ is called the complex shear and the scalar quantity $\kappa$ is called convergence. The spin-parameter $s$ describes the transformation properties of each quantity under rotations of the coordinate frame.

The regime of weak-gravitational lensing is governed by small distortions in the shape of background galaxies, which are observationally measured as complex ellipticities $\epsilon$. Due to the intrinsic ellipticity of galaxies, localized averages over an ensemble of sources are used to separate the lensing signal from random orientations. One finds

\[
\langle \epsilon \rangle = g := \frac{\gamma}{1 - \kappa},
\]

where we introduced the reduced shear $g$ and established the relation to Eqs. 14, at least for regimes well outside the strong-lensing regime of the lens ($|g| < 1$). For a thorough review of weak lensing and its applications we refer to Bartelmann & Schneider (2001, and references therein) and for a discussion of systematic effects in weak lensing studies to Kitching et al. (2012), Massey et al. (2013) and Mandelbaum et al. (2014).

In the strong-lensing regime the lens equation becomes non-linear, multiple images of the same source can form and shape distortions are not small any more. This leads, in some cases, to the formation of spectacular gravitational arcs or even rings in the vicinity of a strong lens. The spatial extent of this regime, close to the core of the lens where densities are highest, is indicated by the critical curve at a given redshift. It is defined by the roots of the determinant of the lensing Jacobian

\[
\det A = (1 - \kappa)^2 - \gamma^*\gamma.
\]

\[\text{Figure 4. The distribution of lensing constraints in the galaxy cluster Abell 383 as seen by CLASH (Zitrin et al. 2011; Postman et al. 2012; Merten et al. 2014).}\]

3.2 Combining lensing constraints in the mesh-free domain

The toolkit developed in Sec. 2 can be used to perform mesh-free lensing reconstructions. That is to recover the underlying total mass distribution of a lens from weak - and strong lensing constraints. Eqs. 14, 15 and 16 show that all lensing quantities related to the discretized lensing potential $\psi = [\psi(\vec{x}_1), \psi(\vec{x}_2), \ldots, \psi(\vec{x}_N)]$ we find

\[
\begin{align*}
\alpha_i^{1/2} &= D_{ik}^{1/2} \psi_k \\
\gamma_i^{1/2} &= G_{ik}^{1/2} \psi_k \\
\kappa_i &= K_{ik} \psi_k
\end{align*}
\]

where $D$, $G$ and $K$ contain the FD weights for the differential operators in Eq. 14. In order to recover the lensing potential from the input constraints we define a $\chi^2$-function, which relates the lensing observations at each evaluation point to the lensing potential. We will define all components of this function in different lensing regimes later on. In order to find the lensing potential which is most likely to have caused the observed lensing effects we minimize the $\chi^2$-function with respect to the potential values at each evaluation point

\[
\frac{\partial \chi^2(\psi_k)}{\partial \psi_k} = 0;
\]

where we use that

\[
\frac{\partial W_{lk} \psi_k}{\partial \psi_l} = \delta_{lk}
\]

for any matrix representation $W$ of a FD stencil.

3.2.1 Weak lensing

It is our goal to combine several lensing constraints into a joint reconstruction algorithm and we start with the weak-lensing contribution. Eq. 15 shows that average measured ellipticities of background galaxies are directly related to the reduced shear of the lens. Hence, we write the weak-lensing term as

\[
\chi^2_\alpha = \sum_{i,j} \langle \epsilon - g(\psi) \rangle C_{ij}^{-1} \langle \epsilon - g(\psi) \rangle,
\]

where the indices $i, j$ run over all weak-lensing evaluation points. It has been shown in Bradac et al. (2005) and Merten et al. (2009) how the minimization of such a $\chi^2$ with respect to the discretized lensing potential can be written as a linear
system of equations, while using Eqs. 18, 19 and Eq. 21. We refer the interested reader to Appendix A of Merten et al. (2009), which shows the full derivation but we summarize the most relevant results in App. C1.

The covariance matrix of the weak lensing data $C_{ij}$ deserves special attention. It is well-known that galaxies carry an intrinsic ellipticity with a standard deviation of $\sigma_\epsilon \sim 0.3$ (e.g. Chang et al. 2013), which is not induced by lensing. One way of incorporating this into the $\chi^2$ minimization is to assume that the galaxy intrinsic ellipticities are uncorrelated, which results in a diagonal covariance matrix with the canonical value of 0.3$^2$ for all its non-zero elements. However, in the presence of noise, this approach leads to a very poor recovery of the lensing potential, as we will show later on. A different approach exploits the fact that the intrinsic ellipticity of galaxies has no preferred orientation$^6$. To a given weak-lensing evaluation point we do not assign the ellipticity value of a single galaxy but we perform a distance-weighted average over an ensemble of nearest neighbors with respect to the point of interest. By doing so, the undirected, intrinsic ellipticity signal averages out and the coherent lensing signal remains. This procedure obviously introduces correlations between the neighboring pixels which were used to define the ellipticity samples. We calculate this sample covariance following Equation 15 of Merten et al. (2009) and take it into account by summing over the full covariance in the $\chi^2$-minimization.

### 3.2.2 Strong lensing

One of the biggest advantages and indeed the biggest motivation for a mesh-free reconstruction algorithm is the fact that the distribution of evaluation points is intrinsically adaptive. This is important because different lensing constraints are confined to quite different length scales, as is clearly seen in Fig. 4. The mesh-free approach allows us to place evaluation points where data is available. In the case of weak lensing, this spans the entire field of the lens, with usually increased resolution towards the center when high-quality data from e.g. the Hubble Space Telescope (HST) is available. Strong lensing is confined to the very core of the lens and allows for a very finely grained recovery of the lensing potential if many strong-lensing features are observable.

One constraint related to strong lensing is an estimate on the position of the critical curve of the lens. This has been discussed in detail in e.g. (Jullo et al. 2007; Merten et al. 2009, 2011, 2014). The corresponding $\chi^2$-term enforces the lensing Jacobian to vanish for pixels which are assigned to be part of the critical curve and which are indicated in the following by a pixel index $c$

$$\chi^2_c(\psi) = \sum_{c=1}^{N_c} \frac{|\det A(\psi)|^2}{\sigma^2_{\epsilon,c}}. \tag{23}$$

The total number of these estimators is $N_c$ and the error $\sigma^2_\epsilon$ derives from a positional error that is assigned to the critical curve estimator. We approximate it via

$$\sigma_\epsilon \approx \frac{\partial \det A}{\partial \theta} \bigg|_{\theta_k} \delta \theta \approx \frac{\delta \theta}{\theta_k}, \tag{24}$$

where $\theta_k$ is an estimate for the Einstein radius of the lens. The minimization of the $\chi^2$-function related to this constraint can also be related to a linear system of equations using Eqs. 18,19 and Eq. 21 and we again leave the actual calculation to Appendix A of Merten et al. (2009) and show the result in App. C2.

The second strong-lensing constraint are multiple-image systems. Our $\chi^2$-minimization term is similar to Bradač et al. (2005) but differs in some details. The general idea is based on the fact that different images $i$ of the same multiple-image system should be mapped back to the same position in the source plane. Therefore, we write a $\chi^2$-term

$$\chi^2_m = \sum_{i=1}^{N_i} \left( \frac{\bar{\beta}_i(\psi) - \langle \beta \rangle}{\sigma_{i,m}} \right)^2 \tag{25}$$

where $\beta_i$ is the source-plane position of each image of the system and

$$\langle \beta \rangle = \frac{1}{N_s} \sum_{i=1}^{N_i} \beta_i \tag{26}$$

is the average source-plane position of all images in the system. The total number of images in a single multiple-image system is $N_s$. In the last two equations we can use Eq. 11 to replace the source-plane position $\tilde{\beta}$ with the observed lens-plane position $\bar{\beta}$ and the deflection angle $\bar{\alpha}$, which carries the wanted dependence on the lensing potential

$$\chi^2_m = \sum_{i=1}^{N_i} \frac{1}{\sigma^2_i} \left( \bar{\beta}_i - \bar{\alpha}_i(\psi) - \frac{1}{N_s} \sum_{j=1}^{N_i} \left( \bar{\beta}_j - \bar{\alpha}_j(\psi) \right) \right)^2. \tag{27}$$

Here, $\sigma_\epsilon$ is the tolerated positional error on each image position in the source plane. The result of minimizing this $\chi^2$-function with respect to the lensing potential is shown in App. C3. In the presence of more than one multiple-image system, one adds a $\chi^2$-term for each system, respectively.

#### 3.2.3 Implementation

In order to find the lensing potential which causes the joint observations of all weak -and strong-lensing terms we sum over the independent contributions of the $\chi^2$ function and find a single linear system of equations. The solution to this linear system is the mesh-free representation of the reconstructed lensing potential, from which all other quantities of interest can be derived using Eqs. 17, 18 and 19.

In order to guarantee a smooth reconstruction in the presence of noisy weak lensing data we introduce an outer-level iteration, following the scheme of Bradač et al. (2005), which is also used in Merten et al. (2009). We define a regularization term in the $\chi^2$-function which controls the reconstruction in such a way that the result will not diverge strongly from a well-defined regularization condition. In our case, this condition is set by pre-defined convergence $\kappa_{reg}$.
In the summation above, $i$ runs over all evaluation points and it should be noted that in this implementation the strength of the regularization $\eta$ can be set for each evaluation point individually. The contribution of Eqs. 28 and 29 to the total linear system of equations is derived in App. C4. The idea of the outer-level iteration is then to start with only few weak-lensing evaluation points and to average ellipticities of a large sample of weak-lensing sources for each of these nodes. This results in a coarse but almost shape-noise free reconstruction. In the following steps, the number of nodes is continuously increased, resulting in smaller ellipticity samples but relying on a reconstruction that is convergence- and shear-regularized on the result of the former reconstruction step. This outer-level iteration effectively reduces the noise-level in the reconstruction, as shown in Bradač et al. (2005) and Merten et al. (2009), and as we will explore in our accuracy tests later on.

Ultimately, we are solving the linear system of equations which is calculated from the minimization of the $\chi^2$-function

$$\chi^2(\psi) = \chi^2_c(\psi) + \chi^2_m(\psi) + \chi^2_{\gamma,\mathrm{reg}}(\psi) + \chi^2_{\eta,\mathrm{reg}}. \quad (30)$$

One notes that the weak-lensing and the critical-line estimator term contain non-linear contributions. We account for this by introducing an inner-level iteration, following the scheme of Schneider & Seitz (1995). During each inner-level reconstruction iteration, non-linear terms in the $\chi^2$-function are isolated and held constant in order to solve the linear system of equations. New estimates for convergence and shear are calculated from this solution and new approximations for the non-linear terms are inserted as constants into the linear system of equations. This iteration converges after 2-5 reconstruction steps.

A complete flowchart of the reconstruction algorithm is shown in Fig. 5. Initially, the weak-lensing catalog is read and depending on the stage of the outer-level iteration, ellipticity values are averaged to define the weak-lensing evaluation points. These are then combined with the strong-lensing evaluation points, which directly derive from the critical-line estimator and multiple-image system catalogs. For all these nodes, the finite-differencing stencils are calculated using RBFs. The convergence and shear maps derived from this interpolation serve as the regularization template for the next step. The very first regularization template depends on the reconstruction and the field of view of the data but, in most cases, a flat and zero convergence and shear template suffices. However, a more sophisticated choice for the initial prior is also possible resulting in more complicated initial convergence and shear regularization templates.

![Flowchart of the reconstruction process using weak- and strong-lensing constraints.](image)

Figure 5. A flowchart of the reconstruction process using weak- and strong-lensing constraints.

## 4 ACCURACY TESTS WITH MOCK LENSES

We use a numerically simulated, cluster-sized lens to thoroughly test our reconstruction algorithm in several stages. This mock lens is described in more detail in Bartelmann et al. (1998) and was already used for accuracy tests in Cacciato et al. (2006) and Merten et al. (2009). The surface-mass density map of this lens is shown in Fig. 6. The side length of the field of view is 5 Mpc/h or 18′ at the lens’ redshift of $z = 0.35$. The Einstein radius of the lens is $\theta_E \sim 30''$ for a source redshift of $z_s = 1.0$. In the following, especially in the figures of this section, we will scale these distances to dimensionless coordinates by mapping them into the unit-square with side length 2. From the known deflection angle fields, we sample the following lensing catalogs to serve as input for our test reconstructions:

- 9000 complex shear values at random positions in the field. This refers to a background-galaxy density of $\sim 25 \, \text{arcmin}^{-2}$.
- The same catalog of 9000 weak lensing shear values but with an added shape-noise component. This noise is sampled randomly in each shear component from a Gaussian distribution with a standard deviation of 0.2 (compare Sec. 3.2).
- 55 multiple images, by randomly placing 5 point sources inside the inner caustic and 10 point sources in between the inner and the outer caustic of the lens.
- 20 critical line estimators by randomly sampling points in the field for which the determinant of the lens Jacobian (Eq. 16) vanishes within the limits of the numerical precision.

All these lensing constraints are placed at a fiducial redshift of $z_s = 1.0$.

In the following we will investigate how well we can reconstruct the underlying mass distribution from strong-lensing, weak-lensing and from joint weak- and strong-lensing constraints. We will also analyze the performance...
of the outer-level iteration process to minimize the effect of shape noise on the weak-lensing reconstruction.

4.1 Strong lensing tests

We use the catalog of 55 multiple images to reconstruct the central part of the mock lens from strong-lensing constraints only. We attempt this reconstruction of the cluster core as a proof of concept and important test of our strong-lensing implementation before we move on to combined reconstructions.

In order to embed this core reconstruction into its surroundings in the lens plane we define a set of 50 support points around the area of interest. During the strong-lensing reconstruction these support points are uniformly regularized to the azimuthally averaged, real convergence level in this area of the core. We do not have this information in a real lensing scenario but one has to derive other means of fixing the convergence level in the core surroundings. This is an important step in all lensing reconstructions since most lensing observables are invariant under the mass-sheet transformation (Falco et al. 1985; Gorenstein et al. 1988)

\[ \kappa(\theta) \rightarrow \kappa'(\theta) = \lambda \kappa(\theta) + (1 - \lambda), \]  

with the free\(^7\) transformation parameter \(\lambda\). Later on we will break this degeneracy by including the WL regime in a field that is wide enough to safely assume that the convergence level approaches zero in the regions most distant from the cluster core.

In Fig. 7 we present the results of our strong-lensing only reconstruction. The top panel shows the real convergence map of the lens as a Voronoi representation of the mesh-free evaluation points. The right panel shows the reconstructed convergence map. The agreement is very good in the core of the lens where the lensing constraints are located. This is quantified in the bottom panel which shows the absolute difference between reconstructed and real convergence on the left and the relative difference to the real map on the right. This panel also shows the distribution of strong-lensing constraints in the field. In most areas of the cluster core the agreement is better than ten percent and generally not worse than twenty percent. The agreement gets worse towards the outskirts of the cluster core where less or no strong-lensing information is available and where the reconstruction is plagued by the fact that the support points are simply regularized to an average convergence value as described above. The average absolute error over the full field is 0.002 and the average relative error over the full field is 0.08. The maximum absolute and relative errors are 0.33 and 1.15, which are of course found in the outskirts of the reconstructed field. We conclude that our strong-lensing implementation works reasonably well in areas where this information is available. Ultimately, we will include the weak-lensing regime to overcome the short-comings of this strong-lensing only approach.

\(^7\) The parameter is free up to the point that the density profile of the lens still yields physical solutions, e.g. in terms of kinematics of cluster member galaxies.
4.2 Weak lensing tests

As another proof of concept for our implementation we perform a pure weak-lensing reconstruction of the mock lens. We use the shear catalog without any shape noise contribution first. From the 9000 ideal shear values we randomly pick 900 to serve as evaluation points of the mesh-free reconstruction. Since the data contains no noise component, no outer-level iteration is needed, but we still assign the fiducial value $\sigma = 0.3$ to each weak-lensing constraint. However, we use the reduced shear as weak-lensing input which demands the inner-level iteration to compensate for the non-linear contributions to the $\chi^2$-minimization. In order to correct for the mass-sheet degeneracy we force the very upper-right corner of the reconstructed region to approach a convergence value of zero and perform a weak-lensing reconstruction using the reduced shear input and three inner-level iteration steps. We present the result in Fig. 8 where the top panel shows the real convergence map of the mock lens on the left and the reconstructed convergence map on the right. Both maps follow the same resolution based on the 900 weak-lensing evaluation points. The bottom panels show the absolute difference between the reconstructed convergence map and the real one on the left and the relative difference to the real map on the right. The general agreement is quite striking, which is not surprising given that the data contains no noise component. The average absolute difference between the maps is 0.002 and the average relative difference is 0.02. In general the reconstruction is better than ten percent with some larger errors towards the edges of the field. The maximum absolute and relative errors in the field are found to be 0.08 and 1.7, respectively, found in areas where the signal is vanishing and where we are limited by our way of breaking the mass-sheet-degeneracy.

This first weak-lensing test is far from reality since the reduced shear values did not include any shape noise. We repeat the reconstruction with the reduced shear catalog which contains shape noise. The result is shown in Fig. 9. Obviously our reconstruction is completely dominated by noise and massive overfitting of the data. In order to avoid these problems we implemented the outer-level iteration scheme which is described in Sec. 2.5. For this reconstruction we define six different refinement levels by starting with 150 weak-lensing evaluation points and gradually add 150 more points until we reach the target resolution of 900 nodes. The weak-lensing catalog for this reconstruction is the full ellipticity sample, containing 9000 measurements and shape noise. For each of these six outer-level iterations we perform a distant-weighted average over 80, 50, 30, 22, 18 and 15 weak-lensing shear values, respectively, in order to extract the lensing signal from the noisy data. For the first step with 150 reconstruction cells we regularize on a flat and zero convergence and shear field as initial step. Later on we regularize on the interpolated results of the former, more smooth reconstruction step to avoid overfitting. For each outer-level resolution we perform three inner-level iterations, which is enough for the reconstruction to converge. The result of this reconstruction is shown in Fig. 10 and shows a clear improvement over the naive approach that Fig. 9 is based on. The average absolute error in the convergence is $-0.005$ and the relative error drops to a value of $-0.04$. The maximum absolute and relative errors are $-0.27$ and $2.44$, respectively, which are found in the outskirts of the field where the lensing signal is weakest and the shear values are largely dominated by shape noise.

4.3 Full scale combined reconstructions

This last effort of the testing program brings the pieces together and performs combined weak- and strong-lensing reconstructions. We again draw 900 ellipticity measurements from the shape-noise free weak lensing catalog and also use the 55 multiple-image systems in a first joint reconstruction. The result of this reconstruction is shown in Fig. 11 and it is immediately obvious that, while the field size is identical to the weak-lensing only reconstruction, the resolution in the central area of the lens is increased drastically due to the additional evaluation points defined by the positions of the multiple images. The reconstruction yields an average absolute convergence error of $-0.007$ and the average of the relative error is 0.004. The maximum absolute and relative convergence errors are $-0.29$ and $-2.4$, respectively. In the bottom right panel of Fig. 11 we also show the reconstructed critical curve of the lens and compare it with the real critical curve for our fiducial redshift of $z_l = 1.0$. Only small deviations between the two curves are present and the fact that the reconstructed critical curve gets split into two parts is only due to the limited number of evaluation points. In general, the accuracy of the reconstruction is at the 5–10% level, with the clear exception of areas just
outside the critical curve where the reconstruction overestimates the convergence by 15–20% especially in the areas around the $\{x,y\} = (0.0, -0.2)$ and the $(0.2, 0.2)$ coordinate. We overcome this shortcoming by making use of the one feature in the reconstruction that we have not used, yet. We add the additional 20 sample points of the critical curve of the cluster. We show the reconstruction that adds these constraints in Fig. 12 and again find an excellent reconstruction but with much reduced inaccuracies around the critical curve of the cluster. The average absolute and relative error is now $-0.005$ and $-0.009$, respectively. The maximum absolute error is $-0.25$ and the maximum relative error is $1.75$. The recovery of the critical curve is still excellent as we also show in the bottom right panel of Fig. 12.

5 CONCLUSIONS

In this work we introduced a framework for the mesh-free interpolation and differentiation of functions. This framework is an important and potentially powerful tool for applications in the field of gravitational lensing, because input data does usually not follow any regular pattern in its spatial distribution and is confined to very different length scales. The particular examples for this problem in this work are the regimes of weak- and strong-gravitational lensing.

Our implementation of mesh-free interpolation and differentiation is based on the concept of radial basis functions (RBFs). Specifically, we use Gaussian radial basis functions, although our methodology is not restricted to this one class of RBFs. We convincingly proved the performance of our implementation in Secs. 2 and Apps. A and B. We showed the importance of a well vetted shape parameter of the Gaussian RBF, depending on evaluation-points layout and application and showed that by using an increasing number of nodes in the nearest-neighbors stencils, higher accuracies can be achieved with the drawback of longer runtimes. If all these parameters are chosen appropriately, the accuracy of our interpolation and differentiation routines is well below the percent level.

Using the new techniques to express mesh-free numerical derivatives we implemented a novel method for mass reconstruction from gravitational lensing. We translated the initial ideas of Bartelmann et al. (1996); Bradáč et al. (2005); Cacciato et al. (2006) and Merten et al. (2009) into the realm of a mesh-free and intrinsically adaptive reconstruction and tested the performance of this approach with a simple mock lens. In Sec. 4 we showed that we are able to reconstruct the lens based on 55 multiple images only, at least in the area close to or within its critical line. Here we achieve accuracies at the 5–10% level as we show in Fig. 7. A lensing reconstruction based on strong lensing only clearly suffers from inaccuracies in the areas further away from the critical curve where no constraints are available. However, this region is covered by complemental weak-lensing constraints, which when used as a stand-alone input, allow a reconstruction also at the 5–10% accuracy level but do not cover the strong-lensing core of the lens. The main problem in weak-lensing analyses though is the presence of shape noise in the input data. We showed the performance of the well-
Figure 11. The Voronoi representation of the joint weak- and strong-lensing reconstruction using 900 reduced shear values and 55 multiple-image system. The top panels show the convergence of the real lens on the left and the reconstructed convergence map on the right. Both maps follow a resolution which is defined by the 955 input constraints. Shown in the bottom panels are differences between the real and reconstructed maps in terms of an absolute error on the left and a relative error with the real map as a reference on the right. Also shown in the bottom right panel is the reconstructed critical line in cyan and the real critical line for a source redshift of 1.0 in black. The black line can barely be seen since it is well overlaid by the reconstructed line.

established two-level iteration scheme of Bradač et al. (2005) and Merten et al. (2009) in order to deal with noisy data in Fig. 10. Finally, we combine weak- and strong-lensing constraints to achieve an accurate reconstruction of the mock lens over all relevant length scales. The real surface-mass distribution of the mock lens is recovered at the 5–10% level and also the critical curve of the lens is recovered almost perfectly, as shown in Figs. 11 and 12.

We are well aware that our simple reconstruction of the mock lens was performed under very ideal conditions and shall indeed just serve as a first-order proof of principles. Our main intention in this work was to lay out the mathematical and algorithmic framework which we will make frequent use of in the future. As a next step, we will test our new method with much more realistic ray-tracing simulations along the lines of Meneghetti et al. (2010) and Rasia et al. (2012), where we will create realistic astronomical images, including all relevant noises, from a lensing scenario with a numerically simulated cluster as deflector. Furthermore, we are planning to apply our method not only to mass-reconstruction applications but extend our work to lens-source plane mapping. While lensing features are usually observed on very regular meshes in the lens plane, due to the pixelization scheme of CCD images, the lens mapping transforms this regular pattern into a very irregular one in the source plane. This is a problem both in ray-tracing simulations (Meneghetti et al. 2008) and in the source-plane reconstructions of lensed sources (see e.g. Dye & Warren 2005; Vegetti & Koopmans 2009; Tagore & Keeton 2014, and references therein) Finally, we will apply our reconstruction method to real data and we will explore the usefulness of our approach in the field of PSF interpolation, which also deals with the irregular pattern of star positions in the fields of astronomical observations. Improvements to our general implementation may stem from the introduction of a spatially varying shape parameter (Fornberg & Zuev 2007), although most recent developments in applied mathematics may allows us to discard the shape parameter altogether when using polyharmonic spline-type RBFs (Bengt Fornberg, private communication).

Figure 12. This figure is identical to Fig. 11 but uses in the underlying reconstruction additional 20 constraints on the position of the critical line of the cluster.

ACKNOWLEDGEMENTS

I want to send a warm thank you to Bengt Fornberg for helping me to understand and implement the concept of radial basis functions for the purpose of finite differencing. I also thank Matthias Bartelmann, Massimo Meneghetti, and Leonidas Moustakas for inspiring discussions. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA and I acknowledge support from NASA Grants HST-GO-13343.05-A and HST-GO-13386.13-A.
REFERENCES

Abdelsalam, H. M., Saha, P., & Williams, L. L. R. 1998, MNRAS, 294, 734, ADS, astro-ph/9707207
Bacon, D. J., Goldberg, D. M., Rowe, B. T. P., & Taylor, A. N. 2006, MNRAS, 365, 414, ADS, arXiv:astro-ph/0504478
Bartelmann, M. 2010, Classical and Quantum Gravity, 27, 233001, ADS
Bartelmann, M., Huss, A., Colberg, J. M., Jenkins, A., & Pearce, F. R. 1998, A&A, 330, 1, ADS, arXiv:astro-ph/9707167
Bartelmann, M., Narayan, R., Seitz, S., & Schneider, P. 1996, ApJ, 464, L115+1, ADS, arXiv:astro-ph/9601011
Bartelmann, M., & Schneider, P. 2001, Phys. Rep., 340, 291, ADS, arXiv:astro-ph/9912508
Bradaˇ c, M., Schneider, P., Lombardi, M., & Erben, T. 2005, A&A, 437, 39, ADS
Bradaˇ c, M. et al. 2009, ApJ, 706, 1201, ADS, 0910.2708
Bridle, S. L., Hobson, M. P., Lasenby, A. N., & Saunders, R. 1998, MNRAS, 299, 895, ADS, arXiv:astro-ph/9802159
Broadhurst, T. et al. 2005, ApJ, 621, 53, ADS, arXiv:astro-ph/0409132
Broadhurst, T. J., Taylor, A. N., & Peacock, J. A. 1995, ApJ, 438, 49, ADS, arXiv:astro-ph/9407032
Cacciato, M., Bartelmann, M., Meneghetti, M., & Moscardini, L. 2006, A&A, 458, 349, ADS, arXiv:astro-ph/0511694
Chang, C. et al. 2013, MNRAS, 434, 2121, ADS, 1305.0793
Cole, D., Fosalba, E., Benitez, N., Broadhurst, T., Frye, B., & Ford, H. 2008, ApJ, 681, 814, ADS, 0803.1199
Diego, J. M. et al. 2014, ArXiv e-prints, 1402.4170, ADS, 1402.4170
Diego, J. M., Tegmark, M., Protopapas, P., & Sandvik, H. B. 2007, MNRAS, 375, 958, ADS, arXiv:astro-ph/0509103
Dye, S., & Warren, S. J. 2005, ApJ, 623, 31, ADS, astro-ph/0411452
Falco, E. E., Gorenstein, M. V., & Shapiro, I. I. 1985, ApJ, 289, L1, ADS
Fasshauer, G. F. 2007, Meshfree Approximation Methods with MATLAB (World Scientific Publishing Company)
Flyer, N., Wright, G., & Fornberg, B. 2014, in Handbook of Geomathematics, ed. T. S. W. Freeden, Z. Nashed (Springer)
Fornberg, B. 1998, SIAM Rev, 40, 685
Fornberg, B., Larson, E., & Flyer, N. 2011, SIAM Journal on Scientific Computing, 33, 869, http://dx.doi.org/10.1137/09076756X
Fornberg, B., Lehto, E., & Powell, C. 2013, Computers & Mathematics with Applications, 65, 627
Fornberg, B., & Zuev, J. 2007, Comput. Math. Appl., 54, 379
Gorenstein, M. V., Shapiro, I. I., & Falco, E. E. 1988, ApJ, 327, 693, ADS
Halkola, A., Seitz, S., & Pannella, M. 2006, MNRAS, 372, 1425, ADS, arXiv:astro-ph/0605470
Hirata, C. M., & Seljak, U. 2004, Phys. Rev. D, 70, 063526, ADS, arXiv:astro-ph/0406275
Jee, M. J. et al. 2007, ApJ, 661, 728, ADS, 0705.2171
Johnson, T. L., Sharon, K., Bayliss, M. B., Gladders, M. D., Coe, D., & Ebeling, H. 2014, ArXiv e-prints, 1405.0222, ADS, 1405.0222
Jullo, E., Kneib, J., Limousin, M., Elíasdóttir, Á., Marshall, P. J., & Verdugo, T. 2007, New Journal of Physics, 9, 447, ADS, 0706.0048
Jullo, E., Pires, S., Jauzac, M., & Kneib, J.-P. 2014, MNRAS, 437, 3969, ADS, 1309.5718
Kitching, T. D. et al. 2012, MNRAS, 423, 3163, ADS, 1202.5254
Kneib, J., Ellis, R. S., Smail, I., Couch, W. J., & Sharles, R. M. 1996, ApJ, 471, 643, ADS, arXiv:astro-ph/9511015
Larsson, E., Lehto, E., Heryodono, A., & Fornberg, B. 2013, SIAM J. Sci. Comp., 35, A2096
Liesenborgs, J., De Rijcke, S., & Dejonghe, H. 2006, MNRAS, 367, 1209, ADS, arXiv:astro-ph/0601124
Mandelbaum, R. et al. 2014, ApJS, 212, 5, ADS, 1308.4982
Massey, R. et al. 2013, MNRAS, 429, 661, ADS, 1210.7690
Meneghetti, M. et al. 2008, A&A, 482, 403, ADS, arXiv:0711.3418
Meneghetti, M., Rasia, E., Merten, J., Bellagamba, F., Ettori, S., Mazzotta, P., Dolag, K., & Marri, S. 2010, A&A, 514, A93+, ADS, 0912.1343
Merten, J., Cacciato, M., Meneghetti, M., Mignone, C., & Bartelmann, M. 2009, A&A, 500, 681, ADS, 0806.1967
Merten, J. et al. 2011, MNRAS, 417, 333, ADS, 1103.2772
—. 2014, ArXiv e-prints, 1404.1376, ADS, 1404.1376
Monna, A. et al. 2014, MNRAS, 438, 1417, ADS, 1308.6280
Muja, M., & Lowe, D. G. 2009, in International Conference on Computer Vision Theory and Application VISSAPP’09 (INSTICC Press), 331–340
Newman, A. B., Treu, T., Ellis, R. S., Sand, D. J., Nipoti, C., Richard, J., & Jullo, E. 2013, ApJ, 765, 24, ADS, 1209.1391
Newman, E., & Penrose, R. 1962, Journal of Mathematical Physics, 3, 566
Oguri, M. 2010, PASJ, 62, 1017, ADS, 1005.3103
Postman, M. et al. 2012, ApJS, 199, 25, ADS, 1106.3328
Rasia, E. et al. 2013, MNRAS, 429, 661, ADS, 1201.1569
Schneider, P., & Er, X. 2008, A&A, 485, 363, ADS, 0709.1003
Schneider, P., & Seitz, C. 1995, A&A, 294, 411, ADS, arXiv:astro-ph/9407032
Seitz, S., Schneider, P., & Bartelmann, M. 1998, A&A, 337, 325, ADS, arXiv:astro-ph/9803038
Smith, G. P., Kneib, J.-P., Smail, I., Mazzotta, P., Ebeling, H., & Czoske, O. 2005, MNRAS, 359, 417, ADS, 0403588
Tagore, A. S., & Keeton, C. R. 2014, MNRAS, 445, 694, ADS, 1408.6297
Vegetti, S., & Koopmans, L. V. E. 2009, MNRAS, 392, 945, ADS, 0805.0201
Williams, L. L. R., & Saha, P. 2011, MNRAS, 502, ADS, 1102.3943
Zitrin, A. et al. 2011, ApJ, 742, 117, ADS, 1103.5618
—. 2009, MNRAS, 396, 1985, ADS, 0902.3971

APPENDIX A: INTERPOLATION WITH RADIAL BASIS FUNCTIONS

In the following we test the robustness and accuracy of the interpolation in 2D using a Gaussian radial basis function.
The evaluation point configurations we use for our performance tests of RBF interpolation and differentiation. The red points in the top left panel show a regular mesh with 900 nodes and no refinement. The top right panel shows a regular mesh of 900 nodes with two refinement levels towards the center. The bottom panels show 900 randomly chosen nodes on the unit disk. The example in the bottom left panel is not refined, while the bottom right panel includes two levels of refinement towards the center of the disk. In each refinement step, the density of random points doubles. The blue circles show an ensemble of 180 evaluation points for each setup, which anchor the interpolant in the interpolation test.

The main free parameters in this analysis are the shape parameter of the Gaussian RBF, the number of nearest neighbors used in the interpolation stencil and the number of evaluation points to perform the interpolation.

We use the analytic test function from Eq. 9 and define four different kind of node layouts, all of which are shown in the four panels of Fig. A1. The first layout is a regular, square mesh with 900 nodes. The second mesh has the same number of total nodes but is refined twice towards the center of the mesh. The third layout is defined by 900 random nodes on the unit disk, as is the fourth layout, with the difference that also this configuration is refined twice towards the center of the disk. The last two examples define mesh-free sets of nodes. In the following, we interpolate the test function on these domains. For illustration purposes we evaluated Eq. 9 on the nodes of all four configurations in Fig. A2.

In order to perform interpolations, we define as a first step a set of 180 evaluation points in each of the four test cases. The evaluation points for each node layout are shown as blue circles in Fig. A1. We calculate the interpolant \( f(x, y) \) of the test function \( f(x, y) \) by using Eq. 2 and define two performance metrics. The average relative error \( \langle (\tilde{f}(x, y) - f(x, y))/f(x, y) \rangle \) for all nodes \( \vec{x} = (x, y) \) and the maximum relative error \( \max((\tilde{f}(x, y) - f(x, y))/f(x, y)) \) for all nodes \( \vec{x} = (x, y) \). We evaluate both metrics as a function of the shape parameter and the number of nearest neighbors used to derive the interpolant at a given node \( \vec{x} \). For the regular, nonrefined mesh we plot the results in Fig. A3 and for the nonrefined, mesh-free setup in Fig. A4. As one can see, the overall performance is excellent once the right shape parameter is found. With the use of 16 nearest neighbors or more, the average relative errors approach the \( 10^{-5} \) level and the maximum error approaches \( 10^{-3} \). The performance is slightly better for the mesh-free setup, which is due to the fact that the RBF is approach is not well suited to treat the edges of the regular mesh. The same holds for the interpolation on the refined node layouts. We show the results for the regular, refined mesh in Fig. A5 and for the random, mesh-free node layout in Fig. A6. The performance in the latter case is similar to the unrefined one, but the performance drops slightly for the regular, refined mesh. Also here the RBF approach is not ideal to treat the abrupt transitions between the different refinement levels, which are not well described by a radially dependent function. However, the overall performance in all four cases is remarkable.
Figure A4. This figure shows the same plot as in Fig. A3, but for a mesh-free, unrefined node layout with 180 evaluation points.

Figure A5. This figure shows the same plot as in Fig. A3, but for a refined regular mesh with 180 evaluation points.

Figure A6. This figure shows the same plot as in Fig. A3, but for a random, mesh-free node layout with two levels of refinement towards the center.

Figure A7. The accuracy of the interpolation on the mesh-free, refined node layout as a function of the number of evaluation points. In the case of 9 and 27 evaluation points, 8 and 24 nearest neighbors were used respectively. In all other cases, 32 nearest neighbors were used. Shown is the average and maximum relative error for all nodes.

As a last test we vary the number of evaluation points in the mesh-free, refined node layout, going from 1% of the total number of nodes, 9, to 50% of the total number of nodes, 450. We plot both performance metrics as a function of the number of evaluation points in Fig. A7, where the shape parameters and the number of nearest neighbors in each case are optimally chosen. The interpolation accuracy is clearly a steep function of the number of evaluation points, where very good results ($<10^{-5}$ average and $<10^{-3}$ maximum relative error) can be achieved with a large number of evaluation points exceeding one quarter of the total number of nodes. It is remarkable however, that even with a very small number of evaluation points, only a few percent of the total number of grid nodes, good results can be achieved at least for the average relative error of all nodes. Accuracies at the percent level are possible although the maximum error can be close to or exceed 100%. We investigate the dependence of the interpolation accuracy on the number and position of the support points a little further in Fig. A8, where we show the relative interpolation error for four different evaluation point setups. Not surprisingly, the local performance of the interpolation strongly depends on the position of the evaluation points. Areas of high interpolation accuracy can be found near the evaluation points while the accuracy drops significantly in areas far from any evaluation point.

APPENDIX B: FINITE DIFFERENCING WITH RADIAL BASIS FUNCTIONS

To test the accuracy of numerical differentiation using radial basis function finite differencing stencils (RBF FD), we
return to Eq. 9 and calculate some of its derivatives

$$\frac{\partial f(x, y)}{\partial x} = 4 \cos(4x) - 3 \sin(3x)$$

$$\frac{\partial^2 f(x, y)}{\partial x^2} = -16 \sin(4x) - 9 \cos(3x)$$

$$\frac{1}{2} \Delta f(x, y) = -8 \sin(4x) - 4.5 \cos(3x) - 2 \sin(2y).$$

In the following we perform tests on the mesh-free refined node layout of App. A only, since it resembles best the conditions in our real lensing applications. We visualize the test function and the three derivatives of interest for this node configuration in Fig. B1.

We use Eq. 4 to calculate the derivatives of the test function. This operation has the Gaussian RBF shape parameter and the number of nearest neighbors as free parameters. We again define the average and the maximum error for all nodes as performance metrics and plot them for the first x derivative in Fig. B2. As expected, the right choice of the shape parameter is crucial and the accuracy steadily increases when using more nearest neighbors to calculate the numerical derivatives. This also applies to the accuracy of the second numerical derivative shown in Fig. B3 and for the accuracy of the numerical Laplacian as shown in Fig. B4.

We note that the accuracy of the second derivatives is about an order of magnitude lower than for the first derivative, but with the right choice of shape parameter and an adequate number of nearest neighbors, average relative errors on the grid of $< 10^{-3}$ and maximum errors of $< 10^{-1}$ can be achieved throughout. To analyze the local performance of the finite differencing stencils we show the relative error for each node and for four different choices of the number of nearest neighbors in Fig. B5. The shape parameter is chosen to be optimal in each case. No clear trends can be found, despite the expected drop of performance near the edges of the grid and a smaller accuracy where the gradient in the derivative is the largest. This drop can possibly be overcome with a locally varying shape parameter (Fornberg & Zuev 2007), but the performance clearly suffices for our purposes.

APPENDIX C: LINEARIZATION OF THE LIKELIHOOD FUNCTION

In the methodology outlined in Sec. 3.2 we need to minimize a complicated $\chi^2$-function with the lensing potential in each evaluation point as a free parameter. This function consists of a contribution from a reduced-shear term, a critical-line estimator term, a multiple-image system term and a regul-
Figure B3. This figure is identical to Fig. B2 but shows the RBF FD performance for the second x derivative.

Figure B4. This figure is identical to Fig. B2 but shows the RBF FD performance for the 2D Laplacian of the test function multiplied by 1/2.

Figure B5. The local performance of the finite differencing stencils. The top left panel shows the relative error for the first x derivative when using 16 nearest neighbors. The top right panel shows the same relative error for 32 nearest neighbors. The bottom left and right panel show the second x derivative and the 2D Laplacian using 32 nearest neighbors, respectively. The shape parameter was chosen to be optimal in each case.

C1 Weak-lensing term

Eq. 22 defines the weak-lensing term in the potential reconstruction in which we isolate the non-linear components and

\[ B_{lk} = B_{lk}^{\text{w}} + B_{lk}^{\text{m}} + B_{lk}^{\text{reg}} + B_{lk}^{\text{seg}}, \]  
\[ \mathcal{V}_{l} = \mathcal{V}_{l}^{\text{w}} + \mathcal{V}_{l}^{\text{m}} + \mathcal{V}_{l}^{\text{reg}} + \mathcal{V}_{l}^{\text{seg}}. \]  

\[ N_{ij} = \frac{C_{ij}^{-1}}{(1 - Z_i \kappa_i)(1 - Z_j \kappa_j)}. \]  

This term is sequentially updated with new convergence values from the previous inner-level iteration and held constant during the current inner-level iteration. Following the calculation in Appendix 1 of Merten et al. (2009) we find the linear system of equations

\[ B_{lk}^{\text{w}} = N_{ij} \left[ \epsilon_i \epsilon_j Z_i Z_j K_{ik} K_{jl} - \frac{1}{2} \epsilon_i Z_i Z_j G_{ik} G_{jl} \right], \]  
\[ \mathcal{V}_{l}^{\text{w}} = N_{ij} \left[ \epsilon_i \epsilon_j Z_i Z_j K_{il} - \frac{1}{2} \epsilon_i Z_i Z_j G_{il} \right]. \]  

In the equations above we adopt Einstein’s sum convention.

C2 Critical-line estimator term

The calculation for this term starts with Eq. 23 and again following the calculation in Appendix 1 of Merten et al. (2009) we find another linear system of equations with coefficient matrix and data vector

\[ B_{lk}^{\text{s}} = \frac{4(\det A)}{{\sigma^2}} \left( Z_i^2 (K_{ik} Z_{il} - G_{ik} G_{il} - G_{ik}^2) \right), \]  
\[ \mathcal{V}_{l}^{\text{s}} = \frac{4(\det A)}{{\sigma^2}} Z_i K_{il}. \]  

Also here we adopt Einstein’s sum convention.
C3 Multiple-image systems term

This calculation is not part of Merten et al. (2009) and we will use a slightly different approach than Bradač et al. (2005). Starting point is Eq. 27 which we partially differentiate after \( \psi_k \) and use Eq. 17 and Eq. 21.

\[
\frac{\partial \chi^2_m}{\partial \psi_l} = \sum_{n=1}^{N_s} \frac{2}{\sigma^2} [\theta_n D_{nl} + D_{nk} D_{nl} \psi_k] \\
+ \frac{1}{N_s} \sum_{i=1}^{N_n} (\theta_i D_{il} - D_{il} D_{nk} \psi_k) \\
+ \frac{1}{N_s} \sum_{i=1}^{N_n} (\theta_i D_{nl} - D_{nl} D_{ik} \psi_k) \\
- \frac{1}{N_s^2} \sum_{i,j=1}^{N_s} (\theta_j D_{il} - D_{jk} D_{il} \psi_k)]
\]

(C8)

After sorting terms with and without \( \psi_k \)-term, we find the linear system of equations

\[
B_{lk}^m = \sum_{n=1}^{N_s} \frac{2}{\sigma^2} \left[ D_{nl} D_{nk} - \frac{1}{N_s} \sum_{i=1}^{N_n} (D_{nk} D_{il} + D_{nl} D_{ik}) \\
+ \frac{1}{N_s^2} \sum_{i,j=1}^{N_s} D_{il} D_{jk} \right]
\]

(C9)

and data vector

\[
V_l^m = \sum_{n=1}^{N_s} \frac{2}{\sigma^2} \left[ \theta_n D_{nl} - \frac{1}{N_s} \sum_{i=1}^{N_n} (\theta_i D_{il} + \theta_D D_{nl}) \\
+ \frac{1}{N_s^2} \sum_{i,j=1}^{N_s} \theta_j D_{il} \right]
\]

(C10)

where \( \theta \) is one component of the observed lens-plane position of the multiple images of the system. In order to obtain the full multiple image term contribution, one has to sum the contributions from both coordinate components by substituting \( \theta \) with \( \theta_1 \) or \( \theta_2 \) and \( D \) with \( D^1 \) or \( D^2 \), respectively.

If there are more than one multiple-image systems in the lens, each system contributes a \( \chi^2 \)-term.

C4 Regularization term

For the derivation of this term we start from Eq. 28 and perform the \( \chi^2 \)-minimization

\[
\frac{\partial \chi^2_{\text{reg}}(\psi_k)}{\partial \psi_l} = \eta_i \frac{\partial}{\partial \psi_l} (\kappa_i \psi_k)^2 \\
= 2 \eta_i (\kappa_i \psi_k) \left( -\frac{\partial}{\partial \psi_l} K_{ik} \psi_k \right) \\
= 2 \eta_i (\kappa_i \psi_k - K_{ik} \psi_k) (-K_{il}) \\
= 2 \eta_i (-\kappa_i \psi_k K_{il} + K_{ik} K_{il} \psi_k).
\]

(C11)

This leads to the linear system of equations

\[
B_{lk}^{\text{reg}} = \eta_i K_{ik} K_{il}
\]

(C12)

\[
V_l^{\text{reg}} = \eta_i \kappa_i \gamma_i \psi_k
\]

(C13)

Additional regularization terms for the shear are also possible and lead to

\[
B_{lk}^{\text{reg}} = \eta_i G_{ik} G_{il}
\]

(C14)

for each shear component. To calculate the contributions for both shear components, one needs to substitute \( \gamma \) with \( \gamma^1 \) or \( \gamma^2 \), respectively.