Grid-based Strong Gravitational Lensing*

BY LÉON KOOPMANS

Kapteyn Astronomical Institute, University of Groningen, Netherlands

Email: koopmans@astro.rug.nl

Abstract

In this lecture I present a short mathematical introduction to non-parametric/grid-based strong gravitational lens modeling. It is assumed that the reader is familiar with the basics of strong gravitational lensing, linear algebra and Bayesian statistics.

1 What is grid-based strong lensing?

As opposed to "parametric" strong lensing, which often indicates that the source and lens-potential model of the lens system can be described by a small set of parameters (e.g. image positions, flux-ratios, time-delays for the lensed images, and lens strength, ellipticity, scale, etc. for the lens potential model), "grid-based" strong lensing means that the source and lens-potential models are described by a *large* set of parameters that *directly* quantify a grid of source-brightness and/or lens-potential values. Hence in short summary: *"Non-parametric" really means "lots of parameters!"*.

1.1 Why use grid-based strong lensing?

In many cases parametric strong lens modeling can not be used. For example, there are lens systems with extended and complex lensed images for which one can not easily relate structure in one lensed image to structure in another lensed image (this is often trivial in case of point images and jets). Besides this, grid-based strong lensing:

- makes very little to no assumptions about the structure of the source (and possibly the lens potential as well; see end of this lecture), although the solutions often require regularization because the number of free parameters can be large.
- makes use of all (or most) information available in the lensed images or even absence of information; i.e. the prediction of lensed images that are not present in the data are penalized.
- allows structure of the source to be separated from structure in the lens potential, in a statistical (i.e. Bayesian) sense.

In some cases, however, full grid-based modeling is not justified or extremely difficult to implement. For example, images that consist mostly of very compact structures (e.g. flat-spectrum radio sources/jets) or have high dynamic ranges (quasars plus faint host galaxies) are often hard to model. This can possibly be overcome using adaptive source grid (not part of this lecture). In addition, under certain assumptions, one can use only the azimuthal structure of Einstein rings to help constrain the lens potential, without the full use of its surface brightness distribution.

1.2 Outline of lecture

Before continuing, I will give a short outline/overview of the lecture:

a) First, a data-model is build, which describes the observed data (e.g. CCD image) as function of a set of (non)linear parameters. The source is described by a set of linear parameters (i.e. their surface brightness values on a predefined source grid), whereas the lens-potential is described by a set of non-linear parameters that parametrically describe the lens potential in the image plane.

*. Adapted for the school on gravitational lensing in Cargese, France, September 2012.

b) To solve for the linear parameters (given a fixed potential model), a quadratic penalty function is defined, using the above data-model, the first part of which is a standard χ^2 term and the second part is a regularization term that results in the smoothest source model allowed by the data. The minimum of this penalty function is found through standard linear-algebra techniques.

c) Once a good starting model has been found, the relative weight (λ) between the χ^2 and regularization terms in the penalty function is found through an Bayesian evidence optimization, subsequently marginalizing over this nuisance parameter, giving the "evidence"[1] of the set of non-linear model parameters.

d) The best non-linear parameters are then found by maximizing the evidence penalty-function, using standard non-linear optimization methods, like Downhill-Simplex, Conjugate-Gradient methods, etc (see Numerical Recipes).

e) Finally, the lens-equations can be linearized to include a linear correction to the lens-potential in the equations. This fully-linear equation can then iteratively be solved.

The above outline indicates what can be done in grid-based lensing techniques, although simplified version are possible (e.g. modeling can be done without Bayesian evidence optimization) or more complicated version (e.g. adaptive grids).

2 Building the data model

The underlying idea of grid-based strong-lens modeling of the lensed images is based on the fact that surface brightness is conserved in gravitational lensing, that is to say, in the absence of other effects (e.g. PSF smearing, dust absorption), the surface brightness at a given point $\vec{x}$ in the image plane is identical to the surface brightness of a point $\vec{y}$ in the source plane, connected by the lens equation $\vec{y} = \vec{x} - \vec{\nabla}\psi(\vec{x})$. In other words only *magnification* and not amplification occurs in lensing! This conservation of surface brightness can formally be written as:

$$s(\vec{y}) = d(\vec{x}) \text{ with } \vec{y} = \vec{x} - \vec{\nabla}\psi(\vec{x}),$$

where $s(\vec{y})$ is the source surface brightness distribution and $d(\vec{x})$ is the observed lensed image (i.e. the data; excluding PSF smearing, but see below for a discussion of this effect). The lens potential is given by $\psi(\vec{x})$. The question is what kind of equation this is *and* how can it be solved? We examine this in more detail in the next section.

2.1 A grid-based data model

Let us assume that the source can be described by a set of surface brightness values on a grid (finite) in the source plane. Hence the source model is an array (or vector) of unknown source surface brightness values $s(\vec{y}_{k,l})$ or in vector notation $\vec{s}_{k+l\times \dim_k} = \vec{s}_{k,l} = s(\vec{y}_{k,l})$, where the element $k + l \times \dim_k$ in the vector $\vec{s}$ is the element (k, l) on the grid $\vec{y}_{k,l}$, with $k = 0...(\dim_k - 1)$ and $l = 0...(\dim_l - 1)$.

Remark 1. Note that we have not yet specified whether the points (k, l) lie on a regular or irregular (possibly adaptive) grid. This specification only becomes necessary when interpolation (or other operations) are necessary on the source plane (see for example below).

In the image plane, we often have a *data set* on a (regular) grid as well (e.g. on a CCD image), i.e. $\vec{x}_{i,j}$. This data set can be written as a vector as well[2], i.e. $\vec{d}_{i+j\times \dim_i} = \vec{d}_{i,j} = d(\vec{x}_{i,j})$, with $i = 0...(\dim_i - 1)$ and $j = 0...(\dim_j - 1)$.

At this stage, how do we build a "data-model" for the observed data $\vec{d}$?

1. The "evidence" is the average likelihood in the parameter-space allowed by the data-model (see below).

2. Note that the pixels of a CCD image are not really surface-brightness values at a given point, but surface brightness values averaged *within* the pixel. This effect can be closely mimicked by re-sampling the model image grid and re-binning it when compared with the real data. See further in this lecture.

This is an important next step and it requires that we have to relate the surface brightness value of a single point in $\vec{d}$ (i.e. in the image plane) with the (as of yet unknown) surface brightness values in the source plane, $\vec{s}$. This relation is the "data-model". The data model basically gives us a model $\vec{d}\,'$ of the observed data $\vec{d}$ for a (yet unknown) source model $\vec{s}$. The latter is to be determined from the data model through some sort of inversion method.

Let us now also define a regular (i.e. rectangular pixels) grid in the image plane, $\vec{x}_{i,j}$, and in the source plane, $\vec{y}_{k,l}$. Let us first concentrate on a single surface-brightness point in the image plane $\vec{d}_{i,j}$ and how it can be modeled using $\vec{s}$. This can be done in a number of steps:

1. First one maps the point $\vec{x}_{i,j}$ with a corresponding data value $d_{i,j}$ on to the source plane using the lens equation $\vec{y}'_{i,j} = \vec{x}_{i,j} - \vec{\nabla}\psi(\vec{x}_{i,j};\vec{p})$, where $\vec{p} = \{p_1 ... p_n\}$ is a set of parameters that describes the form of the lens potential ψ (note also the distinction between $\vec{y}'_{i,j}$ and $\vec{y}_{k,l}$, where the former is position in the source plane related to the image-plane grid point $\vec{x}_{i,j}$ and the latter are fixed grid positions in the source plane).
2. This point $\vec{y}'_{i,j}$ in the source plane, in general, will *not* correspond to any point on the (regular) source grid $\vec{y}_{k,l}$, but will be delimited by a set of four pixels $\vec{y}_{k+\mu,l+\nu}$ with $\mu = 0, 1$ and $\nu = 0, 1$ (it can also fall outside the source grid in which case it is masked).
3. In that case, what is the surface brightness at $\vec{y}'_{i,j}$ given $\vec{y}_{k+\mu,l+\nu}$? This can easily be determined through bi-linear interpolation. Let us assume that $\vec{y}_{k'+\mu,l'+\nu}$ delimit $\vec{y}'_{i,j}$ in such a way that the y_1 and y_2 coordinates of $\vec{y}_{k'+\mu,l'+\nu}$ for $\mu = \nu = 0$ are each smaller than those of $\vec{y}'_{i,j}$. In general we can assume $\vec{y}_{k,l} = (y_{1,\min} + k \times \delta y_1,\, y_{2,\min} + l \times \delta y_2)$, where δy_1 and δy_2 are the pixel scales in the x and y direction in the source plane. Similarly the grid in the image plane can be defined as $\vec{x}_{i,j} = (x_{1,\min} + i \times \delta x_1, x_{2,\min} + j \times \delta x_2)$, where δx_1 and δx_2 are the pixel scales in the x and y direction in the image plane.

Remark 2. The surface brightness value at $\vec{y}'_{i,j}$ can also be obtained through other lower or higher-order interpolation, or other means. The choice for using the four delimiting pixels is, in that sense, somewhat arbitrary, but this choice minimizes the computational effort and correlation between pixel values. The choices to be made are therefore the grids that one plans to use (these can be adaptive; in this lecture we assume a regular grid) and the interpolation scheme (in this lecture bilinear interpolation). The other requirement is then an algorithm to find those pixels in the source plane that delimit the position $\vec{y}'_{i,j}$. This is very easy in the case of regular grid, but more difficult in the case of irregular grids.

4. Let us further define the relative distance: $(t, u) = ((y'_{1,i,j} - y_{1,k',l'})/\delta y_1, (y'_{2,i,j} - y_{2,k',l'})/\delta y_2)$ with $\vec{y}_{k',l'} = (y_1, y_2)_{k',l'}$. Using this definition the surface brightness at $\vec{y}'_{i,j}$ can be written as

$$s'_{i,j} = \sum_{\mu=0}^{1} \sum_{\nu=0}^{1} w_{k'+\mu,l'+\nu}\, s_{k'+\mu,l'+\nu}$$

with

$$\begin{cases} w_{k',l'} &= (1-t)(1-u) \\ w_{k'+1,l'} &= t(1-u) \\ w_{k'+1,l'+1} &= tu \\ w_{k',l'+1} &= (1-t)u \end{cases}$$

All other weights are zero. (see Numerical Recipes).

Exercise 1. Show that the above determination of $s'_{i,j}$ give indeed the correct values of the surface brightness when $\vec{y}\,'$ approaches the position of the four delimiting source pixels.

5. We can write this also as (note the $\cdots^T$ stands for a transpose operation):

$$s'_{i,j} = \vec{l}^{\,T}_{i,j}\, \vec{s}$$

with

$$\vec{l}_{i,j}^{\,T} = (\ldots\, w'_{k'+\mu,l'+\nu}\,\ldots),$$

Where the weights w are placed at the positions $(k'+\mu, l'+\nu)$ as calculated and discussed above. Now it is easy to construct a full matrix L, the "lens operator", that gives a model of $\vec{d}$, i.e. $\vec{s}\,'$, as follows

$$\boldsymbol{L}(\psi) \equiv \begin{pmatrix} \vec{l}_{0,0}^{\,T} \\ \vdots \\ \vec{l}_{i,j}^{\,T} \\ \vdots \\ \vec{l}_{\mathrm{ndim}_i,\mathrm{ndim}_j}^{\,T} \end{pmatrix}$$

such that our data model (at this point) becomes:

$$\vec{d}\,' = \boldsymbol{L}(\psi)\vec{s}\,,$$

where we explicitly indicate the lens potential ψ to show that it is used to build the lens operator.

Exercise 2. Show that if $\vec{s} = \vec{1}$ (i.e. surface brightness is 1 on all pixels) that $\vec{d}\,' = \vec{1}$ as well, regardless of the lens potential ψ.

6. Are we done here? No! Even though we have a data model, real data is often smeared by the PSF. In fact the PSF "distributes" flux from where it really is in the sky to where it is observed in the CCD image (for example). This blurring can also be easily written as an operator (which conserves flux, but not surface brightness!). Hence our final data model becomes:

$$\vec{d}\,' = \boldsymbol{B}\boldsymbol{L}(\psi)\vec{s}$$

Which now includes the effect of PSF smearing. This equation is linear and can be solved through standard algebraic techniques (e.g. Cholesky decomposition).

7. Something discussed shortly is that the data values in e.g. a CCD pixel in fact represents an integral over the PSF-smeared surface brightness inside that pixel. This effect can be mimicked by sub-sampling the data-grid on a grid with $n_s \times n_s$ sub-pixels and calculating the lens operator $\boldsymbol{L_s}(\psi)$ as before for this sub-sampled grid. Similarly, the PSF smearing $\boldsymbol{B_s}$ is done on the sub-sampled grid. After this, a matrix operator $\boldsymbol{G_s}$ simply adds all sub-pixels together inside a data pixel with weights $1/n_s^2$ to mimic the integration:

$$\vec{d}\,' = \boldsymbol{G_s}\boldsymbol{B_s}\,\boldsymbol{L_s}(\psi)\vec{s}$$

Exercise 3. Calculate the "sparseness" of the lens operator $\boldsymbol{L}$, that is to say the fraction of non-zero elements in the matrix. Note that the lens operator is indeed very sparse and can thus be quickly calculated, as apposed to the often dense blurring operator.

2.2 Grid masks

Whereas we now have a linear data model for our data $\vec{d}$, we have omitted one important fact, which is the result of the *finite* sizes of the source, data and potential (see below) grids. Their finite size sometimes (in fact often) means that not all positions $\vec{x}_{i,j}$ map onto the finite source grid, i.e. there are *no* grid-points that delimit the point cast on the source plane (hence interpolation is impossible!). Similarly there might be grid-points in the source plane that are not constraint at all by the data (this case is less important because source regularization, discussed below, deals with this efficiently).

To deal with these two cases, we need to define two masks (when only dealing with the gridded source model): the data mask $\vec{m}^{\,d}_{i,j}$ and the source mask $\vec{m}^{\,s}_{k,l}$. The former mask is 1 everywhere except for those image-grid pixels that do not map inside the source grid. The latter mask is also 1 everywhere except for those source-grid pixels that are not constrained by any data.
The data masks are most important in properly assessing the penalty function that we need to minimize in order to find the correct solution for the source model $\vec{s}$, but will not be explicitly mentioned further in this lecture.

3 Solving the data model

Now that we have a data-model, how can we solve it? The simplest idea is to simply invert the data model (using a pseudo-inverse) to obtain a solution

$$\vec{s}\,' = \overbrace{[\boldsymbol{BL}(\psi)]^{+}}^{\mathrm{Pseudo-inverse}}\ \vec{d}\,.$$

This however is often not possible (the inverse simply does not exist, or is not unique), and if possible, the inversion is ill-posed and noise is often dominant in the inversion. The solution will therefore look *very* bad indeed! To overcome this problem, instead of inverting the above equation, one might want to define a penalty function, based on the difference between $\vec{d}$ (the data) and $\vec{d}\,'$ (the data model), which is subsequently minimized by varying the free parameters of the data model (i.e. $\vec{s}$ and $\vec{p}$).

3.1 Penalty function of the data model

Let us first define a penalty function based on how well the model represents the data

$$\chi^2(\vec{s}\,,\vec{p}\,) = [\vec{m}^{\,d} \circ (\boldsymbol{BL}(\psi(\vec{p}\,))\vec{s} - \vec{d})]^T C^{-1} \overbrace{[\vec{m}^{\,d} \circ (\boldsymbol{BL}(\psi(\vec{p}\,))\vec{s} - \vec{d})]}^{\mathrm{residual\,image}},$$

where $\circ$ indicates a Hadamard (element-wise) product and C^{-1} is the inverse noise covariance matrix with $[C]_{\mathrm{ij}} = \langle n_i n_j \rangle$ where n_i^2 is the noise variance in pixel i. We have explicitly written χ^2 to depend on the non-linear parameters $\vec{p}$ that describe the lens potential, i.e. $\psi(\vec{p}\,)$.

The above penalty is the standard χ^2 penalty function (which we will assume throughout this lecture) which, when the errors are Gaussian, also gives the maximum-likelihood (ML) solution. The masks can implicitly be incorporated in the data or lens operator at each step: if a data point is not modeled/constrained, one can set its value to zero in the vector $\vec{d}$ as well as the corresponding weights in the corresponding $\vec{l}^{\,T}$ so it does not count toward the penalty and the values of $\vec{s}$ are not constrained improperly (since they might still constrain other data points), hence we will not explicitly write the masks anymore. Similarly, if a source pixel is not constrained by the data, its value can be arbitrary in the above penalty function without changing the value of the penalty (since it does not change the data-model). Hence it can be neglected in the optimization. However, for aesthetic and plotting purposes, one should keep track of the masks in any case.

3.2 Regularization of the source solution

Even though the above penalty function is already a large improvement over the simple inversion (when that is possible at all!), the solution of $\vec{s}$ that minimizes χ^2 is often still not optimal (noise is still amplified, since what we in fact are doing is deconvolving the lensed images which is a very unstable process sensitive to noise). To improve upon that situation, one needs to *regularize*[3] the source solution with an additional additive penalty function:

$$P = \chi^2(\vec{s}\,,\psi) + \lambda\,\mathrm{Reg}(\vec{s}\,),$$

where $\mathrm{Reg}(\vec{s}\,)$ is a penalty function based only on the values of $\vec{s}$ and where λ is the regularization parameters that sets the relative weight between the χ^2 and the level of regularization in the penalty function.

There are many choices for the regularization, one of the most used being the maximization of the entropy of the solution. This, however, leads to a non-linear (although still easily solvable) equation, which we will not discuss further in this lecture.

3. Regularization is in itself a debatable topic, quite closely related to placing priors on the allowed solution space in Bayesian statistics. We will not enter further into that sometimes heated debate, but note that the regularized solution often resembles real sources more closely than non-regularized solutions (i.e. we are using the prior called "experience"!).

3.2.1 Quadratic Regularization

In this lecture, we concentrate on quadratic regularization penalty functions, i.e. those that can be written as $\mathrm{Reg}(\vec{s}) = \|\boldsymbol{H}\vec{s}\,\|^2 = \vec{s}^{\,T}(\boldsymbol{H}^T\boldsymbol{H})\vec{s}$. In this case the penalty function becomes

$$P_{\chi^2} = [\boldsymbol{B}\boldsymbol{L}(\psi)\vec{s} - \vec{d}\,]^T \boldsymbol{C}^{-1}\,[\boldsymbol{B}\boldsymbol{L}(\psi)\vec{s} - \vec{d}\,] + \lambda_s\,[\vec{s}^{\,T}(\boldsymbol{H}^T\boldsymbol{H})\vec{s}]$$

In general, the choice of $\boldsymbol{H}$ is driven by our desire to obtain a source model that is as smooth as possible (note that this is pure prior believe on what the source structure should be; in general if two models give equal likelihood values, the smoother model is preferred, based on "experience".). A good choice for the regularization operator $\boldsymbol{H}$ is often therefore either the operation ∇ or ∇^2. Hence either the square of the gradient or the square of the curvature, integrated over the source grid, is minimized.

So what is the solution of $\vec{s}$ that minimizes this penalty function? To obtain that, we need to do some linear-algebra calculus to determine $\partial P_{\chi^2}/\partial\vec{s} = \vec{0}$. This results in the following linear equation:

$$(\boldsymbol{M}^T\boldsymbol{C}^{-1}\boldsymbol{M}^T + \lambda\,\boldsymbol{H}^T\boldsymbol{H})\vec{s} = \boldsymbol{M}^T\boldsymbol{C}^{-1}\vec{d}\,,$$

with $\boldsymbol{M} \equiv \boldsymbol{B}\boldsymbol{L}(\psi)$. Note that the above equation is also a linear equation that can be solved using standard techniques (e.g. Cholesky decomposition). The solution to this equation now minimizes the penalty function P_{χ^2}, including regularization.

Exercise 4. Show that the solution of $\partial P_{\chi^2}/\partial\vec{s} = \vec{0}$ leads to the above linear equation. Remember that $\partial(\vec{x}^{\,T}\vec{a}\,)/\partial\vec{x} = \partial(\vec{a}^{\,T}\vec{x})/\partial\vec{x} = \vec{a}$ and that $\partial(\vec{x}^{\,T}\boldsymbol{A}\vec{x})/\partial\vec{x} = (\boldsymbol{A} + \boldsymbol{A}^T)\vec{x}$.

3.3 Solving for the lens potential parameters

Now that we devised a way to solve for the source model – given a potential $\psi(\vec{p}\,)$ –, how do find the best parameters, $\vec{p} = \{p_1 ... p_n\}$, of the potential itself? This in general is a non-linear problem and therefore of very different nature than what we have done above. There are two distinct cases for the potentials:

a) The lens potential parameters $\vec{p}$ quantify a number of physical (or geometric) characteristics of the lens potential, such as lens strength, ellipticity, etc. In this case, a non-linear optimizer (e.g. the CERN package `MINUIT` is quite useful; or other techniques e.g. gradient methods, downhill-simplex, MCMC, simulated annealing, etc.) can be used to vary the lens potential parameters, each time re-determining the source model $\vec{s}_{\rm min}$ that minimizes P_{χ^2} for that set of parameters and then minimizing $P_{\chi^2}(\vec{s}_{\rm min}; \vec{p}\,)$ as function of the lens-potential parameters $\vec{p}$. Non-linear optimization is in itself a complete lecture, a complex topic, and will not be further discussed during this lecture.

b) The lens potential parameters $\vec{p}$ actually quantify the values of the potential depth itself on some pre-defined set of positions in the image plane. This case will be discussed a bit further in this lecture. In that case, the values of $\vec{p}$ can be found by linearizing all previous equations, solving for small corrections $\delta\psi(\vec{p}\,)$, add this correction to the previous solution $\psi(\vec{p}\,)$, and iterate this until convergence is reached.

3.4 Recapitulation and a-priori/systematic assumptions

At this point, it is time to recapitulate what we have done so far:

- We defined a source model on a finite (ir)regular grid, and written it as a vector, with each grid point having a value of the source surface brightness assigned to it. The observed data (surface-brightness values) is also defined on a grid. Our goals was to relate these two through a data model based on the lens potential.
- We constructed a lensing operator, as function of the lens potential parameters, which when acting on a source model (written as vector) results in a lensed source, defined on the same grid-points in the image plane as the observed data. The latter grid was convolved with a blurring operator to account for the effect of the PSF smearing (re-sampling and binning could be used to improve accuracy if needed). The blurring and lens operators working on the source model define the final data-model.

- We determined a penalty function that tells us how well the data-model fits the observed data-set (i.e. χ^2) and penalizes source models that are too irregular, using a quadratic regularization term added to the penalty function based on either a gradient or curvature minimization of the source structure. This additional term has a weight λ, which is a nuisance parameter, that can not a-priori be determined (but see below when the data comes in!).
- The solution of the source model that minimizes the penalty function, for a given potential, was found to be a linear algebraic equation (only if the regularization term is quadratic) that can easily be solved.
- The penalty function was subsequently minimized with respect the lens-potential parameters (note that this is equivalent to a line-search method in the multi-dimensional space of $(\vec{s}, \vec{p}\,)$), resulting in the ML solution for the source solution and lens-potential parameters (this will be discussed in more detail below).

Systematics and a-priori assumptions:
A major issue that has *not* been yet addressed, however, is how to choose the value of λ, the number of source grid-points, where to place them, etc. These seem to be choices that have to be made a-priori. When made, one simply minimizes the above penalty function and finds the "best" (i.e. ML=maximum likelihood or MP=maximum posterior) solution of the model parameters. However, what if we make a different choice for λ, a smaller or larger number of source pixels, place them at different positions, choose a different PSF or regularization model, or a different parametric lens-potential family model, etc.? How can we compare the outcome in that case and which model is preferred? Obviously one can not simply compare the best ML or MP values, because one can simply add more and more free parameters to the model (e.g. the source or potential) and the ML/P value will generally keep decreasing. Hence a different method is required to properly rank different model assumptions/families. This is what we will now do in the next section, based on Bayesian statistics.

4 Bayesian ranking of model assumptions

Before we go on we need to go back to the penalty function $P_{\chi^2}(\vec{s}_{\rm min}; \vec{p}\,)$ (i.e. the function value for the best source solution for a given $\vec{p}\,$). As we noticed before the penalty function is quadratic and closely resembles (i.e. is) the normal χ^2 penalty plus an additional quadratic term.

If we assume that the errors and quadratic regularization penalty function are Gaussian distributed[4], we can write a likelihood function:

$$\mathcal{P} \propto e^{-\frac{1}{2}(\chi^2 + \lambda \vec{s}^{\,T} \boldsymbol{H}^T \boldsymbol{H} \vec{s})} = \overbrace{e^{-\frac{1}{2}\chi^2}}^{\text{Likelihood}} \times \overbrace{e^{-\frac{1}{2}\lambda \vec{s}^{\,T} \boldsymbol{H}^T \boldsymbol{H} \vec{s}}}^{\text{Regularisation Prior}} ,$$

where we note that $\partial \mathcal{P}/\partial \vec{s} = \vec{0}$ leads to $\partial P_{\chi^2}/\partial \vec{s} = \vec{0}$. The first term is the likelihood function (say $\mathcal{L}$), whereas the second term is the prior on the source brightness distribution.

This result can then be associated with the Bayesian probability (note that more priors can be combined in to a single prior)

$$\overbrace{P(\vec{s}\,|\,\vec{d}\,, \lambda, \boldsymbol{L}(\vec{p}\,), \boldsymbol{H})}^{\text{Posterior}} = \frac{\overbrace{P(\vec{d}\,|\,\vec{s}\,, \boldsymbol{L}(\vec{p}\,))}^{\text{Likelihood}} \times \overbrace{P(\vec{s}\,|\,\lambda, \boldsymbol{H})}^{\text{Prior}}}{\underbrace{P(\vec{d}\,|\,\lambda, \boldsymbol{L}(\vec{p}\,), \boldsymbol{H})}_{\text{Evidence}}}$$

Definition 3. *The "evidence" is the average value of the likelihood over the entire model parameter space that was allowed before the data came in.*

4. In fact the errors do not need to be Gaussian distributed. When the model is constrained well, the local minimum of P_{χ^2} can often be well approximated by a quadratics function, hence a Gaussian likelihood function (this is the result of the central limit theorem). The Gaussian distribution for the regularization is also relative arbitrary, but could in principle be determined from real data from the parent population of the source that is being lensed.

What does this definition really mean? To understand this in a bit more detail let's have a look at the following equation

$$\overbrace{P(\vec{d}\,|\,\lambda,\boldsymbol{L}(\vec{p}),\boldsymbol{H})}^{\text{Evidence}} = \int \underbrace{\overbrace{P(\vec{d}\,|\,\vec{s}\,,\boldsymbol{L}(\vec{p}))}^{\text{Likelihood}} \times \overbrace{P(\vec{s}\,|\,\lambda,\boldsymbol{H})}^{\text{Prior}}}_{\text{Posterior}} d\vec{s}$$

From this is is clear that the evidence is the integral over parameters space $\vec{s}$ and therefore represent the weighted (by the prior) average of the likelihood.

Note, as illustration, that we can roughly approximate this integral by

$$P(\vec{d}\,|\,\lambda,\boldsymbol{L}(\vec{p}),H) \approx P(\vec{d}\,|\,\vec{s}_{\text{ML}},\boldsymbol{L}(\vec{p})) \times P(\vec{s}_{\text{ML}}|\,\lambda,\boldsymbol{H}) \times \sigma_{\vec{s}\,|\vec{d}}$$

where $\sigma_{\vec{s}\,|\vec{d}}$ is approximately the width of the posterior PDF around the best solution $\vec{s}_{\text{ML}}$, i.e. the parameters space allows *after* the data came in. If $P(\vec{s}\,|\,\lambda,H)$, i.e. the prior, is relatively flat, we can approximate it by $\sim 1/\sigma_{\vec{s}}$, where $\sigma_{\vec{s}}$ is the parameter space allows *before* the data came in. In other words:

$$\overbrace{P(\vec{d}\,|\,\lambda,\boldsymbol{L}(\vec{p}),\boldsymbol{H})}^{\text{Evidence}} \approx \overbrace{P(\vec{d}\,|\,\vec{s}_{\text{ML}},\boldsymbol{L}(\vec{p}))}^{\text{Likelihood}} \times \overbrace{\left(\frac{\sigma_{\vec{s}\,|\vec{d}}}{\sigma_{\vec{s}}}\right)}^{\text{Occam's ratio}} .$$

This means that the evidence is, roughly, the ML solution times a factor that tells us how much the allowed model parameters space was reduced from before the data came in to after the data came in.

Hence a model that is overly complex and can describe a wide range of potential $\vec{s}$ (i.e. the model is not very predictive), will have a large value of $\sigma_{\vec{s}}$, and as such is penalized in the evidence. In that sense one can make the analogy with the χ^2/NDF, where models with too many free parameters (compared to data points) will have unlikely small values of their reduced χ^2 values. In that case, however, degeneracies between parameters are never accounted for, which will be the case in the Bayesian approach.

Can the value of λ be determined through posterior maximization? No! Because for any $\lambda > 0$ the minimum value of the posterior will always be larger than the minimum value of the likelihood. Hence varying λ to minimize the posterior will always lead to $\lambda = 0$. What about maximizing the evidence? Suppose that the value of λ is very small (hence nearly no regularization). In that case, the prior space of allowed solutions of $\vec{s}$ is very broad and thus $\sigma_{\vec{s}}$ will be very large (i.e. the regularization prior is very broad). This tends to lower the evidence since very irregular (i.e. non-smooth) models are allowed and probably fit the data best, which is exactly what the prior is supposed to prevent. By increasing the value of λ, the allowed space of prior models $\sigma_{\vec{s}}$ is reduced, smoother models are found, and the evidence tends to increase (as long as the likelihood is not too much affected by forcing more smooth models). If, however, λ becomes too large (very smooth models), even though $1/\sigma_{\vec{s}}$ increases, the smooth model can no longer fit the data and the likelihood of the model starts to decrease. Hence a balance needs to be found between fitting the data (i.e. maximizing the likelihood) and finding the smoothest model (i.e. maximizing the prior). Hence maximizing the evidence as described above will lead to a non-zero value for λ. We will formalize this further in the next section, including an additional prior on λ itself (which does not change the general explanation above).

4.1 Obtaining the regularization parameter through evidence maximization

We can assume that λ represents a model assumption and as such can be ranked with the evidence. Continuing the Bayesian description, we find:

$$P(\lambda|\,\vec{d}\,,\boldsymbol{L}(\vec{p}),\boldsymbol{H}) = \frac{P(\vec{d}\,|\,\lambda,\boldsymbol{L}(\vec{p}),\boldsymbol{H}) \times P(\lambda)}{P(\vec{d}\,|\,\boldsymbol{L}(\vec{p}),\boldsymbol{H})}.$$

We note here that the evidence in the posterior for $\vec{s}$ is equivalent the likelihood for λ (assuming we do not integrate over all the other assumptions)! To assess the best value of λ we therefore have to first determine the evidence for $\vec{s}$. With a flat prior on $\log(\lambda)$ [flat in log space because the scale is not know; this choice of prior can and has been be debated, but is beyond the scope of this lecture], we can then obtain the MP=ML value of λ. Note however that this can be done for any other model assumption as well (e.g. $\boldsymbol{L}$ or $\boldsymbol{H}$). Here we simply concentrated on obtaining the ML value for λ.

One can continue this one step further to assess $P(\vec{d}\,|\boldsymbol{L}(\vec{p}),\ \boldsymbol{H})$ which is the evidence for the models $\{\boldsymbol{L}(\vec{p}),\boldsymbol{H}\}$, and can be used to rank different assumptions for $\boldsymbol{H}$ or the lens models $\boldsymbol{L}(\vec{p})$ (hence also the lens-potential parameters).

Working this out under the approximation of Gaussian errors on the data, Gaussian priors on $\vec{s}$ and a flat prior on λ results in:

$$\begin{aligned}\log\left(P(\vec{d}\,|\,\lambda,\boldsymbol{M}\equiv\boldsymbol{B}\boldsymbol{L},\boldsymbol{H})\right) &= -\frac{1}{2}[\chi^2+\lambda\parallel\boldsymbol{H}\,\vec{s}\parallel^2]-\frac{1}{2}\log\left[\det\left(\boldsymbol{M}^T\boldsymbol{C}^{-1}\boldsymbol{M}+\lambda\,\boldsymbol{H}^T\boldsymbol{H}\right)\right]+\frac{N_s}{2}\log(\lambda)\\ &\quad+\frac{1}{2}\log[\det\left(\boldsymbol{H}^T\boldsymbol{H}\right)]-\frac{N_d}{2}\log(2\pi)+\frac{1}{2}\log[\det\left(\boldsymbol{C}^{-1}\right)].\end{aligned}$$

Where the definition of χ^2 was given above. The MP value of λ can be found by maximizing the above equation. To marginalize over λ, in fact it turns out that putting $\lambda_{\rm MP}$ back into the above equation gives a very good approximation of $P(\vec{d}\,|\,L(\vec{p}),H)$, the evidence for the model. The latter evidence can be used to rank different potential models, model families, PSFs, pixel-scales, regularization models, etc. Of course this does not tell us which models to choose, which still requires insight of the scientist based on physics, but at least it allows them to be ranked objectively.

5 Including the lens potential in the linear equations

This is more tricky, since the lens-potential enters in to $L(\psi)$ in a non-linear manner. We can therefore not simply use the same techniques as above to solve for the potential itself. However, we can *linearize* the equations and obtain a small linear correction $\delta\psi$ to a starting potential, and from there on iteratively improve upon this initial potential model.

To obtain a linearized equivalent equation to what we found for the source, we again write

$$s(\vec{y})=d(\vec{x})\ \text{ with }\ \vec{y}=\vec{x}-\vec{\nabla}\psi(\vec{x}),$$

which is exactly true (in absence of noise), if we know the true source and potential model. Let us now assume that we know the source, but only a reasonable first guess for the potential. In that case

$$\begin{aligned}s(\vec{x}-\vec{\nabla}[\psi(\vec{x})+\delta\psi(\vec{x})])&=d(\vec{x})\\ s(\vec{x}-\vec{\nabla}\psi(\vec{x}))&=d(\vec{x})+\delta d(\vec{x}).\end{aligned}$$

From which we readily obtain, through Taylor expanding the above equation:

$$\delta d(\vec{x})=-\frac{\partial s(\vec{y})}{\partial\vec{y}}\cdot\frac{\partial\delta\psi(\vec{x})}{\partial\vec{x}}=-\vec{\nabla}_y s(\vec{y})\cdot\vec{\nabla}_x\delta\psi(\vec{x}).$$

What does this equation mean? First, we note that $\delta\vec{y}=\delta\vec{\nabla}_x\psi(\vec{x})=\vec{\nabla}_x\delta\psi(\vec{x})$, which means the last terms is simply a linear shift in the source plane due to the change in the potential. The first term in a linear expansion (i.e. the gradient) around the source surface brightness around point $\vec{y}$. Hence $\vec{\nabla}_y s(\vec{y})\cdot\vec{\nabla}_x\delta\psi(\vec{x})=\frac{\partial s(\vec{y})}{\partial\vec{y}}\cdot\delta\vec{y}\approx\delta s(\vec{y})$. The minus sign is there because of the positive definition of $\delta d(\vec{x})$ in the above equations (or similarly positive change in the potential). Hence the change in the image plane corresponds to a similar change in the source plane. Our aim is therefore to find that $\delta\psi(\vec{x})$ that leads to a complete cancellation of $\delta d(\vec{x})$, when applied to $\psi_{\rm init}(\vec{x})$.

Since the above equation is again linear in $\delta\psi$, it can be written as

$$\delta\vec{s} = -\boldsymbol{D}_s(\vec{s}\,)\,\boldsymbol{D}_x\,\delta\vec{\psi}\,,$$

where the first matrix contains the derivatives of the previous best model of $\vec{s}$ and the second is the gradient operator on $\delta\vec{\psi}$.

Exercise 5. Derive the structure of the matrices $\boldsymbol{D}_s(\vec{s}\,)$ and $\boldsymbol{D}_x$ and show that when applied to $\delta\vec{\psi}$ (similarly structured as $\vec{\psi}$) it in fact gives the correct correction $\delta\vec{s}$.

Combining the above linear correction with the equation found before for the source, we find:

$$\overbrace{\boldsymbol{B}[\boldsymbol{L}(\boldsymbol{\psi})\,|\,-\boldsymbol{D_s}(\vec{\boldsymbol{s}}\,)\,\boldsymbol{D_x}]}^{\equiv\boldsymbol{K}\,\text{(block matrix)}}\overbrace{\begin{pmatrix}\vec{s}\\ \delta\vec{\psi}\end{pmatrix}}^{\equiv\vec{r}} = \boldsymbol{K}\,\vec{r} = \vec{d}\,'$$

This equation is linear-algebraic equivalent to $d(\vec{x}\,) = \mathrm{PSF}\star s(\vec{x}\,-\vec{\nabla}[\psi_{\mathrm{init}}(\vec{x}\,)+\delta\psi(\vec{x}\,))]$, including the PSF smearing through convolution.

Of course we can solve this equation in a fully equivalent way as before, leading to

$$(\boldsymbol{K}^T\boldsymbol{C}^{-1}\boldsymbol{K}+\boldsymbol{R}^T\boldsymbol{R})\,\vec{r} = \boldsymbol{K}^T\boldsymbol{C}^{-1}\vec{d}\,,$$

with

$$\boldsymbol{R}^T\boldsymbol{R}\equiv\begin{pmatrix}\lambda_s\boldsymbol{H}_s^T\boldsymbol{H}_s & \mathbf{0}\\ \mathbf{0} & \lambda_{\delta\psi}\boldsymbol{H}_{\delta\psi}^T\boldsymbol{H}_{\delta\psi}\end{pmatrix},$$

the latter being a block matrix that regularizes the solutions. Note that the regularization of the source and potential solutions are independent (i.e. diagonal), whereas the terms of $\vec{s}$ and $\delta\vec{\psi}$ in $\boldsymbol{K}^T\boldsymbol{C}^{-1}\boldsymbol{K}$ are fully mixed, hence the two solutions "compete" in their attempt to fit the data $\vec{d}$ best.

Hence given a good starting model for the potential (e.g. obtained by fitting a simple parametric potential model as discussed earlier in this lecture), one can obtain a best guess for the source. Given these two models, one can then determine an improved model of the source and a correction to the lens potential. That correction is added to the previous potential model and the whole process is repeated until convergence. We note here that it is not clear (yet) what the correct meaning is of the convergence solution.

5.1 Error estimates on the source and potential models

To understand this a bit better, however, let us assume we have converged *and* we have indeed found the true source and potential model that minimizes the penalty function $P_{\chi^2}(\vec{s}\,,\,\vec{p}\,=\vec{\psi}\,)$ for both source and potential model. We can then expand the equations as follows

$$\boldsymbol{B}[\boldsymbol{L}(\vec{\psi}\,)\,|\overbrace{-\boldsymbol{D}_s(\vec{s}\,)\,\boldsymbol{D}_x}^{\equiv\boldsymbol{L}_{\delta\psi}}]\begin{pmatrix}\delta\vec{s}\\ \delta\vec{\psi}\end{pmatrix} = \vec{d}\,-\boldsymbol{B}\boldsymbol{L}(\vec{\psi}\,)\vec{s}\,=\delta\vec{d}\,.$$

One then finds that the (approximate) Hessian of the penalty function becomes:

$$\boldsymbol{\mathcal{H}}\approx\begin{pmatrix}[\boldsymbol{BL}]^T\boldsymbol{C}^{-1}[\boldsymbol{BL}] & [\boldsymbol{BL}]^T\boldsymbol{C}^{-1}[\mathbf{BL}_{\boldsymbol{\delta\psi}}]\\ [\mathbf{BL}_{\boldsymbol{\delta\psi}}]^T\boldsymbol{C}^{-1}[\boldsymbol{BL}] & [\mathbf{BL}_{\boldsymbol{\delta\psi}}]^T\boldsymbol{C}^{-1}[\mathbf{BL}_{\boldsymbol{\delta\psi}}]\end{pmatrix}+\begin{pmatrix}\lambda_s\boldsymbol{H}_s^T\boldsymbol{H}_s & \mathbf{0}\\ \mathbf{0} & \lambda_{\delta\psi}\boldsymbol{H}_{\delta\psi}^T\boldsymbol{H}_{\delta\psi}\end{pmatrix},$$

which can be further simplified as

$$\boldsymbol{\mathcal{H}}\approx\begin{pmatrix}\boldsymbol{L}^T\mathbf{AL} & \boldsymbol{L}^T\boldsymbol{A}\boldsymbol{L}_{\boldsymbol{\delta\psi}}\\ \boldsymbol{L}_{\boldsymbol{\delta\psi}}{}^T\mathbf{AL} & \boldsymbol{L}_{\boldsymbol{\delta\psi}}{}^T\boldsymbol{A}\boldsymbol{L}_{\boldsymbol{\delta\psi}}\end{pmatrix}+\begin{pmatrix}\lambda_s\boldsymbol{H}_s^T\boldsymbol{H}_s & \mathbf{0}\\ \mathbf{0} & \lambda_{\delta\psi}\boldsymbol{H}_{\delta\psi}^T\boldsymbol{H}_{\delta\psi}\end{pmatrix},$$

with $\boldsymbol{A} \equiv \boldsymbol{B^T C^{-1} B}$. The above Hessian follows from the fact that the deviation from the minimum penalty function value can be approximated by $\delta P \approx \delta\vec{d}^{\,T}\boldsymbol{C}^{-1}\delta\vec{d} \; + \lambda_s \delta\vec{s}^{\,T}\boldsymbol{H}_s^T\boldsymbol{H}_s\delta\vec{s} + \lambda_{\delta\psi}\delta\vec{\psi}\boldsymbol{H}_{\delta\psi}^T\boldsymbol{H}_{\delta\psi}\delta\vec{\psi} = \delta\vec{r}^{\,T}\boldsymbol{\mathcal{H}}\,\delta\vec{r}$, since the gradients of the penalty function in the minimum are zero (by definition). We note that if no mixing would occur in the linearized equation between the source and potential, there would be no off-diagonal terms. From this Hessian, an approximate error can be determined for both $\vec{s}$ and $\vec{\psi}$ in the usual manner from the approximation

$$\mathcal{P}(\delta\vec{r}\,) \approx \mathcal{P}_{\min}\, e^{-\frac{1}{2}\delta\vec{r}^{\,T}\boldsymbol{\mathcal{H}}\,\delta\vec{r}}$$

with $\delta\vec{r} = \vec{r} - \vec{r}_{\min}$. Once the best solution has been found, that maximizes the posterior, one can use the above approximation of the posterior in the Bayesian estimate of the regularization parameters λ by integrating over the posterior space of $\delta\vec{r}$ (remember that this gives the likelihood of λ) and then maximize over the posterior of λ, after having multiplied its likelihood by a prior (flat in $\log(\lambda)$; see above). If the data and priors constrain the models sufficiently well, then in general the above approximation (as a result of the central-limit theorem; i.e. the penalty function around the minimum is well-approximated by a quadratic function) provides a good approximation of the true error on the source and potential models.

6 Summary

In this lecture I have given a short introduction to non-parametric/grid-based gravitational-lens modeling techniques applied. The basic idea is to build a linear data-model of the data (i.e. the lensed images), where the source brightness distribution and potential depth values are free 'parameters' and need to be determined from the data. This is done by defining a quadratic penalty function that minimizes the residuals between the data and the data-model by varying the free parameters of the source and potential models. This was then lifted to a higher level through Bayesian statistics, which allows us to rank different models and marginalize over nuisance parameters (e.g. regularization). Based on these lectures, I hope you will have the basis to start building your own grid-based codes!

Some additional reading material and software

- Barnabè, M., Koopmans, L.V.E. 2007. A unifying framework for self-consistent gravitational lensing and stellar dynamics analyses of early-type galaxies. Astrophysical Journal 666, 726
- Koopmans, L.V.E. 2005. Gravitational imaging of cold dark matter substructures. Monthly Notices of the Royal Astronomical Society 363, 1136-1144
- MacKay, D. 1992, Bayesian Methods for Adaptive Models, PhD Thesis, http://www.inference.phy.cam.ac.uk/mackay/PhD.html
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. 1992. Numerical recipes in C. The art of scientific computing. Cambridge: University Press
- Saha, P., Williams, L.L.R. 1997. Non-parametric reconstruction of the galaxy lens in PG 1115+080. Monthly Notices of the Royal Astronomical Society 292, 148
- Suyu, S.H., Marshall, P.J., Hobson, M.P., Blandford, R.D. 2006. A Bayesian analysis of regularized source inversions in gravitational lensing. Monthly Notices of the Royal Astronomical Society 371, 983-998
- Vegetti, S., & Koopmans, L. 2009. Bayesian strong gravitational-lens modelling on adaptive grids: objective detection of mass substructure in Galaxies. Monthly Notices of the Royal Astronomical Society, 392, 945
- Warren, S.J., Dye, S. 2003. Semilinear Gravitational Lens Inversion. Astrophysical Journal 590, 673-682
- Wayth, R.B., Webster, R.L. 2006. LENSVIEW: software for modelling resolved gravitational lens images. Monthly Notices of the Royal Astronomical Society 372, 1187-1207