

# Gravitational lens modelling in a citizen science context

Rafael Küng,<sup>1</sup>★ Prasenjit Saha,<sup>1</sup> Anupreeta More,<sup>2</sup> Elisabeth Baeten,<sup>3</sup> Jonathan Coles,<sup>4</sup> Claude Cornen,<sup>3</sup> Christine Macmillan,<sup>3</sup> Phil Marshall,<sup>5</sup> Surhud More,<sup>2</sup> Jonas Odermatt,<sup>6</sup> Aprajita Verma<sup>7</sup> and Julianne K. Wilcox<sup>3</sup>

<sup>1</sup>Physik-Institut, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

<sup>2</sup>Kavli Institute for the Physics and Mathematics of the Universe, University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi 277-8583, Japan

<sup>3</sup>Zooniverse, c/o Astrophysics Department, University of Oxford, Oxford OX1 3RH, UK

<sup>4</sup>Exascale Research Computing Lab, Campus Teratec, 2 Rue de la Piquetterie, F-91680 Bruyeres-le-Chatel, France

<sup>5</sup>Kavli Institute for Particle Astrophysics and Cosmology, Stanford University, 452 Lomita Mall, Stanford, CA 94035, USA

<sup>6</sup>Kantonsschule Zug, Lüssiweg 24, CH-6300 Zug, Switzerland

<sup>7</sup>Sub-department of Astrophysics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK

Accepted 2014 November 30. Received 2014 November 5; in original form 2014 December 2

## ABSTRACT

We develop a method to enable collaborative modelling of gravitational lenses and lens candidates, that could be used by non-professional lens enthusiasts. It uses an existing free-form modelling program (GLASS), but enables the input to this code to be provided in a novel way, via a user-generated diagram that is essentially a sketch of an arrival-time surface. We report on an implementation of this method, SpaghettiLens, which has been tested in a modelling challenge using 29 simulated lenses drawn from a larger set created for the Space Warps citizen science strong lens search. We find that volunteers from this online community asserted the image parities and time ordering consistently in some lenses, but made errors in other lenses depending on the image morphology. While errors in image parity and time ordering lead to large errors in the mass distribution, the enclosed mass was found to be more robust: the model-derived Einstein radii found by the volunteers were consistent with those produced by one of the professional team, suggesting that given the appropriate tools, gravitational lens modelling is a data analysis activity that can be crowd-sourced to good effect. Ideas for improvement are discussed; these include (a) overcoming the tendency of the models to be shallower than the correct answer in test cases, leading to systematic overestimation of the Einstein radius by 10 per cent at present, and (b) detailed modelling of arcs.

**Key words:** gravitational lensing: strong – methods: numerical.

## 1 INTRODUCTION

The first work on lens modelling (Young et al. 1981a,b) was developed after the discovery of the first two gravitational lenses (Walsh, Carswell & Weymann 1979; Weymann et al. 1980), where a massive galaxy causes a background quasar to appear as two or four images. For the second lens to be discovered (PG1115+080), mass models scored an early success with the prediction that one of the lensed images seen would split further into two at higher resolution. That galaxies must sometimes cause multiple images had long been expected (Zwicky 1937), and it had even been argued that the phenomenon could help measure cosmological parameters (Refsdal 1964, 1966), but apparently nobody was expecting that lenses would need detailed modelling. The first observations, however,

immediately stimulated models. The reason for that lies in the image separation. Recall that image separations are of the order of the angular Einstein radius:

$$\Theta_E \sim \left( \frac{4GM}{c^2 D_L} \right)^{1/2} \simeq 0.1 \text{ arcsec} \left( \frac{M}{M_\odot} \right)^{1/2} \left( \frac{D_L}{\text{pc}} \right)^{-1/2}, \quad (1)$$

where  $D_L$  is the distance to the lens, and  $M$  its mass. A lensing galaxy with  $M \sim 10^{11} M_\odot$  at  $\sim 1$  Gpc would cause image separations of  $\sim 1$  arcsec, which is comparable to the size of the galaxy; typically the lensed images are seen through the galaxy halo. Hence, the lensed images depend on the detailed mass distribution of the lensing galaxy. Galaxy lenses therefore require models of their mass distributions.

Since those early discoveries, more than 400 secure lenses are now known. Modelling of the mass distribution is part of any research using lenses, but so far no modelling study has spanned all known lenses. The largest single one (Koopmans et al. 2009)

\* E-mail: rafael.kueng@uzh.ch

models 58 separate lenses to infer the distribution of dark matter around galaxies. In other work, Leier et al. (2011) combined lens models of 21 galaxies with models of their stellar populations, to find the relation between stars and dark matter, and Sereno & Paraficz (2014) modelled 18 time-delay lenses together to infer cosmological parameters.

Imaging surveys now under way aim to increase the inventory of lenses another ten or a hundred fold (see e.g. Marshall et al. 2005; Oguri & Marshall 2010), with both automated and visual search techniques proposed (e.g. Marshall et al. 2009; More et al. 2012; Gavazzi et al. 2014). For example, Space Warps (Marshall et al., in preparation; More et al., in preparation) is a citizen science project<sup>1</sup> in which volunteers are presented sky-survey images and are invited to identify lens candidates, by eye. Simulated lenses are mixed in with the data, both to help train volunteers on what to look for, and to provide measures of the effectiveness of the search. The motivation for Space Warps is to enable volunteers, some of whom had previously serendipitously identified lens candidates on earlier citizen science surveys, either to make discoveries missed in automatic searches by software robots, or to perform the necessary inspection of an automatically generated sample, for quality control. Robots can be built to be good at detecting lensing system in clean, uncrowded fields with high signal-to-noise ratio, but in more general test situations, robots miss lenses (low completeness) or contaminate the results with non-lenses (low purity; Marshall et al. 2009).

The encouraging early results from the first Space Warps lens search, carried out on the  $\simeq 172$  deg<sup>2</sup> Canada–France–Hawaii Telescope Legacy Survey (CFHTLS) imaging by over 30 000 volunteers (Marshall et al., in preparation; More et al., in preparation) prompted the question: could the modelling of the lenses also be done by the volunteers? If so, modelling could help prioritize lens candidates at an early stage, which would be very useful with new wide-field and sensitive surveys, which will yield thousands of lens candidates. There are several software tools for lens modelling available, and work has been done on generic interfaces (e.g. Lefor 2014). Some early designs for Space Warps included a prototype lens-modelling tool (Naudus, Wallin & Marshall 2010). Moreover, some Space Warps volunteers are quite experienced from earlier projects, having individually spent a thousand hours or more with data, and are very interested in more demanding projects. The interests of citizen science communities are just beginning to be studied (e.g. Raddick et al. 2013), but it is clear that some volunteers welcome open-ended challenges, and sometimes these have led to new scientific results: one example is the discovery of an exceptional extrasolar planet (Schwamb et al. 2013); another is the development of new algorithms for protein folding (Khatib et al. 2011). All these are grounds for optimism. There is, however, a basic difficulty in strong gravitational lensing. Lensed images do not look much like their source, and still less do they resemble the lensing-mass distribution. To model a lens, one needs either to do a lot of random guessing, or to have a good intuition for what works.

In this paper, we propose a way around the difficulty, and report on a modelling test on Space Warps using simulated lenses. The three following sections are devoted to the concept, the implementation, and tests, respectively.

In Section 2, we introduce a markup system for lensed images, which we call a ‘spaghetti diagram’. A spaghetti diagram resembles the visible image system, in a cartoon-like way, and at the

same time it encodes the basis of a mass model. This supplies an intuitive link between the image system and the mass distribution, which look frustratingly different from each other. Spaghetti diagrams are essentially the saddle-point contours originally introduced to gravitational lensing by Blandford & Narayan (1986) as a way of classifying lensed images. They are sometimes shown as part of the output of lens models (for example Rusin et al. 2001; Keeton & Winn 2003; Lubini & Coles 2012). In this work, however, spaghetti diagrams are the *input* through which the modeller tells the SpaghettiLens program what to do.

In Section 3, we describe the SpaghettiLens program, which implements the above scheme. SpaghettiLens is an interface to and extension of the GLASS framework for modelling lenses (Coles, Read & Saha 2014). We will not go into software details in this paper, instead concentrate on lens modelling per se, but we remark that SpaghettiLens is designed to be friendly to the forum style of citizen science projects, and enables incremental collaborative model refinement by different people, without sacrificing any of the technical features of GLASS.

In Section 4, we describe a modelling challenge where a diverse sample of 29 simulated lenses was modelled multiple times by a small number of Space Warps volunteers using SpaghettiLens. The models were then examined in two ways. One was whether the spaghetti diagram was correct. The other was the recovery of the Einstein radius of the lens. In addition, we show some visual comparisons of the actual and recovered lens shape and radial profile, and identify some areas to improve. Profile and shape recovery with GLASS has been studied in more detail in Coles et al. (2014).

Section 5 gives the general outlook and next steps.

## 2 FERMAT’S PRINCIPLE AND SPAGHETTI DIAGRAMS

We first explain the lensing theory relevant to SpaghettiLens, following the formulation of gravitational lensing in terms of Fermat’s principle by Blandford & Narayan (1986).

### 2.1 Geometrical and gravitational time delays

Consider a lens at some redshift  $z_L$  and let  $(x, y)$  be planar coordinates at the lens, transverse to the line of sight. Let  $\Sigma(x, y)$  be the mass distribution. It is a mass per unit area, i.e. density projected along the line of sight. The mass distribution is often given in a dimensionless form

$$\kappa(x, y) \propto \Sigma(x, y) \quad (2)$$

called the convergence. Let there be light, in the form of a more distant source, at redshift  $z_S$ , behind point  $(x_s, y_s)$  on the lens.

We now imagine a virtual photon flying from the source to some  $(x, y)$  on the lens, then changing direction and coming to the observer. Such a direction change would increase the light traveltime compared to coming through  $(x_s, y_s)$ . The increased light traveltime from the geometry of deflection would be

$$t_{\text{geom}}(x, y) \propto (x - x_s)^2 + (y - y_s)^2, \quad (3)$$

assuming the delay is small compared to the total light traveltime.

An additional delay of the photon comes from travelling through the curved space-time at the lens. This gravitational time delay  $t_{\text{grav}}$  is related to the mass distribution of the lens. The relation is generally written as a two-dimensional Poisson equation, but an alternative expression, avoiding calculus, is as follows. The value of  $t_{\text{grav}}$  through  $(x, y)$  equals its average value on the circumference

<sup>1</sup> <http://www.spacewarps.org>

of a small circle centred at  $(x, y)$ , plus a constant times the mass within that circle. The constant is  $2G/c^3$  times the cosmological expansion factor  $(1 + z_L)$ . Thus,

$$t_{\text{grav}}(x, y) = \langle t_{\text{grav}}(x_{\circ}, y_{\circ}) \rangle + (1 + z_L) \frac{2G}{c^3} M(x_{\bullet}, y_{\bullet}). \quad (4)$$

We have used  $(x_{\circ}, y_{\circ})$  to denote the circumference of a circle, and  $(x_{\bullet}, y_{\bullet})$  to indicate the integrated mass within the circle. Appendix A relates this expression to the better-known explicit form for the gravitational time delay.

The light traveltime of a virtual photon is therefore longer by

$$t(x, y) = t_{\text{geom}} + t_{\text{grav}} \quad (5)$$

than it would have been with no lens present. Real photons take paths that make  $t(x, y)$  extremal, that is, having a minimum, maximum or saddle point (Fermat's principle).

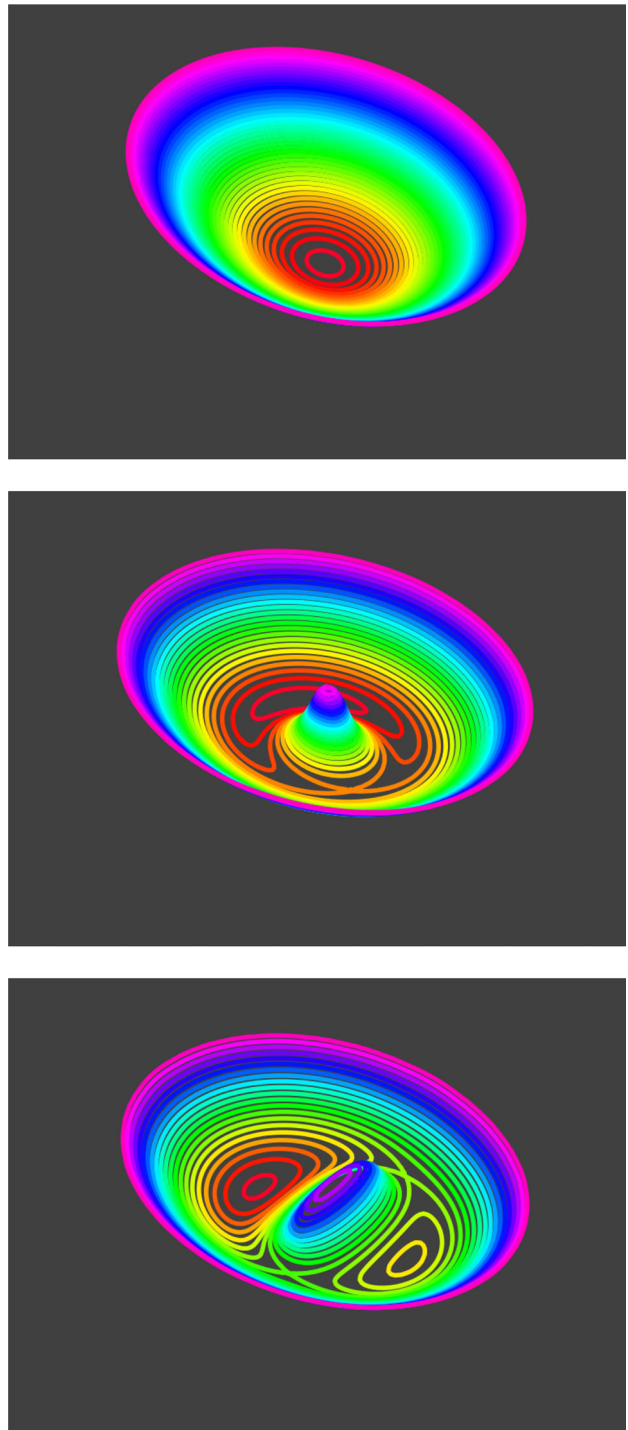
The proportionality factors in equations (2) and (3) depend on the redshifts and cosmological parameters, and are given in Appendix A.

## 2.2 Arrival-time contours

The full function  $t(x, y)$ , also known as the arrival-time surface, applies to virtual photons. In other words, it is an abstract construct and not itself observable. But observable image positions can be derived from the arrival-time surface, so visualizing the surface is useful. Fig. 1 does so. In this figure, a maximum, if present, is easy to see. To locate minima and saddle points, however, one needs to examine the contours of equal arrival time. A saddle point is characterized by a contour crossing itself, forming an X. Minima, on the other hand, have contours looping around them, as do maxima.

The saddle-point contours which form an X are especially interesting, because they set the overall topography of the arrival-time surface. They obviously give the locations of the saddle points, and roughly localize the minima and maxima as well. If more precise locations of the minima and maxima are added, the whole arrival-time surface is already approximately known. Since the arrival-time surface has an exact relation to the lens-mass distribution and the source position, in effect the mass distribution is also automatically approximately specified. In other words, a simple sketch of saddle-point contours along with locations of minima and maxima – which we call a ‘spaghetti diagram’ – is implicitly already an approximation to a lens-mass distribution.

The preceding assumes a point source. To get an idea of what an extended source would do, let us imagine moving the original source slightly. The contours of constant arrival time will naturally move slightly, and so will the images. The movement of the contours will be most noticeable where the contours are far apart, that is where the arrival-time surface is nearly flat. As is evident from Fig. 1, this is the region where the minimum and saddle points lie, or near the images. In this region, points on the source that are close together produce images that are comparatively far apart. In other words, the image is highly magnified. In summary, lower curvature in the arrival-time surface for a point source implies larger magnification of an extended source. Conversely, where the arrival-time surface is strongly curved, the image will be demagnified. We see from Fig. 1 that the arrival-time surface tends to be highly curved near the maximum. Hence, maxima tend to be demagnified. In practice, maxima of the arrival time are nearly always too faint to see. The minima and saddle points dominate.



**Figure 1.** Perspective views and contour maps of example arrival-time surfaces. Contours are coloured in rainbow order (red: least delay, violet: highest delay). The special contours that self-cross at saddle points are the basis of spaghetti diagrams. Upper panel: no lens, hence showing the parabolic shape of the geometrical time delay. The image would be at the bottom, coinciding with the source. Middle panel: a circular lensing mass (offset from the source) has been added, which has pushed the minimum to one side and introduced a maximum and saddle point, each corresponding to an image. The saddle point is characterized by a self-crossing ‘spaghetti’ contour. Lower panel: an elongated lensing mass has been added. There are now two minima, two saddle points, and a maximum, each corresponding to an image.

### 3 A LENS-MODELLING PROGRAM

SpaghettiLens is a mass modelling program that makes use of the Space Warps infrastructure, in particular, the image data base and the discussion forum.<sup>2</sup> The forum is essential for establishing contact between interested members of the Space Warps community and the project science team, and then for enabling collaboration between them. We were able to collaborate together on modelling objects from Space Warps in the usual style of medium-sized astronomical collaborations, with video-conferencing and in-person meetings where possible. Preliminary results were immediately summarized on modelling threads on the forum, and anyone interested was made welcome to join at any time.

Modelling with SpaghettiLens involves three stages, (1) markup of the image, followed by (2) intensive numerical computation carried out on a server in the background, followed by (3) review of diagnostics and possible discussion. Human interaction is essential to the first and third stages, while stage 2 is completely automated. We now describe the three stages.

#### 3.1 Image markup

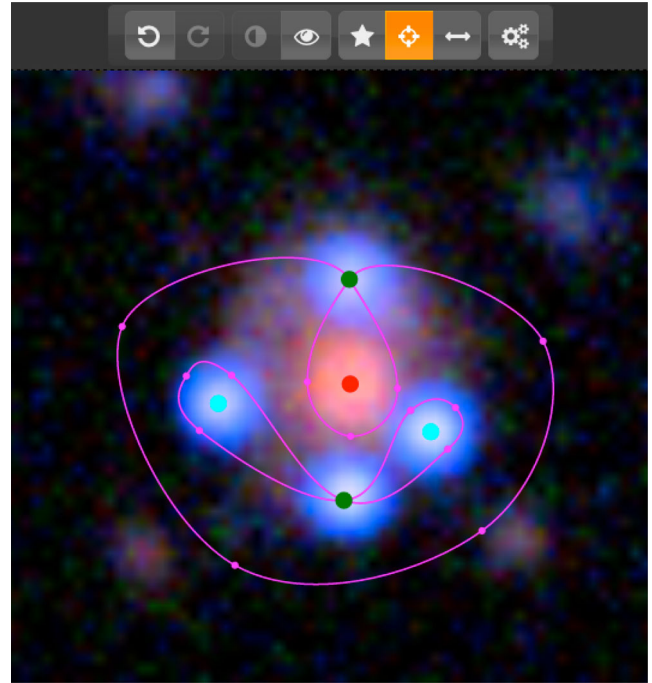
One begins by going to the SpaghettiLens web application<sup>3</sup> and entering the number of a Space Warps image tile. SpaghettiLens then presents the image, along with zoom and pan options and a markup tool to construct a spaghetti diagram. The human modeller now has to make an educated guess for the topography of the arrival-time surface, and input the locations time-ordering of the maxima, minima, and saddle points. The markup tool (which is inspired by fig. 6 of Blandford & Narayan 1986, and is like that figure made interactive and overlaid on data) lets the modeller enter the information by sketching saddle-point contours. Examples can be seen in Fig. 2 and the upper-left panels of Figs 3–10. The loops in the markup tool were the origin of the ‘spaghetti’ metaphor.

The markup tool allows only valid lensing configurations to be entered. The user does not need to think explicitly about the image parities (though the markup tool provides this information using colour codes) or about time-ordering, or worry about the odd-image theorem. The exact placement of the loops in a spaghetti diagram has no significance. Only the hierarchy of which loop is inside which is relevant. The loops are there simply to help modeller’s intuition.

As implemented so far, SpaghettiLens assumes that the lens is dominated by a single galaxy. Accordingly, only one maximum in the arrival-time surface is permitted, and it is taken to be the centre of the main lensing galaxy. The user can, however, mark additional minor galaxies: these are modelled as point masses, the mass being fitted by the program along with the rest of the mass distribution.

#### 3.2 Numerics

Having sketched a spaghetti diagram, the user presses a button to initiate the next stage. SpaghettiLens then translates the spaghetti diagram into input for GLASS, and forwards this input. The task of GLASS, which runs server-side as it is compute-intensive, is to find a mass distribution  $\kappa(x, y)$  that exactly reproduces the given locations of the maximum, minima, and saddle points. This criterion by itself is extremely underdetermined – there are infinitely many mass distributions that will reproduce a given set of maxima, minima



**Figure 2.** Screen grab of SpaghettiLens in action. A Space Warps image has been loaded in, re-centred and zoomed. Five images and the associated ‘spaghetti’ contours have then been suggested, using the marking tools associated with the buttons along the top of the panel. The mass model is generated server-side when the rightmost button is pressed.

and saddle points, but typically they (a) produce lots of extra images, and (b) look very unlike galaxies. Additional assumptions (a priori) are necessary. GLASS uses the following priors (cf. Saha & Williams 1997; Coles 2008).

(i) The mass distribution is built out of non-negative tiles of mass. (Sometimes these tiles are called mass pixels, but we should emphasize that they are unrelated to image pixels, and are much larger.)

(ii) There is a notional lens centre, say  $(x_0, y_0)$  which is identified with the maximum of the arrival time. The source can have an arbitrary offset with respect to the lens centre.

(iii) The mass distribution must be centrally concentrated, in two respects. First, the circularly averaged density must fall away like

$$[(x - x_0)^2 + (y - y_0)^2]^{-1/2}$$

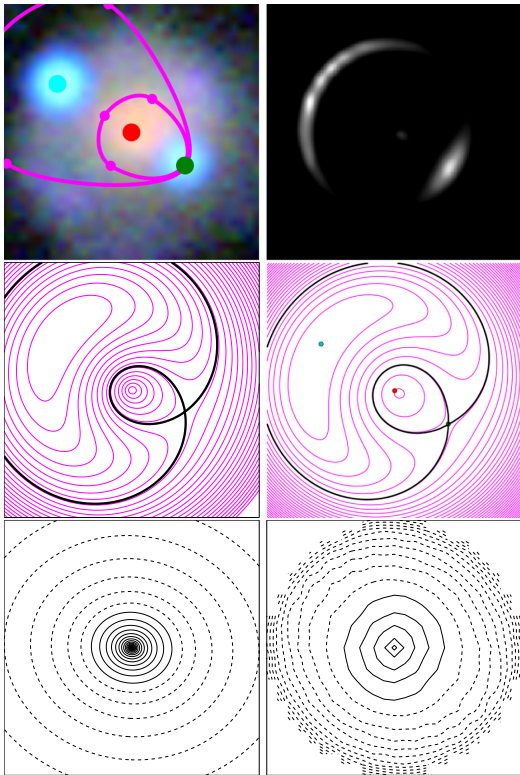
or more steeply. Secondly, the direction of increasing density at any  $(x, y)$  can point at most  $45^\circ$  away from  $(x_0, y_0)$ .

(iv) The lens must be symmetrical with respect to  $180^\circ$  rotations about  $(x_0, y_0)$ . This symmetry assumption can be relaxed if the user wishes.

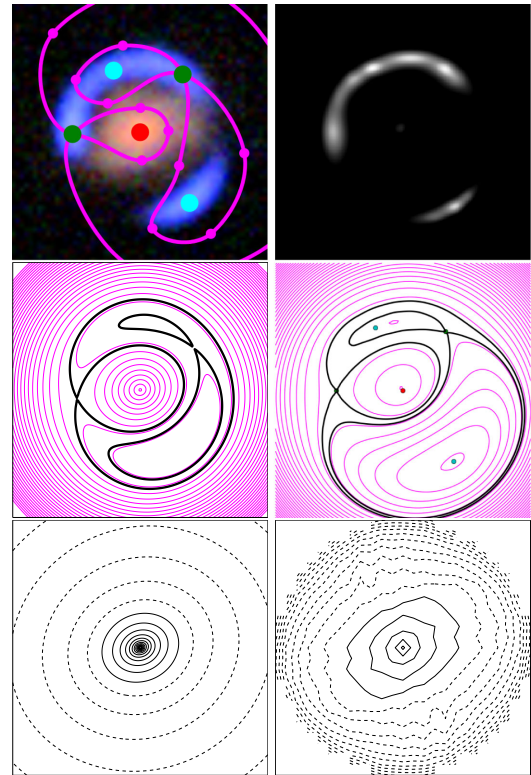
There are still infinitely many models that satisfy both data and prior constraints, but now they are more credible as galaxy lenses. It is then possible to generate an ensemble of models. The sampling technique used by GLASS is described in Lubini & Coles (2012). Typically, ensembles of 200 models are used. That is to say, what we call a SpaghettiLens model is really the mean of an ensemble of 200 models, and its estimated uncertainty is the range covered by the whole ensemble.

<sup>2</sup> <http://talk.spacewarps.org>

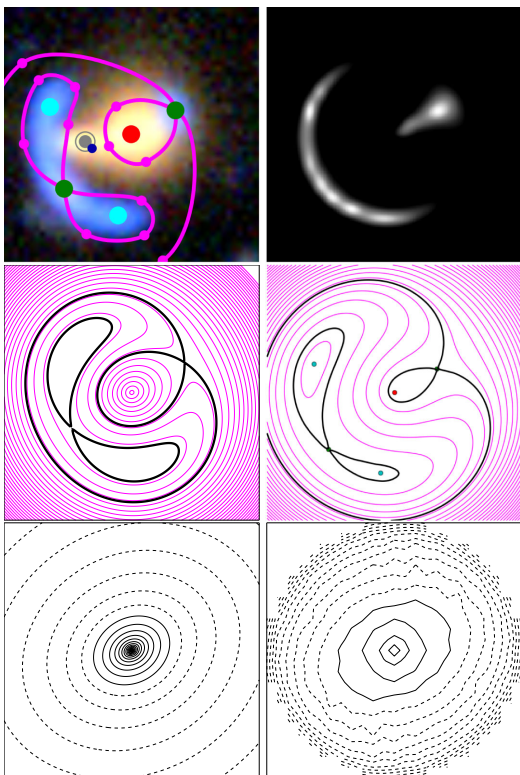
<sup>3</sup> <http://mite.physik.uzh.ch>



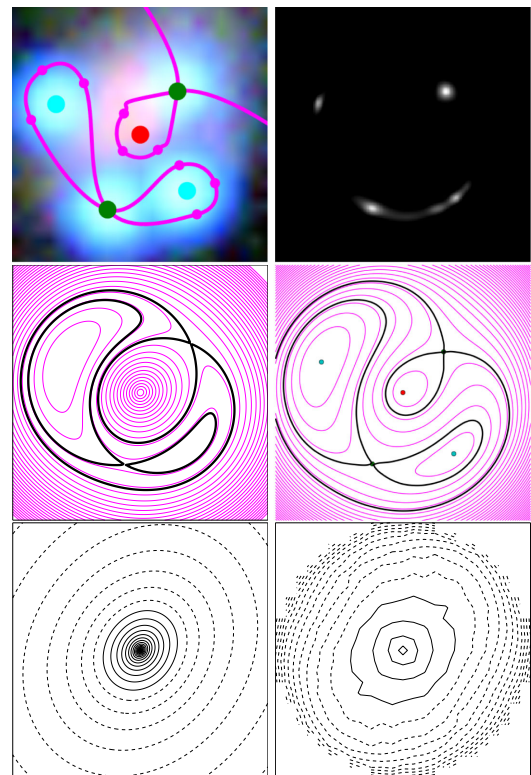
**Figure 3.** A simulated lens that mimics a lensed quasar, and model results. The left-hand panels derive from the simulation, and the right-hand panels are SpaghettiLens output. Details of individual panels are in Section 4.2.



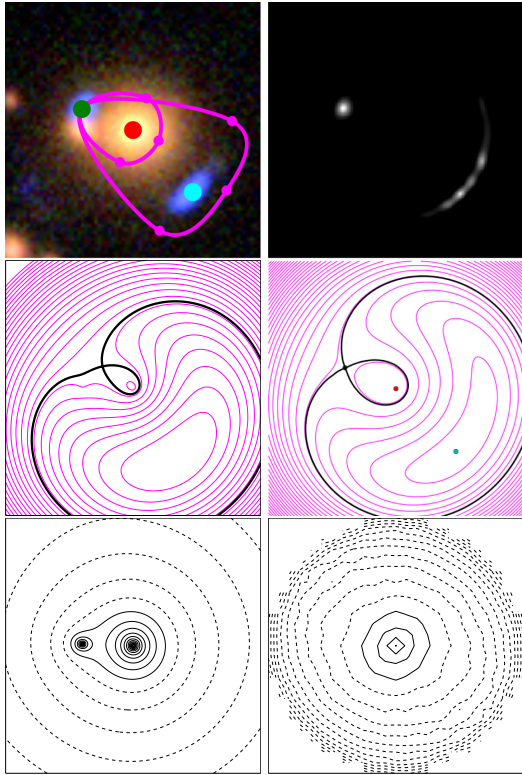
**Figure 5.** Another configuration of arc plus counter-image: an arc and counter-image where the arc is closer to the lensing galaxy than the counter-image. (See Section 4.2 for details.)



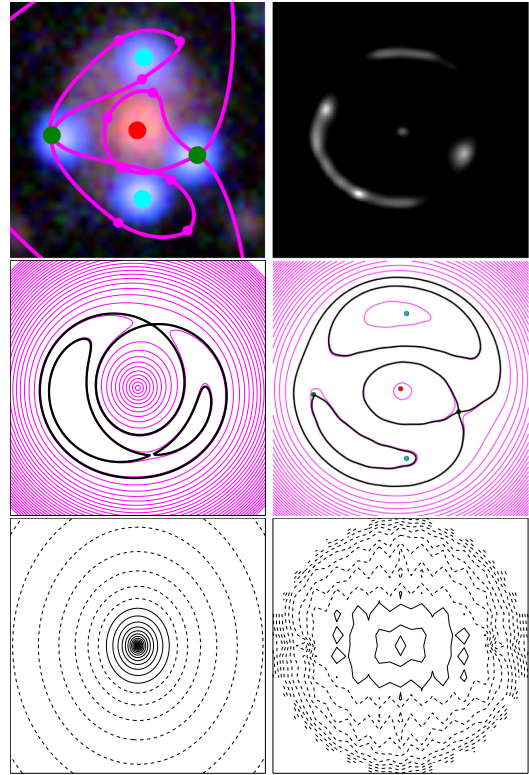
**Figure 4.** Results from a system with an arc plus a counter-image, typical of lensed galaxies. (See Section 4.2 for details.)



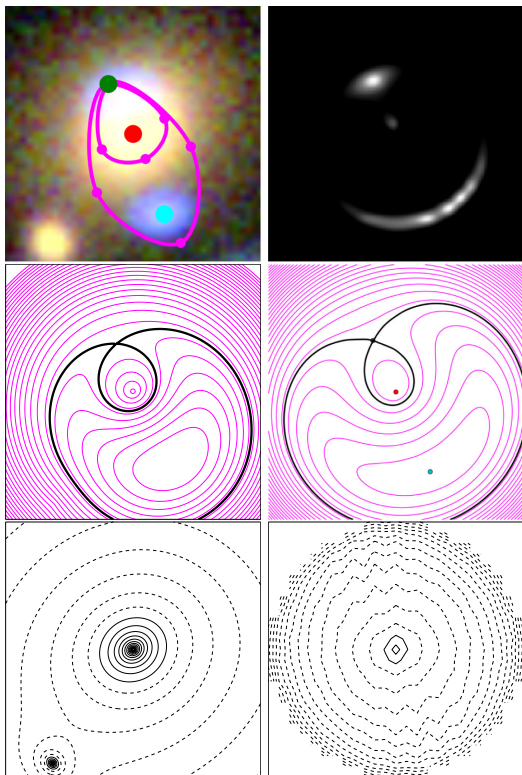
**Figure 6.** A four-image configuration typical of lensed quasars. (See Section 4.2 for details.)



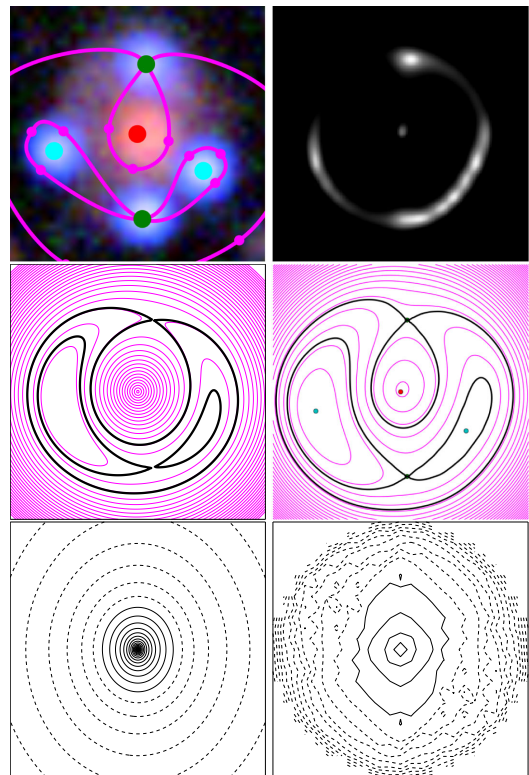
**Figure 7.** A lens with unrecovered mass substructure. (See Section 4.2 for details.)



**Figure 9.** A four-image system with image parities incorrectly identified. The model is poor, but the estimated Einstein radius is not bad. (See Section 4.2 for details.)



**Figure 8.** A sim with unrecovered substructure, resulting in a poor mass model. (See Section 4.2 for details.)



**Figure 10.** The same system as in Fig. 9, this time with image parities correctly identified. (See Section 4.2 for details.)

### 3.3 Diagnostics

After the model ensemble has been generated, SpaghettiLens post-processes it to present results and diagnostics to the user for inspection. This takes the form of three figures.

- (i) A synthetic image of the lensed features.
- (ii) A contour map of the arrival-time surface  $t(x, y)$ .
- (iii) A grey-scale plus contour map of the mass distribution.

The synthetic image generated by SpaghettiLens assumes a simple circularly symmetric source with linearly decreasing surface brightness profile. The user can change the contrast level on the image, which (though it is not saved) amounts to adjusting the size of the source. These synthetic images are still very crude, and not always useful for assessing models. The best indicator, in practice, of whether the modelling was successful is contour map of  $t(x, y)$ , with saddle-point contours highlighted. It is, in effect, the computer's refinement of the spaghetti diagram input by the user. If the arrival-time surface looks qualitatively similar to the spaghetti diagram, that generally indicates a successful model. The mass distribution also provides indications; successful models generally lead to smooth-looking mass distribution, whereas an irregular or checkboard pattern in the mass map signals a bad model.

After examining this feedback, the user can choose to save the model to the SpaghettiLens archive, at which point it is assigned a unique URL. They can also modify the input and try again, or discard the attempt altogether. After archiving, there can be discussion among modellers, through the Space Warps forum or by any other means, and revision of the model. This is achieved simply by sharing the model's URL; following its hyperlink takes one to the SpaghettiLens app, pre-loaded with the correct data image and input spaghetti. Any archived model can be revised by any user: they can modify the spaghetti configuration slightly or drastically, or change options like the size of the mass tiles. Particularly, interesting lens candidates lead to trees of models in this way. Discussion among modellers tends to prune a model tree, focusing attention on the most interesting models.<sup>4</sup>

## 4 A LENS-MODELLING CHALLENGE

We now describe a test of the lens-modelling system, under conditions that mimic as closely as possible the modelling of real lens discoveries. The lenses to be modelled were the simulated lenses (known as 'sims') already sprinkled on to the Space Warps field. Once a small user base had grown around SpaghettiLens, a modelling challenge was announced through the Space Warps forum. The challenge set consisted of 29 sims, chosen to represent the different visual morphologies of Space Warps sims. Modellers then contributed 119 models for these sims (at least two for each sim). Models were reported on the same forum used to model real candidate lenses. Modellers were free to consult and refine each other's models, but had no information on how the sims were generated.

Once the modelling was complete, the models were compared with the originals. There were two main tests: a check of whether the spaghetti diagrams were correct for the lens in question, and a comparison of the effective Einstein radii of the sims and the models.

<sup>4</sup> See 'Collaborative gravitational lens modelling...' in <http://letters.zooniverse.org> for an example.

### 4.1 The simulated lenses

The Space Warps sims are described in detail in More et al. (in preparation), but relevant here is that the sims were of three kinds, as follows.

- (i) Lensed quasars: the lens is modelled as a singular isothermal ellipsoid (SIE) and a constant external shear whereas the quasar is represented with a circular Gaussian source whose size is given by the point spread function in each imaging band.
- (ii) Galaxy-scale lenses: the lens model is the same as above whereas the background galaxy is modelled as an elliptical de Vaucouleurs.
- (iii) Group-scale lenses: the lens model includes SIE models for the central galaxy and the inner group members, plus a circular NFW (Navarro, Frenk & White 1996, 1997) to represent the underlying dark matter distribution and the background galaxy model stays the same as galaxy-scale lenses.

The GRAVLENS program (Keeton 2001b) was used. Formulas for the lenses appear in Keeton (2001a). The SIE lenses follow equations 33–35 of that work, with core radius set to zero. The NFW lens is in equations 48 and 50, while shear is the  $\gamma$  term in equation 76.

The information in this section was not revealed to the main developer of SpaghettiLens (RK, who also chose the challenge set) or to the modellers (EB, CC, CM, JO, PS, and JW) while modelling was in progress. That is, the modellers had no advance knowledge of what kind of parametrization had been used to make the sims. After the modelling stage, AM released the details of the sims for post-modelling analysis. Results from the latter now follow.

### 4.2 Some example models

Of the 119 models proposed, we now discuss eight examples in some detail. Results from these are shown in Figs 3–10. The first four of these show the most common image morphologies, the other four explain some problem cases.

Each of Figs 3–10 figures has the following layout.

Marked-up CFHTLS image	Model synthetic image
$t(x, y)$	model $t(x, y)$
$\kappa(x, y)$	model $\kappa(x, y)$

The model synthetic image presented in this paper is not the original one, but an interpolated version generated by an updated version of SpaghettiLens. The image presented during the original experiment was of lower resolution. The two plots in the middle showing  $t(x, y)$  have uniform, but arbitrary spaced contour lines. The  $\kappa(x, y)$  plots in the bottom row show solid lines for  $\kappa > 1$  and dashed lines for  $\kappa < 1$ . The spacing is logarithmic, with 10 contours for every decade – that means contour spacing is a quarter magnitude in optical terms.

Let us now consider these cases in turn.

- (i) Fig. 3 shows the simplest case, with two clear images produced by a nearly circular lens. The centre of the lensing galaxy is a maximum, the image nearer to the galaxy is a saddle point, and the image further away is a minimum. All these were correctly identified. As noted above, in Section 3, the precise shape of the loops in the spaghetti diagrams is unimportant, only the implied image locations, parities and time-ordering matters.  $\kappa(x, y)$  shows, that the model has a more shallow mass distribution than the simulation. This is a persistent issue throughout all models and is discussed in Section 4.4.

(ii) Fig. 4 shows an example of an arc that has split into three images. This kind of configuration, with a counter-image close to the lensing galaxy and a more distant arc/triplet on the other side, generally arises from an elongated mass distribution when the source is displaced along the elongated direction. The spaghetti diagram in this case has another markup element, a grey point and circle overlaid on a probable secondary lensing galaxy. This is an instruction to SpaghettiLens to allow a point mass at that location, distinct from the main mass map.

(iii) Fig. 5 shows another example of an arc plus counter-image, but (in contrast to Fig. 4) the arc is closer to the lens than the counter-image. This configuration arises if the source displacement is perpendicular to the long axis of the lensing mass. Comparing the two panels in the middle row, we see that the modeller interpreted arc as consisting of three images, whereas the sim shows a single saddle point associated with the arc. But the identification is not really erroneous – we just need to take into account that the source is extended. In fact, in the sim, the brightest part of the source is only doubly imaged, but the source extends into a region that produces four images. In the  $t(x, y)$  of the sim, the hairpin-bending contours are typical of double on the verge of splitting into a quad.

(iv) Fig. 6 shows another quad. This kind of configuration arises when the mass is elongated and the source is displaced at an angle to the elongation. The minima and saddle points are correctly identified, and the orientation of the ellipticity of the mass distribution is correctly reproduced.

(v) Fig. 7 shows a lens with substructure in the form of a smaller secondary galaxy. The galaxies in such group or cluster sims were based on galaxies visible in the images, but the modellers were not told in advance whether this was the case. The minimum and saddle point are correctly identified. The mass distribution misses the substructure, but overall appears reasonable.

(vi) Fig. 8 shows a sim with substructure, like Fig. 8. In comparison to the above, the resulting mass model is poor.

(vii) Fig. 9 shows a quad. In this one, the identification of the minima and saddle points was incorrect, and mass distribution comes out elongated east–west instead of north–south. The mass distribution also appears somewhat jagged and the saddle-point contours are not as clean as in the previous examples; these are often indicators of a problem with the model. The enclosed mass is, however, none the worse – the reason is probably that in a relatively symmetrical image configuration, the Einstein radius is quite well constrained by the images in a fairly model-independent way.

(viii) Fig. 10 shows another model of the same system, the only one done by an expert in this sample. The image parities are correct. The elongation has the right orientation, but is too shallow.

### 4.3 Test of image identification

The first post-modelling test was a qualitative comparison of the original arrival-time surfaces and the input spaghetti diagrams given by the modeller. This tested first, for correct identification and location of the lensed images, and secondly, for the correct parities and ordering of the lensed images in respect of the arrival time.

While we expected the identification of lensed images to be trivial, given the generally clean appearance of the sims in the test, we expected the parities and time-ordering to be more difficult. While the SpaghettiLens tutorials had provided general guidelines, to be consistently correct with the time-ordering, a modeller needs to develop some intuition for arrival time surfaces. This is an area where experience and tutorials training could improve results at a later

**Table 1.** Table of image-identification errors and the number of models containing each. A model can contain more than one type of error.

Total	119	100%
Errors in image locations	9	8%
Errors in image parities or time ordering	49	41%
Inaccurate image placement over an arc	21	18%
Identified two images of four	5	4%
Identified two nearby images as one	3	3%
Missed faint images	1	1%
Proposed too many images	1	1%
Modelled a three-image arc as one image	4	3%
Modelled one image as a three-image arc	5	4%
Swapped minimum and saddle in double	2	2%
Swapped minima and saddles in quad	38	32%
Swapped early and late saddles in a quad	7	6%

stage, and correspondingly, feedback on the difficulties modellers encounter can help improve the tutorial materials.

Table 1 presents a summary of the test. The evaluation was done manually, comparing the input to SpaghettiLens with the actual arrival-time surface of the sim. This amounts to comparing the middle-left and middle-right panels in each of Figs 3–10, and similarly for the other 111 models.

The images of the system were considered to have been located correctly, if all the images were identified and were approximated within about 5 per cent of SpaghettiLens frame used to draw the spaghetti diagram. That frame size is adjustable by the user, but in practice it is somewhat larger than the spaghetti diagram. Such image-placement errors were found in only nine models. That does not include inaccurate image placement over an arc, which was considered a separate category of error.

In addition to simple image-placement errors, 10 types of errors were recognized and are listed in Table 1. Most of the problems were due to unclear arc-like structures. Critical errors like the failure to identify all five images in a five images system, or to include too many images, were rare.

The assignment of the parity of the images was a more difficult task, and was successful in only about 60 per cent of the cases. The most common error was swapping of minima and saddle points in a quad; Fig. 9 shows an example. Another, less common, error was flipping the spaghetti diagram, thus swapping the time-ordering of the two saddle points.

Incorrect image parities and time-orderings tended to produce poorer-looking models, such as the checker board patterns in the mass map in Fig. 9. Interestingly, however, the enclosed-mass profiles were quite robust. We will consider this aspect in the next section.

### 4.4 Test of mass-profile recovery

The second test was to compare the mass distributions  $\kappa(x, y)$  of the sims and of the SpaghettiLens models. A visual comparison is presented for the eight models in Figs 3–10, in the lower-left versus lower-right panels. We will summarize the mass distributions drastically in a single number, the effective Einstein radius. Other measures for comparison of free-form lensing-mass distributions appear in Coles et al. (2014), but comparing Einstein radii is already useful.



There is no standard way of defining the Einstein radius of a general non-circular lens. We adopt the simple definition

$$(\kappa)_{\Theta_E} = 1, \quad (6)$$

that is, the effective Einstein radius  $\Theta_E$  is such that the mean  $\kappa$  is unity inside a circle of radius  $\Theta_E$  centred at the lens centre.

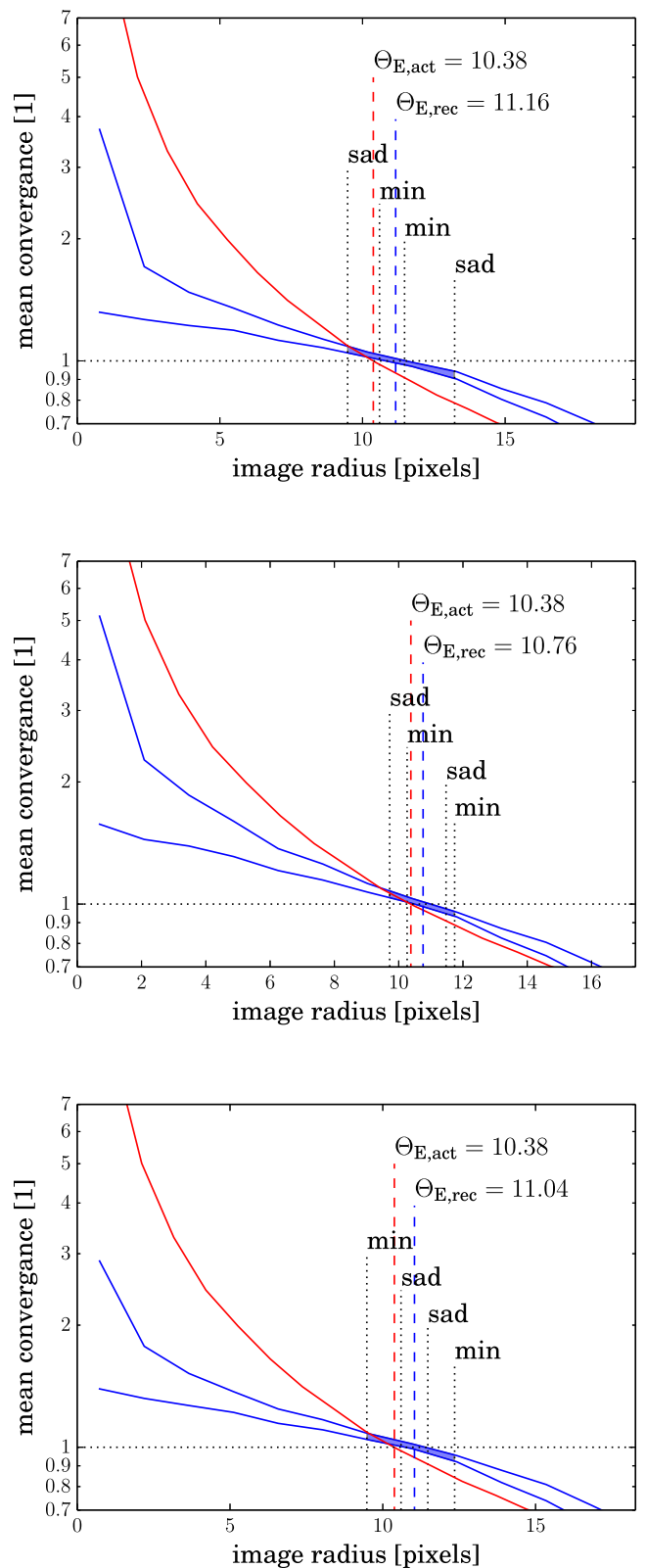
To illustrate, Fig. 11 compares the circularly averaged mass profiles of three different models of one particular lens; two of the models are shown in Figs 9 and 10. Each panel in Fig. 11 shows the mean  $\kappa$  within a circle of given radius. The red curve is the correct profile for the sim. The two blue curves are the minimal and maximal mean enclosed  $\kappa$  from the internal ensemble in SpaghettiLens. Radial locations of the images are marked, along with the image parities. The region between the blue curves is shaded between the radii of the innermost and the outermost images: this is the confidence region from the modelling. The definition (6) for  $\Theta_E$  corresponds to crossing the dashed horizontal line at 1: the red curve crosses the dashed line at the actual Einstein radius  $\Theta_{E,\text{act}}$ ; the recovered Einstein radius  $\Theta_{E,\text{rec}}$ ; and its uncertainty are given by the blue curves crossing the dashed line. We see that in all three panels, the blue curves are shallower than the red curve and  $\Theta_{E,\text{rec}}$  is more than  $\Theta_{E,\text{act}}$ , by more than the model uncertainties. Now, steeper mass profiles tend to give wider image separations – recall that the image separation for a circular isothermal lens is  $2\Theta_E$ , whereas for a point mass it is more (see e.g. Courbin, Saha & Schechter 2002) – so  $\Theta_{E,\text{rec}}$  being too high is really a consequence of the GLASS models being too shallow for the sims.

Fig. 12 shows that  $\Theta_{E,\text{rec}}$  of the models tend to be too high. However, this is entirely due to the GLASS model density profiles being too shallow, as illustrated above. We can separate out the performance of the SpaghettiLens interface and its users by comparing their results with the Einstein radii of SpaghettiLens models made by an expert (PS). Discounting the models which were flagged by the volunteers as poor, the mean Einstein radius overestimate was 10 per cent, with a 15 per cent standard deviation (shown by the light grey band in Fig. 12). The expert models show a similar bias, with standard deviation 10 per cent (the dark grey band in Fig. 12). One source of this systematic error is that it is difficult to centre the lens accurately: an offset leads to a flatter mass profile for the model compared to the simulation.

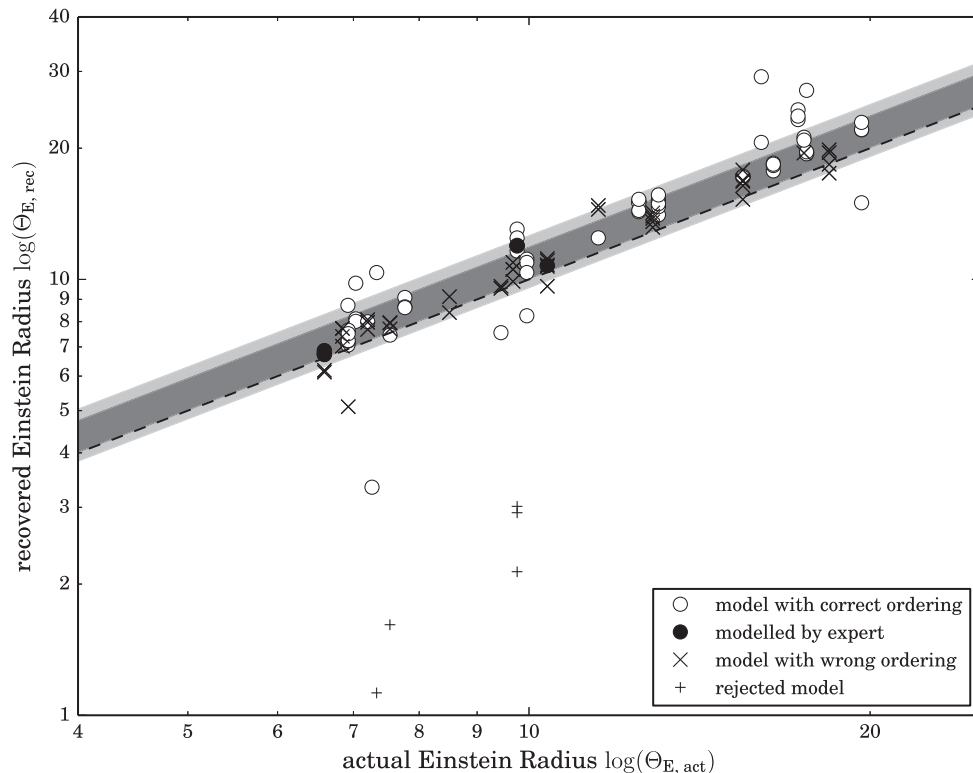
## 5 OUTLOOK

This work has developed the concept of saddle-point contours in the traveltime of virtual photons, originally introduced by Blandford & Narayan (1986) for understanding image structure in strong gravitational lenses, into a technique for mass-mapping lenses. Despite being highly abstract, saddle-point contours look like schematic arcs, and hence lend themselves to an intuitive markup tool for lenses or lens candidates, which we call a spaghetti diagram. At the same time, saddle-point contours encode information about possible mass distributions, which can be translated into input for an existing lens-modelling engine (GLASS; by Coles et al. 2014).

SpaghettiLens is an implementation of these ideas, enabling experienced but non-professional lens enthusiasts to model newly discovered lens candidates from the Space Warps citizen science platform. The tests in this paper indicate that such modelling would be both feasible and scientifically interesting: given a suitable modelling tool, and appropriate guidance, a small team of non-professional volunteers was able to model a sample of 29 test lenses, and measure their Einstein radii with comparable accuracy to a professional expert.



**Figure 11.** Mean  $\kappa$  inside a circle around the lens centre, as a function of the radius of the circle. (See Section 4.4 for details.) The upper panel corresponds to the model shown in Fig. 9, in which the minima and saddle points have been incorrectly swapped. The middle corresponds to Fig. 10, where the image parities were correct. The lower panel corresponds to another model, where the image parities were correct but the time-ordering was incorrect.



**Figure 12.** Model-recovered versus actual Einstein radii  $\Theta_{E, \text{rec}}$  and  $\Theta_{E, \text{act}}$ . Plus signs indicate models flagged by the modeller as failures by commenting negatively about it in the forum. Light and dark grey bands show standard deviation of volunteers (15 per cent) and expert (10 per cent).

There is, however, plenty of room for improvement.

(i) SpaghettiLens tends to overestimate the Einstein radius (evident from Fig. 12), and the model density profiles tend to be shallower than the SIE model used for generating the sims. The likely explanation is that while the sims are steeply peaked at the centre, the pixelated mass model fixes a comparatively large area near the central at constant density. Allowing smaller pixels in the centre region, thus enabling a steeper centre (similar to the ‘high resolution’ feature implemented in Coles et al. 2014) may remove this bias. The use of simply parametrized, appropriately steeply-profiled models would also avoid the problem.

(ii) Currently, SpaghettiLens does not attempt to model the source shape; the user identifies the brightest points on the image, and these are taken as images of a point-like source, whose positions must be reproduced exactly. For generating a synthetic image, a conical source profile is assumed. Fitting for the source profile to optimize resemblance to the observed lensed image after the lens model has been generated, is algorithmically straightforward (cf. Warren & Dye 2003; Suyu et al. 2006) and planned to be implemented. This would alleviate another problem with SpaghettiLens, which is that there is as yet no quantitative figure of merit for any given model: assessment of each model is a judgment call based on the synthetic image, and on whether the mass distribution and the arrival-time surface show suspicious features. Another possibility would be use the SpaghettiLens models as a feeder to a different lens-modelling program that already implements source-profile fitting.

(iii) Another limitation so far in SpaghettiLens is that the lens is assumed to be dominated by one galaxy, which puts most galaxy-group lenses beyond the reach of the modeller. Since complicated group lenses are some of the most interesting candidates present,

removing this limitation is most desirable. From the users’ point of view, it would mean that spaghetti contours with more than one maximum can be allowed. For examples, see fig. 5c in Rusin et al. (2001) and fig. 4b in Keeton & Winn (2003).

(iv) At present, a single false-colour composite is used as the data. An option could be added to use all available filters, individually or in combination, at the user wishes.

(v) As mentioned above, the option of revising an already-archived model is already available. Desired now are tools for comparing different models of a given system, both visually and through different statistical measures. As evidenced by a current collaborative modelling effort, a particularly interesting candidate can lead to an extended discussion and dozens of models, that in some way sample the high likelihood region of model parameter space.

(vi) Better tutorial materials are also needed, and this would address some of the problem areas found in the modelling challenge. For example, we saw in Section 4.3 that volunteers are most prone to making errors in two situations: when in identifying an arc-like structure while placing the points, and in identifying the correct ordering of the points in nearly symmetric configurations. Better and more detailed introductory materials would also allow the community of modellers to grow faster and without individual instructions by experts or experienced volunteers.

The SpaghettiLens program was developed by RK, with design suggestions from JC, CC, and PS, and feedback from all co-authors. The simulations were created by AM, in consultation with PM, SM, and AV. Modelling was done by EB, CC, CM, JO, PS, and JKW, with post-modelling analysis by RK and PS. All authors participated in writing and editing the manuscript.

## ACKNOWLEDGEMENTS

We thank the Swiss Society for Astrophysics and Astronomy and the Swiss Academy of Sciences. The work of AM and SM was supported by World Premier International Research Center Initiative (WPI Initiative), MEXT, Japan. AV is supported by a research fellowship from the Leverhulme Trust. RK is supported by the Swiss National Science Foundation. This work was supported in part by the US Department of Energy under contract number DE-AC02-76SF00515.

## REFERENCES

- Blandford R., Narayan R., 1986, *ApJ*, 310, 568  
 Coles J., 2008, *ApJ*, 679, 17  
 Coles J. P., Read J. I., Saha P., 2014, *MNRAS*, 445, 2181  
 Courbin F., Saha P., Schechter P. L., 2002, in Courbin F., Minniti D., eds, *Lecture Notes in Physics*, Vol. 608, *Gravitational Lensing: An Astrophysical Tool*. Springer-Verlag, Berlin, p. 1  
 Gavazzi R., Marshall P. J., Treu T., Sonnenfeld A., 2014, *ApJ*, 785, 144  
 Keeton C. R., 2001a, preprint ([astro-ph/0102341](#))  
 Keeton C. R., 2001b, preprint ([astro-ph/0102340](#))  
 Keeton C. R., Winn J. N., 2003, *ApJ*, 590, 39  
 Khatib F., Cooper S., Tyka M. D., Xu K., Makedon I., Popović Z., Baker D., Players F., 2011, *Proc. Natl. Acad. Sci.*, 108, 18949  
 Koopmans L. V. E. et al., 2009, *ApJ*, 703, L51  
 Lefor A. T., 2014, *Astron. Comput.*, 5, 28  
 Leier D., Ferreras I., Saha P., Falco E. E., 2011, *ApJ*, 740, 97  
 Lubini M., Coles J., 2012, *MNRAS*, 425, 3077  
 Marshall P. J., Moustakas L. A., Hogg D. W., Bradač M., Fassnacht C. D., Blandford R. D., 2005, *BAAS*, 37, 1498  
 Marshall P. J., Hogg D. W., Moustakas L. A., Fassnacht C. D., Bradač M., Schrabback T., Blandford R. D., 2009, *ApJ*, 694, 924  
 More A., Cabanac R., More S., Alard C., Limousin M., Kneib J.-P., Gavazzi R., Motta V., 2012, *ApJ*, 749, 38  
 Naudus P., Wallin J., Marshall P., 2010, *BAAS*, 42, 383  
 Navarro J. F., Frenk C. S., White S. D. M., 1996, *ApJ*, 462, 563  
 Navarro J. F., Frenk C. S., White S. D. M., 1997, *ApJ*, 490, 493  
 Oguri M., Marshall P. J., 2010, *MNRAS*, 405, 2579  
 Raddick M. J. et al., 2013, *Astron. Educ. Rev.*, 12, 010106  
 Refsdal S., 1964, *MNRAS*, 128, 307  
 Refsdal S., 1966, *MNRAS*, 132, 101  
 Rusin D. et al., 2001, *ApJ*, 557, 594  
 Saha P., Williams L. L. R., 1997, *MNRAS*, 292, 148  
 Schwamb M. E. et al., 2013, *ApJ*, 768, 127  
 Sereno M., Paraficz D., 2014, *MNRAS*, 437, 600  
 Suyu S. H., Marshall P. J., Hobson M. P., Blandford R. D., 2006, *MNRAS*, 371, 983  
 Walsh D., Carswell R. F., Weymann R. J., 1979, *Nature*, 279, 381  
 Warren S. J., Dye S., 2003, *ApJ*, 590, 673  
 Weymann R. J. et al., 1980, *Nature*, 285, 641  
 Young P., Deverill R. S., Gunn J. E., Westphal J. A., Kristian J., 1981a, *ApJ*, 244, 723  
 Young P., Gunn J. E., Oke J. B., Westphal J. A., Kristian J., 1981b, *ApJ*, 244, 736  
 Zwicky F., 1937, *ApJ*, 86, 217

## APPENDIX A: RELATION TO STANDARD LENSING FORMALISM

The description of the arrival-time surface in Sections 2 and 2.2 omitted some details for the sake of a more intuitive explanation. The convergence  $\kappa$  and the geometric time delay  $t_{\text{geom}}$  were left

as proportionalities (equations 2 and 3), and the gravitational time delay  $t_{\text{grav}}$  was given in an implicit form (equation 4). Here, we fill in the details.

The original formulation of the arrival-time surface appears in equations 2.1–2.6 of Blandford & Narayan (1986). Their equations can be rearranged as follows:

$$t_{\text{geom}} = \frac{(1+z_L)}{2c} \frac{d_S}{d_L d_{LS}} [(x-x_s)^2 + (y-y_s)^2]$$

$$\nabla^2 t_{\text{grav}} = -(1+z_L) \frac{8\pi G}{c^3} \Sigma(x, y)$$

$$\kappa(x, y) = \frac{4\pi G}{c^2} \frac{d_L d_{LS}}{d_S} \times \Sigma(x, y). \quad (\text{A1})$$

Here, the symbols  $d_L$ ,  $d_S$  and  $d_{LS}$  are angular-diameter distances, respectively, from observer to lens, observer to source, and lens to source. We have replaced angular positions on the sky with positions on the lens plane as

$$(x, y) = d_L(\theta_x, \theta_y). \quad (\text{A2})$$

In the concordance cosmology:

$$d_{LS} = \frac{c}{H_0} \frac{1}{1+z_S} \int_{z_L}^{z_S} \frac{dz}{\sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}} \quad (\text{A3})$$

and similarly  $d_L$  and  $d_S$ .

The first line of equation (A1) is  $t_{\text{geom}}$  from equation (3) with the proportionality filled in, and with the source offset at  $(x_s, y_s)$  rather than at the origin. The last line of equation (A1) fills in the proportionality factor in equation (2) for the convergence (or dimensionless surface density)  $\kappa$ .

The middle line of equation (A1) is a Poisson equation for the gravitational time delay, and is equivalent to the implicit expression (4). One way to verify the equivalence is to consider the small circle in equation (4) as a region where  $\Sigma$  is constant, and approximate  $t_{\text{grav}}$  by its Taylor expansion to  $O(x^2, y^2)$ . Substituting in equations (A1) gives the Taylor coefficients in terms of  $\Sigma$ , and result satisfies the expression (4). Alternatively, we can proceed with a discrete form of the Poisson equation from (A1). Discretizing on a grid with spacing  $\Delta$ , we have

$$t_{\text{grav}}(x, y) = \frac{1}{4} [t_{\text{grav}}(x + \Delta, y) + t_{\text{grav}}(x - \Delta, y) + t_{\text{grav}}(x, y + \Delta) + t_{\text{grav}}(x, y - \Delta)] + (1+z_L) \frac{2G}{c^3} \pi \Delta^2 \Sigma(x, y). \quad (\text{A4})$$

This is recognizable as a formula for solving the two-dimensional Poisson equation from equation (A1) by relaxation. Let us now replace the average over four neighbouring points by the circular average  $\langle t_{\text{grav}}(x_\circ, y_\circ) \rangle$  and replace  $\pi \Delta^2 \Sigma(x, y)$  by the enclosed mass  $M(x_\bullet, y_\bullet)$ . These replacements are valid in the limit of a small grid. The result is the implicit equation (4).