Igloo[1]: a lens simulation and fitting program

V. 7 (July 2004)

A quick guide

Igloo will calculate a picture of the image plane, given a position of a source (or sources) and parameter(s) of a lensing galaxy or galaxies. It will also fit parameters of lensing galaxies to obtain the best match with observations.

How to adjust parameters

On starting igloo, a screen pops up containing various buttons. The top row of buttons refers to parameters of the source; the red row in the middle refers to parameters of galaxies. Source parameters are, in order: X and Y position on the grid, major and minor axis and position angle (to cope with extended sources), F (a spare parameter which doesn't do anything) and redshift. Galaxy parameters are, in order: X and Y position on the grid, Einstein radius, position angle of galaxy major axis (defined east of north), ellipticity of galaxy mass distribution, core radius S, an unused parameter a, redshift and model type. Model types between 0 and 0.99 refer to point sources; models between 1.00 and 1.99 (the default) give singular isothermal ellipsoid mass distributions (a la Kormann, Bartelmann & Schneider) and values from 2.00 to 2.99 give Blandford & Kochanek SIE potential distributions. Values of 3.00 to 3.99 give a cusp model with inner and outer slopes a1 and a2. At the moment the core radius works only for the B-K potential, although Kovner and others have pointed out this becomes unphysical at large ellipticities. The KBS and BK models should give pretty similar results for low ellipticities.

X and Y are defined on a 128 by 128 grid. The grid size is soft-wired in (by a #define statement) so can be changed fairly easily. Each grid pixel represents a given angle, so that the plane represents what is seen on the sky. The Einstein radius (b) is defined by angle and not in absolute value; if you want to convert a velocity dispersion to b, you need the gory-details section later in this document.

To adjust any parameter upwards by a small amount, click the left mouse button; to increase by a large amount, click the middle button. ("Large" and "small" amounts are increments which for the moment are hardwired into the program, although they do have fairly sensible values). Extremely small increments are achieved by "s". To make the left and middle buttons decrease instead of increase, hit the right-hand mouse button and proceed as before.

You may want more than one source or one galaxy. To do this, hit the "addS" and "addG" buttons and adjust the parameters of the resulting source or galaxy that will pop up. Up to 8 of each are currently allowed.

How to look at the image plane

Once you are happy with the parameters, hit "eval". You then get several things. First, a greyscale plot is produced of the image plane Green blobs indicate positions of galaxies and red blobs positions of sources. You can move them around with the mouse without touching the X and Y number buttons. A hit in the picture area with the left-hand mouse button moves the nearest source to that point, and a hit with the middle button moves the nearest galaxy to that point. The right-hand button does the same thing as the "eval" key.

[1] Interactive Gravitational Lens Optimisation Olgorithm

How to read in a model

You will need a file which looks like the one below. All lines beginning with # are comments. The first line is the number of observed images, and subsequent lines give the source number to which the image belongs, the RA (relative to a reference point), error in RA, declination, error in declination, flux, error in flux, and parity (+1 or −1). Note that the RA does not follow astronomical convention; higher RAs are to the right of the picture.

The lines that follow describe the parameters of the galaxy, including its type (0=point source, 1=SIE, 2=elliptical potential, 3=cusp model), its position (`xlens` and `ylens`), the critical radius in arcseconds, the ellipticity and the position angle (`theta`) of the galaxy. The external shear is specified in magnitude and direction. To specify more than one galaxy, give the command "new" on a line by itself. All subsequent parameters will be taken to refer to the new galaxy.

The "0" and "1" which appear after the values of some parameters indicate whether these parameters are to be optimised by default. Optimised parameters will register as purple on the screen.

```
#### Image parameters ####
#  RA  RAerr DEC    DECerr Flux  Flerr  parity
4
1 +0.7726 0.0100 -0.1463 0.0100 10.50 1.10 +1
1 +0.7216 0.0100 -0.3063 0.0100 8.50 0.90 -1
1 -0.0333 0.0100 -0.8163 0.0100 4.70 0.50 +1
1 -0.3904 0.0100 +0.3137 0.0100 0.90 0.10 -1
#
### Galaxy parameters ###
#
1
type    = 1.000000
xlens   = -0.000619 0.010000 1
ylens   = -0.010170 0.010000 1
crit    = 0.696130  1
ellip   = 0.365829  1
theta   = 50.316353  1
#
### External shear ###
#
shear = 0.000000 0
angle = 0.000000 0
#
### Redshifts ###
#Source Lens
zl = 0.500000
zs = 1.000000
### Annealing ###
iter = 200
starttemp=200
deltemp = 100
```

Subsequent use of the "eval" key will show the images in yellow (positive parity) or brown (negative parity).

How to optimise

The first thing to do is to adjust the galaxy position, ellipticity and PA in order to get the parities right. The optimisation routine imposes a large penalty on any parities of the wrong sign. The parameters need to be adjusted so that the ellipse produced by "pmag" encloses all the brown images and none of the yellow ones.

Adjust the `doop` parameter and use the `opt` button as detailed below.

Gory details

The bottom end

Most of the bottom-level work is done in `ggetu.c`. This calculates the convergence and shear for any given point in the image plane, for the given galaxy parameters. It implements equations from Kormann et al. (1994), Blandford & Kochanek (1987) and Schneider, Ehlers & Falco (1992), as follows.

For a point source, the deflection potential $\psi = \ln r/b$, where r is the angular distance from the deflector. Hence the convergence κ and shear γ are given by

$$\kappa = (\psi_{,xx} + \psi_{,yy})/2 = 0$$

$$\gamma_1 = (\psi_{,xx} - \psi_{,yy})/2$$
$$\gamma_2 = (-\psi_{,xy})$$

where subscripts refer to derivatives, and x and y are orthogonal Cartesian coordinates in the image plane (NB in most works, including that of Kormann et al., these are referred to as x_1 and x_2 and y is reserved for coordinates in the source plane). The angular source position is recovered from the lens equation, $x_s = x - \psi_{,x}$ and similarly for y.

For a KSB lens (singular isothermal ellipsoid mass distribution), the code simply implements equations from KSB. First the grid position is rotated to the galaxy axis, and then equations 21b, 21c, 27a and 29 together with equation 10 are used for the convergence and shear.

For a BK lens, the convergence and shear are obtained from the derivatives of the BK isothermal potential: $\psi = (1-e)x^2 + (1+e)y^2$ in exactly the same way as for a point-source lens.

Cusp models are handled using the method of Chae (2003) and the Fortran code supplied in that paper.

The U matrix, defined as in SEF, equation 9.10, is then calculated and passed back. For a single lens plane, $\kappa = \Sigma\kappa_i$ and $\gamma^2 = (\Sigma\gamma\cos\theta)^2 + (\Sigma\gamma\sin\theta)^2$ (e.g. equation 9.14 of SEF is a special case of this).

One step up from all this is the `gesm.c` program, whose purpose is to turn the convergence and shear which has been got separately for each of the n galaxies whose redshifts are less than the source redshift, into offsets and magnifications. The sortindex routine returns the galaxies in redshift order, and the rest of the program is devoted to implementing equations 9.1, 9.3, 9.11 and 9.12 of SEF. In each case the recursion relation (9.3 and 9.12) is done first to calculate the U matrix and offset α for each galaxy separately. Note that the angular versions of the equations are used. The Einstein radius throughout is defined in pixels, and therefore in an angle calculable by multiplying by the current value of arcseconds/pixel. An absolute value in Mpc can be got by multiplying by D_l, the angular diameter distance of the galaxy.

For the record, the relation between this and velocity dispersion is (see any GL textbook):

$$b_{\rm Mpc} = 4\pi\frac{v^2}{c^2}\frac{D_{ls}D_l}{D_s}$$

On the subject of ellipticities

There are three ellipticities to consider:

- ϵ_m, the ellipticity of the mass distribution
- ϵ_ψ, the ellipticity of the potential distribution
- ϵ_p, the ellipticity appearing in the potential equation

In each case let the axis ratio be represented by f, thus $f_m = 1 - \epsilon_m$ and so on.

From Kormann et al. (equations 26 and 32):

$$f_\psi = \mathrm{acos}(f_m)/\mathrm{acosh}(1/f_m)$$

The relation between ϵ_ψ and ϵ_p can then be got by considering the potential equation for an SIE:

$$\psi^2 = (1-\epsilon_p)x^2 + (1+\epsilon_p)y^2,$$

which implies an axis ratio of $\sqrt{(1-\epsilon_p)/(1+\epsilon_p)}$. This must equal $1-\epsilon_\psi$, so rearranging we get

$$\epsilon_\psi = 1 - \sqrt{\frac{1-\epsilon_p}{1+\epsilon_p}}$$

oe solving for ϵ_p,

$$\epsilon_p = \frac{1-f_\psi^2}{1+f_\psi^2}$$

This conversion has to be done in tge `getu` subroutine to convert ellipticity of mass distribution, used in most of the program, to ϵ_p which is required for the BK lens (model 2).

How the optimisation is done

The basis of the optimisation is the AMOEBA downhill-simplex method of Press et al. (1992; Numerical Recipes in C). It is controlled by the method of simulated annealing (also from Press et al.) the purpose of which is to stop the routine falling into the first local minimum it finds. The update subroutine re-draws the image plane every time the fit gets better. The goodness-of-fit is evaluated by the funk routine in `gopt.c`; this contains the code for maintaining or excluding the optimisation in the image plane as well as the source plane.

References

Blandford & Kochanek (BK) 1987. ApJ 321, 658

Kassiola & Kovner 1993. ApJ 417, 450

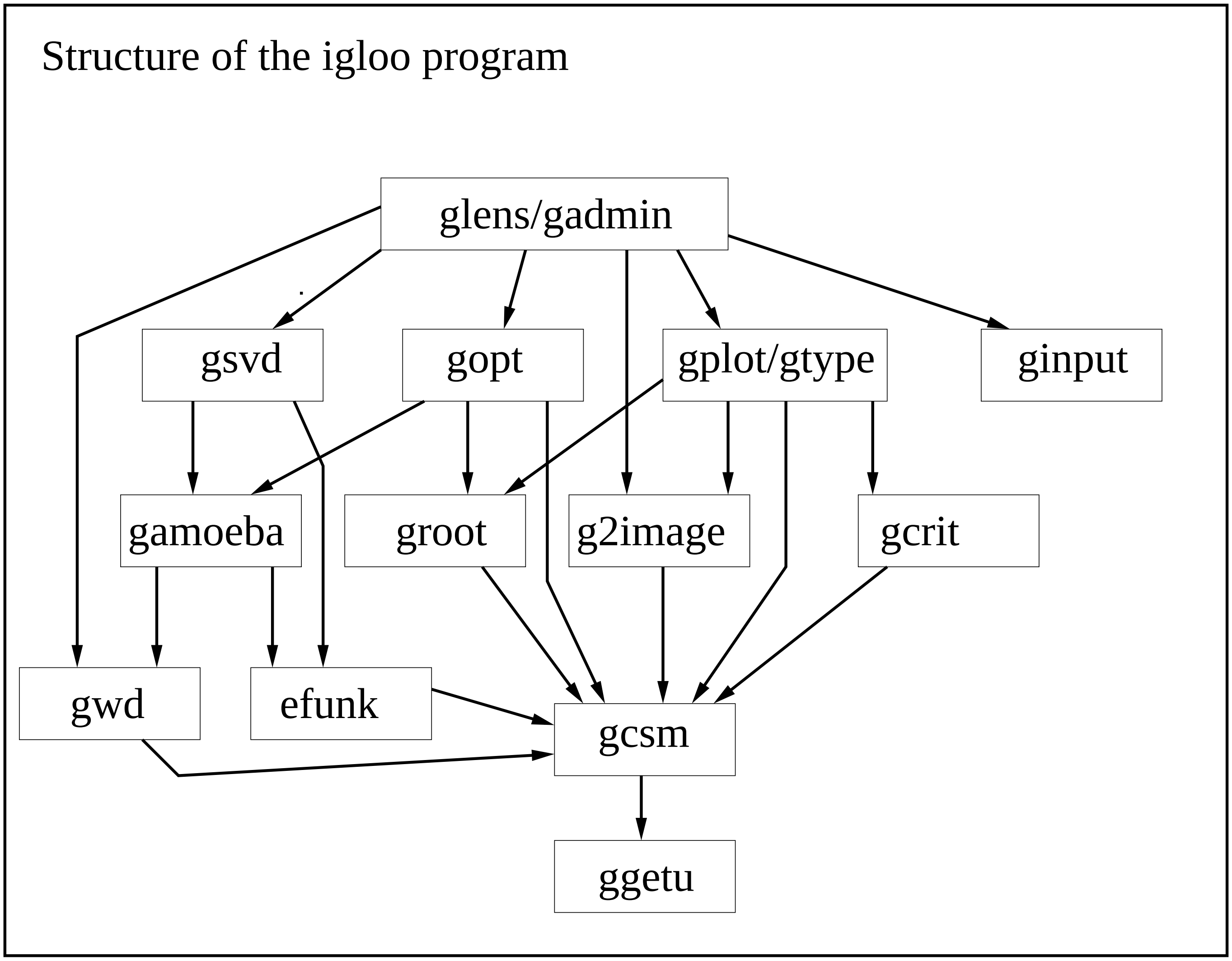
Kormann, Bartelmann & Schneider (KBS) 1994. A&A, 284, 285.

Nair 1997. MNRAS, astro-ph/9803076.

Press et al. 1992. Numerical Recipes in C.

Schneider, Ehlers & Falco (SEF). “Gravitational lenses”, publ. Springer

A slower guide button by button



Function buttons

addG

Purpose: adds an extra galaxy to the model with default parameters. Up to a maximum number (MAXG, defined in gdefs.h) may be added before the program will refuse to add more.

addS

Purpose: adds an extra source to the model with default parameters. Up to a maximum number (MAXS, defined in gdefs.h) may be added before the program will refuse to add more.

exit

Purpose: exit program

type

Purpose: run a few programs which do not have their own button. Available commands are

`findsrc` – asks for an image-plane position in arcsec offset from the centre of the grid and calculates the source position using the current model

`findimg` – calculates the image positions corresponding to a source position input by the user `findimX` where X is any letter other than g, finds the image positions corresponding to the current first source.

Note that the image finding will not give the correct positions or fluxes for core images if `core` is set to zero, although this mode is faster (and should always be used for any optimization where core fluxes are not available). `core`=1 takes several times as long but will find core images down to a magnification of 10^{-6} currently set by the MINCOREMAG parameter in gdefs.h. Any image within IGNORECORE pixels from the centre is assumed to be the core image.

`magt` returns the full magnification tensor at a point in the image plane specified by the user (in arcsec offset from the centre).

eval

Plots a picture of the image plane using the current model and source positions by a direct call to `g2image`. Each pixel is assigned a flux value

$$f_s e^{-3d^2/w_s^2}$$

where f_s is the flux of the source, d is the distance between the projection of the image plane pixel and the source, and w_s is the characteristic size of the source.

opt

When hit using any key or mouse button, starts off an optimization using the current value of `doop`. This may be any of the following:

- N changes the parameters to be optimized without doing an optimization
- F does a pseudo-image plane optimization with positions and fluxes
- P does a PIP optimization with positions only
- I does an image-plane optimization with positions and fluxes, using the `core` parameter to decide whether to consider core images
- H does an IP optimization with positions only
- W does an optimization of the Warren & Dye model to force the best agreement of the calculated image plane with an image and psf supplied by the user. See the remarks on the `wd` button.

The user is invited to set parameters to be optimized using the left-hand (A) mouse button to select a parameter, middle button (D) to deselct and right-hand button (X) to begin the optimization. By default the source position is not optimized unless explicitly requested; otherwise the program takes the weighted average of the back-projected source positions. The weighting is either done by the inverse square of the data errors or using the magnification matrix; changing the mode requires a change in the `gopt.c` code.

The optimization will be restarted from the best-fit a total of `aiter` times. It is carried out using simulated annealing with a starting temperature `stemp`, a temperature decrement `dtemp` and a number of iterations `iter` all of which can be set by the user. The stopping criterion is set by the TOL (tolerance) and FTOL (fractional tolerance) parameters in gdefs.h.

By default, if `alter` is greater than 1, the first iteration is run without optimization of source positions unless this has already been selected by the user; subsequent iterations are done with optimization.

rmod

Reads in a model file in the standard format descriibed above.

plot

Plots grey-scale images of various things according to the button pressed. Available buttons are K (convergence), M (magnification), P (potential), D (deflection), X (deflection in x direction), Y (deflection in y direction), G (shear), 1 (component γ_1 of shear), 2 (for γ_2). Potential is slow using cusp models as a whole grid of deflection angles has to be computed and interpolated.

wd

Purpose: runs the Warren & Dye 1993 algorithm to produce a picture of the source plane for a given image plane. It requires two fits files, one for the observed image plane (without any emission from the central galaxy) and one for the PSF. It also requires a data file for the model input. The image and PSF must be square and the same size, and the program has only been tested for 128×128 images.

Vital parameters for this program are set using the brown buttons. The source plane size is `spix` pixels each of `ssca` milliarcseconds. Regularization for the matrix inversion is performed using `reg` which is the logarithm of the regularization parameter. Currently only simple (flat) regularization is available. Finally, the `svs` parameter is the logarithm of the smallest allowed ratio of matrix values which are considered non-singular. It is likely to need to be of the order of -8 to -9.

The execution time for this routine is controlled by two things: the time to produce convolved images, which is proportional to the square of `ssca`, and the inversion time which is proportional to the sixth power of `ssca`. For solution of a 128x128 image these two are equal, at about half a minute each, for `ssca` = 25. It is best to start with relatively coarse pixels if the model is being adjusted and transit to finer pixels afterwards.

The source plane may be rectangular. This saves a lot of time but requires a change in the first line of the gwd.c program. The program attempts to centre the source plane on the back-projection of the brightest image point. To adjust this, the line in the program which sets the xoff and yoff parameters must be changed by hand.

Use: any button

funk

Gives the current projected image positions and χ^2 values for the current model according to the projection method given by `doop` (see the section on `opt` for further details).

caus

Plots the cut and caustic for the current model.

tdel

Plots time delay contours for the current model.

writ

Writes the current data and model out to a standard format data file called output.txt. This is useful for saving models at different stages of optimization.

dump

Writes the current image plane out to a file called igloo_out.fits.

prin

Prints a postscript file containing the current parameters, the image plane and the cut and caustic, together with the time delay surface. A good alternative to a screen-dump.

var1

Varies a specified optimized parameter between specified limits, and optimizes all other optimized parameters at each iteration using the currently specified optimization method `doop` (see the `opt` button for more on this). Produces an output file pgplot.ps of the resulting χ^2 as a function of the varied parameter.

var2

As `var1` but with two varied parameters, hence the output file is a contour plot. Can be very slow, as `aiter` should be used to give the best chance of hitting the minimum at each point.

svd

Performs the Evans & Witt inversion to determine the angular structure of a SIE galaxy with the current parameters. The left-hand button (or A) gives the galaxy shape using the current source position. The other two buttons will perform an optimization, allowing the image positions to vary by three times their error given by the data, in order to obtain the smoothest possible model. The `svs` parameter is important for rejecting singular values in the matrix. It needs to be set by experiment for each problem but should normally be between -4 and -9.