Compound gravitational lensing with `complens`

Jakob Jönsson
Department of Physics
Stockholm University
jacke@fysik.su.se

March 22, 2011

Abstract

This document describes the code `complens` which allows gravitational lensing compounded by host and subhaloes to be computed.

1 Introduction

We use the method of Metcalf & Madau [1] to compute the compound lensing effect of host and subhalo. Sinde the simulation region covers only a small part of the area of the host halo in the lens plane, it is necessary to treat the contributions from the halo outside the simulation region in some way. The fraction of the host halo that consists of substructure is denoted f_Σ. If this fraction of substructure is considered only in the simulation region, then the surface density of the host halo can be written as

$$\kappa_{\rm halo}(\vec{x}') = \begin{cases} \kappa_{\rm host}(\vec{x}') & \text{outside simulation region} \\ [1 - f_\Sigma(\vec{x}_0)]\kappa_{\rm host}(\vec{x}') & \text{inside simulation region} \end{cases} \tag{1}$$

where $\vec{x}_0$ is the center of the simulation region and f_Σ is supposed not to change within the simulation region. In addition the host halo is supposed to be smooth. According to the principle of linear superposition, the small deflection angles add. The lensing equation inside the simulation region can thus be written as

$$\vec{y}' = \vec{x}' - \vec{\alpha}'_{\rm halo}(\vec{x}') - \vec{\alpha}'_{\rm sub}(\vec{x}'). \tag{2}$$

It is convienient to let the center of the simulation region and the image of the smooth macro lens coincide. In the absence of substructure the image appears at $\vec{x}_0$ and the lensing equation is

$$\vec{y}_0 = \vec{x}_0 - \vec{\alpha}_{\rm host}(\vec{x}_0). \tag{3}$$

Inserting equation (3) into equation (2) and changing coordinates to $\vec{x} = \vec{x}' - \vec{x}_0$ and $\vec{y} = \vec{y}' - \vec{y}_0$ leads to

$$\vec{y} = \vec{x} - \vec{\alpha}_{\rm halo}(\vec{x} + \vec{x}_0) + \vec{\alpha}_{\rm host}(\vec{x}_0) - \vec{\alpha}_{\rm sub}(\vec{x} + \vec{x}_0). \tag{4}$$

It is desirable to have an expression whitout $\vec{\alpha}_{\rm halo}(\vec{x})$. If the surface density inside the simulation region can be approximated as constant, then the following approximation can be used

$$\kappa_{\rm host}(\vec{x}') - \kappa_{\rm halo}(\vec{x}') = f_\Sigma \kappa_{\rm host}(\vec{x}') \approx f_\Sigma \kappa_{\rm host}(\vec{x}_0). \tag{5}$$

After adding and subtracting $\alpha_{\rm host}(\vec{x}+\vec{x}_0)$ to equation (4) and rearranging the terms, the following equation is obtained

$$\begin{aligned} \vec{y} = \vec{x} + \vec{\alpha}_{\rm host}(\vec{x}+\vec{x}_0) - \vec{\alpha}_{\rm halo}(\vec{x}+\vec{x}_0) + \\ + \vec{\alpha}_{\rm host}(\vec{x}_0) - \vec{\alpha}_{\rm host}(\vec{x}+\vec{x}_0) - \vec{\alpha}_{\rm sub}(\vec{x}+\vec{x}_0). \end{aligned} \tag{6}$$

A relation between the surface density and the deflection angle can be obtained by integration of Poisson's equation $\nabla^2\psi(\vec{x}) = 2\kappa(\vec{x})$

$$\vec{\alpha}_{\rm host}(\vec{x}+\vec{x}_0) - \vec{\alpha}_{\rm halo}(\vec{x}+\vec{x}_0) = \int (\kappa_{\rm host}(\vec{x}+\vec{x}_0) - \kappa_{\rm halo}(\vec{x}+\vec{x}_0)) d\vec{x}. \tag{7}$$

The approximation (5) and equation (7) then gives

$$\vec{\alpha}_{\rm host}(\vec{x}+\vec{x}_0) - \vec{\alpha}_{\rm halo}(\vec{x}+\vec{x}_0) \approx f_\Sigma(\vec{x}_0)\kappa_{\rm host}(\vec{x}_0)\vec{x}. \tag{8}$$

If equation (8) is inserted into equation (6), the final lens equation for the simulation region is finally obtained

$$\vec{y} = \vec{x} - \vec{\alpha}_{\rm host}(\vec{x}+\vec{x}_0) + \vec{\alpha}_{\rm host}(\vec{x}_0) + f_\Sigma \kappa_{\rm host}(\vec{x}_0)\vec{x} - \vec{\alpha}_{\rm sub}(\vec{x}+\vec{x}_0). \tag{9}$$

The contribution from the substructure, $\vec{\alpha}_{\rm sub}$ is computed by adding the contribution from all sub halos.

2 Running `simsource`

The program `simsource` allows the user to simulate a jet with a posotion relative to the location of the macro image. The user can change the width and length of the jet and its angle. The resulting line is a superposition of Gaussian profiles. To run `simsource` type something like:

```
./simsource FILE=source.txt L=300 W=10 ANG=45
```

where width (`W`) and length (`L`) are given in pc and the angle (`ANG`) in degrees. Note the capitals and equal. The code will produce an image of the source given in the file `source.txt`. signs. In addition to these basic parameters the parameters of the grid in can also be set. The extent of the grid along the x-axis can be set by `XMIN` and `XMAX`, while the numbers of divisions is set by `NX`. To change the properties along the y-axis simply change `X` to `Y`. The fineness and extent of the grid must be set to match the scales we wish to resolve for the lensed image.

3 Running plotg

To view images a program called `plotg` have been written. This program uses the PGPLOT library. To make an encapsulated postscript file (`source.eps`) showing the simulated source type:

```
./plotg source.txt source.eps
```

This program plots the image in units of milli arc seconds. It takes `XMIN=xmin` `XMAX=xmax`, `YMIN=ymin`, and `YMAX=ymax` as optional parameters that allows the limits of the image to be controlled.

When using `complens` to compute the image of a lensed image a few optional flags can be used. For example the command

```
./plotg image.txt -s image.eps
```

will also plot the position of the substructure in the image plane. Using `-c` will overplot contours on the image, while `-C` will only plot contours. Flags can be combined. Using the flag `-sc`, for example, produces a plot where the location of the substructures are indicated and contours are plotted on top of the image.

4 Running complens

The program `complens` computes the compounded effect of a hosthalo and subhalos. The basic call to the program is as follows

```
./complens source.txt SUBFILE=substruct.txt image.txt
```

where `source.txt` is a file describing the source. The ouput of the program is `image.txt` describing the lensed image. The file `substruct.txt` contains a description of the substructure. Let's look at an example:

```
# SUBSTRUCT 2 -200. 0. 1.e7
```

In this example we have a $10^7\ M_\odot$ point mass located at coordinates $(-200, 0)$ relative to the macro image of the host halo and in units of pc. The general substructure syntax is

```
# SUBSTRUCT isub x y par1 ...
```

where `isub` is an integer identifying the halo model for the substructure and `par1`, `par2`, `...` are the parameters specifying the properties of the halo. Table 1 lists the syntax for different types of substructures. Note that different models take a different number of parameters. The units of the input should be $M_\odot$ (M, $m_{\rm vir}$), km s^{-1} (σ), or pc ($r_{\rm t}$). The concentration parameter, $c_{\rm vir}$, is dimensionless.

Table 1: Parameters for different subhalo models.

Halo model	`isub`	`par1`	`par2`	`par3`
SIS	0	σ		
Truncated SIS	1	σ	r_t	
Point Mass	2	M		
NFW	3	c_vir	m_vir	
Truncated NFW	4	c_vir	m_vir	r_t

Table 2: List of input parameters to `complens`. The parameters are passed using `./complens source.txt PARAMETER=value image.txt`

Input `PARAMETER`	Purpose	Default
`FSIGMA`	Fraction of substructure	0.01
`SUBFILE`	Name of substructure-file	substruct.txt
`HOSTY`	Location of image in source plane (pc)	1063
`HOSTSIG`	Velocity dispersion of host halo (km s^{-1})	240
`SOL`	Solution to host halo lens equation	1
`QUIET`	If set to 1, print messages to screen	0
`GRIDN`	Number of grid points	400
`GRIDSIZE`	Size of grid (pc)	1000
`ZSOURCE`	Redshift of the source	2.0
`ZLENS`	Redshift of the lens	0.5

A few flags can also be passed to the program. These are listed in table 3.

The hosthalo is modelled as a singular isothermal sphere (SIS). Its only parameter the velocity dispersion can be changed by `HOSTSIG=sigma`. The dimensionless lensing equation for a SIS is $y = x - \frac{x}{|x|}$, with the two solutions $x = y + 1$ and $x = y - 1$ for $0 < y < 1$. For $y > 1$ only one solution at $x = y + 1$ exists. The value of y can be changed by `HOSTY=y`. Similarly the soultion can be set to either `SOL=1` or `SOL=2`.

Figure 1 shows an example of the effect of substructure modelled as a $10^7\ M_\odot$ point mass at different locations in the image plane. The leftmost top image shows the source, which is an X. The second top image to the left shows the effect of the hosthalo. Subsequent images shows the effect of adding the substructure. The position of the substructure changes from the left to the right. Since the point mass is a very efficient lens, it leads to image splitting.

5 An example

Lets look at a realistic example. The following input simulates a small jet (4×10 pc) tilted $45°$.

```
./simsource FILE=source.txt XMIN=-10 XMAX=10 YMIN=-10 YMAX=10 L=10 W=4 ANG=45
```

Figure 1: The effect of adding a 10^7 $M_{\odot}$ point mass to a galaxy modelled as a SIS with $\sigma = 240$ km s^{-1}. The top left image shows the source. The second top image from the left shows the effect of the host halo. Subsequent shows the effect of adding the substructure (indicated by the white circle) at different locations in the image plane.

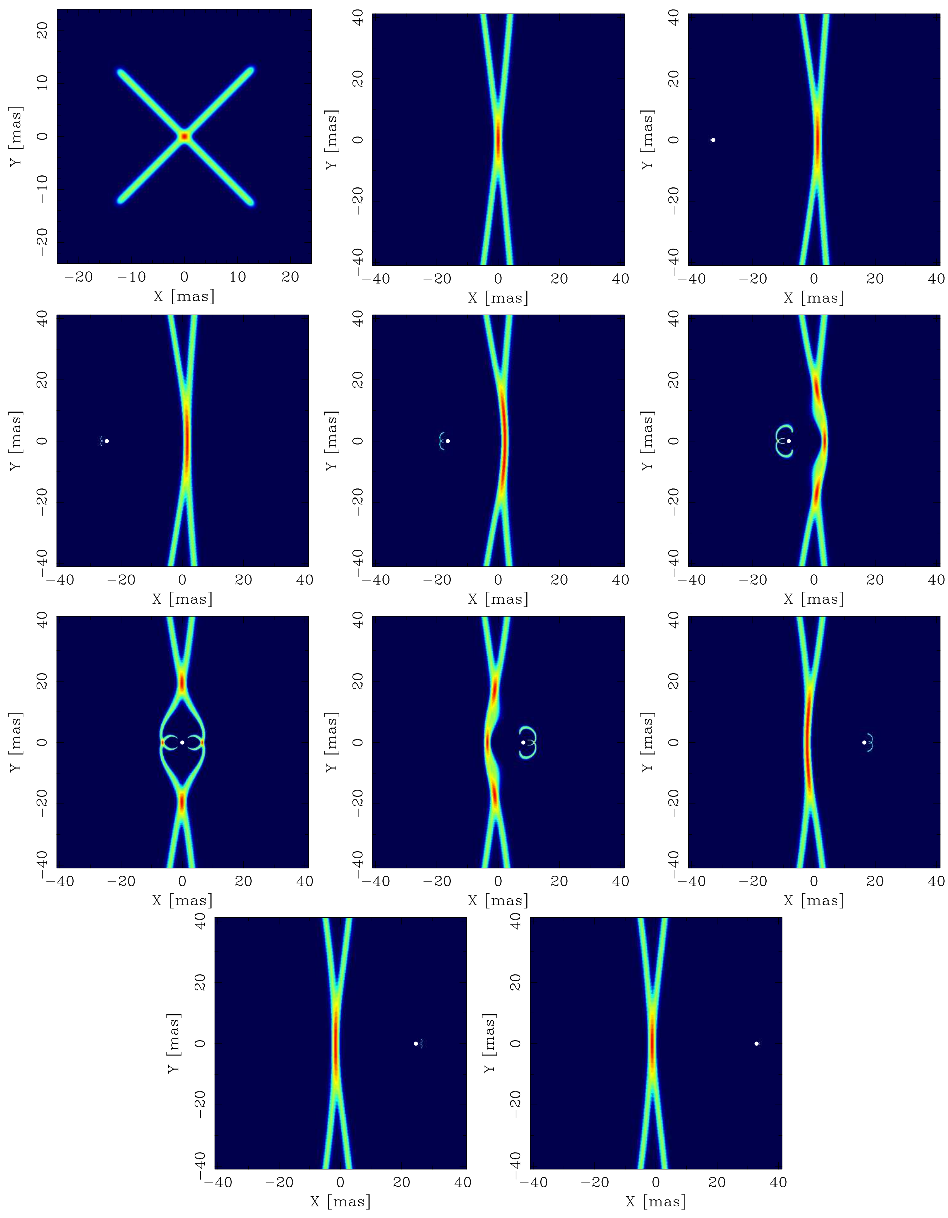
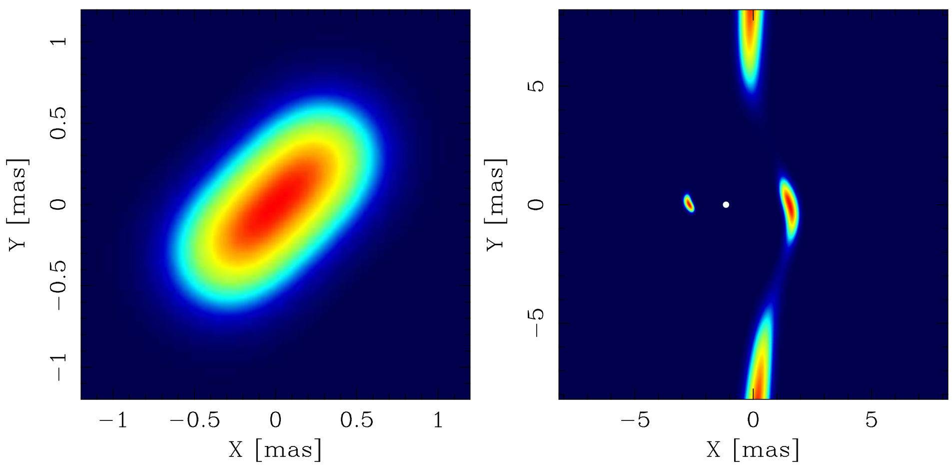


Table 3: List of flags accepted by **complens**. The flags are passed using `./complens source.txt -f image.txt`

Flag (`f`)	Purpose
`nosub`	No substructure added
`nohost`	No hosthalo deflection added
`fcomp`	Compute f_Σ from the substructure

Figure 2: Example of a small jet lensed by substructure.



To compute the lensed image we use the following input

```
./complens source.txt GRIDSIZE=100 GRIDN=1000 HOSTY=300 image.txt
```

The distance between source and lens is assumed to be 300 pc correspoding to a magnification of $\mu \simeq 30$. The size of the grid is 1000 pc and the number of gridpoints is 1000. In this case the file `substruct.txt` is a single line

```
# SUBSTRUCT 2 -7. 0. 1.e6
```

Corresponding to a $10^6\ M_\odot$ point mass located 7 pc to the left of the macro image. The source and the resulting image are shown in Figure 2

6 Image file

The `complens` program requires a file describing the source as input and delivers a another file as output. The input file (`source.txt`) looks something like this:

```
# Additional information
# SRC
```

```
 # XRANGE   -200.00000       200.00000             400
 # YRANGE   -200.00000       200.00000             400
 # ZSOURCE    2.0000000
 # Description of source:
   0.0000000
   0.0000000
   0.0000000
   ...
```

The source is described as a matrix written as a column (truncated) beneath `Desription of the source`. The additional information tells `complens` that the file desribes a source (indicated by the `SRC` keyword) the range of th grid (`XRANGE XMIN XMAX NX`) and the redshift of the source (`ZSOURCE`). `complens` takes `ZSOURCE` as an optional parameter, but it is included in the source file in order to plot the image with `plotg` in units of arc seconds. The piece of code writing the matrix of flux as a single column is

```
do i=1,NX
   do j=1,NY
      write(dsk,*) flux(i,j)
   enddo
enddo
```

The image files looks like:

```
 # IMG
 # XRANGE   -249.37344       250.62656             400
 # YRANGE   -249.37344       250.62656             400
 # ZLENS   0.50000000
 # HOSTX01    7235.2524
 # HOSTX02    0.0000000
 # SUBSTRUCT   -199.99982       0.0000000       2.0000000
0.0000000       0.0000000       10000000.       0.0000000
0.0000000
   0.0000000
   0.0000000
   0.0000000
   ...
```

Here we have the keywords `IMG` and `ZLENS` instead of `SRC` and `ZSOURCE`, respectively. There are also two keywords giving the coordinates of the macro image in the lens plane (`HOSTX01` and `HOSTX02`). This coordinates are also the center of the image grid. For each substructure there is a line with substructure information. The information is very general:

```
 # SUBSTRUCT  X Y isub sigma r m cvir mvir
```

but only the relevant parameters are non zero (in this case `isub` and `m`).

References

[1] Metcalf, R. B., & Madau, P. 2001, ApJ, 563, 9