Hands-on Workshop: How to calculate Quasar Microlensing

Joachim Wambsganss
Zentrum für Astronomie der Universität Heidelberg (ZAH/ARI)

XIth School of Cosmology
September 17 - 22, 2012 — IESC, Cargèse

ELSEVIER

Journal of Computational and Applied Mathematics 109 (1999) 353–372

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Gravitational lensing: numerical simulations with a hierarchical tree code

Joachim Wambsganss

Astrophysikalisches Institut Potsdam, An der Sternwarte 16, 14482 Potsdam, Germany

Received 12 May 1998; received in revised form 6 January 1999

Abstract

The mathematical formulation of gravitational lensing — the lens equation — is a very simple mapping $\mathscr{R}^2 \rightarrow \mathscr{R}^2$, between the lens (or sky) plane and the source plane. This approximation assumes that all the deflecting matter is in one plane. In this case the deflection angle α is just the sum over all mass elements in the lens plane. For certain problems — like the determination of the magnification of sources over a large number of source positions (up to 10^{10}) for very many lenses (up to 10^6) — straightforward techniques for the determination of the deflection angle are far too slow. We implemented an algorithm that includes a two-dimensional tree-code plus a multipole expansion in order to make such microlensing simulations "inexpensive". Subsequently we modified this algorithm such that it could be applied to a three-dimensional mass distribution that fills the universe (approximated by many lens planes), in order to determine the imaging properties of cosmological lens simulations. Here we describe the techniques and the numerical methods, and we mention a few astrophysical results obtained with these methods.

Efficient Inverse Ray Shooting: A Tree-Code Approach

(Wambsganss 1990, 1999)

Deflection angle for n lenses:

$$\tilde{\boldsymbol{\alpha}}_i = \sum_{j=1}^{n} \tilde{\boldsymbol{\alpha}}_{ji} = \frac{4G}{c^2} \sum_{j=1}^{n} M_j \frac{\boldsymbol{r}_{ij}}{r_{ij}^2}$$

Number of computational operations:

$$N_{\rm total} = N_{\rm op} \times N_{\rm pix} \times N_{\rm av} \times N_* \simeq 10 \times 2500^2 \times 500 \times 10^6 \approx 3 \times 10^{16}$$

Calculation of deflection angle for N* lenses split into two parts:

$$\tilde{\boldsymbol{\alpha}} = \sum_{i=1}^{N_*} \tilde{\boldsymbol{\alpha}}_i \approx \sum_{j=1}^{N_{\rm L}} \tilde{\boldsymbol{\alpha}}_j + \sum_{k=1}^{N_{\rm C}} \tilde{\boldsymbol{\alpha}}_k =: \tilde{\boldsymbol{\alpha}}_{\rm L} + \tilde{\boldsymbol{\alpha}}_{\rm C}.$$

The N's denote the following:

- N_* is the number of all lenses,
- $N_{\rm L}$ the number of lenses to be included directly,
- $N_{\rm C}$ the number of cells (= pseudo-lenses) to be included.

Efficient Inverse Ray Shooting: A Tree-Code Approach

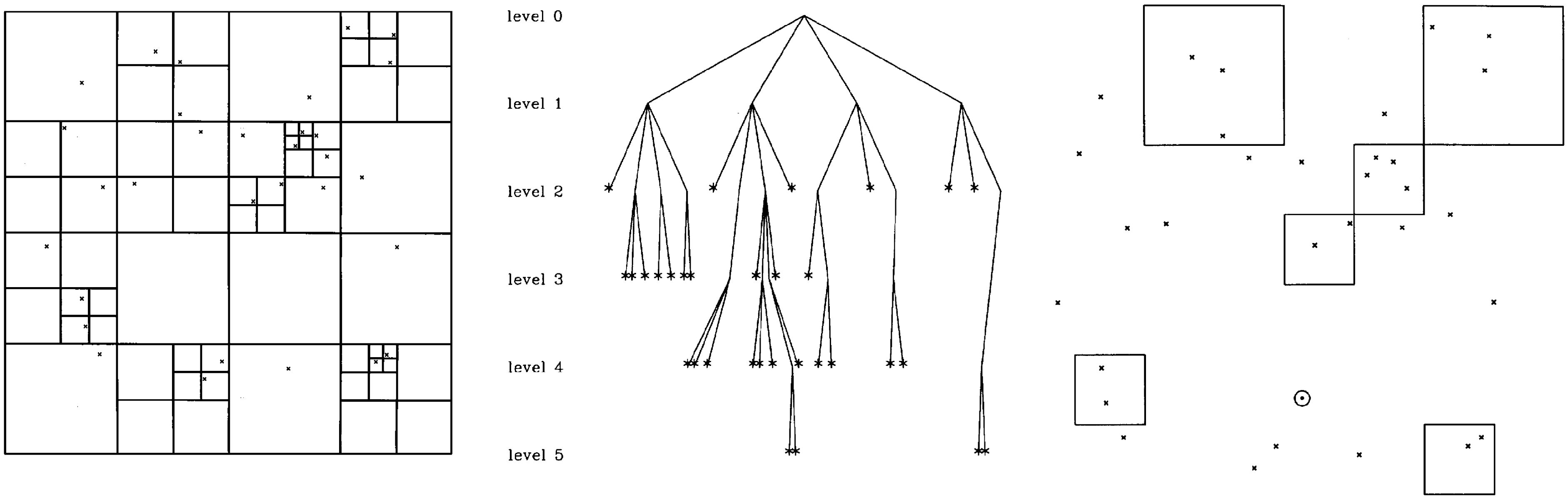
(Wambsganss 1990, 1999)

Lens Equation:

$$\boldsymbol{y} = \begin{pmatrix} 1-\gamma & 0 \\ 0 & 1+\gamma \end{pmatrix} \boldsymbol{x} - \sigma_c \boldsymbol{x} - \sum_{i=1}^{N_*} \frac{m_i(\boldsymbol{x}-\boldsymbol{x}_i)}{(\boldsymbol{x}-\boldsymbol{x}_i)^2}$$

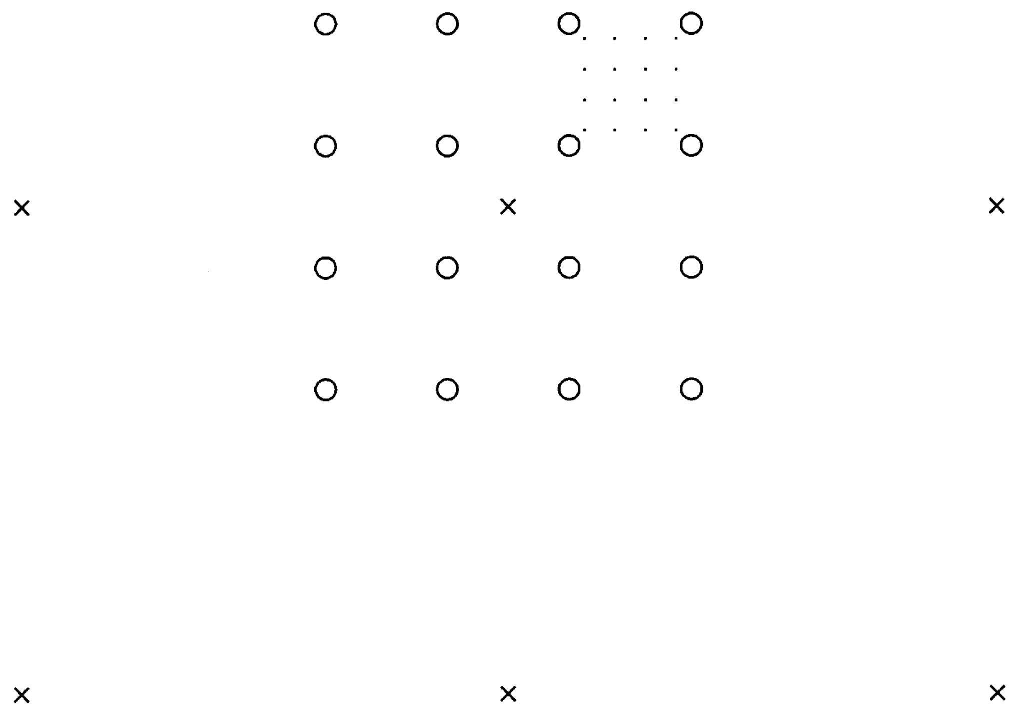
Tree code approach:

$$\tilde{\boldsymbol{\alpha}} = \sum_{i=1}^{N_*} \tilde{\boldsymbol{\alpha}}_i \approx \sum_{j=1}^{N_\mathrm{L}} \tilde{\boldsymbol{\alpha}}_j + \sum_{k=1}^{N_\mathrm{C}} \tilde{\boldsymbol{\alpha}}_k =: \tilde{\boldsymbol{\alpha}}_\mathrm{L} + \tilde{\boldsymbol{\alpha}}_\mathrm{C}.$$



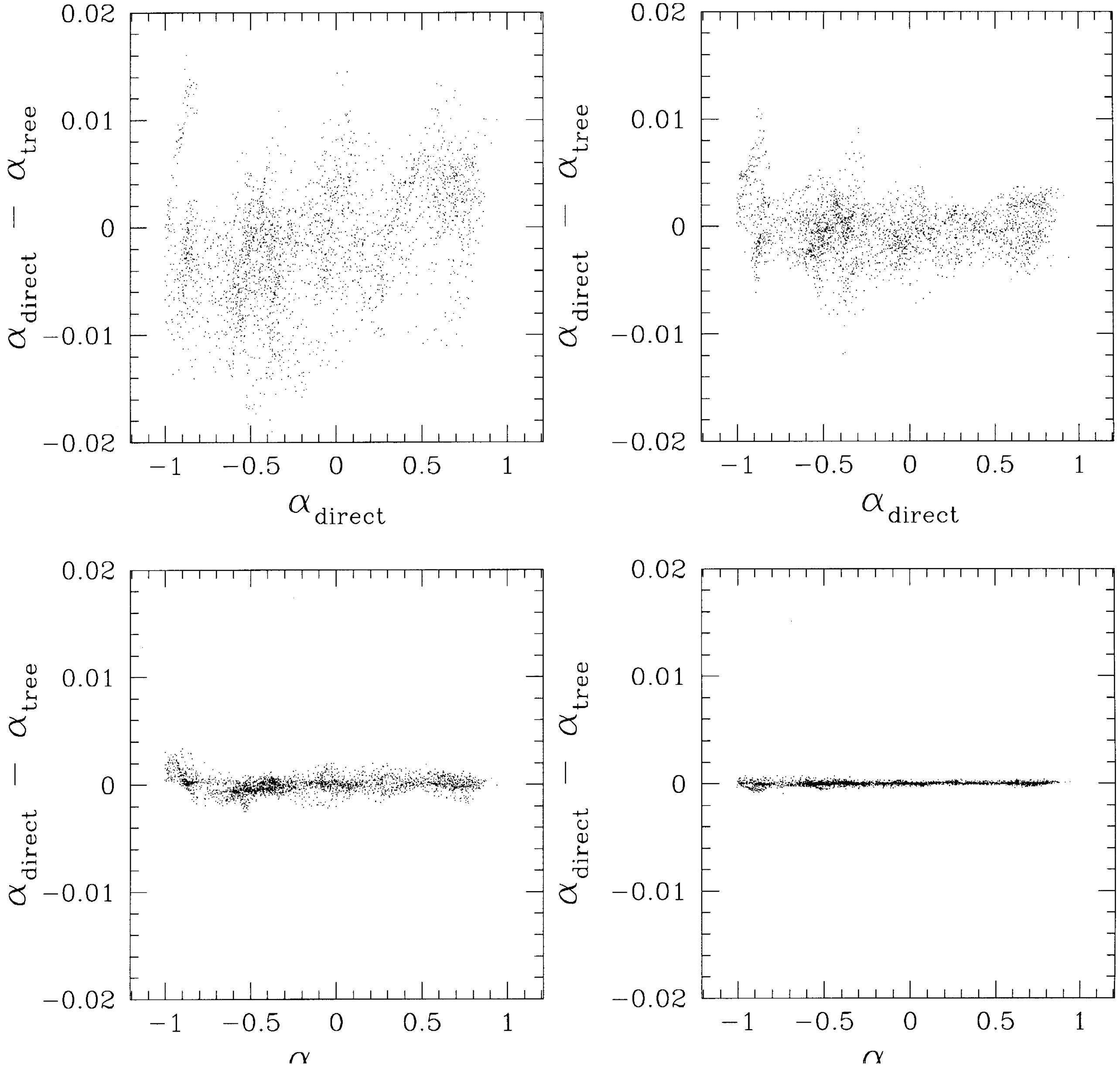
Efficient Inverse Ray Shooting: A Tree-Code Approach

(Wambsganss 1990, 1999)



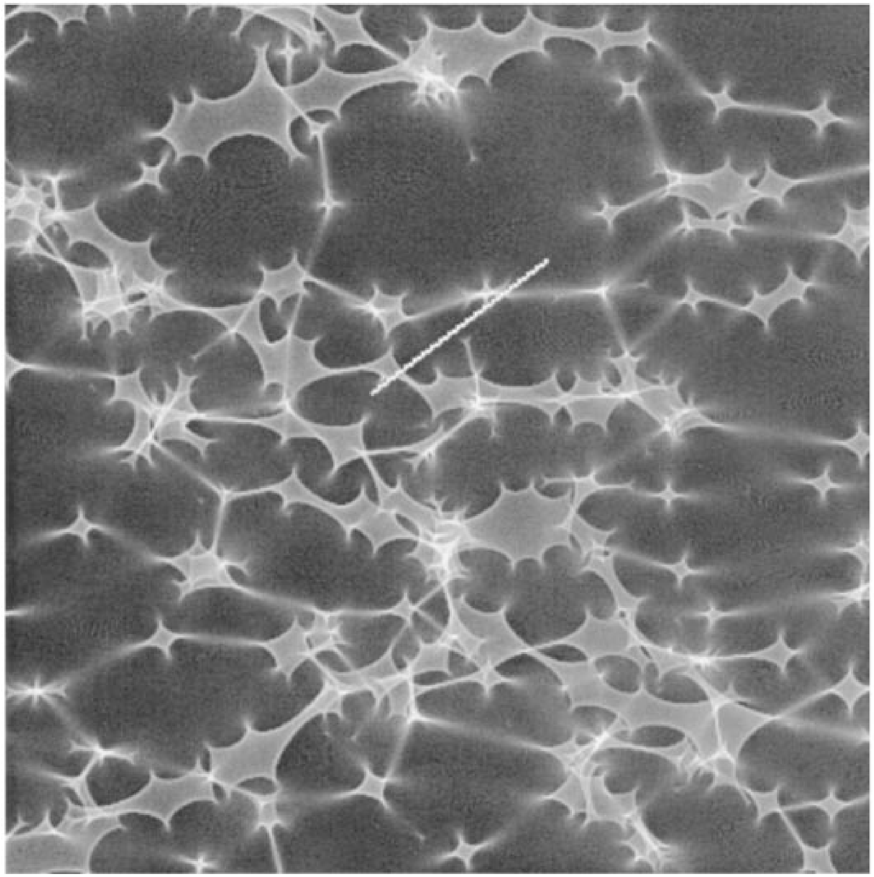
Efficient Inverse Ray Shooting: A Tree-Code Approach

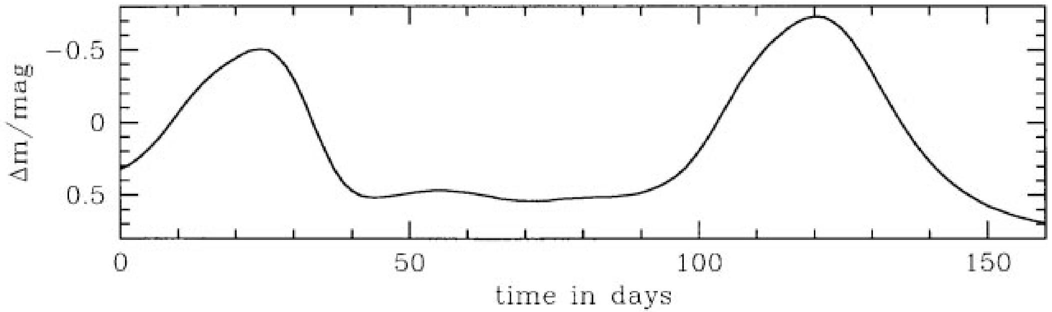
(Wambsganss 1990, 1999)



Efficient Inverse Ray Shooting: A Tree-Code Approach

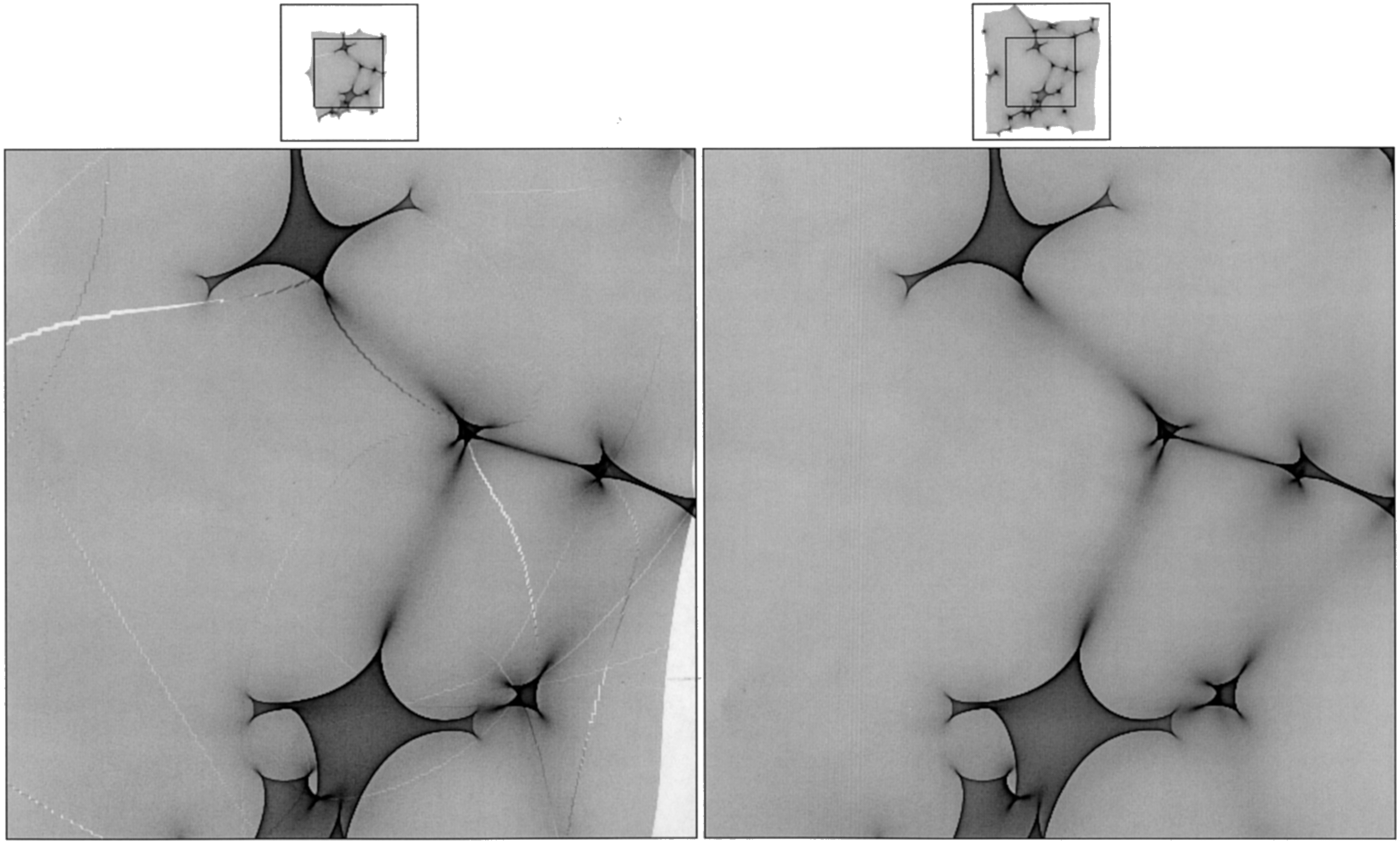
(Wambsganss 1990, 1999)





Efficient Inverse Ray Shooting: A Tree-Code Approach

(Wambsganss 1990, 1999)



Quasar Microlensing: How to do simulations!

1) copy file Wambsganss-MicrolensingCode-Cargese-2012.tar to your disk

2) untar this file ... should produce directory:
Wambsganss-MicrolensingCode-Cargese-2012

Wambsganss-MicrolensingCode-Cargese-2012	10,3 MB
_pdf	1,7 MB
JCAM_jkw.pdf	1,6 MB
ML-code-wambsganss.pdf	97 KB
cfitsio	8,4 MB
cputime.c	216 bytes
dat.111	19 KB
detko.f	2 KB
dis_1000.pro	500 bytes
dis_light.pro	1 KB
input	996 bytes
input.f	8 KB
jobnum	4 bytes
lightcurve.f	7 KB
ls-all	214 bytes
main.f	11 KB
Makefile	2 KB

Quasar Microlensing: How to do simulations!

1) copy file   Wambsganss-MicrolensingCode-Cargese-2012.tar   to your disk

2) untar this file ... should produce directory:

Wambsganss-MicrolensingCode-Cargese-2012

3) cd cfitsio

4) ./configure

5)   make   (still in directory cfitsio)

6)   ..   (now in directory Wambsganss-MicrolensingCode-Cargese-2012)

7)   make   (should produce executable   “microlens”   )

8)   run the program by typing: ./microlens

Quasar Microlensing: How to do simulations!

1) copy file Wambsganss-MicrolensingCode-Cargese-2012.tar to your disk

2) untar this file ... should produce directory:

Wambsganss-MicrolensingCode-Cargese-2012

3) cd cfitsio

4) ./configure

5) make (still in directory cfitsio)

6) .. (now in directory Wambsganss-MicrolensingCode-Cargese-2012)

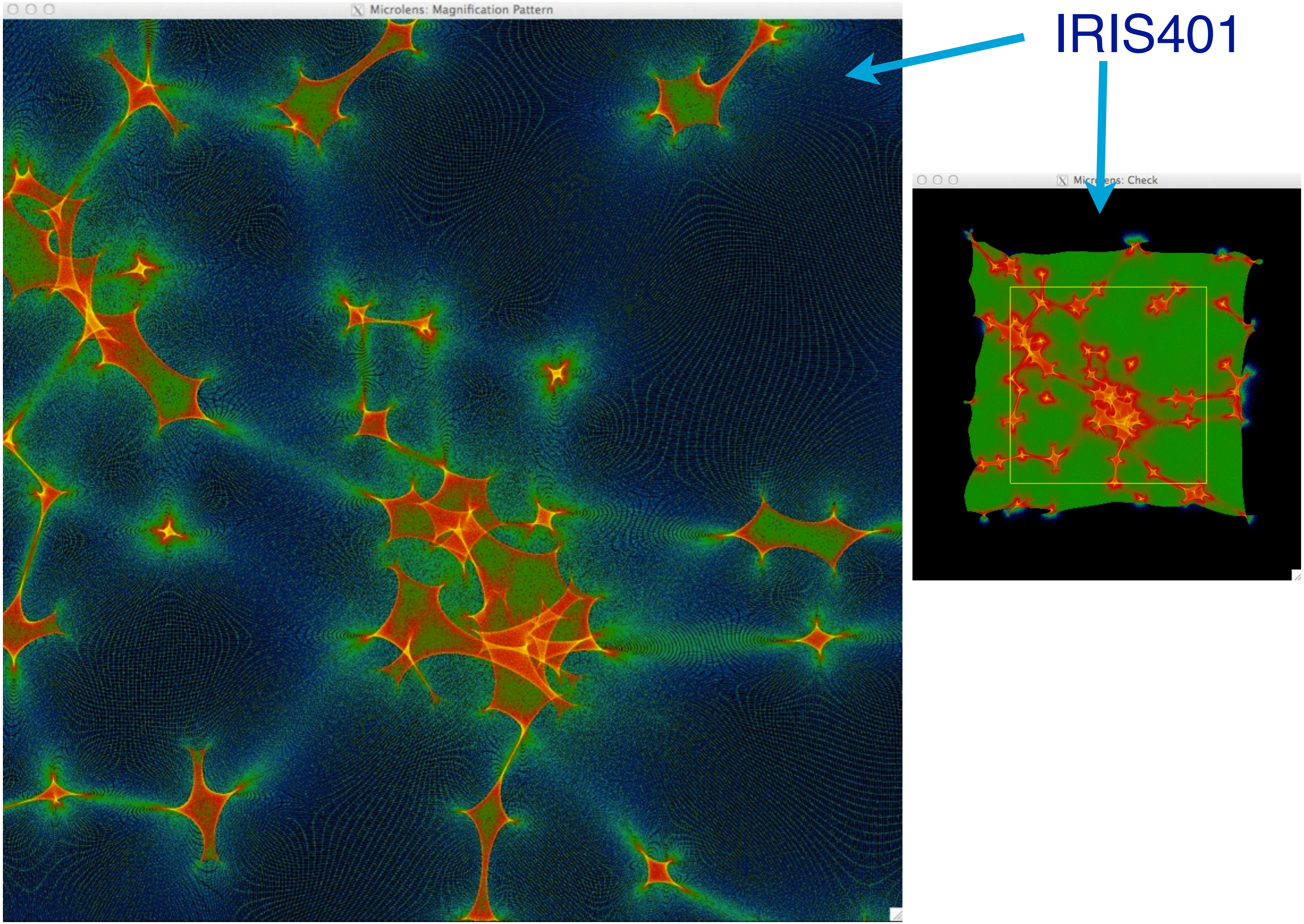
7) make (should produce executable "microlens")

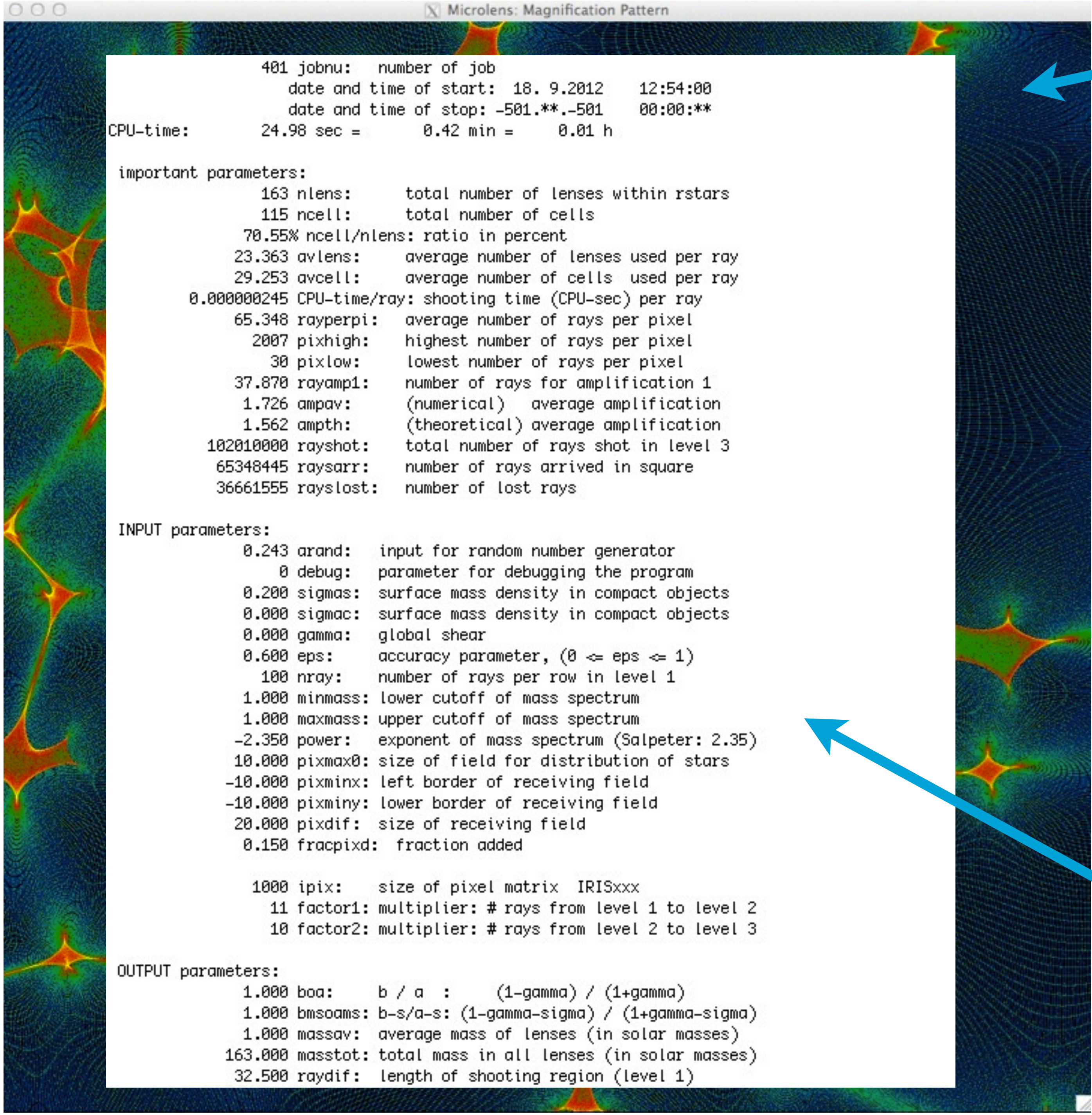
8) run the program by typing: ./microlens

9) newly produced files:

dat.401	log-file
IRIS401	magnification pattern (unformatted)
IRIS401.fits	magnification pattern (FITS format)

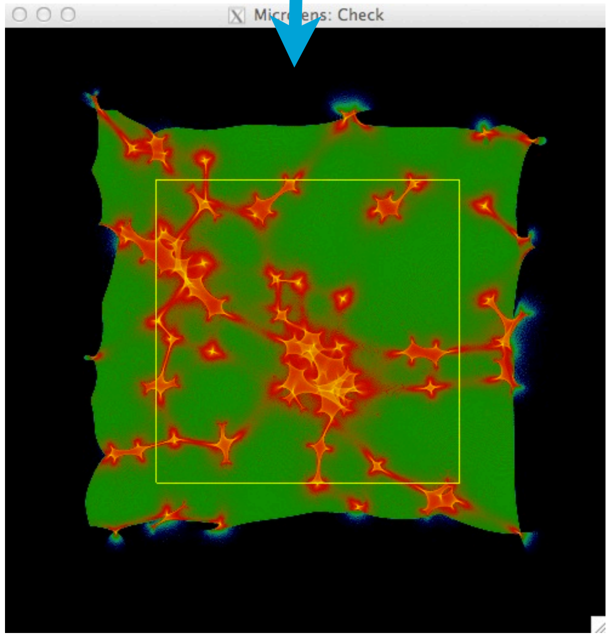
10) display magnification pattern with IDL: ./rnew dis_1000


Microlens: Magnification Pattern
IRIS401



```
Microlens: Magnification Pattern

                  401 jobnu:   number of job
                     date and time of start:  18. 9.2012    12:54:00
                     date and time of stop: -501.**.-501    00:00:**
CPU-time:         24.98 sec =       0.42 min =     0.01 h

 important parameters:
                  163 nlens:      total number of lenses within rstars
                  115 ncell:      total number of cells
                70.55% ncell/nlens: ratio in percent
               23.363 avlens:     average number of lenses used per ray
               29.253 avcell:     average number of cells  used per ray
          0.000000245 CPU-time/ray: shooting time (CPU-sec) per ray
               65.348 rayperpi:   average number of rays per pixel
                 2007 pixhigh:    highest number of rays per pixel
                   30 pixlow:     lowest number of rays per pixel
               37.870 rayamp1:    number of rays for amplification 1
                1.726 ampav:      (numerical)   average amplification
                1.562 ampth:      (theoretical) average amplification
            102010000 rayshot:    total number of rays shot in level 3
             65348445 raysarr:    number of rays arrived in square
             36661555 rayslost:   number of lost rays

 INPUT parameters:
                0.243 arand:   input for random number generator
                    0 debug:   parameter for debugging the program
                0.200 sigmas:  surface mass density in compact objects
                0.000 sigmac:  surface mass density in compact objects
                0.000 gamma:   global shear
                0.600 eps:     accuracy parameter, (0 <= eps <= 1)
                  100 nray:    number of rays per row in level 1
                1.000 minmass: lower cutoff of mass spectrum
                1.000 maxmass: upper cutoff of mass spectrum
               -2.350 power:   exponent of mass spectrum (Salpeter: 2.35)
               10.000 pixmax0: size of field for distribution of stars
              -10.000 pixminx: left border of receiving field
              -10.000 pixminy: lower border of receiving field
               20.000 pixdif:  size of receiving field
                0.150 fracpixd:  fraction added

                 1000 ipix:    size of pixel matrix  IRISxxx
                   11 factor1: multiplier: # rays from level 1 to level 2
                   10 factor2: multiplier: # rays from level 2 to level 3

 OUTPUT parameters:
                1.000 boa:     b / a  :     (1-gamma) / (1+gamma)
                1.000 bmsoams: b-s/a-s: (1-gamma-sigma) / (1+gamma-sigma)
                1.000 massav:  average mass of lenses (in solar masses)
              163.000 masstot: total mass in all lenses (in solar masses)
               32.500 raydif:  length of shooting region (level 1)
```

IRIS401



dat.401

Quasar Microlensing: How to do simulations!

11) in order to extract a lightcurve: compile lightcurve.f

(I use: gfortran lightcurve -o lightcurve)

12) run lightcurve routine:

./lightcurve

13) output produced:

out_line (lightcurve data, pixels convolved with source profile)

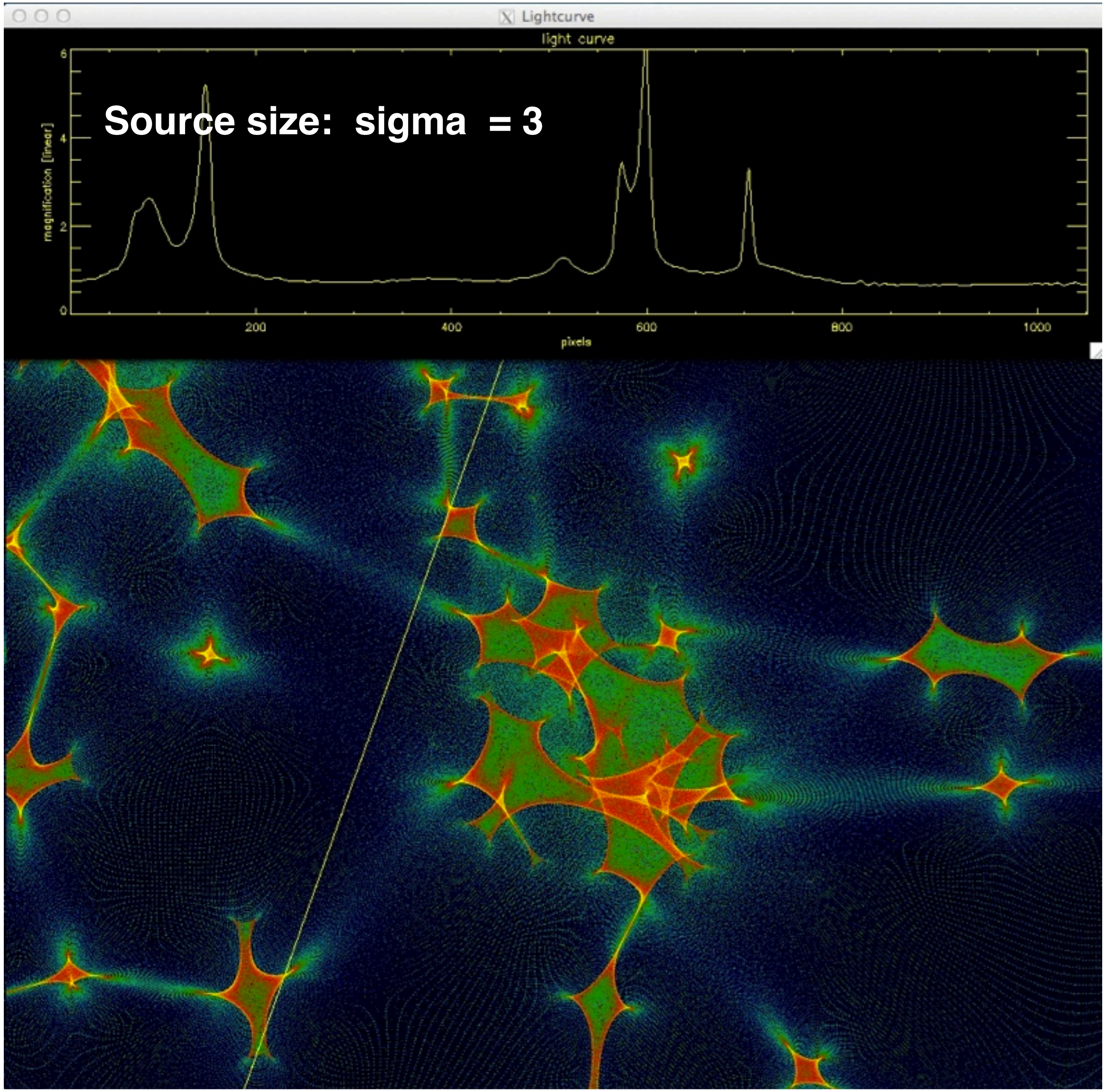
IRIS401-track (magnification pattern WITH track marked)

14) display magnification pattern with track AND lightcurve:

dis_light

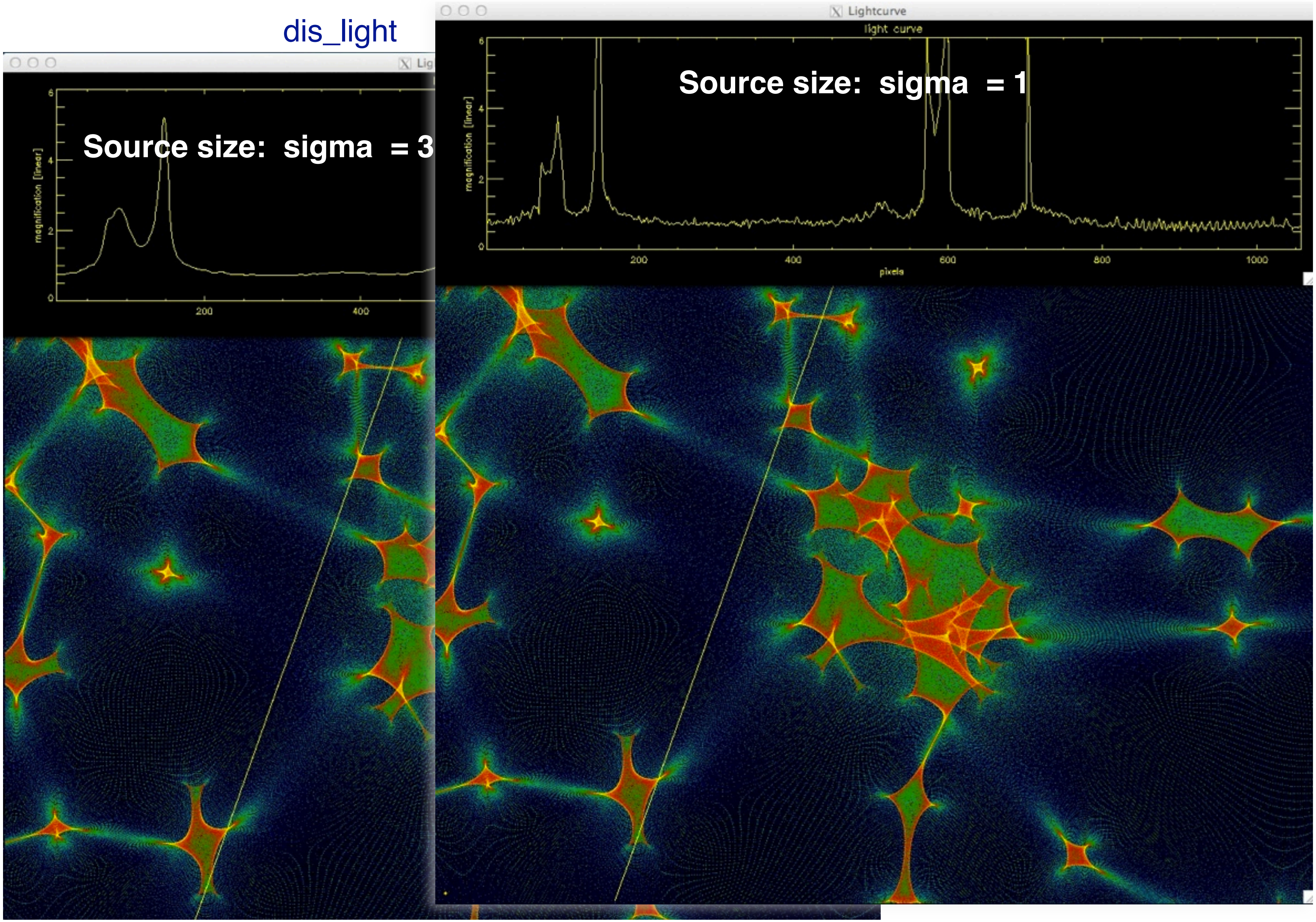
14) display magnification pattern with track AND lightcurve:

dis_light



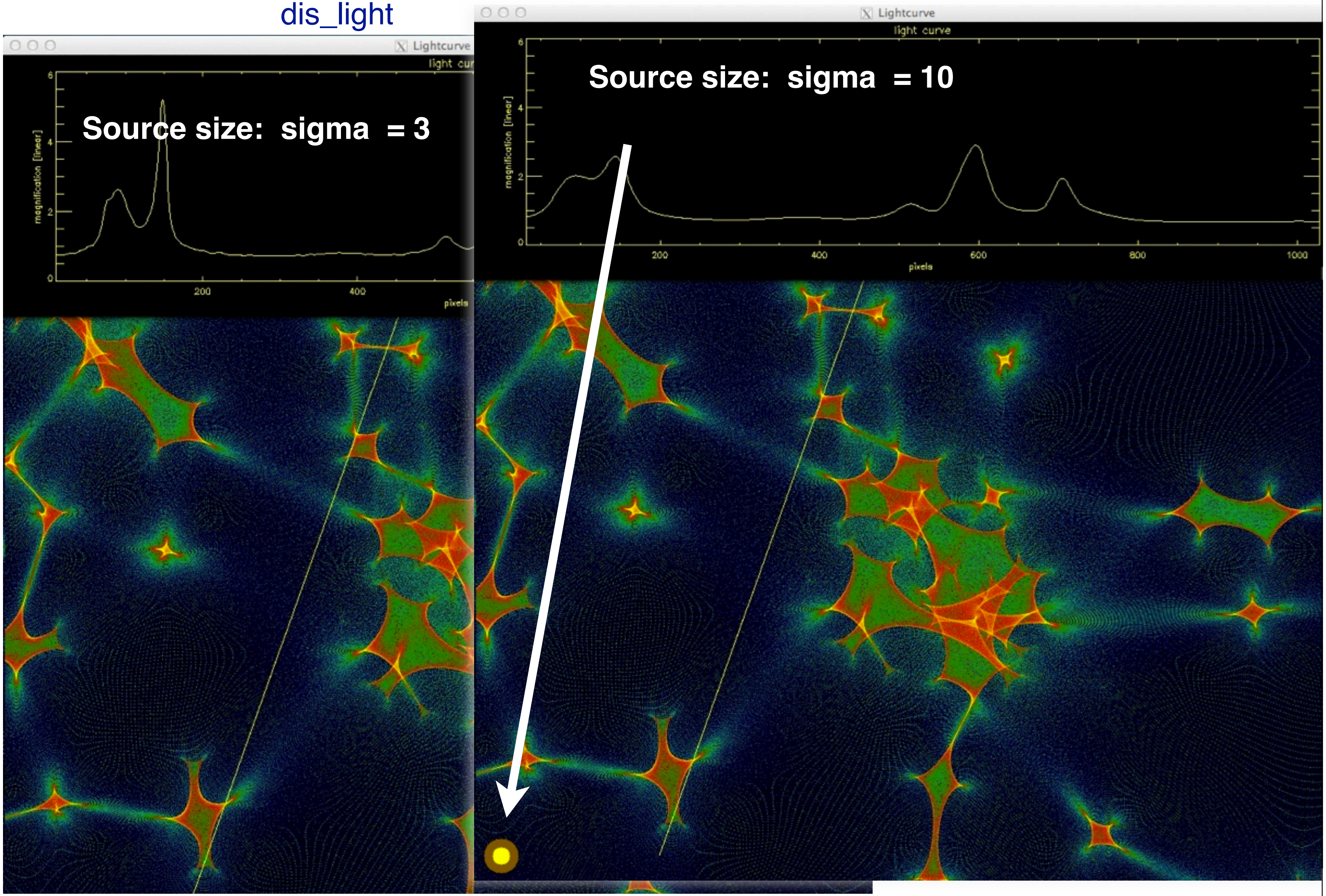
14) display magnification pattern with track AND lightcurve:

dis_light



14) display magnification pattern with track AND lightcurve:

dis_light



Quasar Microlensing: How to do simulations!

11) in order to extract a lightcurve: compile lightcurve.f

(I use: gfortran lightcurve -o lightcurve)

12) run lightcurve routine:

./lightcurve

13) output produced:

out_line (lightcurve data, pixels convolved with source profile)

IRIS401-track (magnification pattern WITH track marked)

14) display magnification pattern with track AND lightcurve:

dis_light

15) modify input file for microlens:

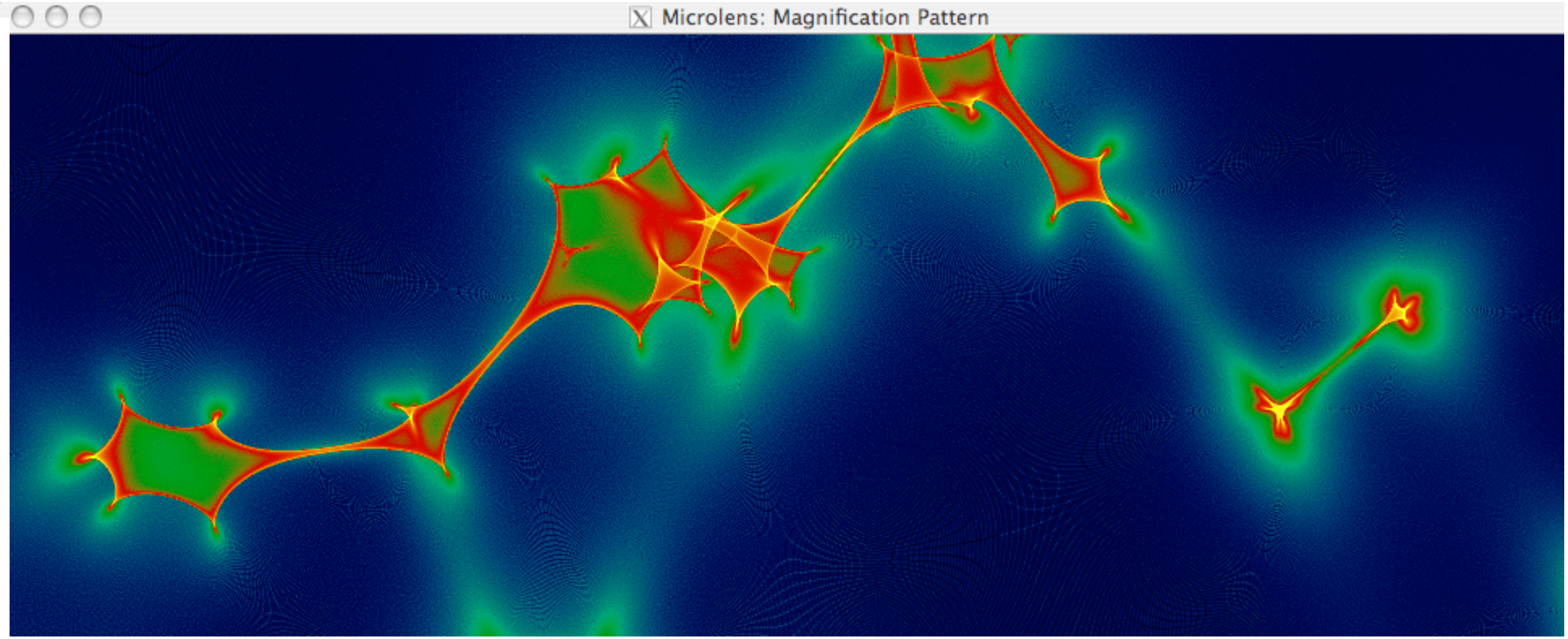
Quasar Microlensing: How to do simulations!

surface mass density kappas (or sigma)

15) modify input file for microlens: replace nray = "20" by "100" ... and run again!

```
   0.243          drand   (random seed number;   0.345
       0          debug   (parameter: 0 - standard run)
   0.200          kappas  (matter in stellar objects; 0.000: 1 lens; 0.001: 2 lenses)
   0.000          kappac  (continously distributed matter)
   0.000          gamma   (external shear)
   0.600          eps     (accuracy parameter)
      20          nray    (# of rays per row in level 1)
1.000000          minmass (lower cut off in mass spectrum)
1.000000          maxmass (upper cut off in mass spectrum)
  -2.350          power   (index in mass spectrum)
  10.000          pixmax0   (radius of field (ex.: 10.000)
 -10.000          pixminx   (ex.: -2.000)
 -10.000          pixminy   (ex.: -2.000)
  20.000          pixdif    (ex.:  4.000)
   0.150          fracpixd  (ex.:  0.020, 0.050, 0.250 ....,  2.000)
       0          iwrite (iwrite >0: write these values on fort.99)
*******************************************************************
```

Quasar Microlensing: **Now YOU do simulations!**



Deal:

1) You can use the code “microlens” freely
2) On first scientific paper using “microlens”: J.W offered co-authorship
3) This (and subsequent) papers cite:

    Wambsganss, J.: 1999, Journ. Comp. Appl. Math. 109, 353
    Wambsganss, J.: 1990, PhD Thesis, Ludwig-Maximilians-University Munich (also available as MPA report 550)