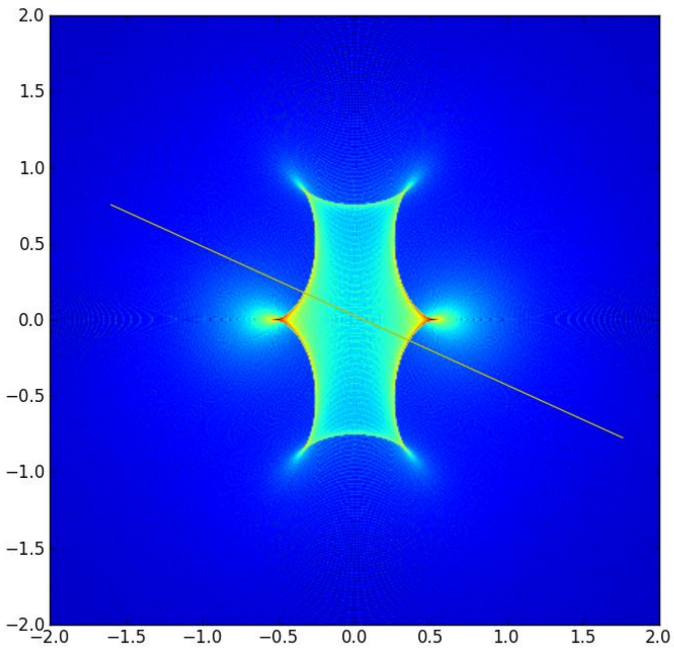
Inverse Ray Shooting Tutorial (III)

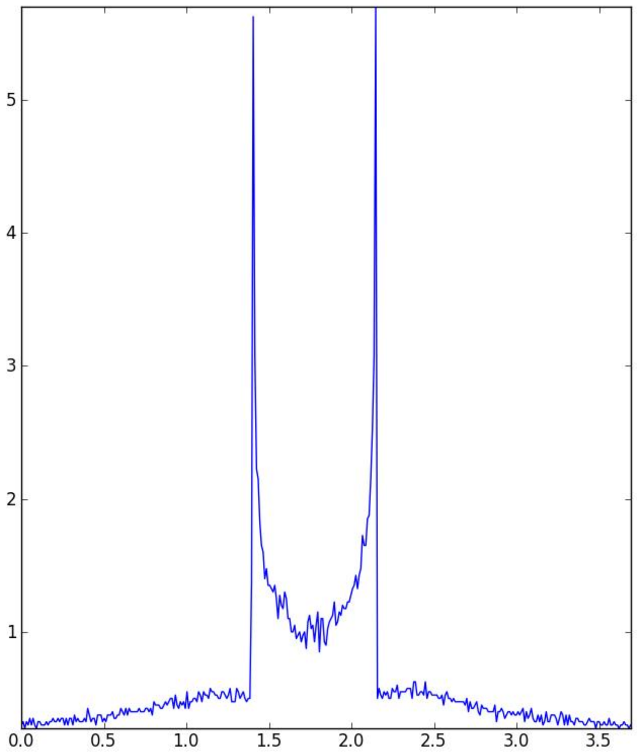
Jorge Jiménez Vicente
Dpto. Física Teórica y del Cosmos
Universidad de Granada
Spain

Session III

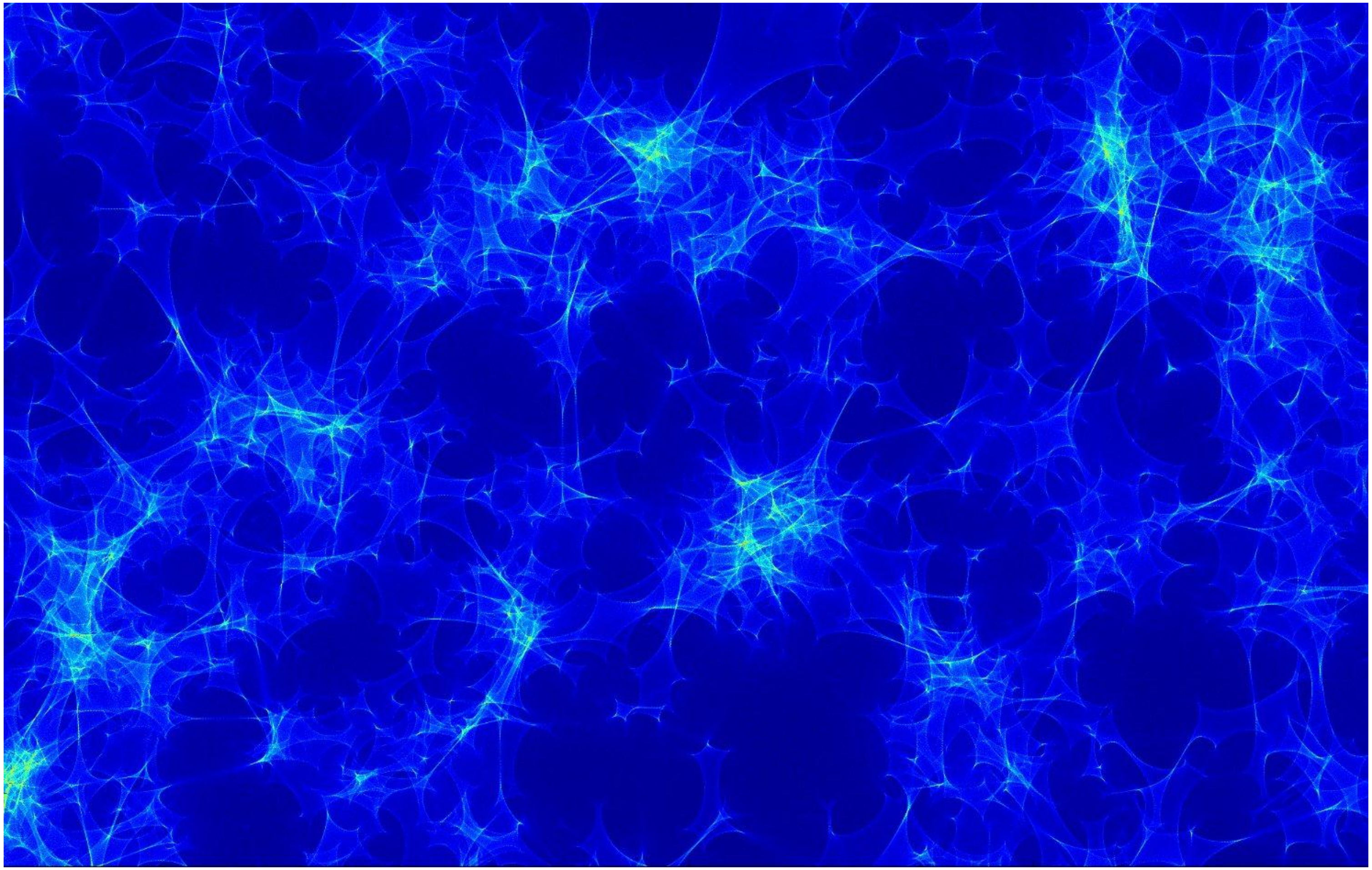
- Light Curves
- Quasar microlensing maps
- Source size effects
- Beyond simple IRS: Treecodes & IPM

Today's goal #1

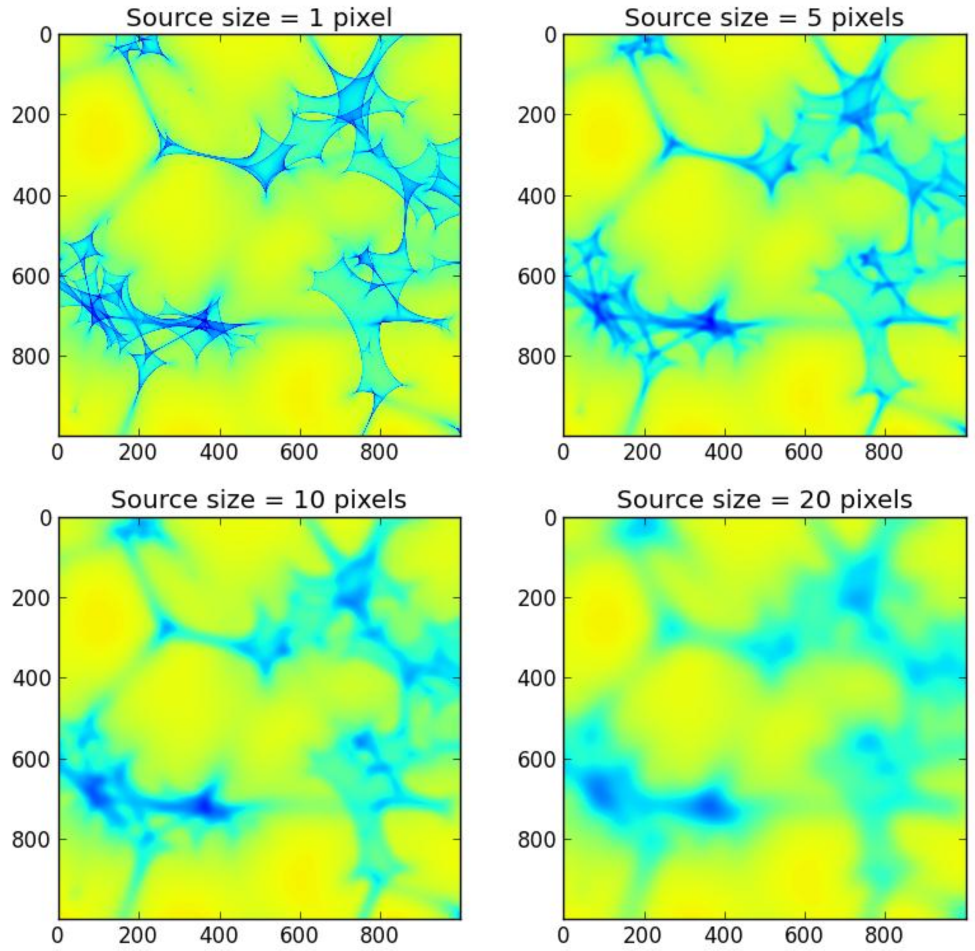




Today's goal #2



Today's goal #3



Light curves

- The magnication of a gravitational lens system may change with time because:
  - Source moves
  - Lens(es) move
  - Both move

Light Curves in binary systems

- Let's try to reproduce the light curves of some of the microlensing events from the MACHO Project → Alcock et al. 2000, ApJ, 541,27.
- Have a look at:
  - The light curve
  - The source plane configuration
- Try to reproduce it:
  - Generate the magnication map as we did yesterday
  - Try first to produce light curves in the horizontal and vertical direction for a whole row or column.
  - Try to produce light curves in any direction and of any length.
  - You may find useful the function **auxfun.prof** to calculate the profile (or have a look at it to help you doing it yourself)

Quasar microlensing

- The lens equation takes the form:

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

- We need to randomly distribute $N_* = \kappa_* A_x / \pi \langle M \rangle$
- How large should Ax be?
- Several recipes:
  - Ellipse
  - Circle
  - Square → Max(1.5*yl/(1-κ-γ) ,1.5*yl/(1-κ+γ))

Quasar Microlensing Mag Maps

- Set parameters of the system:
  - Size of magnification map → yl,ny
  - Lensing parameters: $\kappa,\gamma,\alpha \leftrightarrow \kappa_*, \kappa_s ,\gamma$
  - Calculate sizes of region to distribute stars and shooting region
- Prepare the stars by randomly locating them as described before.
- Shoot your rays on a per row basis
- Collect them at the source plane

ugr Universidad de Granada

Size effects

- When source is larger than a pixel in our magnication map, different parts of the source suffers different magnications → Effectively it is like a blured magnification map.
- You can:
  - Put your finite source at many places within the map
  - Convolve the magnification map with the source profile → Much more convenient.

Source Profile

- Mortonson & Schechter 2005.→Size is everything

  The Astrophysical Journal, 628:594-603

- Statistical properties of microlensing for different source profiles are mostly determined by source size (half light radius).
- It is relatively safe to use a gaussian profile for the source as a representative profile.

Size effects (II)

- Convolve the magnification map with a gaussian source of a given size.
- We can use the **auxfun.gconv** function to do it on our maps (have a look at it to know how it works)
- Compare light curves
- Compare histograms
- What can you see?

Size effects… and more

- Smaller sources suffer more microlensing → Microlensing can help us to estimate the size of the source
- Light curves/Statistical properties of microlensing contain (combined) information on not only source size but:
  - Mass fraction of the lens in stars/compact objects
  - Velocities (mainly transverse velocity of lens)
  - Temperature structure of the source (chromaticity) if we have wavelength resolved microlensing.
  - Etc….
  - See for example:
    - Kochanek, C. S. 2004, ApJ, 605, 58
    - Mediavilla et al. 2009, ApJ 706, Issue 2, pp. 1451-1462
    - Jiménez-Vicente et al. 2012, ApJ 751, Issue 2, article id. 106
    - Muñoz et al. 2012, ApJ 742, Issue 2, article id. 67

Beyond simple IRS

Number of computational operations:

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

- Three ways to improve efficiency:
  - Reduce last factor → Treecodes
  - Reduce N_{av} → IPM
  - Use faster hadware

TreeCodes & N-Body calculations

- Take benefit from the fact that gravitational potential of far lenses is smooth.
- Treat far lenses as pseudo-particles characterized by their total mass (and maybe higher multipolar moments.)
- J. Barnes and P. Hut (December 1986) used it for N-Body calculations.

  "A hierarchical O(*N* log *N*) force-calculation algorithm".

  Nature **324**: 446–449

Barnes-Hut Algorithm

- Divide space (plane) into a tree of cells.
- Subdivide every cell until all end up with 1 or 0 particles.
- There are cells/pseudoparticles at different levels of the tree.
- Forces of nearby particles are included directly
- Forces of far away particles are calculated via larger cells/pseudoparticles.
- Which size of cell?
  - Use parameter θ=s/d (s=size of cell, d=distance)
  - If θ is larger than some value (~0.5) use individual particles
  - Otherwise it is safe to use pseudoparticles/cells.

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

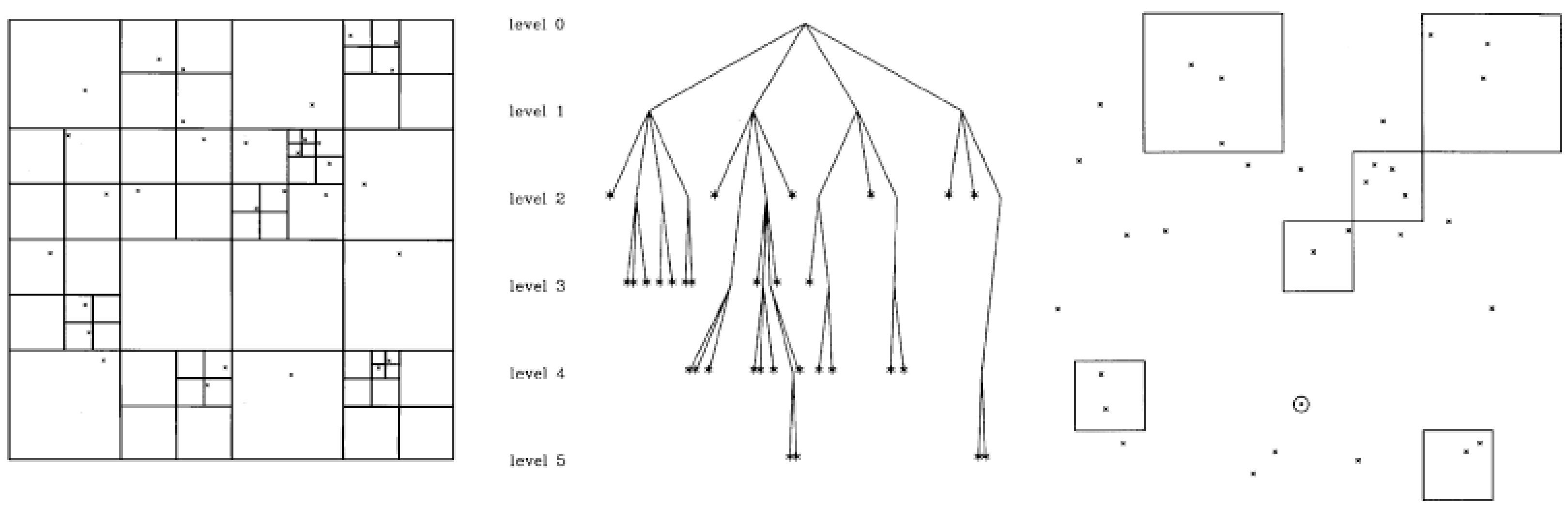
(Wambsganss 1990, 1999)

Lens Equation:

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

Tree code approach:

$$\tilde{\alpha} = \sum_{i=1}^{N_*} \tilde{\alpha}_i \approx \sum_{j=1}^{N_L} \tilde{\alpha}_j + \sum_{k=1}^{N_C} \tilde{\alpha}_k =: \tilde{\alpha}_L + \tilde{\alpha}_C.$$



When are treecodes best?

- For problems with many bodies the time invested on tree construction, pseudocell multipolar expansion calculation, … pays off at the end of the day.
- For problems with not too many particles, treecodes do not pay off.
- How many are "too many"?
  - Not a precise number but around a **few thousand.**

Inverse Polygon Mapping

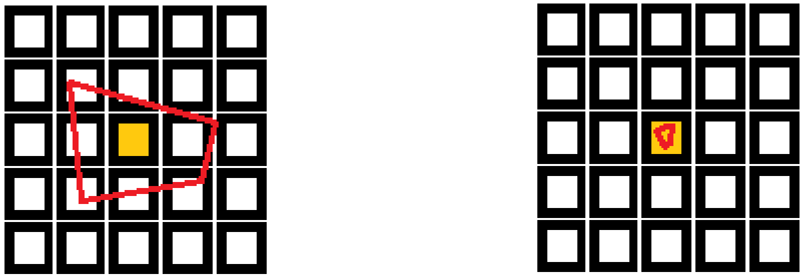
- Mediavilla et al. (2006, 2011)
- Remember "The Zen of Python":

  Sparse is better than dense.

- In plain IRS, the whole area transported backwards by a ray to the source plane is assigned to a single pixel of the magnification map. ← Quite inefficient
- By shooting many rays, the area transported by each ray is small, and so we can keep the error more or less under control.
- Why not do it in a more clever way?

IPM (2)

- We can apportion the area of the cell among the corresponding pixels in the source plane.
- This way, we do not need to throw so many rays.



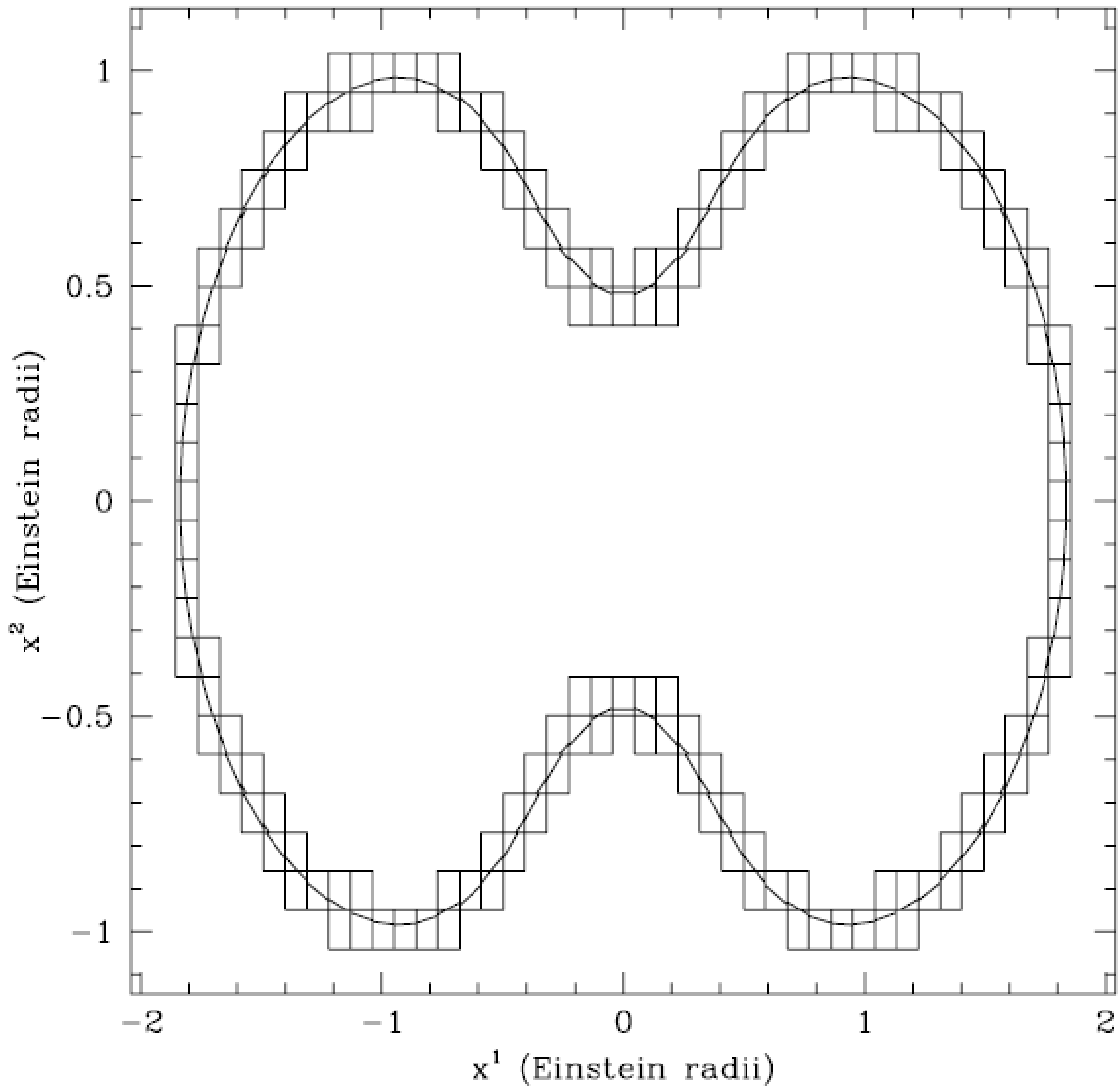


Figure 1. Critical cells and straight line approximation to the critical curve corresponding to a binary lens. The critical cells are 0.1 Einstein radius in size. The critical curve divides each critical cell into two non-critical subcells.

Non Critical cells

Transformed Non critical cells

THE ASTROPHYSICAL JOURNAL, 741:42 (8pp), 2011 November 1

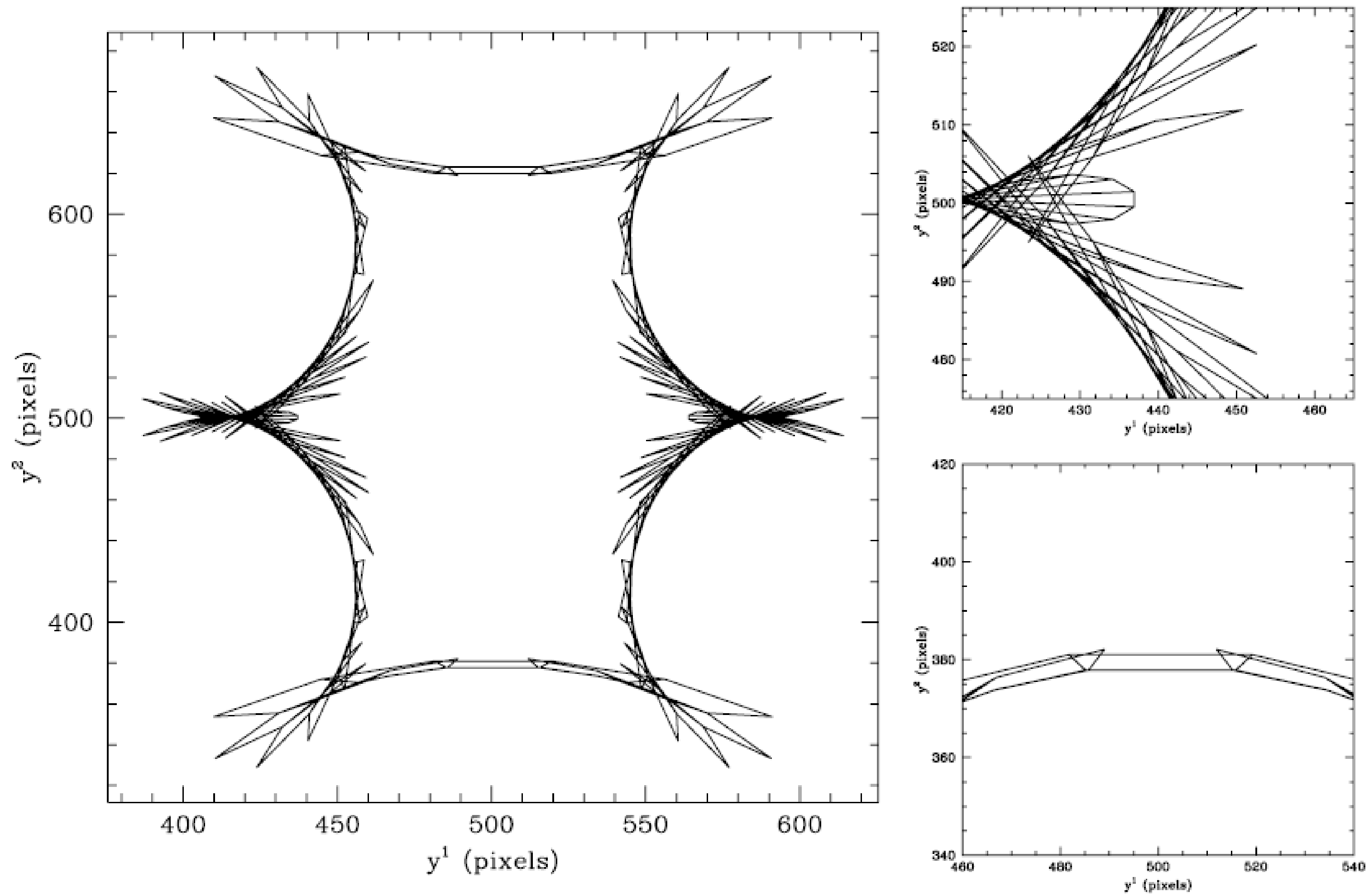


Figure 2. Large panel: inverse lens mapping of the non-critical subcells of Figure 1. Bottom small panel: detail of a caustic fold (notice the collapse of cells in the direction perpendicular to the caustic). Top small panel: detail of a caustic cusp (notice the distortions of the cells and their collapse in the direction perpendicular to the caustics that join at the cusp). See the text.

A critical cell

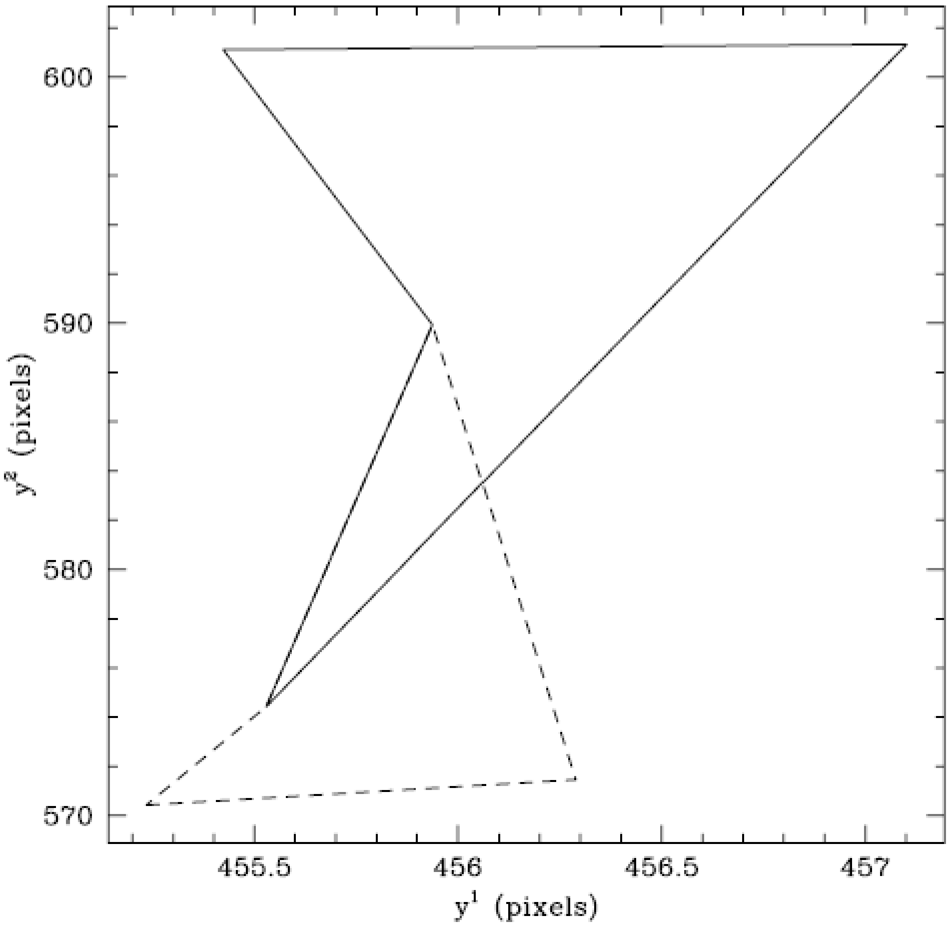
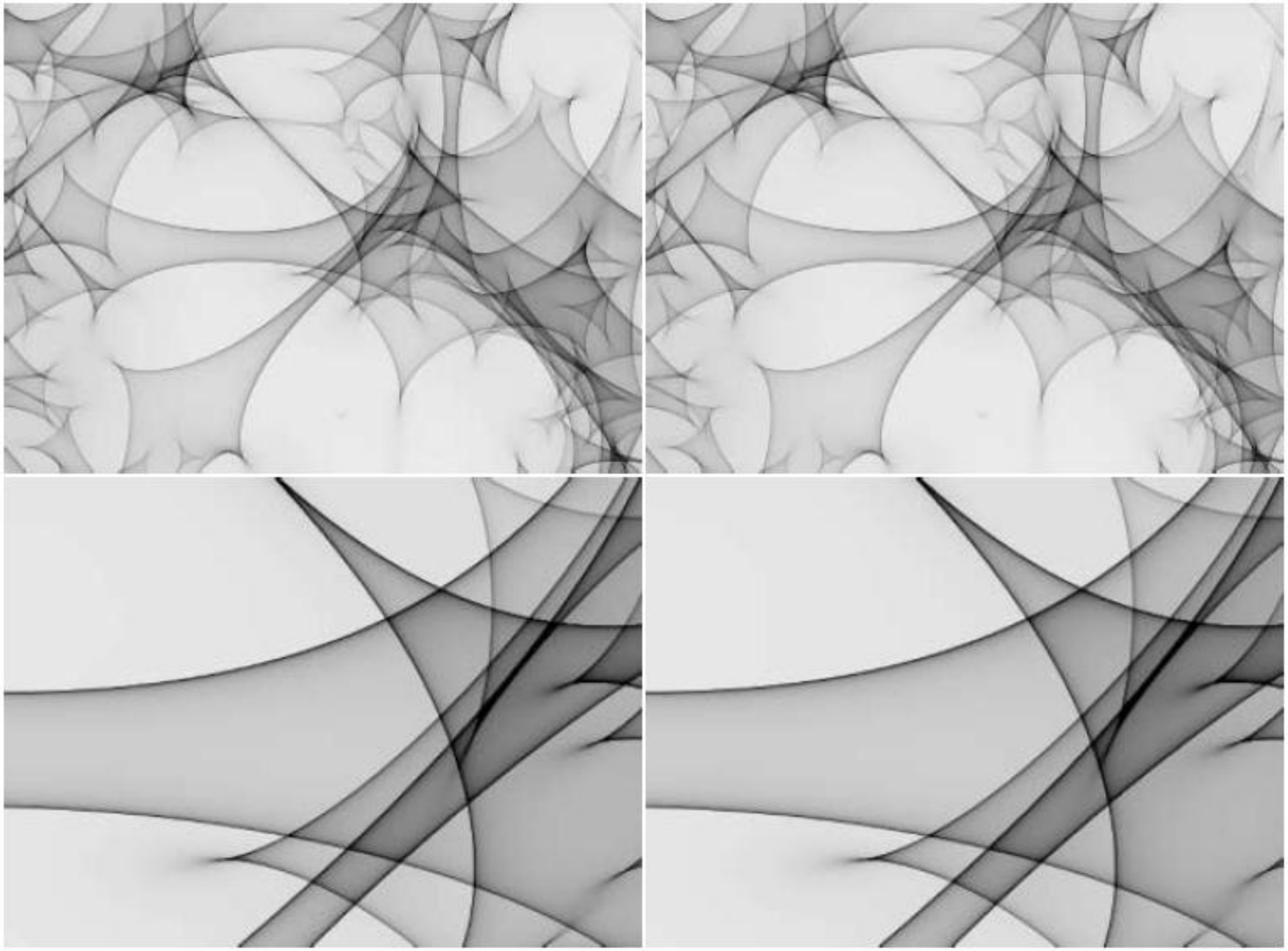


Figure 3. Detail of Figure 2 zooming the transformed subcells of one critical cell. One of the subcells is plotted with a solid line and the other with a dashed line except in the common side (the caustic). Notice the overlapping of the transformed subcells, or, in other words, the auto-overlapping of the transformed critical cell.

IPM vs IRS



IPM (III)

- Critical cells are detected via non linearity
- You have several choices:
  – Ignore them
  – Use IRS for those cells
  – Adaptive subdivision of critical cells
- This way, maps with extreme acuracy can be obtained with 1 ray/pix or even less !!!!
- We may speed up calculation by a factor of a **few hundred** !!!!

ugr Universidad de Granada

GPUs

- GPUs have become very popular these days.
- They provide very fast and relatively cheap hardware.
- You have to invest a bit in learning how to deal with them …
- Perfect for parallel computing (IRS is the super-mega-hyper-parallelizable problem)
- Thompson et al. (2010) New Astronomy, Volume 15, Issue 1, p. 16-23

Recipes

- You do not have too many particles
  →Go for IPM

- There are many deflectors
  →Go for TreeCode

- GPUs are a also a valid alternative… specially for mass production.

What's next?

- IPM + Treecode → Coming soon…

- IPM + GPU?

- Treecode → Fast Multipole Method

L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. J. Comput. Phys. 73, 325–348 (1987).

O(N^2) → O(N log N) → O(N)