



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 $\mathcal{R}^2 \rightarrow \mathcal{R}^2$, 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. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Astrophysics; Gravitational lensing; Tree code

1. Introduction

Gravitational lensing deals with the deflection of light by matter. Originally proposed by Einstein in his General Theory of Relativity, the deflection angle at the solar limb was measured in the 1919 solar eclipse by Dyson, Eddington and Davidson [6] and confirmed Einstein’s prediction of $\alpha_{\odot} = 1.75$ arcsec. In the 1930s Einstein, Russell, Zwicky and others investigated lensing theoretically [5,23,42]. In the 1960s various authors dealt with astrophysical applications of the gravitational lens effect. Most excitingly it was shown by Refsdal [21] that lensing offers the possibility to measure the Hubble constant which can be used to determine size and age of the universe.

E-mail address: jwambsganss@aip.de. (J. Wambsganss)

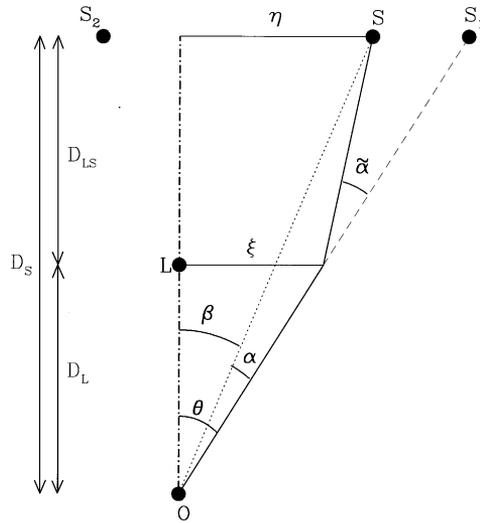


Fig. 1. Simplified gravitational lens scenario: The letters S, L and O mark the positions of source (at position η), lens and observer, respectively; S_1 and S_2 indicate the image positions. The distances observer–source, observer–lens, and lens–source are shown on the left-hand side (D_S , D_L , D_{LS}). The deflection angle $\tilde{\alpha}$ at impact parameter ξ and the position angles β of the source S and $\theta (= \beta + \alpha)$ of image S_1 are defined as well. For $\theta, \beta, \tilde{\alpha} \ll 1$, the lens equation can be derived from this geometrical arrangement (cf. Eq. (1)).

Only in 1979 the first gravitational lens was observed: Walsh and colleagues [29] found two quasars Q0957+561A,B separated by 6.1 arcsec on the sky, with identical redshifts. They turned out to be two images of one quasar. The image splitting is produced by a massive galaxy which happens to lie in the line of sight to the quasar. This discovery started gravitational lensing as an observational area of astrophysics. Today this field is booming: By now almost 2000 papers have appeared in the field of gravitational lensing. Almost a dozen different lensing phenomena have been discovered, e.g. multiple quasars, giant luminous arcs, arclets, Einstein rings, quasar microlensing, “Macho” microlensing, weak lensing, galaxy–galaxy lensing. There exists a comprehensive textbook for an overview on gravitational lensing [26]; mathematical aspects of lensing — in particular singularity theory — are treated in [18]. Two recent reviews describe the current situation in lensing [16,31].

A simplified gravitational lensing scenario is displayed in Fig. 1. The “ingredients” are: A light-emitting source S in the source plane at a distance D_S ; one (or more) lens(es) L in the lens plane at a distance D_L ; and the observer O. The light rays from the source are attracted by the lens(es), so that the observer receives only deflected rays. The light rays are well approximated by straight lines. This can be pictured as the action taking place in single planes. For certain geometrical arrangements more than one ray (or light bundle) can reach the observer from different directions. Then multiple images S_1 and S_2 are seen.

The relations between source position, deflection angle and image positions are formulated in the lens equation in angular coordinates:

$$\beta = \theta - \alpha(\theta), \tag{1}$$

where θ is the (angular) position of the image, α is the two-dimensional (scaled) deflection angle, and β is the position of the unperturbed source. This relation can easily be derived from the geometric approximation (cf. Fig. 1):

$$\theta D_S = \beta D_S + \tilde{\alpha} D_{LS}, \tag{2}$$

under the assumption that $|\theta|, |\beta|, |\tilde{\alpha}| \ll 1$. This condition is fulfilled in practically all astrophysically relevant situations. Here D_S, D_{LS} are the (angular diameter) distances observer–source and lens–source, respectively, and $\alpha = \tilde{\alpha} \times (D_{LS}/D_S)$.

The deflection angle for a point lens is given by

$$\tilde{\alpha}(\xi) = \frac{4GM}{c^2} \frac{\xi}{|\xi|^2}, \tag{3}$$

where M is the mass of the lens, ξ is the impact vector of the ray (cf. Fig. 1), G is the gravitational constant and c is the velocity of light. For an ensemble of (point) lenses the deflection angle is just the sum over all individual deflections, for a continuous mass distribution it is the integral over the surface mass density, weighted by the inverse distance (more details see [16]).

For the special case in which the source lies exactly behind the lens ($|\beta|=0$), due to the symmetry a ring-like image occurs whose angular radius is called *Einstein radius* θ_E :

$$\theta_E = \sqrt{\frac{4GM}{c^2} \frac{D_{LS}}{D_L D_S}}, \tag{4}$$

here we used the relation $\xi = D_L \theta$, where D_L is the distance observer–lens (cf. Fig. 1). The Einstein radius defines the angular scale for a lens situation. For a galaxy with a mass¹ of $M = 10^{12} M_\odot$ at a redshift of $z_L \approx 0.5$ and a quasar at redshift $z_S \approx 2.0$ the Einstein radius is about 2 arcsec (we used here a Hubble constant of $H_0 = 100h = 50 \text{ km s}^{-1} \text{ Mpc}^{-1}$ and an Einstein–deSitter universe).

In the case of quasars multiply imaged by galaxies, lensing occurs in fact on (at least) two mass scales. The lensing galaxy can produce two or more images with typical separations of order arcsec. The matter distribution inside the galaxy is not smooth, but “grainy”: galaxies consist at least partly of stars. And each star acts as a gravitational lens as well. That means we should see zillions of “micro-images”. The deflection angles and Einstein radii for individual stars are only of order microarcsec, so multiple micro-images can not be resolved, they rather form an unresolved “macro-image”. But microlensing also affects the total brightness of a quasar image, and so microlensing effects can be observed indirectly (see below).

Due to space limitations these introductory remarks can barely present the basic facts and relations in the field of gravitational lensing. We could mention here only the essential concepts very briefly. Comprehensive overviews and introductions to the field can be found in each review on lensing, e.g. the above-mentioned texts [16,18,26,31]. Below we only consider gravitational lensing simulations using a hierarchical tree code for the determination of the deflection angles. There are complementary, Fourier-based methods which address the problem from different point of view (e.g. [3,10]).

This paper consists of two main parts. In Section 2 we describe the techniques used to make the microlensing calculations with very *many point lenses in a single lens plane* very efficient. We show some results in the form of magnification patterns obtained with the tree-code enhanced ray shooting

¹ A practical mass unit in astronomy is the mass of the sun: $1M_\odot = 2 \times 10^{33} \text{ g}$.

technique, as well as a microlensed lightcurve. In Section 3 we describe our generalization of this technique to studies of lensing by extended *three-dimensional mass distributions* which more or less fill the universe. We approximate these mass distributions by many lens planes (of order 100) and apply the method to cosmological simulations.

2. Quasar microlensing: light deflection by two-dimensional distributions of point lenses

2.1. The astrophysical problem

For an observed gravitational lens situation of two or more images of a quasar, the standard task is to build — with some knowledge of the light distribution of the lensing galaxy — a mass model of the lens that can reproduce the positions of the quasar images and ideally the intensity ratios as well. In this approach the quasar is usually considered as a point source, and the mass distribution of the lens is smooth. But — as stated above — the individual stars in the lensing galaxy also affect the measured brightness of a quasar image [4,8]. Hence a perfect match of the intensity ratios is not expected. The exact (micro-lens) magnification depends on the positions of the stars.

Since the stars move relative to each other and the galaxy as a whole moves relative to quasar and observer, this lens configuration changes with time. As a consequence, the observed brightness of the quasar varies with time. Furthermore, the Einstein radius of a solar mass star — the “typical” length scale — corresponds to about 0.01 parsec.² This is of the same order as the “size” of the quasar: the continuum emitting region of a quasar, presumably the accretion disk around the massive central black hole is expected to have a similar size. This is an interesting coincidence, because this way microlensing can probe the innermost structure of quasars, it should even be possible to determine the brightness profile of quasars with microlensing [9].

The exact arrangement of stars in a distant galaxy can not be known, so microlensing simulations deal with the problem in a statistical way: with stars/microlenses distributed according to a given surface mass density — which is usually obtained from the “macro-model” of a certain gravitational lens system — the magnification as a function of position (in the source plane) is determined and its statistical properties are analyzed.

Such numerical simulations of the microlensing effect are done in order to learn both about the background source and about the objects that act as microlenses: the measurable flux changes of the quasars have different character for different microlens masses.

2.2. The numerical problem

In the microlensing calculations one determines the flux magnification corresponding to a certain position in the source plane. For an extended source one has to convolve such a magnification pattern with the source profile.

This method, known as inverse or backwards ray shooting [12,25,30], works as follows: one “shoots” light rays from the observer to the lens plane, calculates the angle of deflection due to the individual stars acting as microlenses (and considers external shear — the tidal force due to

² The “parsec” is a practical length unit in astronomy: 1 parsec $\approx 3.1 \times 10^{18}$ cm.

asymmetrically distributed matter far away from the region considered — and continuously distributed matter). Then these deflected light rays are followed until they hit the source plane (cf. Fig. 1). The rays are “collected” in the source plane in small squares. These “pixels” contain various numbers of rays; the number of rays per pixel is directly proportional to the magnification of a source with the size and shape of such a square. This two-dimensional density distribution of light rays – a magnification pattern — can be visualized easily (see below).

In order to “shoot” a light ray i through a single-lens plane, one has to sum over the deflection angles of all individual stars acting as microlenses. The deflection angle $\tilde{\alpha}_i$ for a number of n point lenses hence is just a summation of the deflection angle by each point lens j :

$$\tilde{\alpha}_i = \sum_{j=1}^n \tilde{\alpha}_{ji} = \frac{4G}{c^2} \sum_{j=1}^n M_j \frac{\mathbf{r}_{ij}}{r_{ij}^2}. \tag{5}$$

Here M_j is the mass of point lens j , \mathbf{r}_{ij} is the projected (vector) distance between the positions of light ray i and point lens j , and r_{ij} is its absolute value, $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$; here (x_i, y_i) is the position of ray i , and (x_j, y_j) is the position of lens j .

By far most of the computing time in the microlensing calculations is needed for the calculation of these deflection angles. As seen above, the calculation of the deflection angle α_{ij} of one lens on one light ray takes about ten mathematical operations (N_{op}). In order to obtain a high resolution, a large number of pixels N_{pix} is required (e.g., 2500×2500 pixels), and for statistical reasons one would like a high density of rays per pixel on average (e.g., $N_{av} \gg 100$).

The (minimal) number of individual lensing stars N_* that have to be taken into account depends on the surface mass density σ_* , the amount of external shear γ that is included, and on the ratio ε between the “diffuse flux” (i.e., the rays that are deflected into the receiving area from stars far outside the region where microlenses are considered) which one may neglect and the total flux. Details about this concept of diffuse flux can be found in [13,25]; an approximated expression for the minimum number of stars to be included for a certain value of ε is

$$N_* \simeq \frac{3\sigma_*^2}{(1 - \sigma_*)^2 - \gamma^2} \varepsilon^{-1}. \tag{6}$$

For low values of σ_* (≤ 0.4) a few hundred stars may be enough, but for the interesting cases closer to $\sigma_* = 1$ the number of stars increases dramatically. The number of stars that should be included for no external shear ($\gamma = 0$) with the requirement that more than 99% of the total flux is in the receiving field ($\varepsilon = 0.01$) for, e.g., $\sigma_* = 0.5, 0.8, 0.98$ are 300, 4800, 720300, respectively. A brief estimate of the number of mathematical operations for such a direct calculation with high resolution and for a high surface mass density (2500^2 pixels, 500 rays per pixel, 10^6 stars) results in:

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

Even with the fastest computers such a brute force calculation would take months or years! In other words: such a direct calculation can not be performed for values of σ_* close to one.

However, the deflection angle has an r^{-1} dependence on the distance between ray and lens (see Eq. (3)). That means, the farther away a lens is from the light ray, the less important it is. This is a standard situation in problems in which gravity is involved. It forced people to develop more efficient methods for the calculation of forces, most of them involved specific algorithms, but even specific hardware was developed. We use here the tree-code approach [1]: lenses are treated differently

according to their distance to the light ray. The “hierarchical tree code”, which will be described in the next section, does exactly this, it bunches lenses together in cells of different sizes. For the calculation of the deflection angle (or gravitational force in the three-dimensional case) it uses cells — pseudo-particles — whose sizes depend on the distance to the light ray considered.

The actual computation by the inverse ray shooting method is done on a regular grid of rays in the sky/lens plane. This grid is mapped onto the source plane, whereby the deflections are considered for each ray. In the general case, a smoothed out or continuously distributed surface mass density σ_c contributes to the deflection as well (as an additional constant deflection). The general microlensing equation then is

$$\mathbf{y} = \begin{pmatrix} 1 - \gamma & 0 \\ 0 & 1 + \gamma \end{pmatrix} \mathbf{x} - \sigma_c \mathbf{x} - \sum_{i=1}^{N_*} \frac{m_i (\mathbf{x} - \mathbf{x}_i)}{(\mathbf{x} - \mathbf{x}_i)^2}, \quad (7)$$

where the macromodel of the gravitational lens has to specify the values of shear γ and surface mass density σ_c (cf. [26]); \mathbf{x}_i are the positions of the lenses. That means that the rays shot in a square are mapped onto a rectangle with a side ratio $(1 - \sigma - \gamma)/(1 - \sigma + \gamma)$ (where $\sigma = \sigma_c + \sigma_*$, and σ_* is the surface mass density in compact objects). As we want the *receiving* area to be a square, we choose the shooting field, i.e., the area in the lens plane in which rays are mapped, to be a rectangle with the side ratio $(1 - \sigma + \gamma)/(1 - \sigma - \gamma)$.

We produced such maps in the source plane with very different side lengths, depending on the purpose and the desired resolution: the smallest receiving square in the source plane we considered had a sidelength of $L = 0.8\eta_0$ (where $\eta_0 = \theta_E \times D_S$ is the physical length of the Einstein ring in the source plane). This was useful for the high-resolution study of a single caustic crossing event [36]. The largest sidelength considered was $L = 2000\eta_0$ in an investigation of microlensing by small objects of planetary mass [24]. In practice, we have to use shooting regions that are much larger than $L/(1 - \sigma \pm \gamma)$ because due to the graininess of the matter many light rays are scattered out of the receiving square, and others from farther out are scattered in (cf. Eq. (6)).

2.3. The numerical approach

2.3.1. The hierarchical tree code

The basic idea of the use of the hierarchical tree code in microlensing calculations is: lenses close to the considered light ray have to be treated individually. The action of lenses farther away from the ray is approximated by considering only the centers of mass of, say, two or three lenses next to each other, and lenses even farther away from the light ray could be bunched together in even larger groups. The concept of the hierarchical tree code was first developed by [1] for the use in stellar dynamical problems.

Applied to the two-dimensional case of microlensing, the use of the hierarchical tree code can be explained as follows: all lenses are put in a hierarchy of square cells. The “root” is the largest cell comprising the total field with all lenses. If there is more than one particle in a cell, it is subdivided into four subcells with half the side length (in stellar dynamical calculations in three dimensions, cubes are used and divided in eight subcubes). This process is repeated recursively until all subcells contain either zero or one particle. So the particles are organized in a nested hierarchy of cells. For each cell the following quantities are determined: total mass, center of mass, quadrupole moment and higher multipole moments. This setup of the cell structure has to be computed only once at the

beginning of a microlensing calculation. Even for the largest number of lenses to be included (order 10^6) the setup needed only a tiny fraction of the total computing time (e.g., few CPU minutes versus many CPU hours).

2.3.2. Labelling of cells/lenses

A non-trivial problem connected with the setup of the cell structure and the identification of the individual cells and lenses is the numbering of cells and lenses, which is important for effectively storing and finding the data. We solved it in the following way: The “root square” (here normalized such that it has unit side length) at level 0 covers all lenses and gets index 0. The four squares in level 1 with side length $\frac{1}{2}$ get indices 1–4. The 16 squares in level 2 with side length $\frac{1}{4}$ get indices 5–20. Each square can be uniquely identified with the algorithm $k = \sum_{i=1}^n a_i \times 4^{n-i}$, where k is the index of the square, n is the number of the level of the square (starting with the root square at level 0; a cell with side length $\frac{1}{8} = 2^{-3}$ is at level 3), and $a_i = 1, 2, 3,$ or 4 for the cell positions top left, top right, bottom left, bottom right, respectively, in the level below. If the cell contains just one lens, this lens gets the respective index. With this algorithm

- squares at level 1 get indices between 1 and 4,
- squares at level 2 get indices between $5 = 4^1 + 1$ and $20 = 4^1 + 4^2$,
- squares at level 3 get indices between $21 = 4^1 + 4^2 + 1$ and $84 = 4^1 + 4^2 + 4^3$,
- ⋮
- squares at level n get indices between $4^1 + \dots + 4^{n-1} + 1$ and $4^1 + \dots + 4^n$.

These indices of cells and lenses have — especially at the high end — many unoccupied numbers in between; e.g., the four highest indices for a certain realisation of 163 lenses used for the case $\sigma=0.2$ are 76 354, 76 356, 192 070 and 192 072. For an effective use of the memory we store the data of N_* lenses at N_* successive vector elements. In the example above one would store these four indices in an index vector as vector elements $\text{index}(160)=76\ 354, \text{index}(161)=76\ 356, \text{index}(162)=192\ 070$ and $\text{index}(163)=192\ 072$. The number of cells N_C filled with more than one lens turned out (empirically) to be about $N_C \approx 0.73N_*$ for large values of N_* . This number N_C finally limited the number of lenses we could include in the calculations. For a lens we had to store just three numbers (two coordinates and mass), for each cell we needed to store 17 numbers: in addition to coordinates of center of mass and total mass there were the 14 coefficients of the multipole components up to sixth order. The largest fields we considered so far contained about $N_* \approx 10^7$ lenses.

Fig. 2a shows an example for the cell structure of 31 arbitrarily distributed lenses. The hierarchy in this case goes down to level 5. The cell structure can be visualized with a “tree”, as is shown in Fig. 2b. The ending parts, i.e., the cells that contain exactly one lens, are called leaves (astrophysicists like drawing this tree “upside-down”: the root at the top and the leaves at the bottom). In Fig. 2c it is shown which lenses are treated individually and which ones are bunched together as “cells” for a certain ray position, marked with a circled dot.

The criterion that decides which nearby stars are to be included by direct summation, and which of the more distant stars are to be bunched together in cells of which size depends on the distance of the cell/lens to the light ray considered, and on the side length of the respective cell. In order to do that one has to “descend” the tree from root to leaves. One always compares the size of the cell with the distance (of center of mass) of it to the ray considered (in other words: the accuracy

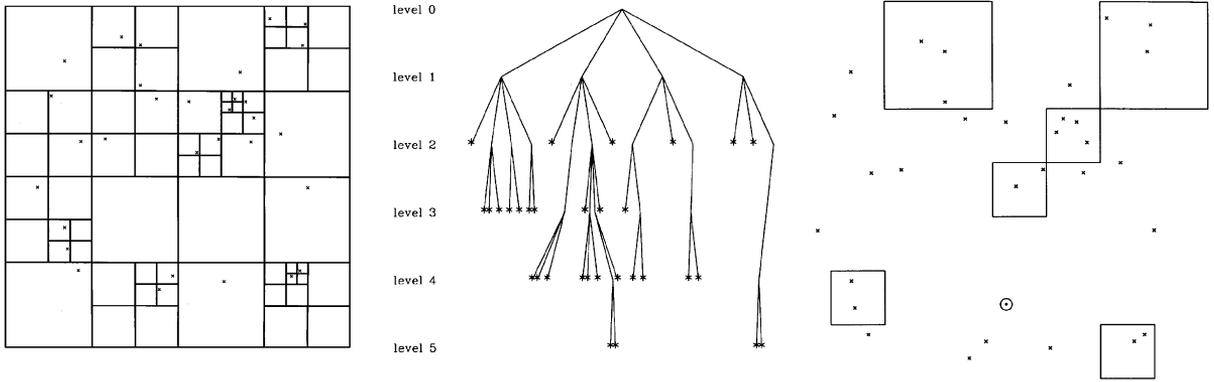


Fig. 2. Example of the use of the hierarchical tree code in a microlensing scenario with 31 randomly distributed lenses. (a, left) Cell structure of the point lenses; (b, middle) Illustration of the cell structure as “tree”; (c, right) Cell/lens configuration to be used for a certain ray position (marked with a circled dot).

criterion is something like the opening angle of the cell as seen from the position of the ray). If this ratio is smaller than a chosen value of the “accuracy parameter” δ or if the cell contains only one lens, this cell is used. Otherwise, the cell is resolved into its (up to four) subcells, whose side lengths again are compared with the distances of their center-of-mass positions to the considered ray, and so on. Typically, δ ranges between 0.4 and 0.9. If a particular cell is used for the determination of the deflection angle, it is considered as a pseudo-lens with the total mass of all lenses inside, located at the center of mass determined by all these particles.

2.3.3. *Lenses and pseudo-lenses*

In the approach just described, we use the hierarchical tree code in order to approximate the angle of deflection $\tilde{\alpha}$ by two parts according to the directly included lenses $\tilde{\alpha}_L$ and the cells $\tilde{\alpha}_C$:

$$\tilde{\alpha} = \sum_{i=1}^{N_*} \tilde{\alpha}_i \approx \sum_{j=1}^{N_L} \tilde{\alpha}_j + \sum_{k=1}^{N_C} \tilde{\alpha}_k =: \tilde{\alpha}_L + \tilde{\alpha}_C. \tag{8}$$

The N ’s denote the following:

- N_* is the number of all lenses,
- N_L the number of lenses to be included directly,
- N_C the number of cells (= pseudo-lenses) to be included.

For cases in which $N_L + N_C \ll N_*$, the calculation is speeded up considerably.

For the light ray shown in Fig. 2c, six cells with more than one lens are used (in total they contain 16 lenses), whereas all other lenses are treated individually, that means here $N_* = 31$, $N_L = 15$ and $N_C = 6$. This is not very remarkable yet. But in a calculation with about $N_* = 10^6$ lenses (for $\sigma = 1.17, \gamma = 0.83; L = 150\eta_0; \delta = 0.6$), the average number of cells used was 210, the average number of lenses used directly was 40.

The computing time for this hierarchical tree method (in two dimensions) increases like $\mathcal{O}(\log N_*)$, whereas a direct summation would increase as $\mathcal{O}(N_*)$. In practice, gains in CPU-time of factors of

1000 and more can be reached easily for 10^6 microlenses, when compared with the direct determination of the deflection angles.

The mathematical “problem” to be solved over and over again for the determination of the nearby lenses that must be included directly and the cells that have to be included as pseudo-lenses is the following: if a cell contains more than one lens and is too large to fulfill the accuracy criterion, it has to be resolved into its subcells. The index k of this lens is known, and so are the indices $4k + 1, 2, 3, 4$ of the four subcells. Now one has to find out if the subcells with these indices are empty (i.e., the corresponding index does exist neither in the index vector of the cells nor in that of the lenses), if they contain just one lens (the index vector of the lenses contains this index), or if they are filled with more than one lens (index exists in the index vector of the cells). This search for the indices is done in the ordered index vectors of the cells and lenses with a slightly modified fast searching routine based on [19].

The determination of the cell/lens structure to be used for a certain ray is rather time-intensive. For low numbers of lenses ($N_* \approx 50$) the additional amount of computing time for this climbing down the tree, i.e., the comparisons between cell size and distance of cell to ray is higher than that for a direct brute-force calculation with all lenses. For high numbers of lenses, however, this method saves an appreciable amount of time compared to the direct calculation.

2.3.4. Increased accuracy with higher-order multipoles

In its simplest application the tree code assumes that all matter inside a cell is located at its center of mass. However, we use higher-order multipoles of the mass distribution in order to increase the accuracy. That means instead of using the cell as a point-like particle, we add the contributions of quadrupole moment M_2 , octopole moment M_3 and even higher moments of the cell. This increases the accuracy of the determination of the deflection angle considerably.

In Fig. 3 we show the deviation of the deflection angle determined with this tree-code method from the directly determined, i.e. exact deflection angle: $(\alpha_{\text{direct}} - \alpha_{\text{tree}})$ as a function of α_{direct} in arbitrary units. The panels correspond to determinations of α_{tree} including multipoles of different order; they represent: all matter in center of mass, i.e. only monopole term (top left panel), monopole term plus quadrupole moments (top right), monopole term plus all multipoles up to order 4 (bottom left), and monopole term plus all multipoles up to order 6 (bottom right). It can be seen that for this case the largest fractional deviation between the two methods of determining the deflection angle is of order 2×10^{-3} , with the rms deviation quite a bit smaller.

For the actual calculations we use multipoles up to order 6, corresponding to the lower right panel. This way the value of the accuracy parameter δ can be increased to values as high as 0.9. This in turn then decreases considerably the number of lenses and pseudo-lenses to be used, hence it speeds up the calculation dramatically.

2.3.5. Further speed-up: fixed cell/lens structure for different rays

For our purpose of using the hierarchical tree code in gravitational lens calculations there is very much redundancy. Due to the high density of rays, many adjacent rays use exactly the same cell/lens pattern, i.e., the same lenses for direct calculation and the same cells as pseudo-lenses. This means, one could save the “climbing-down-the-tree” for quite a number of rays. To take advantage of this feature we do the following: For the position of a certain test ray, we climb down the tree and

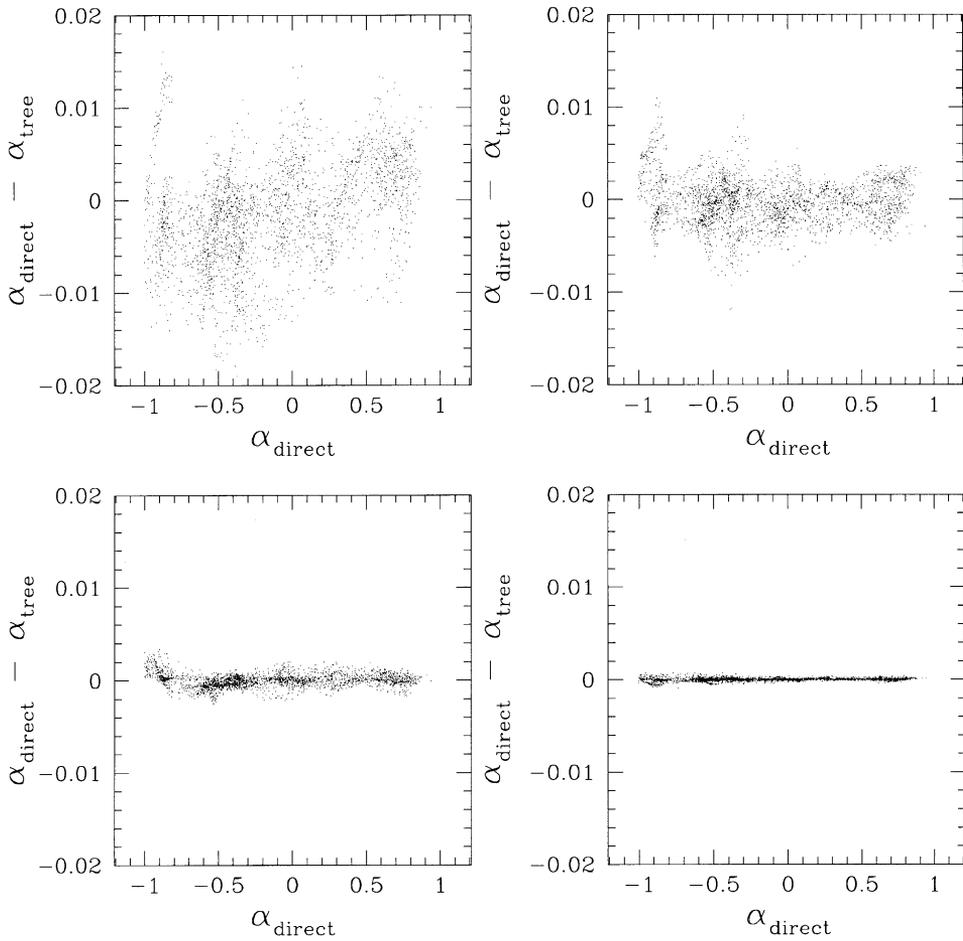


Fig. 3. The accuracy increase of the deflection angle obtained with the inclusion of higher-order multipole moments (see Section 2.3.4) is shown here. In each panel the difference between the directly determined deflection angle ($\tilde{\alpha}_{\text{direct}}$) and the one obtained with the tree code ($\tilde{\alpha}_{\text{tree}}$) is plotted as a function of the former (in arbitrary units). For the top left panel all matter is assumed to be concentrated in the centers of mass of the cells (i.e. only the monopole term is included). In the top right panel, the quadrupole moment is included. In the bottom panels moments up to order 4 (left) and 6 (right) are included. It is obvious how much the accuracy increases with the inclusion of higher-order moments.

determine the lenses and cells to be included in the calculation. Then we put a square grid around this particular position with n_{fix}^2 rays (see Fig. 4), with typically $n_{\text{fix}} \sim 10$. Now, we can shoot all these rays with exactly the same cell/lens configuration. This again speeds up the code by almost a factor n_{fix}^2 .

2.3.6. *Final speed-up: expansion and interpolation*

The code can be accelerated even more: the deflection angle $\tilde{\alpha}_C$ (cf. Eq. (8)) due to the cells (containing the far away lenses) changes only very slightly across the n_{fix} rays with fixed cell/lens

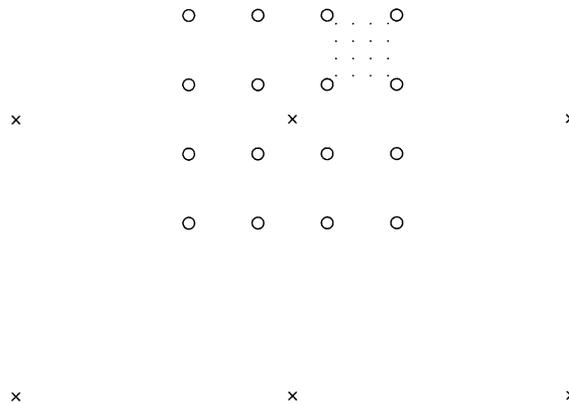


Fig. 4. Illustration of the way we “shoot” light rays: For each (test-)ray marked with a cross the cell/lens configuration is determined. For n_{fix}^2 (test-)rays (marked with circles) the deflection angle due to this fixed cells is determined (here $n_{\text{fix}} = 4$; in the calculations we usually took $n_{\text{fix}} = 10$). On the lowest level n_{int}^2 rays (marked with dots) are “really” shot, with the deflection angle of the cells determined by interpolation and that of the N_L close lenses calculated directly (here $n_{\text{int}} = 4$; in the calculations we took $n_{\text{int}} = 10$).

structure. So we simplify the calculation of the deflection caused by the typically 50–100 cells in the following way: We calculate the deflection angles of the fixed cells for all n_{fix}^2 test rays (cf. Fig. 4). Then, we compute the coefficients of a fourth order polynomial through four points for each set of four rays out of the n_{fix}^2 test rays forming a small square. Now we calculate the deflection angles due to the cells for n_{int}^2 rays inside this square by interpolation of the deflection angles at the four corners (typically $n_{\text{int}} = 10$). We just have to add the additional contributions of the N_L lenses to be treated individually (see Fig. 2), typically between 20 and 100. Both these parts, interpolation of the deflection angle of the distant cells and individual treatment of the N_L lenses close to the light ray considered, can be vectorized very effectively. As this part of the whole program has to be run over and over again, most of the computing time (between 90 and 98%) goes into this optimized part of the calculation.

Summarized: one has to climb down the tree hierarchy just in one out of $n_{\text{fix}}^2 \times n_{\text{int}}^2$ rays, and one has to determine the contributions to the deflection angle of all cells just in one out of n_{int}^2 rays. Only the deflection angles of the lenses very close to the light ray considered (typically between 20 and 100) have to be determined for each individual ray.

2.3.7. Tests and accuracy checks

During the development of the code, every “improvement” was tested at different stages. The most rigorous test was the comparison of the “brute force calculation”, i.e., including every lens directly (for individual rays this was feasible) with the deflection angles obtained by climbing down the tree, by using the fixed cell/lens configuration and by making use of the interpolation method (cf. Fig. 2). The deviation between the direct and the approximate calculation has to be compared with the side length of one pixel, because this size determines the resolution. We find, that a typical or average deviation of less than a tenth of the pixel size is already a very good approximation. In most practical cases, the “noise” obtained by the method with such a limit is below the numerical

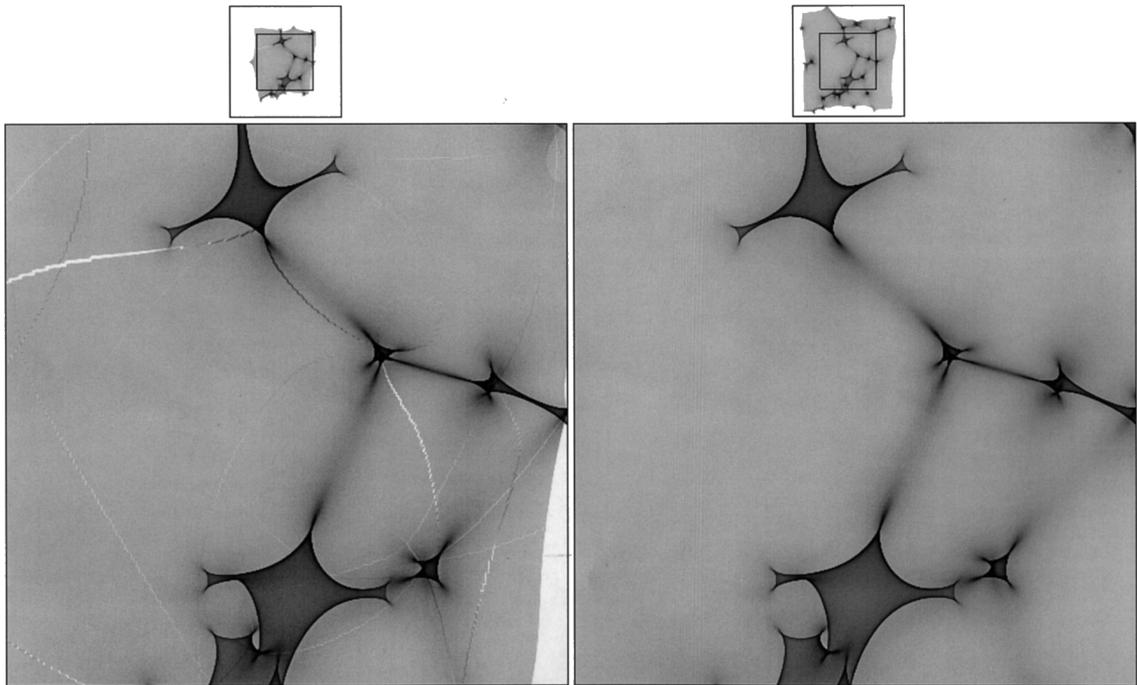


Fig. 5. “Wrong” magnification pattern for $\sigma = 0.2$ (left panel): here the accuracy parameter is too large: $\delta = 1.2$ and the shooting region is too small. Compare this frame with the correct one (right panel). A magnification pattern indicates the magnification as a function of position in the source plane. Darker gray means higher magnification. On top of the two patterns: respective “secondary” receiving fields that cover a larger region with worse resolution; they serve as a test for the size of the shooting region. The black squares in the center mark the positions of the real magnification patterns shown below.

noise (“discreteness effect”) due to the mapping of a regular grid of rays onto another regular grid of pixels (it depends on the density of the rays).

We have tested different accuracy parameters δ for many rays before each calculation, to be sure about the average and maximum errors. Usually values of $\delta = 0.8$ or 0.9 were sufficient; however, for magnification patterns of small sidelength and consequently high resolution, we had to go down to values of $\delta = 0.4$, which meant an appreciable increase in computing time.

In the course of the test calculations we have found out that a second very efficient accuracy test can be performed just by eye. If the accuracy parameter δ is chosen too large, then the change from one square with fixed cell/lens distribution to the next results in a discontinuous magnification pattern with very characteristic lines of discontinuity: the “images” of those neighboring squares are not really adjacent, but partly overlap or are torn apart a bit. These overlapping or separated regions can be seen very easily on the magnification patterns, even if their widths are only a fraction of a pixel size. Those “wrong” features indicate that δ is chosen too large. Because this check is very usefully applied in real calculations, Fig. 5a displays such a “wrong” frame, were δ is too large.

Another “safety measure” is a check of the size of the magnification pattern considered. To be sure that the shooting region is chosen large enough, we collect light rays in *two* receiving fields:

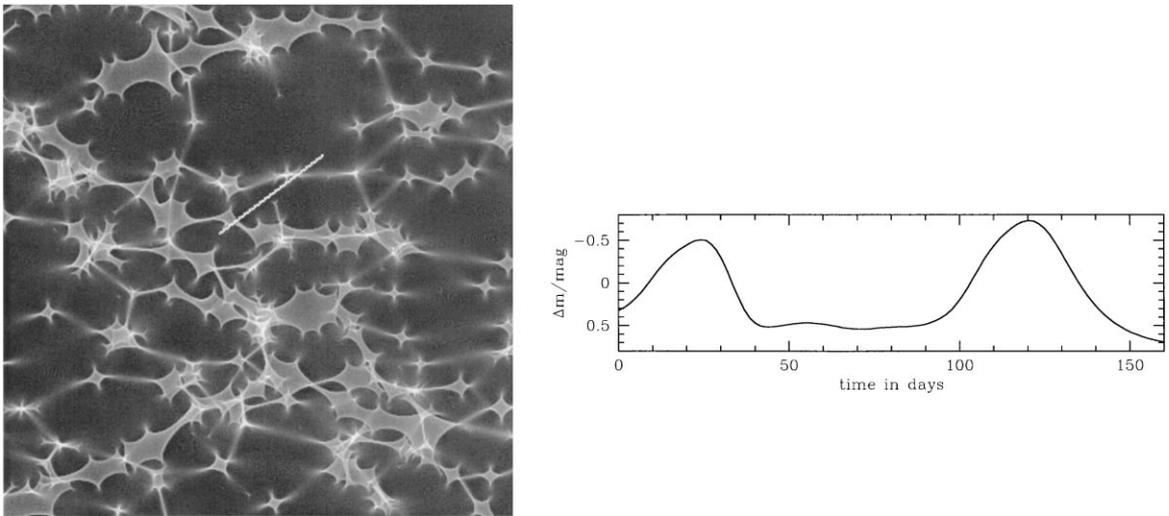


Fig. 6. Left: Magnification pattern for surface mass density of $\sigma = 0.32$ and external shear of $\gamma = 0.18$ (parameters for image A of the double quasar Q0957+561). The sidelength corresponds to 40 Einstein radii. The “lenses” here are objects of planetary mass ($m_{\text{Lens}} = 10^{-5} M_{\odot}$), and the white bar indicates location and length of the track along which the light curve (displayed in the right panel) is evaluated. The length of the light curve (for a quasar with a Gaussian brightness profile of size 3×10^{14} cm) corresponds to 160 days (for an assumed transverse velocity of 600 km/s). More details on this project can be found in [24].

additionally to the magnification patterns described above a second field was used, with a much smaller number of pixels, but covering a much larger area in the source plane (and hence with much lower resolution). We checked this secondary field by eye after each calculation, and if there were any doubts concerning “boundary” effects, we repeated the computation with a larger shooting area. On the small panels at the top of Fig. 5 such poor resolution fields are shown. The black squares in the centers mark the position of the “real” magnification patterns displayed below 5a.

2.4. The astrophysical results

The ray-shooting program described above has been applied to a number of gravitational lens problems in recent years (see, e.g. [7,11,15,20,27,28,37–39]). Most of these application studied microlensing light curves or microlensing magnification distributions, often applied to the two lens systems Q2237+0305 and Q0957+561. Variants of the code, allowing for moving lenses [14,35], study of jet-like source shapes [40], and the explorations of planet lensing [41] have been used as well.

We describe here the “standard” application of the microlensing code, namely producing a magnification pattern for a microlensing situation with given surface mass density σ and external shear γ . In Fig. 6a part of such a magnification pattern is shown for the parameters of image A of the double quasar Q0957+561. In Fig. 6b we show one example light curve to indicate how a one-dimensional cut through such a magnification pattern, convolved with a quasar brightness profile, represents a microlensed light curve. The microlensing properties of this lensing system has recently been studied

in [24], and many more details can be found there. One particular result of this study — which compared the (lack of) observed microlensing with simulations of microlensing by objects of different masses — is that the halo of the lensing galaxy cannot consist entirely of compact objects in the mass range $10^{-7}M_{\odot}$ – $10^{-4}M_{\odot}$, because those objects would produce significant fluctuations on a time corresponding to the observed one. A FORTRAN-version of this microlensing code is available on request from the author.

3. Cosmological lensing: light deflection by three-dimensional mass distributions

3.1. The multi-plane lens equation

In the previous chapter we described the application of the hierarchical tree code to a microlensing scenario where a number of “point” lenses are distributed in a single plane. Here we treat a situation that is more general in two aspects:

- matter is distributed in three-dimensional space, which we approximate by many (of order 100) different lens planes;
- we do not deal with “point lenses” any more, but rather with extended mass distributions.

The first point means we have to follow light rays through a whole set of lens planes. In the text book [26] a whole chapter deals with multiple light deflection; there, all details about the theory of many planes can be found. Here we just describe the multiplane lens equation as we use it:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{y} = \mathbf{x} - \sum_{i=1}^N \frac{D_{iS}}{D_S} \tilde{\boldsymbol{\alpha}}_i(\mathbf{x}), \tag{9}$$

where \mathbf{x} is the position vector of a light ray in the first-lens plane (or equivalently in the image/sky plane) and the \mathbf{x}_i ’s are iteratively determined in the higher-lens planes; \mathbf{y} is its corresponding position in the source plane (note that these definitions are different from those in Section 2!); $\tilde{\boldsymbol{\alpha}}_i$ is the (two-dimensional) deflection angle in lens plane # i ; D_S and D_{iS} are the angular diameter distances between observer and source plane and between i th lens plane and source plane, respectively. The lens planes are labelled from 1 to N in order of increasing redshift (cf. Fig. 7).

The part of the equation that we are essentially interested in describes the mapping between the image/sky plane and the source plane. The relation between the positions \mathbf{x} in the image/sky plane and \mathbf{y} in the source plane is given by $A(\mathbf{x})$, the Jacobian matrix of the mapping. In the case of multiple lens planes the Jacobian

$$A_{ij} \equiv \frac{\partial y_i}{\partial x_j}, \tag{10}$$

does not have all the nice properties it has in the single-lens plane case, in particular, it is not symmetric any more in the general case: $a_{12} \neq a_{21}$. Furthermore it is not curl-free: $\text{curl} A \neq 0$. Many other properties, however, are still valid. The magnification of an image at position \mathbf{x} is given as

$$\mu(\mathbf{x}) = [\det A(\mathbf{x})]^{-1}. \tag{11}$$

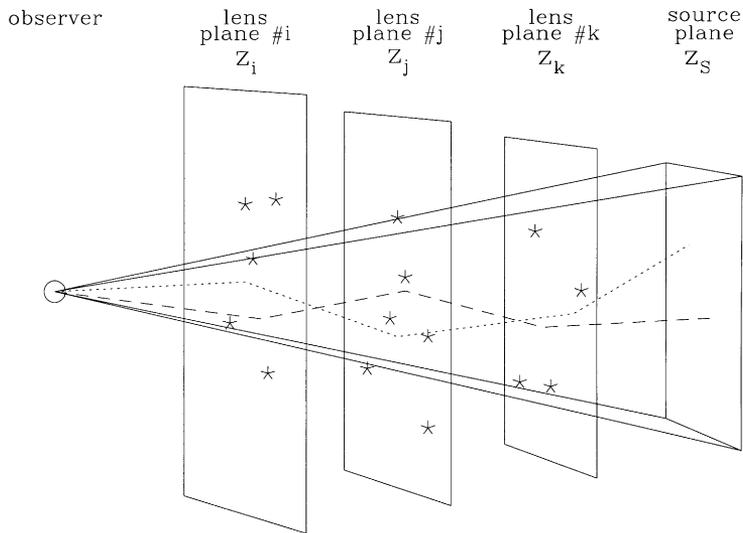


Fig. 7. Set up for a situation dealing with a three-dimensional distribution of lenses: The deflecting matter — here symbolized as little stars — is collected in a number of discrete lens planes with increasing redshift $z_i < z_j < z_k$ (here only three planes are shown). The light rays (dashed lines) get deflected in each lens plane and hit the source plane at z_s . The “maximum opening angle” of the bundle of light rays followed is given by the highest redshift plane. The increase of physical size of our finite lens planes (which have constant “comoving” size) with decreasing redshift reflects the way we treat the expansion of the universe.

Other combinations of the components of $A(\mathbf{x})$ represent useful quantities as well. The effective surface mass density σ (which is basically the sum of the physical surface mass densities of the individual lens planes, weighted by the corresponding ratios between the angular diameter distances) for any position \mathbf{x} in the sky plane is given as

$$\sigma(\mathbf{x}) = 1 - 0.5[a_{11}(\mathbf{x}) + a_{22}(\mathbf{x})]. \tag{12}$$

Similarly, the two components of the shear $\gamma(\mathbf{x}) = [\gamma_1(\mathbf{x}), \gamma_2(\mathbf{x})]$ are

$$\gamma_1(\mathbf{x}) = -0.5[a_{11}(\mathbf{x}) - a_{22}(\mathbf{x})], \tag{13}$$

$$\gamma_2(\mathbf{x}) = -0.5[a_{12}(\mathbf{x}) + a_{21}(\mathbf{x})]. \tag{14}$$

The matter is obtained from large-scale structure simulations (see, e.g. [2]); a lens plane is the projection of all the matter in a cylindrical volume with dimensions $a \times a \times b$, where typical values of the sidelength a correspond to a comoving physical size of about 10 Mpc, and the “depth” b is about 100 Mpc. The lens planes are two-dimensional distributions of matter which are given in a grid, usually at N_{grid}^2 positions (we considered cases with $N_{\text{grid}} = 800$). We treat this matter distribution as N_{grid}^2 point lenses (of order 10^6), or more accurately, as “smeared out” in lens pixels, i.e. in a square region of about 10 kpc at a side.

As described in the previous chapter, we calculate the deflection angle in each lens plane with a hierarchical tree code in two dimensions. Here the matter distribution is treated as essentially N_{grid}^2 point lenses, positioned at the points of the regular matter grid. That means the lens positions are always the same. Therefore, the lens/cell configuration to be used for a grid of light rays has to be

determined just once, and can be used again in all subsequent lens planes, since the relative positions of the lenses are exactly the same. Hence the “climbing down the tree” in order to determine the cell/lens-configuration to be used has to be done only once, and can be used for all other lens planes accordingly.

For point lenses the deflection angle formally diverges when the distance between light ray and lens position becomes zero, $r_{ij} \rightarrow 0$. However, in this case of a continuous matter distribution, this is an artificial divergence: the underlying mass distribution is smooth, we only approximate it by a large number of “points”. We avoid this artificial divergence, by always determining the deflection angle at the points directly in the center between four lens positions so that nearest neighbor effects cancel. That means, for each lens plane we determine the deflection angles for a regular grid of “test-rays”. But, since the real positions of the light rays are offset from the positions of these “test-rays”, we calculate the actual deflection angles of the rays that are followed through the planes using a bicubic interpolation between the four test-rays surrounding the real ray position. In this way we obtain a smooth two-dimensional field of deflection angles for each lens plane, as is to be expected for a continuous matter distribution.

3.2. Application to various cosmological models

In contrast to the microlensing scenario described in the previous chapter, the three-dimensional version of the program is applied to “macroscopic” scales, i.e. lensing effects that can be directly observed. The angular size of the (square) bundle of light rays that we consider is determined by the angular size of the highest redshift lens plane that we use. As an example, the angular size of a cube with comoving size $L = 10h^{-1}$ Mpc at a redshift $z = 3$ is about $\beta \approx 350$ arcsec (depending on the exact cosmological model used).

In these multi-plane lensing simulations, the (average) angular size of the field of rays is fixed, it is the same for all lens planes. However, as the physical size of the underlying cosmological cubes is expanding with decreasing redshift, the angular size of the lens planes increases rapidly with decreasing redshift. This means that the field of rays intercepts only a small part of the lens plane for small redshifts (cf. Fig. 7), i.e. for low redshift lens planes, one lens-pixel (which has fixed comoving size) covers quite a large angular region.

For each complete run of the light rays through all the lens planes, the positions of the light rays in the image/sky plane and the corresponding positions in the source plane are stored. This allows to determine the coefficients a_{ij} of the Jacobian matrix A , and to derive properties like magnification, shear, effective surface mass density as a function of position. It is possible to identify regions that are multiply imaged, and determine positions, shapes and magnifications of the multiple images. In addition, we can put in a distribution of extended sources in the background, “map” them forward and determine their distorted shapes, lens-induced ellipticities, arclets, giant luminous arcs or Einstein rings. A more complete description of the program with a few applications can be found in [34].

3.3. Illustration of astrophysical applications

In Fig. 8 we present an illustration of the type of results obtained with this multi-plane ray shooting method. The density of backwards traced light rays in the source plane is proportional to

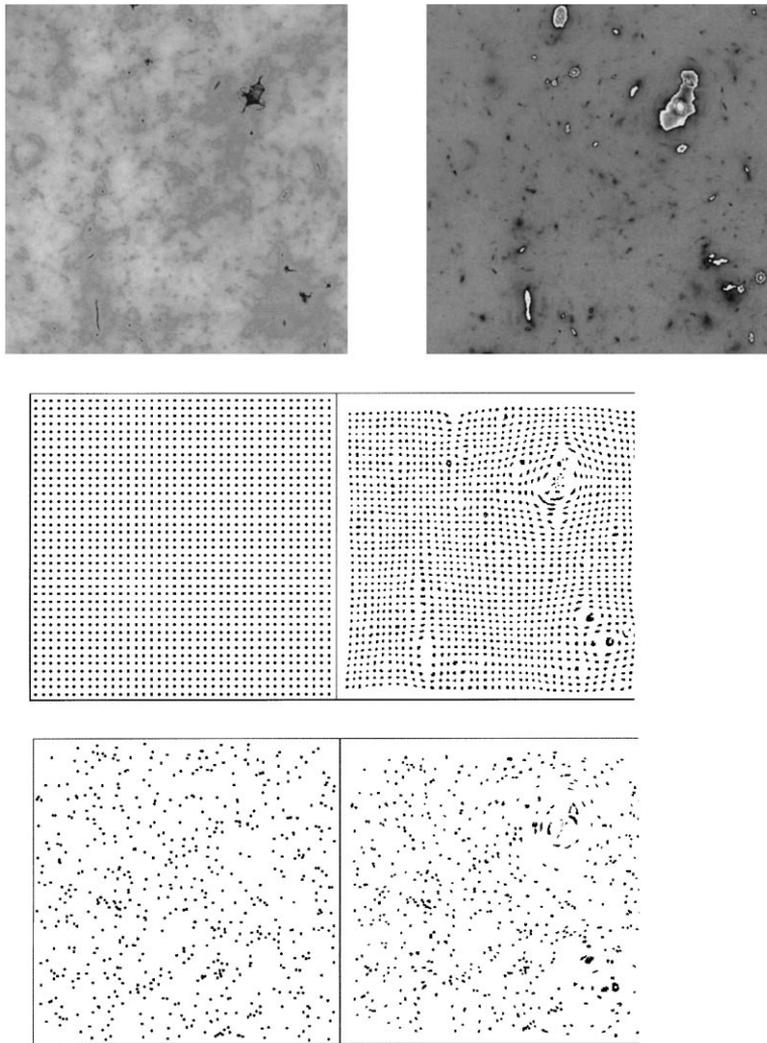


Fig. 8. The magnification distribution for a realistic situation of multi-plane lensing is illustrated (all panels have a sidelength of about 6 arcmin). Top left: Magnification pattern in a source plane at redshift $z = 1$; the darker the gray, the higher the magnification for a source at this location. Top right: Magnification distribution in the image plane/on the sky. Middle row: Source configuration in a source plane at $z = 1$: a regular grid of circular sources (left), and corresponding image configuration (right): distorted, shifted, and magnified images, sometimes even multiply mapped. Bottom row: as above, but for randomly placed sources.

the magnification at these locations. In Fig. 8a such a magnification distribution is shown for one particular line of sight of a standard CDM universe. It is obvious that the magnification changes considerably over the area (which is about 6 arcmin at a side), with a few dark caustic lines indicating very high magnification and embracing multiply imaged parts of the sky. The magnification in the image plane (Fig. 8b) looks qualitatively similar. Only in high magnification regions there are differences: the sharp boundary lines between black and white indicate the “critical lines”, along

which highly magnified images are positioned. In the two pairs of panels below, the distortion effect of the universe is indicated. The left panels show a source population of equally sized circular galaxies at redshift $z = 1$. The right-hand panels show the appearance of these source distributions on the sky, when viewed through the matter that produced the magnification distributions above. The areas of the images reflect their relative magnifications. Note the dramatic giant arc images and/or Einstein ring images that occur at locations that correspond to high magnifications in the panels above.

3.4. Results on weak and strong lensing

The multi-plane ray shooting program as described above has been applied to various cosmological models. Results for a standard cold dark matter model ($\Omega_{\text{matter}} = 1$) showed that it should produce a relatively high fraction of multiple quasars with large angular separations; lensed quasars with images separated by 30 and 40 arcsec or even larger were found in the simulations. Comparison with observations (largest separation multiple quasar known: $\theta = 7$ arcsec) showed that this is a strong argument against this standard cold dark matter model [32].

In another application for a cosmological model with a cosmological constant ($\Omega_{\text{matter}} = 0.4$, $\Omega_{\Lambda} = 0.6$), we evaluated the “weak lensing” effect. This is the small magnification or demagnification that affects basically every line of sight. As a consequence, we do not measure the “true” brightness of distant sources, but rather a slightly modified brightness, which depends on the matter along the line of sight. This means that a population of “standard candles” would get a certain dispersion in apparent brightness. This is important, because cosmologists try to use distant sources as standard candles for the determination of the geometry of the universe, in particular supernovae of type Ia [17,22].

We determined the amount of lensing-induced dispersion for a population of standard candles at various redshifts [33]. It turns out for the model considered we expect statistical fluctuations with a standard deviation of about $\sigma = 0.02$ mag (0.04 mag) for sources at a redshift of $z = 0.5(1.0)$. This is still beyond the observational accuracy at the moment.

In addition to the two applications briefly described here, many more uses are underway, e.g.,

- Study of geometry, magnification ratio, and separation of multiple images;
- Study of correlations between single/multiple quasars and strong foreground mass concentrations;
- Study of effects of chance alignments of dense matter clumps along the line of sight: how often is there more than one-lens plane important;
- Study of the effects of three-dimensional lensing on the cosmic microwave background.

4. Summary

In gravitational lensing one often wants to know the effect of a complicated matter distribution on a (population of) background source(s). This concerns positions, magnifications, distortions of sources as well as identification of multiple images. A straightforward technique to do this is the “ray-tracing method”, which just follows light rays that are deflected by the given matter distribution.

This method is too expensive computationally to be applied to many interesting questions. Therefore, we developed more effective ray-tracing techniques which are explained here.

In the first part of this article we describe the technique and the use of a hierarchical tree code in a two-dimensional situation for the determination of deflection angles in gravitational lensing. This is illustrated by applying it to quasar microlensing. The results obtained with the described ray-shooting code comprise fields as far apart as limits on the size of quasars and the possibility to detect planetary companions to stars in the Milky Way.

In the second part we generalize this two-dimensional method to a multi-lens-plane scenario: a real three-dimensional matter distribution is approximated by many-lens planes, and light rays are followed through them up to a source population at high redshift. A more detailed description of the method can be found in [34]. This method is applied to study various cosmological models, with the particular purpose to find differences in the “lensing properties” between various model universes, in order to be able to distinguish between them when comparing with observations. We investigated, e.g., the frequency of multiply imaged quasars for a standard cold dark matter universe, or the magnification distribution of standard candles. Many more applications to gravitational lensing are under investigation, e.g. the weak lensing effect by large-scale structure, or the influence of lensing on the cosmic microwave background.

Acknowledgements

It is a pleasure to thank Bohdan Paczyński, Peter Schneider, Renyue Cen, and Jeremiah P. Ostriker, who advised me at various stages of development of these gravitational lens programs. In addition, I want to thank both referees for their careful reading of the manuscript, and in particular Peter Schneider for his suggestions which improved the paper considerably.

References

- [1] J. Barnes, P. Hut, *Nature* 324 (1986) 446.
- [2] R. Cen, *Astrophys. J. Suppl. Ser.* 78 (1992) 341–364.
- [3] K.-H. Chae, V.K. Khersonsky, D.A. Turnshek, *Astrophys. J.* 506 (1998) 80–92.
- [4] K. Chang, S. Refsdal, *Nature* 282 (1979) 561–564.
- [5] A. Einstein, *Science* 84 (1936) 506.
- [6] F.W. Dyson, A.S. Eddington, C.R. Davidson, *Phil. Trans. Roy. Soc.* 62 (1920) 291.
- [7] E.E. Falco, J. Wambsganss, P. Schneider, *MNRAS* 251 (1991) 698–706.
- [8] J.R. Gott III, *Astrophys. J.* 243 (1981) 140–146.
- [9] B. Grieger, R. Kayser, S. Refsdal, *Astron. Astrophys.* 194 (1988) 54–64.
- [10] B. Jain, U. Seljak, *Astrophys. J.* 484 (1997) 560–573.
- [11] M. Jaroszyński, J. Wambsganss, B. Paczyński, *Astrophys. J.* 396 (1992) L65–L68.
- [12] R. Kayser, S. Refsdal, R. Stabell, *Astron. Astrophys.* 166 (1986) 36–52.
- [13] N. Katz, S. Balbus, B. Paczyński, *Astrophys. J.* 306 (1986) 2–8.
- [14] T. Kundić, J. Wambsganss, *Astrophys. J.* 404 (1993) 455–459.
- [15] G.F. Lewis, J. Miralda-Escude, D.C. Richardson, J. Wambsganss, *MNRAS* 261 (1993) 647–656.
- [16] R. Narayan, M. Bartelmann, *Gravitational lensing*; in: A. Dekel, J.P. Ostriker (Eds.), *Formation of Structure in the Universe*, Cambridge University Press, Cambridge, 1999, p. 360.
- [17] S. Perlmutter et al., *The supernova cosmology project*, *Astrophys. J.* 483 (1997) 565–581.

- [18] A.O. Petters, H. Levine, J. Wambsganss, *Singularity Theory and Gravitational Lensing*, Birkhäuser, Boston, 1999, in press.
- [19] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1986.
- [20] K. Rauch, S. Mao, J. Wambsganss, B. Paczyński, *Astrophys. J.* 386 (1992) 30–35.
- [21] S. Refsdal, *Mon. Not. Roy. Astron. Soc.* 128 (1964) 307.
- [22] A. Riess, W.H. Press, R.P. Kirshner, *Astrophys. J.* 473 (1996) 88–109.
- [23] H.N. Russell, *Sci. Amer.* 156 (1937) 76–77.
- [24] R. Schmidt, J. Wambsganss, *Astron. Astrophys.* 335 (1998) 379–387.
- [25] P. Schneider, A. Weiss, *Astron. Astrophys.* 171 (1987) 49–65.
- [26] P. Schneider, J. Ehlers, E.E. Falco, *Gravitational Lenses*, Springer, Berlin, 1992.
- [27] P. Schneider, J. Wambsganss, *Astron. Astrophys.* 237 (1990) 42–53.
- [28] C. Seitz, J. Wambsganss, P. Schneider, *Astron. Astrophys.* 288 (1994) 19–29.
- [29] D. Walsh, R.F. Carswell, R.J. Weymann, *Nature* 279 (1979) 381–384.
- [30] J. Wambsganss, 1990 Ph.D. Thesis, Munich available as MPA Report 550.
- [31] J. Wambsganss, 1998 Living Reviews in Relativity, 1/1998-12, 1–80 (<http://www.livingreviews.org/Articles/Volume1/1998-12wamb/>).
- [32] J. Wambsganss, R. Cen, J.P. Ostriker, E.L. Turner, *Science* 268 (1995) 274–276.
- [33] J. Wambsganss, R. Cen, G. Xu, J.P. Ostriker, *Astrophys. J.* 475 (1997) L81–L84.
- [34] J. Wambsganss, R. Cen, J.P. Ostriker, *Astrophys. J.* 494 (1998) 29–46.
- [35] J. Wambsganss, T. Kundić, *Astrophys. J.* 450 (1995) 19–26.
- [36] J. Wambsganss, B. Paczyński, P. Schneider, *Astrophys. J.* 358 (1990) L33–L36.
- [37] J. Wambsganss, B. Paczyński, *Astron. J.* 102 (1991) 864–868.
- [38] J. Wambsganss, *Astrophys. J.* 386 (1992) 19–29.
- [39] J. Wambsganss, *Astrophys. J.* 392 (1992) 424–431.
- [40] J. Wambsganss, B. Paczyński, *Astrophys. J.* 397 (1992) L1–L4.
- [41] J. Wambsganss, *Mon. Not. Roy. Astr. Soc.* 284 (1997) 172–188.
- [42] F. Zwicky, *Phys. Rev. Lett.* 51 (1937) 679.