

An Undergraduate's First Encounter With LENSTOOL

Mike McCourt

*Stanford University, Department of Physics, Stanford, CA 94309**

(Dated: October 22, 2006)

Abstract

In Marusa’s example, she claims to be “green.” Taking a hot-shot to be bright red, then, I must be ranked somewhere around ultra-violet. This is a documentation of my struggle to play in the visible spectrum. My goal is to model the cluster RXJ 1347.5-1145 using `LENSTOOL v6.2`, by Jean-Paul Kneib.

The pages which follow are written by an undergraduate, with undergraduates in mind and will hopefully be helpful to future students in my situation. Accordingly, it is quite a basic coverage and probably won’t be very helpful for any of you veterans. It should, however, have enough information to get a rank novice up to speed and to let him or her slap together a working model.

Needless to say, I’m not a teacher and I don’t know enough about this material to really anticipate what you’ll need to know. What I can do, however, is to document all the mistakes I’ve made and learned from and to write up all that I wish I’d known since the beginning. Many of the snags that caught me ended up being quite avoidable, and I’ll do my best to point them out, mostly in Section IV. Because I caught so many snags, there is almost certainly more information in here than you want, and I’ve surely left out some that you will need. With some luck, however, this manual will help save you a bit of time and a bit of pain.

In the first two sections, I attempt to share what I’ve come to understand about gravity lensing. I’ve found it helpful to keep a picture in mind as I work. Perhaps you’ll find this helpful, as well. If you have your own ideas about lensing and you don’t wish to hear me pontificate, skip straight to Section IV, where I go elbow deep into the tedious, technical details of using `LENSTOOL`.

I. INTRODUCTION TO LENSING

I’ve found modelling to be a tricky business (after more than a month, now, I still haven’t got a model that I’m happy with) and I’ve found it helpful to keep the “big picture” in mind. This picture is simplistic to the point of being inadequate, but I think it *is* accurate and it has helped me keep a handle on what was happening. This was particularly welcome towards the beginning of the summer, as I faced the barrage of new programs and Linux commands that you might now be fighting your way through. It seems to me, therefore, appropriate to give a quick introduction to gravity lensing itself. I have only a basic understanding of General Relativity, so please skip ahead if you are an expert. Chapters 3 and 4 of Marusa’s

book by Elhers, Schneider and Falco might be a better place for you to start.

I'll start by working out an equation of motion for light rays in a Schwarzschild geometry, parroting Hartle's section 9.4. I start from the null geodesic equation, rewrite the equation in terms more familiar to those of us who are used to Newton's laws, and then draw an analogy with geometric optics like we've seen in Physics 65.

The Null Geodesic Equation

Lensing is a simple consequence of Einstein's General Theory of Relativity— it seems, however, safer to introduce terms within the more familiar framework of Special Relativity. To describe the location of a ray or photon, it is customary to assign the 4-vector x^α . To describe the orbit of this ray, we need to parameterize its path. It makes little sense to use *time*, as we all did in Math 52, because the interval (and therefore the proper time elapsed) between any two points on the light's path is zero. We therefore write, for the curve $x = t$

$$\partial_\lambda x^\alpha = u^\alpha, \tag{1}$$

where λ is some other parameter and u^α is the 4-vector. In places where space is flat, the metric is the Minkowski one and $u^\alpha = (1, 1, 0, 0)$. Note that this is a null 4-vector and we have

$$\eta_{\mu\nu} u^\mu u^\nu = 0, \tag{2}$$

Where

$$\eta_{\mu\nu} \equiv \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

is the Minkowski metric describing the flat-spacetime of Special Relativity. This is a result which generalizes to other geometries, as well.¹ The precise form of the 4-vector u^α will depend on the parameterization λ , but *all* will have a null length. It is often simplest to choose the parameterization

$$\partial_\lambda u^\alpha = 0, \tag{4}$$

where λ is known as an "affine parameter." This is effectively a statement of Newton's Second Law and, with this parameterization, it becomes clear that photons travel in straight lines.

Quasi-Newtonian Formulation

For more general geometries, it is useful to list the conserved quantities, or *Integrals of the Motion*:

$$\begin{aligned} e &\equiv -\xi^\alpha u_\alpha \\ \ell &\equiv \eta^\alpha u_\alpha, \end{aligned} \tag{5}$$

where $\vec{\xi}$ and $\vec{\eta}$ are the Killing vectors associated with the time independence and the axial symmetry of the metric. These quantities are conserved, and you can pick the normalization of λ such that \vec{u} coincides with the momentum \vec{p} of the photon and that e and ℓ correspond to the photon's energy and angular momentum, respectively. The fact that the tangent vector to the path must be null gives us our final integral of the motion:

$$\vec{u} \cdot \vec{u} = g_{\mu\nu} \frac{dx^\mu}{d\lambda} \frac{dx^\nu}{d\lambda} = 0. \tag{6}$$

If you take $g_{\mu\nu}$ to be the Schwarzschild metric and consider an orbit which takes place in the equatorial plane $\theta = \pi/2$, the Killing Vector equations (5) become

$$e = \left(1 - \frac{2M}{r}\right) \frac{dt}{d\lambda} \tag{7}$$

and

$$\ell = r^2 \sin^2 \theta \frac{d\phi}{d\lambda}, \tag{8}$$

and (6) then gives

$$-\left(1 - \frac{2M}{r}\right) \left(\frac{dt}{d\lambda}\right)^2 + \left(1 - \frac{2M}{r}\right)^{-1} \left(\frac{dr}{d\lambda}\right)^2 + r^2 \left(\frac{d\phi}{d\lambda}\right)^2 = 0 \tag{9}$$

After using (7) and (8) to eliminate $dt/d\lambda$ and $d\phi/d\lambda$ and multiplying by $(1-2M/r)/\ell^2$, this looks like (Hartle, p. 205):

$$\frac{1}{b^2} = \frac{1}{\ell^2} \left(\frac{dr}{d\lambda}\right)^2 + W_{eff}(r). \tag{10}$$

Where

$$b^2 \equiv \frac{\ell^2}{e^2}, \tag{11}$$

and

$$W_{eff}(r) \equiv \frac{1}{r^2} \left(1 - \frac{2M}{r}\right). \tag{12}$$

This is very similar to a Physics 110 treatment of orbits, where W_{eff} is the effective potential in which the photon moves. It is interesting to note that the path of the light ray can depend only on the *ratio* ℓ/e , and on neither of the two independently. This is very much unlike the case for particle orbits, and I still find it counter-intuitive. I'm told that it has to do with freedom in the choice of affine parameters (if λ is your parameter, then $a\lambda$ gives an equally good description, for any constant scalar a); hopefully that makes more sense to you than it does to me.

Interestingly, the $1/r^3$ term in $W_{eff}(r)$ can also be used to explain the precession of Mercury's perihelion, one of General Relativity's crowning achievements. I included a few graphs I made showing a few particle orbits and a figure from Hartle showing a few light-ray orbits in this Schwarzschild geometry. Both show the mechanics of the "effective potential." Though strictly different in origin, this potential behaves much like its Newtonian counterpart and I hope this will help give you some intuition.

Quantifying Deflection

We would like to quantify the ratio $d\phi/dr$ and integrate it to get a total deflection angle $\delta\phi_{def}(r)$ as the photon moves past the lensing mass through some (Schwarzschild) distance r . We can do this by solving (8) for $d\phi/d\lambda$ and (10) for $dr/d\lambda$ and dividing the two (this trick still seems like magic to me...).

$$\frac{d\phi}{dr} = \pm \frac{1}{r^2} \left(\frac{1}{b^2} - W_{eff}(r) \right)^{-1/2}. \quad (13)$$

If you want to try and integrate this numerically, the \pm makes it a bit tricky, but it's a good opportunity to flex your Mathematica muscles. Fortunately, the situation is entirely symmetric about the sign flip, and we can integrate from ∞ to r_{min} and simply double the result. Hartle quotes (though I must admit to having some difficulty reproducing it)

$$\Delta\phi = 2 \int_{r_1}^{\infty} \frac{dr}{r^2} \left[\frac{1}{b^2} - \frac{1}{r^2} \left(1 - \frac{2M}{r} \right) \right]^{-1/2}. \quad (14)$$

With the change of variables $r = b/w$, the integral becomes

$$\Delta\phi = 2 \int_0^{w_1} \left[1 - w^2 \left(1 - \frac{2M}{b}w \right) \right]^{-1/2} dw, \quad (15)$$

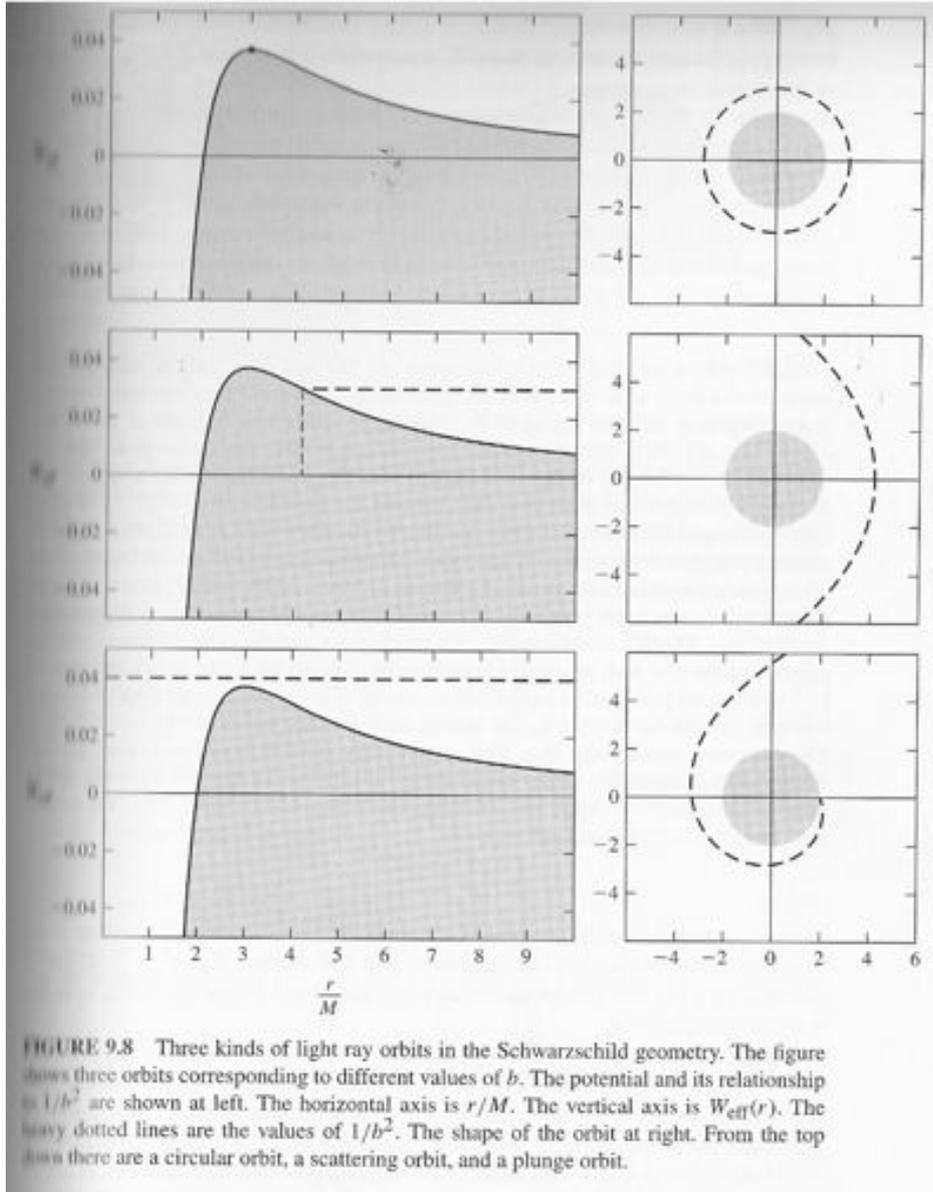


FIG. 1: Hartle's figure showing light deflections in the effective potential.

where w_1 is the value which makes the quantity in the brackets vanish.

When the ratio M/b is small, the integral can be Taylor-expanded (see Hartle for the details), yielding the deflection

$$\delta\phi_{def} = \frac{4M}{b} = \frac{4GM}{c^2b}. \quad (16)$$

An important thing to take from the above equation is that a perfect point mass *cannot* produce multiple images of a perfect point source. Of course, these are both unrealistic

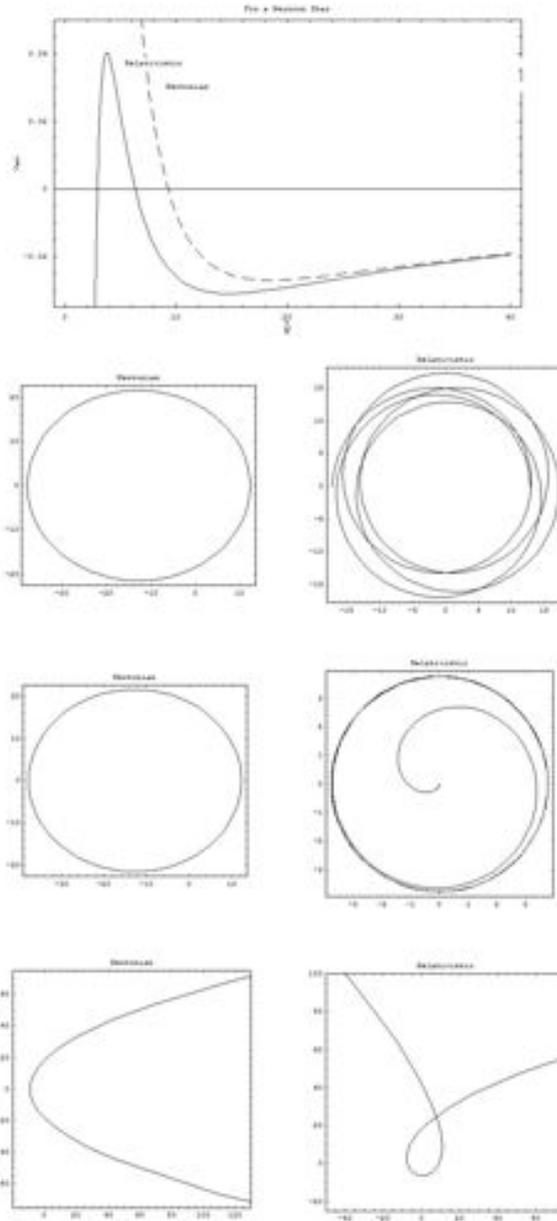


FIG. 2: Another plot of the two potentials and a few comparisons between Newtonian and Relativistic gravity for particle trajectories.

approximations, but there *is* a worthwhile scenario to consider here.

The deflection angle must always be measured in the plane of the light's path and the point mass. When the light source is immediately *behind* the lensing mass, this situation is degenerate and there is no special plane in which to determine the angle. The light, then, is deflected in *all* directions by this angle and its image becomes a ring with the angular

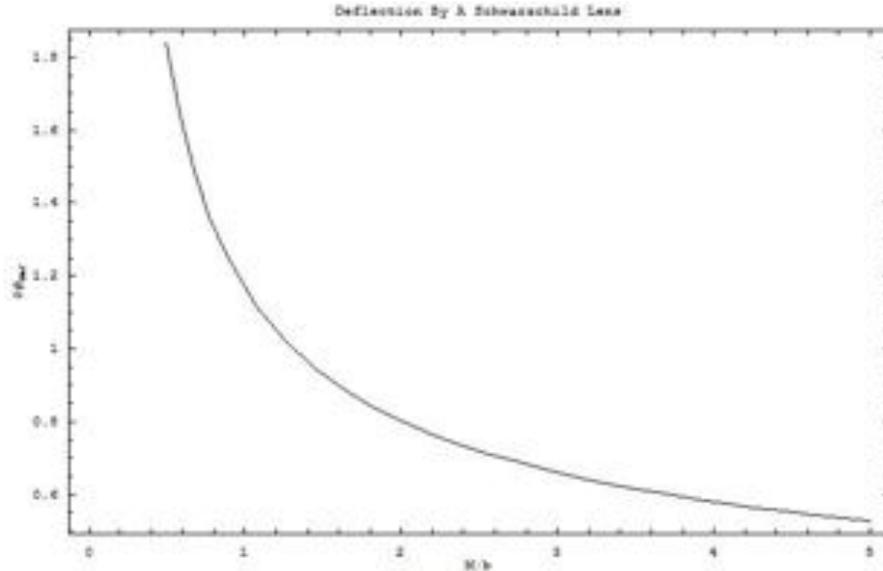


FIG. 3: plot of $\delta\phi_{def}$ against M/b for a Schwarzschild Lens.

diameter $\delta\phi_{def}$. This diameter depends only on the redshift of the source and the mass of the lens and can give a very accurate estimate of the mass inside. This is an unlikely scenario and few such examples have been found, but they *do* (at uncertainties of a few percent) constitute some of the most accurate measurements made in extragalactic astronomy. This is an example of how gravitational lensing can be used to do useful astronomic analysis.

Maybe I should also note that this result carries some historical significance; in May of 1919, Sir Arthur Eddington went to the island Principe to photograph the Sun during an eclipse. Comparison of the locations of stars in the photograph with their known positions indicated that they had been shifted by just the $\delta\phi_{def}$ we predicted. This wasn't the first experimental test of General Relativity, but it was the first test of a *prediction* of the theory. The result was fundamentally different than any that could be reproduced in a Newtonian theory of gravity (by a perfect factor of 2, intriguingly) and the finding won much support for Einstein's theory of gravity.

II. AN ATTEMPT AT THE GENERAL SCENARIO

We now have a very clear picture of how light rays are lensed in a Schwarzschild geometry. This result bears some significance, both historically and scientifically, but it would be extremely useful to work out a result for general geometries. Most objects in the universe

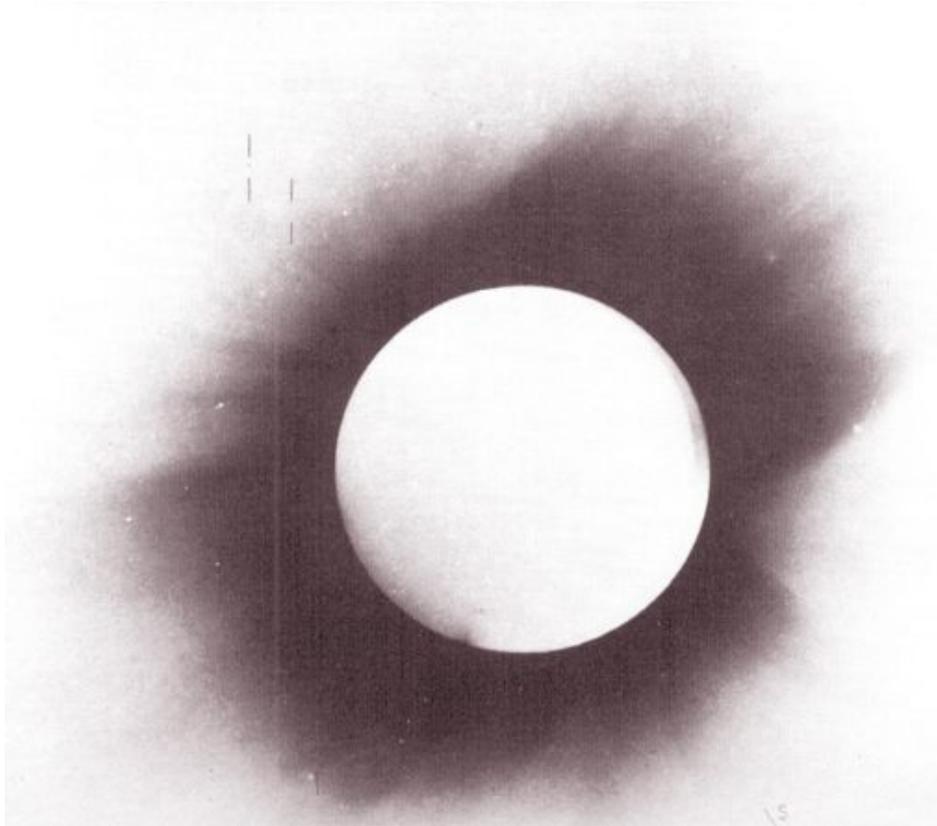


FIG. 4: Sir Eddington’s original data, marking the locations of stars with black lines.

are not point masses, however, and the equations of General Relativity are nonlinear, so we cannot simply “build up” a general solution from the one we have (the way we can for electrostatics and Newtonian Gravity, for example...). Needless to say, the general case is much, much harder than the Schwarzschild one, and this is about where I come to really appreciate the difference between a Ph.D. and half a degree. This section, then, will be a quick summary of two of the chapters of Elhers, Schneider and Falco. I’ll try to explain what I think I understand and I’ll mention what I think is important, but if you are able to follow the book, it will be a much better place for you to start.

Suitably humbled, then, I’ll continue.

Maxwell’s Equations And The Field Strength Tensor

Understanding how light behaves in a particular geometry comes down to solving Maxwell’s equations on the manifold describing that geometry. We know that Maxwell’s

equations *must* be relativistically correct (that's how the whole business of relativity got started), but we need to generalize them to a curved geometry. In the process, we'll make the equations explicitly independent of coordinates, which I think makes their meaning much clearer. So we start by writing the equations:

$$\begin{aligned}
\nabla \times \vec{B} - \partial_t \vec{E} &= \vec{J} \\
\nabla \cdot \vec{E} &= \rho \\
\nabla \times \vec{e} - \partial_t \vec{B} &= 0 \\
\nabla \cdot \vec{B} &= 0
\end{aligned} \tag{17}$$

in component notation:

$$\begin{aligned}
\epsilon^{ijk} \partial_j B_k - \partial_0 E^i &= J^i \\
\partial_i E^i &= J^0 \\
\epsilon^{ijk} \partial_j E_k - \partial_0 B^i &= 0 \\
\partial_i B^i &= 0
\end{aligned} \tag{18}$$

To simplify these, we can define the Maxwell Field Strength Tensor:

$$F_{\mu\nu} \equiv \begin{bmatrix} 0 & -E_1 & -E_2 & -E_3 \\ E_1 & 0 & B_3 & -B_2 \\ E_2 & -B_3 & 0 & B_1 \\ E_3 & B_2 & -B_1 & 0 \end{bmatrix} = -F_{\nu\mu}, \tag{19}$$

which can be expressed in terms of a (4-)vector potential A_μ :

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu, \tag{20}$$

with the guage invariance

$$\begin{aligned}
A_\mu &\rightarrow A_\mu + \partial_\mu \lambda(x) \\
F_{\mu\nu} &\rightarrow F_{\mu\nu} + \partial_\mu \partial_\nu \lambda - \partial_\nu \partial_\mu \lambda = F_{\mu\nu},
\end{aligned} \tag{21}$$

because partial derivatives generally commute. Maxwell's Equations then become

$$\begin{aligned}
\partial_\mu F^{\nu\mu} &= J^\nu \\
\partial_{[\mu} F_{\nu\lambda]} &= \frac{1}{3} (\partial_\mu F_{\nu\lambda} + \partial_\nu F_{\lambda\mu} + \partial_\lambda F_{\mu\nu}) = 0,
\end{aligned} \tag{22}$$

in terms of the 4-current $J^\nu = (\rho, \vec{J})$, where $\rho(t, x, y, z)$ is the local charge density and \vec{J} is the current 3-vector.

These are the equations of motion for electromagnetic fields on curved manifolds and contain all the (classical) information about how light will behave. Maxwell's equations were originally derived partly through experiment and partly through inspiration, but I found it very helpful to go back and rederive them from an action principle. This is an extremely powerful method which generalizes to different types of fields and is good to know about if you hope to follow current literature in cosmology. To start, you make the prescient choice of the Lagrangian:

$$\mathcal{L} \equiv -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + A_\mu J^\mu. \quad (23)$$

The first of (22) can then be taken from the action principle

$$\frac{\partial \mathcal{L}}{\partial A_\nu} - \partial_\nu \left(\frac{\partial \mathcal{L}}{\partial(\partial_\mu A_\nu)} \right) = 0, \quad (24)$$

while the second follows (nearly) trivially from the antisymmetry of $F^{\mu\nu}$ and the fact that partial derivatives commute. So we have turned the fairly unintuitive (though beautiful) set of Maxwell's equations into a definition and a statement of extremal action. These fundamentally describe the same physics, but the tensor notation is extremely efficient and explicitly independent of coordinates and is therefore very appealing.

The efficiency of tensor notation can be even better demonstrated by constructing the Energy-Momentum Tensor (see Schutz for more information):

$$T^{\alpha\beta} \equiv F^{\alpha\gamma}F_{\gamma}{}^{\beta} + \frac{1}{4}g^{\alpha\beta}F_{\gamma\delta}F^{\gamma\delta}, \quad (25)$$

and observing that

$$\begin{aligned} T^{00} &= \frac{1}{2}(\vec{E}^2 + \vec{B}^2) \\ (T^{01}, T^{02}, T^{03}) &= \vec{E} \times \vec{B} \\ T^{ab} &= -E_a E_b - B_a B_b + \frac{1}{2}\delta_{ab}(\vec{E}^2 + \vec{B}^2) \end{aligned} \quad (26)$$

You'll recognize these three equations as the Energy Density, the Poynting Vector and the Maxwell Stress Tensor of the electromagnetic fields. So all four of Maxwell's equations are fit into two, and these three quantities are all unified into one tensor. Independence of coordinates means that we can always set up and solve the problem at hand in the rest-frame (locally Minkowski) coordinates of the test particle, then transform them back into the coordinates of interest.

Optics In Curved Space-Time I

The next logical step, I think, to extend our picture of how lensing works is to use our result for single rays in a Schwarzschild geometry to try and see how a *bundle* of nearby rays behaves in this same geometry. All we need to do is solve the very same geodesic equation for a *system of rays*. This should enable us to work out things like magnification, distortion and shear. The geodesic equation, again, is as follows:

$$\frac{d^2 x^\alpha}{d\lambda^2} + \Gamma_{\beta\gamma}^\alpha \frac{dx^\beta}{d\lambda} \frac{dx^\gamma}{d\lambda} = 0. \quad (27)$$

This can be rewritten as:

$$k^\alpha{}_{;\beta} = 0, \quad (28)$$

where $k^\alpha = \partial_\lambda x^\alpha$ is the wave-vector of a ray. Notice that the form of k^α is strictly dependent on the parameterization used.

Now we can express our bundle of rays as a list of functions $x^\alpha = f^\alpha(\lambda, y^a)$. (Any coordinates are fine, but you could pick y^1 giving the phase difference, and y^2 and y^3 giving the separation angles distinguishing your particular ray from the one picked as the origin). If two consecutive rays are separated by some small δx^α , the wave fronts (planes of constant phase) are given by:

$$\begin{aligned} k_\alpha \delta x^\alpha &= 0 \\ g_{\alpha\beta} \frac{\partial x^\beta}{\partial \lambda} \delta x^\alpha &= 0 \end{aligned} \quad (29)$$

The book claims that the transformation

$$\begin{aligned} \delta \tilde{x}^\alpha &= \frac{\partial f^\alpha}{\partial y^a} \delta y^a + k^\alpha \delta h \\ &= \delta x^\alpha + k^\alpha \delta h \end{aligned} \quad (30)$$

corresponds to a Lorentz boost into a different frame. Lamentably, I was unable to reproduce the result. It seems to me like all it is is a phase shift of adjacent rays... By taking this transformation to be true, however, and plugging into it *three* nearby rays, you can show that

$$g_{\alpha\beta} \delta \tilde{x}_1^\alpha \delta \tilde{x}_2^\beta = g_{\alpha\beta} \delta x_1^\alpha \delta x_2^\beta, \quad (31)$$

meaning, I think, that different observers will agree on the relative phases (and therefore on the orientation of the wave-front) of nearby rays. This seems too weird to be true, though, so it might be an incorrect interpretation...

I wasn't personally able to wade much deeper into the theory end of things. It's interesting stuff and I beseech you to supplement this if you care to. I've taken full stock of my limitations, for now, and have merely to quote "It can be shown that:"

$$k_{\alpha;\beta} = \sigma_{\alpha\beta} + \theta P_{\alpha\beta} + P_{(\alpha k\beta)}. \quad (32)$$

With a prescient choice of $\sigma_{\alpha\beta}$ and θ and P . If we consider 2 rays separated by a length ℓ , we know that

$$\begin{aligned} \ell^2 &= g_{\alpha\beta} \delta x^\alpha \delta x^\beta \\ &= 2k_{\alpha;\beta} \delta x^\alpha \delta x^\beta \\ &= 2(\sigma_{\alpha\beta} + \theta P_{\alpha\beta} + P_{(\alpha k\beta)}) e^\alpha e^\beta, \end{aligned} \quad (33)$$

Which is equivalent to:

$$\boxed{\frac{1}{\ell} \frac{d\ell}{d\lambda} = \theta + \sigma_{\alpha\beta} e^\alpha e^\beta}. \quad (34)$$

Here the first term corresponds to an isotropic expansion of your image, while the second is a distortion, or "shear," which turns a circular cross-section into an elliptical one. This is really the only thing to take out of this section; I find it a really neat result and it turns out to be quite descriptive of how gravity lensing works.

Optics In A Curved Space-Time II

*This is where we finally start to solve for general geometries. The approach is somewhat reminiscent of geometric optics and involves no tensors and gives equations which those of us who don't hold degrees can pretend to understand. Really, just by looking at the following figure, you can immediately derive the lens equation attacked by **LENSTOOL**. If you keep in mind that $\alpha(\vec{\theta})$ is a quantity which must be inferred from the data, then you'll see how the data can be coupled to the geometry (and, consequently, the mass distribution) of the lens.*

By requiring the distances to add in the source plane (these are all at the same redshift, so distances *do* add...), you can immediately see that:

$$\theta D_l = \beta D_s + \alpha D_{ls}. \quad (35)$$

This is the so-called "lens-equation" that you solve when you model a cluster.

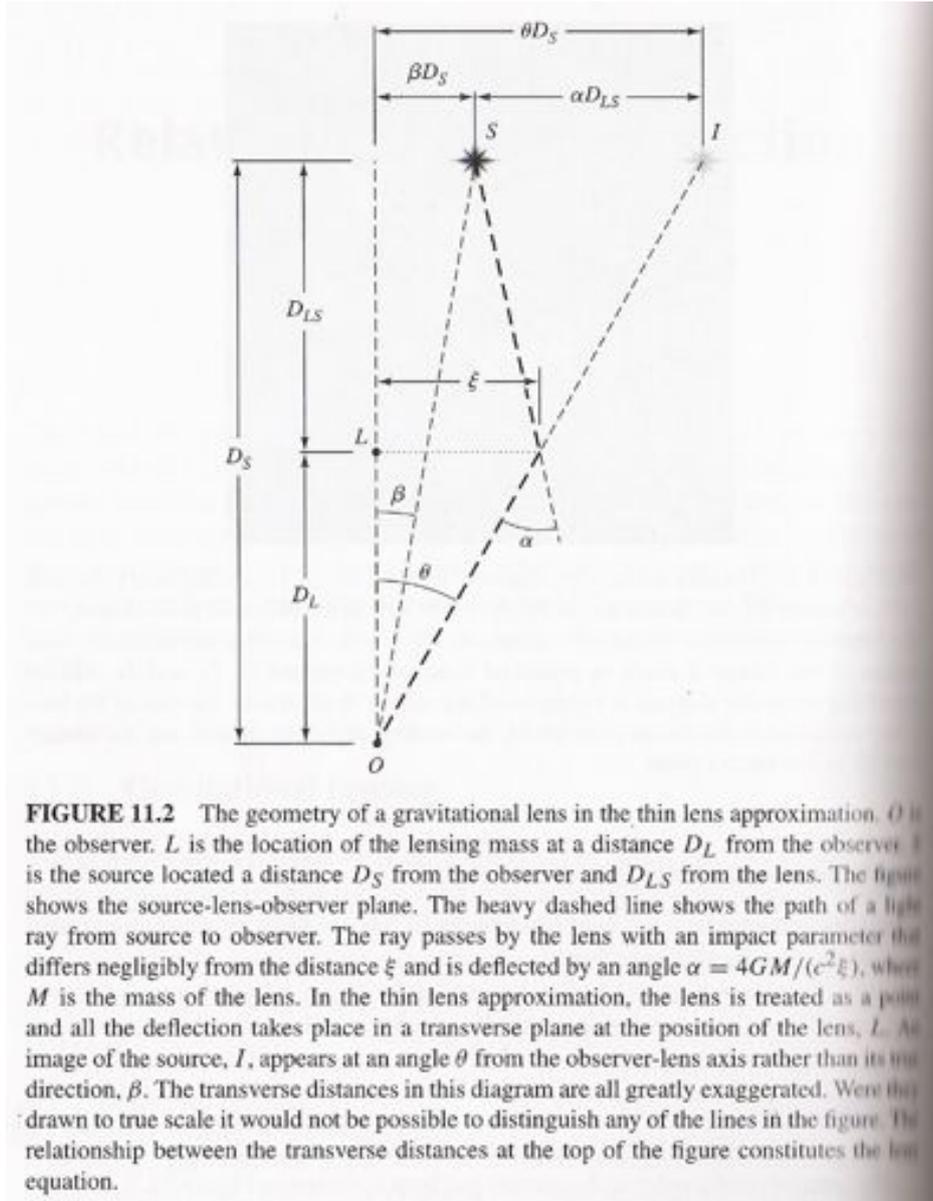


FIG. 5: Geometry of a lensing system.

That part was easy. . . . What is unfortunate, however, is that *both* $\vec{\alpha}$ and $\vec{\beta}$ are unknown. Working out the relation to $\vec{\theta}$ is difficult and what follows is all taken from Marusa's textbook.

I would dishonest if I were to pretend to be able to derive the following, so I'll just tell you to look in Elhers, Schneider and Falco if you're curious:

$$\alpha(\vec{\theta}) = \frac{1}{\pi} \iint_{R^2} \kappa(\vec{\theta}') \frac{\vec{\theta} - \vec{\theta}'}{|\vec{\theta} - \vec{\theta}'|^2}, \quad (36)$$

where

$$\kappa(\vec{\theta}) \equiv \frac{\Sigma(D_l \vec{\theta})}{\Sigma_{cr}}, \quad (37)$$

is the surface mass density, with

$$\Sigma_{cr} \equiv \frac{c^2}{4\pi G} \frac{D_s}{D_l D_{ls}} \quad (38)$$

is the ‘‘critical’’ mass density above which multiple images can be produced.

From (16), we know the form of the angle $\vec{\alpha}$ for a point-mass geometry, provided the impact parameter b of the photon isn’t too close to R_s :

$$\vec{\alpha} = \frac{4GM}{c^2 b} \equiv \frac{2R_s}{b}. \quad (39)$$

With more general mass distributions, we must infer $\alpha(\vec{\theta})$ from the multiply-imaged systems. Because we don’t know locations in the source plane ($\vec{\eta}$ or $\vec{\beta}$ in the figure), this leads to a messy system of the non-linear equations. In principle, though, the two are related and a thorough knowledge of $\vec{\alpha}(\vec{\theta})$ correlates to a complete understanding of $\kappa(\vec{\theta})$.

Because it is easier to work with scalar quantities than with vectors, and because $\nabla \ln |\vec{\theta}| = \vec{\theta}/|\vec{\theta}|^2$ for any vector $\vec{\theta}$ in R^2 , we can, without loss of information, consider only the quantity ψ :

$$\psi(\vec{\theta}) \equiv \frac{1}{\pi} \iint_{R^2} \kappa(\vec{\theta}') \ln |\vec{\theta} - \vec{\theta}'| d^2 \vec{\theta}'. \quad (40)$$

Any theory defined in terms of the lensing potential ψ will be isomorphic to, and potentially much simpler than, one in terms of $\vec{\alpha}$. To get our old quantities back, note that $\nabla^2 \ln(\vec{\theta}) = 2\pi\delta(\vec{\theta})$, and we can use

$$\begin{aligned} \nabla\psi &= \vec{\alpha}(\vec{\theta}) \\ \nabla^2\psi &= 2\kappa. \end{aligned} \quad (41)$$

It’s easy to get lost in all this, so maybe I should restate the problem. The game is to work out $\vec{\alpha}(\vec{\theta})$ from the data and to then solve (numerically) the differential equation for ψ , which then lets you compute κ , the quantity of interest.

To quantify the distortion of images, you can assume that the image is much smaller than the scales over which κ changes (you can always break up your image if this is not the case)

and take distortion effects only to first order terms. Define the transformation matrix:

$$\begin{aligned}
A(\vec{\theta}) &= \frac{\partial \vec{\beta}}{\partial \vec{\theta}} = \frac{\partial}{\partial \vec{\theta}} (\vec{\theta} - \vec{\alpha}(\vec{\theta})) \\
&= \left(\delta_{ij} - \frac{\partial \nabla \psi}{\partial \vec{\theta}} \right) \\
&= \left(\delta_{ij} - \frac{\partial^2 \psi}{\partial \theta_i \partial \theta_j} \right) \\
&= \begin{bmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{bmatrix}, \tag{42}
\end{aligned}$$

where $\gamma_1 \equiv \frac{1}{2}(\psi_{,11} - \psi_{,22})$ and $\gamma_2 \equiv \psi_{,12}$.

We can now define a magnification factor²

$$\mu = \frac{1}{\det A}. \tag{43}$$

The image of a point source will be brightened by a factor of $|\mu|$ at its location. Note that there are certain places where $\det A$ vanishes and where the magnification formally diverges. The locus of these points makes your critical curves, and μ changes sign across the curve. I'm not good enough to prove it to you mathematically, but maybe you can convince yourself that a sign flip in magnification corresponds to a parity inversion in the morphology of the image (have you ever scaled things negatively in Photoshop?). This is an important feature characterizing image breaks. Images near these curves are also highly magnified and highly distorted, making them easier to pick out by eye. This magnification also means that some otherwise invisible objects have been spotted by the Hubble telescope.³ Lensing preserves spectra, so the atomic constituents of these high-redshift sources can easily be investigated,⁴ but it requires good lensing analysis to undo the distortions and work out the real morphologies of the early galaxies.

Now I can rewrite A:

$$A(\vec{\theta}) \equiv (1 - \kappa) \begin{bmatrix} 1 - g_1 & -g_2 \\ -g_2 & 1 + g_1 \end{bmatrix}, \tag{44}$$

where

$$g \equiv \frac{\gamma}{1 - \kappa} = \frac{\gamma_1 + i\gamma_2}{1 - \kappa} \equiv g_1 + ig_2. \tag{45}$$

The factor of $(1 - \kappa)$ should be familiar; it is our isotropic expansion of the image, θ . The matrix behind it stretches your image; by finding its eigenvectors and values, you

can find that a circular source is mapped to an ellipse with radii $|(1 + \kappa)(1 + |g|)|^{-1}$ and $|(1 + \kappa)(1 - |g|)|^{-1}$ and an angle matching the phase of g . This matrix is our $\sigma_{\alpha\beta}$ from above.

Your critical curves should always be smooth and well-behaved, but the caustics (mappings of the critical curves back into the source plane) needn't be. To see why, parameterize one of your critical curves with the function $\vec{\theta}(\lambda)$ and map its tangent vector into the source plane via:

$$\begin{aligned} \frac{d\vec{\beta}(\vec{\theta}(\lambda))}{d\lambda} &= \frac{\partial\vec{\beta}}{\partial\vec{\theta}} \frac{d\vec{\theta}}{d\lambda} \\ &= A(\vec{\theta}(\lambda)) \cdot \dot{\vec{\theta}}(\lambda). \end{aligned} \tag{46}$$

This vector is the tangent to your caustic curve and is null whenever $\dot{\vec{\theta}}(\lambda)$ is parallel to the null eigenvector of A (the direction along which A^{-1} is singular). At these locations, the caustic curve needn't be smooth and can show a sharp cusp.

That's the extent of my knowledge of the theory. Admittedly, there's much more to lensing, but I think that this is an adequate starting point. Remember that you don't really need to know *all* the theory to do good modelling.

Hopefully, by now, you should have a decent picture in your head of what is happening when a source is gravitationally lensed. When it comes to actually *inverting* all these equations and working out a mass distribution from raw data, this gets really hard and starts to make my head hurt. I'm glad to have LENSTOOL to work out the relations for me.

III. INTRODUCTION TO LENSTOOL

Why Use LENSTOOL

It might now seem like a simple matter to work out a mass from a measured deflection (or, at least, it might not seem obvious why it should take a fast linux machine many hours to do so...). Unfortunately, the mass distributions with which we are concerned are seldom Schwarzschild, even to an approximation, and the equations become much more difficult to solve from here. The equations are nonlinear and nasty and we use the optimization routine in LENSTOOL to approximate the mass distribution we seek.

Additionally, unless you are working on microlensing, where there can be some observable, relative movement between the source and the lens, it is usually impossible to know $\vec{\eta}$, the “true” position of the source. We can only extract useful information about sources, therefore, if they are multiply imaged and we know the source’s redshift.⁵ These are the constraints that can be fed into LENS`TOOL`. (see the help file for my program `ImageFinder` for more on the characterization of multiple images.)

It is probably a good idea to take a day or so to just look at the image (use `ds9`’s `rgb` mode and change color scales frequently) to get an idea about what images might be of the same source. Remember from above that surface brightness and color are conserved, that image morphology is inverted across critical curves, and that images near these breaks are probably the easiest ones to find.

General Observations About LENS`TOOL`

Hopefully, it will seem obvious to you, but it took me some time to realize that the operation of LENS`TOOL` actually has nothing to do with any of your images—it operates entirely with virtual, fictitious ellipses, which may or may not bear any relation to your actual data. (This is my plug to be careful with your coordinates; it’s all spherical trig and I found it very easy to get things mixed up. Use `pell` to check your catalogs often and always use the ruler tool in `ds9` to measure distances.(there’s a factor of \cos of *Dec* to worry about, but the ruler tool takes care of you here.)).

Before anything else, you need to tell LENS`TOOL` what type of model to start out with, which of the parameters you wish to vary and how much freedom to allow them as they vary. To do this, you must tell the program the locations, shapes and magnitudes of all the cluster members you wish to consider—you needn’t include all of them; just the 20 brightest (or so) and any that happen to lie nearby your image breaks. All but the brightest one or two should prove utterly irrelevant to the total mass of your cluster, but even the dim ones can shift critical curves more than you might expect. For this application, it’s probably easiest to find these members by eye, but Douglas Stanford wrote a very effective program to do this automatically. I explain how to tell LENS`TOOL` to vary these parameters in the section below on the `PAR` file.

After specifying the cluster members, you must produce a catalog of your multiple images,

which will be used as constraints for the minimization, and a catalog of the arclet files, for which counterimage locations will be predicted. For these, you really only need to know the positions of your images; though shapes and sizes may not help much, they come free from SExtractor and they certainly can't hurt. If you want to get a quantitative evaluation on how good your multiple image pairs are, it's helpful to have color information, as well. The format for inputting this information into the program is simple enough; you can have a look either at my example files or the appendix of the LENSTOOL manual for rules. Note that the cluster members file is sorted by magnitude.

You should also have a look at the pamphlet written by Marusa, Professor Burchat and Phil on "Weak Lensing for Students" for information on using SExtractor for producing catalogs of images from and for doing photometry on your data.

IV. PAR FILE

Here is where you put all the information for your model. All of it. A month's worth of work fits onto a single page. If that makes you feel small, think about the distance to the cluster you're modelling. Then think about dark matter... Then dark energy...

I should also note that I took much, if not most, of what follows from Marusa's document, so this will seem repetitive if you've already read hers. That's probably not too terrible of a thing, however, because I had to read Marusa's example a few times before having a clue what to do with LENSTOOL.

This is my last opportunity to warn you that the next 10 pages (or so) will be exceedingly boring (more so, even, than the last 10 pages...). Though I *did* leave out certain details such as "Must be an odd number, typically 20 or 30,"⁶ my goal is to be as thorough and as clear as possible, which sometimes means going through potentially irrelevant details. Take what you need and skip what you don't, but if you bear with me for an hour or so, you'll have a parameter file. Keep in mind, as you go, that you really only need to do this once.

runmode

This is the most straightforward section, where you tell LENSTOOL how to run.

reference 3 206.87766 -11.752628

This gives your point of origin. Unfortunately, LENSTOOL works in relative arc-seconds, and not in world coordinates. Often, you can specify your coordinate system and have LENSTOOL do the conversion, but sometimes you cannot (with **critic** and **potentiel**, for example). The *integer* 3 (take care not to write 3.0) stands for “degrees”, meaning that the ra and dec which follow are given in degrees. Once you pick your origin, of course, take care that you don’t forget it.

arclet 3 *image.cat*

Here you list the arcs for which LENSTOOL will predict counterimages. Comparison of these counterimages with the actual data should be your best means of evaluating your model (tell me if you find another...). It will also provide you with a welcome method for locating new pairs of images. Most often, I just used my **multfile** for this (see below), but sometimes it’s good to keep them separate (when you consider arcs which may not be multiply imaged, for instance).

Note that I used 3 again to specify that the images are given in the WCS and to have LENSTOOL do the appropriate conversion.

mult_wcs 1

It seems redundant, but my model didn’t work without it. Maybe you’re a programmer and you know this, but throughtout this file, 1 means “true” and 0 means “false.”

source 0

In here you specify the positions of the sources (η), from above. You would use this identifier to make demonstrations or simulations, but we don’t know the source positions, so use 0 here. Source positions will be calculated from the image positions, once a model is obtained.

inverse 3 100

This is where you tell the program to optimize; this is what chews up your processor, but it is also what gives you answers. You can turn it off (0 in front) if you’re doing a simulation, or if you already have a model and just want to try different arclet files. If you want to *produce* a model, however, you *must* turn it on (this, too, took me a while to figure out). The options are as follows:

- If $int1 = 0$, no optimization is done.
- If $int1 = 1$, the program optimizes with the parabolic method, running for $int2$ steps.
- If $int1 = 2$, the optimisation minimizes only for the galaxy scale parameters sigma and cut radius. The optimisation method for the lens model parameters is the parabolic one, with $int2$ playing the same role.
- If $int1 = 3$, LENSTOOL optimizes with its Bayesian method. The optimisation stops when the counter comes back to 0. This optimisation method is very slow (\sim one hour per degree of freedom) but is less sensitive to local χ^2 minima than the parabolic method. $int2$ is neglected by this method (but still needs to be there...)

The optimisation process will create a file “best.par.” This is an inputfile containing LENSTOOL’s favorite parameters. The χ^2 for the model appears in the header as a comment.

Additionally, if you run the Bayesian minimization, the program will output a series of files labeled “Bayes.(#)-(name).dat,” where (#) gives the name of the clump for which the numbers apply (names are all numbers, to LENSTOOL) and (name) gives the parameter which the numbers approximate. A single line, taken from *all* the different files corresponds to a single, *acceptable* model (it can be disheartening when you get hundreds...). It is a good idea to plot these against eachother and see how well constrained you are in parameter-space.

minchi 1

This is the value of χ^2 at which the minimizer will stop, if he gets there before taking $num2$ steps. I don’t understand LENSTOOL’s χ^2 ’s very well, so I always left this at an ever-hopeful 1.00... It doesn’t seem to matter very much, but feel free to change it, if you know what you’re doing.

verbose 1

This tells the program how much output to give to the console. Setting it at 1 left me with plenty of reading material; 0 gave me next to none. I think 2 gives you something in between, but I haven’t tried it.

end

grille

Here you tell LENSTOOL how many cluster members there are. It seems like this could have been combined with another section, but...

polar 0

It seems silly to use polar coordinates for things which aren't axially symmetric, and real mass distributions are *never* axially symmetric. I think I read that somewhere...

nlens 32

This is just the total number of items you put in your cluster members file (see below). In my experience, it hasn't been *that* important to get this number right...

nlens_opt 2

This is the number of objects for which you run the minimization. Remember how long this can take and keep the number as low as you can. It's also good not to give the optimizer too much freedom at first, because it likes to do crazy things. (You'll soon find out what I mean)

nlens_crit 32

This doesn't appear in the LENSTOOL manual and I must admit I'm not quite sure what it does. I know that it has to do with the drawing of critical curves and I know that it doesn't seem to cause any problems when I just leave it at **nlens**...

end

cline

Here you tell LENSTOOL how to draw the critical curves. For each one you tell it to draw, you get 4 files: inner and outer critical and caustic curves. Recall that critical curves are the locations of infinite magnification, while caustic curves are the mappings of these locations

onto the source plane. Outer curves give tangential shears, while inner ones give radial ones. Remember that your critical curves should always be smooth, but the caustics are free to (and undoubtedly will) go wild.

If it's not completely obvious to you what to do with these files, you're not alone. Running `pcl` is probably the easiest way to look at them (I made the mistake of asking how it works and just got very confused); it lays the curves right over your data in `ds9` (assuming you got the coordinates right). `pcl` requires a file `frames.pl` to be copied from "`$GROUP_SOFT_DIR\lenstool\perl`" to wherever you're storing your modelling files. This file also has an origin in it (`$ra` and `$dec`) which needs to be set to your definition in **runmode**.

Alternatively, you can use `SuperMongo` to plot your curves. This method, though, is substantially more difficult and doesn't even look as nice.

nplan *4 0.90 0.95 1.10 1.15*

The first *integer* lists the number of planes for which to draw the curves. The following *floats* list the redshifts of these planes. If you're playing around with these much, it's a good idea to turn off the minimization. (Make sure, though, that you run `best.par`, and not your modelling file. Otherwise, you'll be looking at curves for outdated parameters.)

dmax *150.0*

This lists the maximum radius (in arc-seconds) from the center (as defined in **runmode**) to calculate the curves. I've noticed absolutely *no* difference in run-time based on this parameter, so be generous here; it's a hard error to catch when your curves exceed this boundary and start having weird shapes.

pas *2.0*

This is the accuracy with which the critical curves are calculated, but it is only used for certain algorithms.

algorithm *marchingsquares*

There are 2 options here: *marchingsquares* and *snake*. Snake resembles (to my imagination) Euler's method: it starts out at some point and traces out the curve using derivatives and such. There is no guarantee that the curve will close. If you are using

certain releases of LENSTOOL, this is your only option.

If you are using release 6.2 (or, presumably, some others, as well. . . check Jean-Paul’s website), you have the choice of *marchingsquares*. This method breaks up your image into a series of cells and calculates the (I think?) determinant of A at each of the vertices. Because there are only a finite number of different ways that a curve can go (or not go) through four vertices (I think the number is 16, but all that matters is that it’s finite), it is possible to figure out where the curve goes by considering just this magnification. The program probably has a lookup table for each of the different cases and determines which sides of the cell the line must go through. I’m not sure whether the algorithm starts with small cells, or starts with big ones and subsequently subdivides them, or whether it pulls some tricks with derivatives, but it’s fast and seems to give very good resolution.

This is a really neat way to find level sets, but it isn’t immune to ambiguities, particularly if the curve changes rapidly over the scales of the cell sizes. Even with the sharp cusps on the caustic curves, however, LENSTOOL’s implementation works very well.

end

potentiel (name)

It’s neat that you can spell most of these things in French and have them still work. This one didn’t seem to work for me when I spelled it in English.

Note that you can include a name in the section definition. LENSTOOL will ignore your name completely and redefine it with a number, but it helps keep your input file intelligible.

profil 12

12 gives an NFW profile, the supposedly universal one (don’t ask why it isn’t the only option. . .). It’s a bit of a pain to convert its parameters to real numbers, so have a look at my appendix on the matter, if you use this one. Marusa always suggested I use profil 3, or the “Non-Singular, Isothermal Sphere” (I think you need to specify **alpha** = 0.5 for that one to work. . .), but I was never able to get a sensible model from it. Supposedly, though, it’s easier to convert to real numbers.

x_centre_wcs 206.87766

y_centre_wcs -11.752628

Bet you can guess what these do. Note that, when you minimize for them (see next section), you specify relative measurements, not WCS ones.

ellipticite 0.070

angle_pos 110.052

core_radius_kpc 181.308

v_disp 1658.312

NFW profiles don't have central velocity dispersions (they are singular at the origin), so this isn't really a "honest" number. It's a characteristic one, having to do with the central density, so you can't compare this number directly with velocity dispersions coming out of different models. (see Section VII).

z_lens 0.45

Simple enough.

end

limit (name)

Here you tell the minimizer how to consider what you defined in the last section. In all cases, the first *integer* tells the program how to interpret what follows (0 means no variation at all, while 1 means "two hard bounds;" 2, 3, and 4 indicate varying degrees of hardness). The following *floats* tell the program your lower and upper bounds, as well as the precision with which to find the answer, respectively.

x_centre 1 -5 5 0.1

y_centre 1 -5 5 0.1

Marusa has told me over and over again that no purely dark clumps exist and that the dark matter halos are always very nearly centered on the baryonic bright spots. Still,

every time I set these limits, I hope to find that untrue. Lamentably (and, perhaps, obviously), Marusa seems to be right.

ellipticite *1 0.01 0.3 0.01*

The minimizer likes to make radically elliptical profiles, for some reason or another. These occur, to my understanding, very infrequently in nature, so it's ok to put strict limits on this one.

angle_pos *1 0.0 180.0 1.00*

Doesn't really matter, since the ellipticity's so small, but LENSTOOL really appears to like this one. I think it gave me 5 digits of precision.

core_radius *1 15.0 60.0 0.100*

Note that this is in arc-seconds, and not in kpc, as defined above. You'll need to exercise some cosmology (see Section VIII) to do this conversion.

end

You need to give one of each of the above, **potentiel** and **limit**, for each main component you wish to minimize for. This should probably only be one or two members, so it's no big deal.

potfile

Here come the rest of the cluster members. These ones will be minimized as an "ensemble" and should have only negligible contributions to the cluster's total mass (if you find this to be untrue, you should minimize for them), but they will affect your critical curves and, consequently, your model's predictions for multiple images.

filein *3 RXJ1347_members.lenstoolcat*

A long and ridiculous filename, I know... Note, again, that 3 means I store these objects in WCS coordinates. This file has one object per line, and the lines are sorted by magnitude. Magnitudes are a logarithmic measure (for somewhat silly historical reasons... see Section X), so keep in mind that it's really only the first few that

matter. Also keep in mind that it's fairly important to get these numbers "right,"⁷ so try to do your SExtractor work carefully.

type 81

This is Kneib's PIEMD "Pseudo-Elliptical Isothermal Mass Density Profile." See Section VII on what that means and how to convert it to a mass. I've found that this runs the fastest and is therefore the best for this application. It doesn't seem to matter a whole lot, though... take your pick here.

zlens 0.451

mag0 20.05

This is just the magnitude that LENSTOOL uses for a scale. Here this means that a galaxy with magnitude 20.05 will have a velocity dispersion (see **sigma** below) of 160 km/s.

corekpc 0.09

sigma 1 160. 200.

cutkpc 0 55. 55.

slope 0 4.0 4.0

vdslope 4.0 4.0

end

cosmology

You can probably tell when Kneib wrote LENSTOOL by the cosmology he uses in his example... Maybe you'll be able to tell how old I am by the numbers I quote...

H0 70.0

This is Hubble's Constant, H_0

omega *0.27*

This is the matter-density of the universe, Ω_m .

lambda *0.73*

This is the vacuum-energy density, or the cosmological constant, or Ω_Λ .

end

image

This is where *all* of your constraints go. Really, this is most the work you have to do.

multifile *3 image.cat*

You know the drill.

mult_wcs *1*

Again, it seems redundant. . .

critic *1 20.3982 18.8421 120.0 0.01 1.0*

These are tricky, but only for coordinate reasons. You specify a point that the critical curve must go through, as well as the angle of its tangent at that point. The coordinates are relative to the center you gave in **runmode**, so use **ds9**'s ruler with one end on your center and the other on your break (make sure the mode's set to WCS) to get your x and y values. You can then use the ruler tool to draw another line *through* the break to get its angle. Remember that **LENSTOOL** takes the angle to be measured counter-clockwise from the x-axis, while **ds9** always gives you one less than 90° . If you look at the picture with "Align WCS" chosen, you'll immediately see what to do.

The last two floats give the uncertainty in the position (a \pm value in arc-seconds) and the redshift of the curve, respectively.

sigposArcsec *0.2*

this is your uncertainty in defining the point through which the curve must go. It seems redundant to me.

z_m_limit 0 A1b 1 0.50 6.00 0.1

Here you minimize for the redshift of a multiply imaged pair. Note that only the last of the pair appears here; it won't work if you put both in. On the other hand, *both* need to have a redshift of 0.0 in your **mult_file**.

forme -1 This controls where the minimization is done: in the source plane, or in the image plane. Image plane (-1) is better, but source plane (1) is much, much faster.

end

fini

“Fini...” Take this opportunity to go get yourself a beer... Take the day off... It's been a long time getting here.

You're essentially done, at this point, really... it's just a matter now of turning your model into numbers.

V. MULTIPLE IMAGE FILES

Here you enter the constraints which LENSTOOL uses to optimize its model. This is really where all your information about lensing (save, perhaps, your **critic** definitions...) lies and this should be done with some care. It's a difficult problem saying whether two dots on the screen are, in fact, multiple images of the same source and it's good practice to resist temptations and to be skeptical at the onset of things. Once you feel like you have a pretty good model, you can use the model to *predict* where the counterimages of certain points should be and use that to help you decide whether your guesses are, in fact, multiply imaged.

I spent most of my summer thinking about how to automate this process and ended up with the program **ImageFinder**. The process is still far from automatic; it requires a decent seed model, which must still be gotten by “hand.” As far as I can tell, the only way to get a model by hand is to stare at it for many hours. The program is also far from automatic because it simply associates a scalar likelihood to each image based on similarities in color, surface brightness, and geometric “sense.” (see the program's help file for more on what I

mean). Lensing *is* achromatic; it should conserve these quantities, but that isn't necessarily reflected in the data. If the source is point-like, the surface brightness will be influenced by the magnification in its region. This would, in fact, be a good patch to add to the program. Additionally, colors seem to get skewed slightly, though I have no idea why. Remember that everything comes down to counting photons and that some of these images are built up from very few photons, so uncertainties can be fairly high.

Feel free to try my program, but know that it requires you to

Seed it with a model

This must be done in whichever way you see fit, which should be completely independent of the operation of the program.

Choose among the choices it returns

Do this by eye; I found it easier to keep only the images near critical curves, because these are easiest to distinguish. Pay special attention to morphology and parity inversions.

The program can be a useful tool, but treat it with skepticism. E-mail if you find any bugs or quirks or if you think of a better way to evaluate matches.

I should maybe mention the form of the file, as well. There is one object per line, and the lines go: *name ra dec a b theta z*. LENSTOOL considers the objects to be images of the same source, if their names are identical, up to the last character. An example of the naming convention I use is *F1a*, which would be paired with *F1b*, but independent of *F2a* or, more straightforwardly, *E1a*. Pairs of images *must* have identical redshifts (You can set both to *0.0*, if you minimize for this quantity. Remember to only include the first, though, in your **z_m_limit** definitions.).

VI. CLUSTER MEMBERS FILE

The most important things here are the magnitudes. To get these right, you need to set your parameters in SExtractor correctly. I'll leave an explanation of this to the pamphlet by Marusa, Phil and Professor Burchat, which covers SExtractor much more effectively than I can.

The length of this file actually has a considerable effect on the run-time of LENSTOOL, particularly if you minimize in the image plane. This is because LENSTOOL inverts the lens equation to calculate your image’s position in the source plane, then maps these locations back to the image plane and evaluates the closeness of the predictions. The more complicated your mapping is, the longer this process will take. You will likely find yourself running LENSTOOL hundreds and hundreds of times, so you can save yourself some time by keeping this file as lean as you can. I picked the members out by eye and only included the ones which happened to lie near my critical curves and images. If you have endless patience and want to be complete here, Douglas Stanford wrote a program which very effectively picks out cluster members from a SExtractor catalog. You can use this to produce your file.

VII. CONVERTING KNEIB’S PROFILES TO REAL MASSES

I found this to be a terrible hassle, and I haven’t quite figured all of it out yet...

It may also be the case that LENSTOOL does this automatically. Please tell me if you figure out how.

The Navarro, Frenk and White Profile (Type 12)

This profile has consistently been obtained through N-Body simulations and models the distribution purely of dark matter. The standard NFW model is characterized by the distribution

$$\rho(r) = \frac{\rho_c}{(r/r_s)(1 + r/r_s)^2}. \quad (47)$$

This is unfortunate notation, because ρ_c here is $\rho_{crit}\delta_c$, where ρ_{crit} is the ρ_c I quote in Section VIII. δ_c , in case anyone asks, is defined as

$$\delta_c = \frac{200}{3} \frac{c^3}{\ln(1 + c) - c/(1 + c)}, \quad (48)$$

in terms of the “concentration parameter” c . (This notation isn’t so unfortunate if you set the *real* c forever equal to 1 and deny that it even exists...). Clusters have, I think, concentration parameters of around 5-10...

LENSTOOL spits out a parameter **r_s**, which is straightforward enough to interpret, but it also gives you a **sigma**, which is an elusive quantity. This, effectively, subsumes the

cosmological and concentration dependence and is defined through the equation:⁸

$$\frac{\rho_c}{\rho_{crit}} = \frac{v_c^2}{H_0^2 r_s^2}. \quad (49)$$

I'm about 90% confident that v_c here is the **sigma** given to you by LENSTOOL. Don't hate me, though, if I'm wrong. I like to work in units of solar masses and kiloparsecs and gigayears (which, for the most part, agrees with LENSTOOL's inputs and outputs). Whatever system *you* use, though, know that **sigma** is given to you in $km/s\dots$. This seems obvious, but it can be hard to locate that extra factor of 10^{18} .

You can convert this distribution to an elliptical one by replacing $x_1 = r \cos \theta$ and $x_2 = r \sin \theta$ with x_ϵ , using the following:⁹

$$\begin{cases} x_{1_\epsilon} &= \sqrt{a_{1_\epsilon}} x_1 \\ x_{2_\epsilon} &= \sqrt{a_{2_\epsilon}} x_2 \\ x_\epsilon &= \sqrt{x_{1_\epsilon}^2 + x_{2_\epsilon}^2} \\ \phi_\epsilon &= \arctan(x_{2_\epsilon}/x_{1_\epsilon}) \end{cases},$$

where

$$\begin{aligned} a_{1_\epsilon} &= 1 - \epsilon \\ a_{2_\epsilon} &= 1 + \epsilon, \end{aligned} \quad (50)$$

$$(51)$$

and ϵ is the **ellipticity** given to you by LENSTOOL.

The fact that NFW profiles diverge like r^{-1} makes them very difficult to integrate numerically. You can save yourself from this headache by using polar coordinates and centering your profile at the origin (the r^2 factor from the Jacobian determinant saves you), but this only works, to my knowledge, if you're using a *single* profile. My best solution was just to define the profile piece-wise (cut it off at $\rho =$ a billion or so) and then integrate it numerically in C++. Please write an addition to this document if you find a better way to do this.

Pseudo-Elliptical Mass Density Profile (Type 81)

Ready? I'm not. This whole section actually might be quite wrong, so do check me before trusting or using anything I say...

From Section II, we know that $\nabla^2\psi = 2\kappa$. This profile gives you $\nabla^2\psi$ in these terms:

$$\begin{aligned}\frac{\partial^2\psi}{\partial x^2} &= \Re \left\{ \frac{\partial I}{\partial x} \right\} \\ \frac{\partial^2\psi}{\partial y^2} &= \Im \left\{ \frac{\partial I}{\partial y} \right\},\end{aligned}\tag{52}$$

where

$$I = \frac{(1 - e^2) E_0}{2i\sqrt{e}} \ln \left[\frac{\frac{1-e}{1+e}x - i\frac{1+e}{1-e}y + 2i\sqrt{e}\sqrt{r_0^2 + \frac{x^2}{(1+e)^2} + \frac{y^2}{(1-e)^2}}}{(x - iy + 2ir_0\sqrt{e})} \right],\tag{53}$$

where

$$e = \frac{a - b}{a + b},\tag{54}$$

but the ellipticity given to you by LENSTOOL is

$$\epsilon = 3\epsilon_\Sigma = \frac{a^2 - b^2}{a^2 + b^2}.\tag{55}$$

Also, you need to exercise some cosmology (see Section VIII) to work out E_0 ¹⁰:

$$E_0 = 4\pi \frac{D_{ls}}{D_s} \frac{\sigma_\infty^2}{c^2} = 6\pi \frac{D_{ls}}{D_s} \frac{\sigma_0^2}{c^2},\tag{56}$$

where σ_0 is the **v_disp** given by LENSTOOL and D_{ls} and D_s are angular-diameter distances from the lens to the source and from the observer to the source, respectively.

By now, you've sweat bullets just to get κ , and now you need to convert this back to a density you can work with. Fortunately, all you need is one more angular diameter distance.

Recall that

$$\Sigma(\vec{\xi}) \equiv \kappa(\vec{\theta})\Sigma_{cr},\tag{57}$$

where

$$\Sigma_{cr} \equiv \frac{c^2}{4\pi G} \frac{D_s}{D_l D_{ls}}.\tag{58}$$

And you *still* have to integrate this quantity over the entire lens plane. Are we having fun, yet?

VIII. CONVERTING BETWEEN ANGLES AND DISTANCES

For a more complete discussion of this, read the paper “Distance Measures In Cosmology”, by David Hogg, 2002.

Measuring things in Cosmology becomes a tricky business, because space isn't Euclidean, because distances are changing with time and because you're seeing things as they *were* millions of years ago, rather than as they *are* now. "Distance" becomes a somewhat ambiguous measure, because different methods of its determination, all of which agree in Euclidean geometry, don't necessarily agree in cosmological applications. Fortunately, there *is* an unambiguous number, z , which can be measured and which can be used to characterize the distance to an object. Coupled with an assumption of a cosmological model, this number can be converted into any distance measure you like. This is really very fortunate, when you think about it. The principle methods for determining distance are by measuring brightness or angular size, while the simplest definition is just an integral over r in the Friedmann-Robertson-Walker metric.

I think it might be helpful to give a 5-minute introduction to cosmology; as always, feel free to skip it.

Up to Physics 161 & 262 levels, cosmology is the study of the Friedmann-Robertson-Walker metric:

$$ds^2 = -dt^2 + a^2(t) [dr^2 + r^2 (d\theta^2 + \sin^2 \theta d\phi^2)]. \quad (59)$$

The metric is a solution to Einstein's field equations and stems from the assumption that the universe is perfectly isotropic and homogenous (imagine going through a giant, cosmic blender). This solution isn't very restrictive; it allows for both finite and infinite universes and allows the universe to be flat, positively or negatively curved.

It is, perhaps, ironic that one of the few things that the solution absolutely requires is the reprobation of the age-old "static universe;" the FRW universe can be either expanding or contracting, but it cannot stay still.¹¹ The metric permits many different types of pasts and futures (you've probably heard of "Big Bang," "Big Bounce," "Loitering" and "Big Chill" universes). Distinguishing these radically different fates are only a few parameters: the so called matter-density of the universe,

$$\Omega_m = \frac{8\pi G\rho_0}{3H_0^2} = \frac{\rho_0}{\rho_c}, \quad (60)$$

(where ρ_c is the critical (total) density for which the universe is flat), the vacuum energy density (with *cosmological constant* Λ),

$$\Omega_\Lambda = \frac{\Lambda c^2}{3H_0^2}, \quad (61)$$

and a “density” characterizing the total curvature of the universe:

$$\Omega_k = 1 - \Omega_m - \Omega_\Lambda. \quad (62)$$

ρ_c is chosen so that flat universes are characterized by $\Omega_k = 0$. The quantity H_0 , from (60) and (61) is just the quantity $H \equiv \dot{a}/a$, evaluated at $t = t_0$, or the present day.

Because the metric is in flux (as demanded by $a(t)$), the wavelengths of photons aren’t conserved. We observe a cosmological redshift, dependent in some way on *distance* (however it is measured) and can use this, along with a cosmological model (the vector $(\Omega_m, \Omega_\Lambda, \Omega_k)$) to work out how long this light has been in transit and, accordingly, how far away the source is. The number z is defined as

$$z = \frac{\nu_e}{\nu_o} - 1 = \frac{\lambda_o}{\lambda_e} - 1 = \frac{1}{a(t_e)} - 1, \quad (63)$$

where $a(t_0)$ is taken to be unity. Like everything else in physics, this relation is linear for small times (or distances) and therefore reproduces Hubble’s famous result.¹² This is, perhaps, obvious, but the fact that we always observe a *redshift* means that the universe is expanding. Were it the other way around, we’d see blueshifts ($z < 0$). It’s tempting to associate these frequency shifts with doppler shifts via

$$1 + z = \sqrt{\frac{1 + v/c}{1 - v/c}}, \quad (64)$$

but such treatment isn’t quite accurate. In general, you can’t compare velocities between distant points, because there’s no meaningful way to bring a vector from one place to another; the result will depend on the path you take.¹³ There can be a Doppler shift on top of the cosmological one, and accurate measurements of them gives another means of estimating mass.¹⁴ The path taken, in this case, is the geodesic one taken followed by the light being measured.

For what follows, I’ll define

$$D_H \equiv \frac{c}{H_0} \approx 3000h^{-1}\text{Mpc} \quad (65)$$

$$E(z) \equiv \sqrt{\Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda} \quad (66)$$

$$D_C(z) \equiv D_H \int_0^z \frac{dz'}{E(z')}, \quad (67)$$

using Hogg’s notation. With these, we are ready to define the **Comoving Distance**. This relates the distances among things at the same redshift. The physical distance between two objects at a redshift z_1 , separated by some small angle $\delta\theta$ is simply $D_M(z_1)\delta\theta$.

$$D_M = \begin{cases} D_H (\Omega_k)^{-1/2} \sinh \left[(\Omega_k)^{1/2} D_C/D_H \right] & \text{if } \Omega_k > 0 \\ D_C & \text{if } \Omega_k = 0 \\ D_H (-\Omega_k)^{-1/2} \sin \left[(-\Omega_k)^{1/2} D_C/D_H \right] & \text{if } \Omega_k < 0 \end{cases} \quad (68)$$

The distances D_l , D_s and D_{ls} that pop up in the lens equation are all angular-diameter distances:

$$D_A = \frac{D_M}{1+z} \quad (69)$$

This gives, for instance, the number you will use to convert between **core_radius** and **core_radius_kpc**. Simple enough. Be careful, though, because these distances don’t add. Specifically, the distance between two points of redshift z_1 and z_2 is given by (assuming $\Omega_k \geq 0$):

$$D_{A12} = \frac{1}{1+z_2} \left[D_{M2} \sqrt{1 + \Omega_k \frac{D_{M1}^2}{D_H^2}} - D_{M1} \sqrt{1 + \Omega_k \frac{D_{M2}^2}{D_H^2}} \right] \quad (70)$$

$\neq D_A(z_2) - D_A(z_1) !!!$

You really only need to calculate these distances if you’re doing a lensing calculation by “hand;”¹⁵ LENSTOOL takes care of these measures for you. It’s good, however, to know that this is where your cosmology dependence comes in, and that this is why lensing can be used to constrain the values of the parameters Ω_m and Ω_Λ . It’s also good to know that a redshift dependence comes into your calculations through the angular-diameter distances; this is the source of the mass-sheet degeneracy (see Marusa’s paper), which can be a source of consternation if you don’t have adequate redshift information about your multiply imaged sources. Finally, it’s interesting to see the discrepancy in (70). Cosmology is a very weird thing!

IX. C++

If you’re like me, and you have a windows laptop, and you like to program at home, you may run into problems trying to port your code to Linux...

General Problems I Had With Porting Code

The most annoying problem I had (the one which took the longest to figure out) is that Linux and Windows machines read end-of-lines differently. If you really want to make good, portable code, you need to think hard about this (I think you can do it by reading everything as binary, rather than ASCII, but I haven't really thought it through). My solution was just to write two input files (one in notepad and one in emacs) and to use whichever was appropriate. . . it's likely that they will only be used on a Linux machine, anyways.

Another problem you may run into, but which the `g++` makes makes much more clear, is that you cannot use `stdlib.h` on a Linux machine. Use `unistd.h` instead. This library has no function `itoa()`, so if your program uses this function, you will need to implement your own version.

Compiling Your Code On A Linux Machine

When you program on Windows, your compiler probably manages and compiles projects for you. It also probably has auto-completes and error-checking and nice formatting and all sorts of other wonderful things. For Linux, you will probably be using the compiler `g++`, which you run from the console. If your program is a single file, you can just run the following command

```
g++ -o Program.x Source.cpp
```

`g++` is your compiler, `-o + Program.x` means that the output will be saved under the name `Program.x`, and `Source.cpp` is where all of your code is.

If you subscribe to the object-oriented thing and your program consists of a billion files, you will certainly need to use `make` to compile it. `make` will probably save you some time, though, even if your source code has only 2 or 3 files. `make` is a program which will compile and link all of your program files. It compiles only the files which have nonexistent or out of date object files, so it really is much more efficient than telling `g++` to recompile everything every time. It runs a script saved in the file "Makefile" (no extension), which you call with the command `make`. Output from the compiler will follow, terminated by the message (if you specify it) "make complete."

I found `make` to be really confusing, at first, until I discovered that it really isn't a procedural programming language... it's a sort of artificial-intelligence-like thing. If you're curious about how it works, Professor Paul Kunz has an excellent tutorial on it. If you're not that curious, I tried to comment the makefile for my `ImageFinder` program well enough to make it easily modified. Have a look at my file, if you like, but I strongly recommend you look through a good tutorial on it (try either typing `make help` or looking through Dr. Kunz's tutorial). It takes about half a day to learn this way, but I found it very hard to learn `make` by example.

X. CULTURE

Here I list random facts which aren't necessarily useful, but which might be good things to throw around at parties. They might also be good to know if you ever come across an astrophysics crossword puzzle.

Magnitudes

The magnitude system has its roots in the ancient Greek astronomer Hipparchus who, in the second century BC, ranked the stars in the sky by their brightnesses. The brightest stars were of the *first magnitude*, while the dimmest were of the sixth. It has since been discovered that the response of the human eye to faint light is approximately logarithmic and that the dimmest visible stars yield approximately one percent the fluxes of the brightest. Astronomers used this information to define a bolometric *apparent magnitude*:

$$m \equiv -2.5 \log_{10}(f/f_x). \quad (71)$$

The sign in the equation makes a small value of m correspond to a big flux. (The Sun on Earth, for a point of comparison, has $m \sim -27$, while the dimmest things visible to the Hubble telescope have $m \sim 30$.)¹⁶ The scale characteristic f_x was set to $2.53 \times 10^{-8} \text{ watt/m}^2$ to give visible stars magnitudes $0 < m < 6$. The brightest quasars have $m \sim 13-15$ and, contrary to my initial hopes, aren't visible to the naked eye.

There is also a bolometric *absolute magnitude* defined as

$$M = m - 5 \log_{10} \left(\frac{d_L}{1 \text{ Mpc}} \right) - 25. \quad (72)$$

I honestly have no idea which magnitude is the one that LENSTOOL takes... I just happily used whatever came out of SExtractor and asked no questions.¹⁷ I expect it shouldn't matter, so long as you're consistent with your definition of **mag_0**.

Dark Matter

It has been known since the 1930's that there must be much "missing material" in the universe; simultaneously reflecting an important understanding of this material's properties as well as our marked ignorance of its constitution, we no longer call the material "missing" but "dark." "Dark Matter," is a variety of matter which appears to be made of wholly unfamiliar particles and which, intriguingly, appears to be much more prevalent in the universe than more familiar material is. Dark matter has factored into such diverse fields in cosmology as structure formation, big-bang nucleosynthesis, gravity lensing, the anisotropy of the cosmic-microwave background and the dynamics of galaxies and clusters and therefore lies at the nexus of these converging sources of data. It seems now that a good understanding of the universe necessitates a good understanding of dark matter. Though we know little about its constitution, we know many of its properties and dark matter plays a central role in our understanding of cosmology.

In an impressive display of human understanding, astrophysicists and cosmologists working in the late twentieth century managed to compile a complete inventory of the matter in the universe and concluded that some 25% of it must be this mysterious material.¹⁸ Detailed measurements of the cosmic microwave background has enabled astrophysicists to work out several important properties of dark matter. This understanding of dark matter indicates a impressive knowledge about the universe and has enabled substantive advances in many fields of astrophysics and cosmology, but, until recently, all our evidence for its *existence* has been circumstantial. Much of science is founded on circumstantial evidence and probability,¹⁹ which is fine, but concrete evidence helps to win over dedicated skeptics. For this reason, a proverbial "smoking gun" would boost confidence in cosmological theories. Precisely this smoking gun, it turns out, has been provided by Marusa.²⁰

Because dark matter only interacts with ordinary matter through gravity, its detection necessarily comes down to the detection of stronger-than-expected gravitational fields. It

turns out, however, that dark matter almost always tends to clump around ordinary matter, making it difficult to say for certain that our methods for detecting the field strength and our method for calculating the expected field strength don't factor into the discrepancy. This is interesting to think about, because a modified (or new) theory of gravity *might* obviate the need for dark matter and provide some new insight into the universe.

By studying a cluster which had recently undergone a catastrophic merger, Marusa was able to see dark matter and its ordinary counterpart separated. Because we know that dark matter doesn't interact with electromagnetic forces, we should expect it to be immune from the friction and drag we're compelled to deal with. If a cluster consists mostly of gas and dark matter, then, we can expect a collision to treat the two constituents separately; the dark matter should pass right through itself, while the gas will make the most thunderous crash you've ever imagined.²¹ The two components will move in the same potential well, but will have different energies and should be found to have different orbits at different radii.

I mentioned the possibility that an alternate theory of gravity could obviate the need for dark matter. It seems reasonable to demand that such a theory direct its field in the direction of the gravitating mass.²² By discovering a place where dark matter and ordinary matter were separated, Marusa essentially found that any such theories which explain her data *must* violate our demand.²³ Note that this type of analysis *requires* lensing analysis; other methods go out the door whenever you have a violent merger, and (even more importantly) lensing is all you have when you can't *see* your gravitating mass.

It seems, then, that our current story of the universe is coming together fairly well. The difficult task still remains, though, of finding the dark matter *itself*, more than mere evidence for (or even proof of) its existence. This is a task being attacked with some of the most sensitive equipment ever made (both accelerators *and* telescopes, probing the fantastically small alongside the fantastically large) and is bringing into cooperation the fields of particle physics and cosmology. Of particular importance to *you*, the study of dark matter has motivated the study of large galaxy clusters, which promise to lend insight into its nature. SNAP and LSST will be conducting massive surveys for these structures in the near future, and many (if not all) that are found will require strong lensing analysis.

Nonetheless, it should still make you suspicious that our current cosmological model puts

five times as much dark matter in the universe as ordinary stuff. It took much work from an extremely patient T.A. (God bless you, Mustafa) to convince me that the whole theory of dark matter isn't a cop-out or a fudge factor. We live in an odd place in the universe, where the baryon density happens to be some 10^{22} times the average. There really is no reason we should observe dark matter with $\Omega_{DM} \sim 0.3$, especially if it doesn't interact with the electromagnetic force, our *principle means of detection*. That being said, it *is* important to make sure we are on the right track and that we haven't been somehow fooled into seeing dark matter. Alternate theories have been posited, and are certainly worthy of consideration:

Our entire theory of dark matter rests on the detection of stronger-than-expected gravitational fields near galaxies and clusters of galaxies. This requires both a measurement of the actual field as well as a calculation of what we should expect to find. Implicit in this is an assumption of the accuracy of general relativity on such large scales, as well as some assumptions about fluid dynamics and about celestial mechanics. Even after only two years in physics, I'm familiar with the empirical fact that exceptions tend to arise on *small*, rather than large scales,²⁴ but it's worth considering modifications to these theories. I know of modified mechanical theories (MOND theories (for *MOdified Newtonian Dynamics*)), but not of any hydrodynamical ones. To my understanding, MOND theories tend to lie on the outskirts of acceptance and are seen as more contrived than the theory of dark matter. I know (though I don't know *why*) that MOND theories calibrated to explain galaxies have thus far had difficulty explaining clusters of galaxies and generally still need to allow for some form of dark matter. Λ CDM cosmology, on the other hand, appears to be accurate on all tested scales.

The notion of a different theory of gravity is an interesting one and one that piques my curiosity, but it seems more fruitful to stick with the dark matter paradigm than to enter Einstein's playing field. Any theory of gravity must satisfy *all* observations, and it turns out that the theory of General Relativity plus Λ CDM cosmology has withstood an admirable barrage of tests and has yet, to my knowledge, to be found inconsistent with any data. Attempting to alter it then seems a matter of personal bias and not empirically motivated science.

The story of dark matter is still very much alive, continuing through Marusa's discovery to the present day, but it begins with the Swiss astrophysicist Fritz Zwicky in the Eighteenth

Century. Zwicky studied the Coma Cluster from the Mt. Wilson Observatory at Caltech and, in 1933, found that the mass he calculated from measured velocity dispersions was some 400 times that which he calculated from total brightness. With admirable insight, he concluded that there must be some form of “missing matter” and published his results in 1937.

Zwicky’s evidence was much more humble than Marusa’s, but it is insightful and I think it’s helpful to have a look at it. If you assume some bunch of point masses (call it a galaxy), you can take Newton’s Law of Gravity and calculate the potential energy of the whole system:

$$\begin{aligned}
 \ddot{x}_i &= G \sum_{j \neq i} m_j \frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|^3} \\
 W &\equiv \int \vec{F} \cdot d\vec{l} = -\frac{G}{2} \sum_j \sum_{j \neq i} \frac{m_i m_j}{|\vec{x}_j - \vec{x}_i|} \\
 &\equiv -\alpha \frac{GM^2}{r_h},
 \end{aligned} \tag{73}$$

where M is the total mass of the galaxy, r_h is the radius within which half of the total mass is enclosed and α is chosen to make the equation true. (I seem to recall from Physics 161 that $\alpha \approx 0.4$ for typical disk galaxies; I think it should be 1 for a perfectly spherical one). Continuing with classical mechanics, we have, for total kinetic energy and moment of inertia:

$$\begin{aligned}
 T &\equiv \frac{1}{2} \sum_i m_i \left| \dot{\vec{x}}_i \right|^2 = \frac{1}{2} M \langle v^2 \rangle \\
 I &\equiv \sum_i m_i |\vec{x}_i|^2.
 \end{aligned} \tag{74}$$

Taking the second derivative of I ,

$$\begin{aligned}
\ddot{I} &= 2 \sum_i m_i (\ddot{\vec{x}}_i \cdot \dot{\vec{x}}_i + \dot{\vec{x}}_i \cdot \ddot{\vec{x}}_i) \\
&= 2 \sum_i m_i \dot{\vec{x}}_i \cdot \ddot{\vec{x}}_i + 4T \\
&= \left[\sum_i m_i \dot{\vec{x}}_i \cdot \ddot{\vec{x}}_i + \sum_j m_j \dot{\vec{x}}_j \cdot \ddot{\vec{x}}_j \right] + 4T \\
&= G \left[\sum_{i,j} m_i m_j \frac{\dot{\vec{x}}_i \cdot (\vec{x}_j - \vec{x}_i)}{|\vec{x}_j - \vec{x}_i|^3} + \sum_{j,i} m_i m_j \frac{\dot{\vec{x}}_j \cdot (\vec{x}_i - \vec{x}_j)}{|\vec{x}_i - \vec{x}_j|^3} \right] + 4T \\
&= G \left[\sum_{i,j} \frac{m_i m_j (\dot{\vec{x}}_i + \dot{\vec{x}}_j) (\vec{x}_i - \vec{x}_j)}{|\vec{x}_j - \vec{x}_i|^3} \right] + 4T \\
&= G \left[\sum_{i,j} \frac{m_i m_j}{|\vec{x}_j - \vec{x}_i|} \right] + 4T \\
\ddot{I} &= 2W + 4T.
\end{aligned} \tag{75}$$

Galaxies are old things and therefore should have very slowly changing moments of inertia (else they'd have blown up or shrunk to nonexistence). Therefore, we know

$$\begin{aligned}
T &= -\frac{W}{2} \\
\frac{1}{2} M \langle v^2 \rangle &= \frac{\alpha}{2} \frac{GM^2}{r_h^2} \\
M(r) &= \frac{r_h}{\alpha G} \langle v^2 \rangle (r).
\end{aligned} \tag{76}$$

By measuring the average velocity dispersion within some radius, therefore, you can work out the mass enclosed by that sphere. This is known as the virial theorem and is the formula Zwicky used to obtain his famously discrepant results.

FITS Files

A good knowledge of FITS files isn't really (so far as I can tell) crucial for strong lensing analysis and, consequently, I don't know very much about them. "FITS" stands for "Flexible Image Transport System," and the files can store many different types of information: a single file can have image data in multiple bands, as well as tables and lists giving information about how the data was taken. The "images," therefore, can be multi-dimensional and can

very difficult to interpret, especially if you don't know exactly how they were written. I was lucky enough to face only the 2-dimensional variety; I had a different file for each band of data.

My only experience with FITS files is opening them in `ds9`, but I know that weak lensing analysis requires the ability to manipulate them. `SExtractor` requires your FITS fits files in different bands to be of exactly the same size, as well. I'm actually completely unaware of any method to edit or manipulate these files, but feel like I should warn you that this problem may come up.

Clusters of Galaxies

This is a lot of information to have gotten through and it's easy to lose sight of why we do it. Clusters of Galaxies are currently our most sensitive, and potentially our most important probes on cosmology. Their study promises to shed much light, both on what we think we know, as well as on what we know we don't know, about the universe.

First, clusters can tell us about the formation of structure in the universe. Recent N-Body simulations indicate that clusters emerge as nodes in the gravitationally evolving web of baryons and dark matter which defines this structure.²⁵ The time evolution of these simulations indicates that, in addition to being the largest virialized structures in existence today, clusters mark the regions in which structure first began to form at high redshift. Clusters evolve on time-scales \sim to the Hubble time, so we necessarily see them in some stage of formation. By characterizing a good number of them, astrophysicists have found ample evidence for dynamical and interesting youth. Nonetheless, the bulk measures derived from clusters *are* tightly correlated, making the population an even more attractive probe.²⁶

Additionally, cluster formation is sensitive to the cosmological parameters Ω_Λ , Ω_m and σ_8 and a precise characterization of the population of clusters can place stringent constraints on these values.²⁷ Calculations for relating these constraints are in good shape and plans to conduct large surveys for the collection of data from clusters are underway, but an accurate, reliable method of inferring bulk measures (such as total mass) directly to observables needs to be worked out. This is a difficult problem and it is where a lot of interesting science lies. This is, so far as I can tell, a major motivation for the study of gravitational lensing.

Cosmology is now a data-rich subject, with empirical support from at least four inde-

pendent channels: the cosmic microwave background, light element nucleosynthesis, Type Ia supernovae and large-scale cosmic structure. The latest WMAP data has indicated that the observed temperature fluctuations in the CMB are extremely well-fit to the acoustic peaks expected in an inflationary, hot big-bang picture and spotlights a fundamentally unexpected model containing roughly 2/3 dark energy, 1/3 dark matter and only about 4% ordinary, baryonic matter. The nature of the principal dark components remains a mystery and is the subject of much research going on in both astro and particle physics. It seems a very exciting time to be working in either of these fields.

* Electronic address: mkmcc@stanford.edu

¹ Indeed, this is how you typically define the geometry.

² Note that A maps some vector $\vec{\theta}$ from the *image plane* to the *source plane*. Therefore, to get the magnification as we observe it in the image plane, we need the determinant of the inverse of A . Jacobian matrices have the property that the determinant of the inverse is simply the reciprocal of the determinant.

³ Kneib, ????

⁴ The time required to take a spectrum goes like the square of the flux; a magnification of 5 cuts this time to a 25th and a magnification of 20 cuts it to a 400th. The first is the difference between do-able and not do-able, while the last is unfathomable. It is fortunate that nature lets us cheat in this way.

⁵ LENSTOOL claims to be able to constrain against single arclets, as well, but I'm not sure how. The manual doesn't say much more than that, either. Ask Marusa about it, if you need to.

⁶ Lenstool Manual, page 21

⁷ Marusa told me that you should be getting values between $\sim 0.1 - 0.5$ magnitudes.

⁸ Golse & Kneib, 2002

⁹ Golse & Kneib, 2002

¹⁰ I'm unsure of the origin of the factor of 3/2 here, but I got this equation from the LENSTOOL manual, so I think it to be true.

¹¹ A nearly static scenario is presented by the "loitering" universe, which grows like $t^{3/2}$ for a while, then can sit for a long time before exponentially exploding. Observations have ruled out

this possibility, but the solution has historical interest: during its loitering phase, the solution has much in common with the Einstein Static Universe. . . .

¹² For *extremely* small distances, the expansion becomes, again, complicated. This is because the metric no longer resembles the perfectly homogenous and isotropic FRW one. I've been told (but have *absolutely* no clue why) that the metric expands on Earth like $(H_0)^4$.

¹³ <http://torus.math.uiuc.edu/jms/java/dragosphere/> Try moving the vector around different closed paths; the precession of the vector turns out to be proportional to the area enclosed by your loop.

¹⁴ Cohen & Kneib, 2002

¹⁵ This is, to my knowledge, *impossible* for all but the simplest scenarios, but it a neat thing to try as a simple approximation.

¹⁶ The dimmest objects visible with an 8m, ground-based telescope have $m \sim 27$, so the Hubble really is a considerable improvement.

¹⁷ The *SExtractor For Dummies* manual has *a lot* of information on magnitudes. . . this information might be in there.

¹⁸ It turns out to be even worse than that; some 70% of all matter turns out to be even *less* familiar: gravity, to it, is a *repulsive* interaction

¹⁹ indeed, certain philosophers will argue that *everything* is.

²⁰ astro-ph 0608407

²¹ Stars, I think, are an intermediate case. . . They are too sparse to hit eachother, really (the probability is extremely small), but they do experience drag in the gas. I'm not sure how strong this drag is, but it might be possible that stars, gas and dark matter are *all* separated. . .

²² "Common Sense" is a matter of the experiences we have, and every experience we have with gravity tells us that it must point towards the gravitating mass.

²³ Apparently, other ways around this (called *Scalar-Tensor Theories*) have been found, but I don't really even understand what they are.

²⁴ Can anyone give me a philosophical basis for this, or must it remain emperical?

²⁵ Springel, et. al 2005. Search google for the "Millennium Simulation" and download the videos. They're mesmerizing.

²⁶ Evrard, chapter 1 of the book *Clusters of Galaxies*.

²⁷ White & Fabian, 1995; Bahcall & Fan, 1998