# Calculating geodesics for arbitrary metrics

Guillermo Andree Oliva Mercado, BSc.

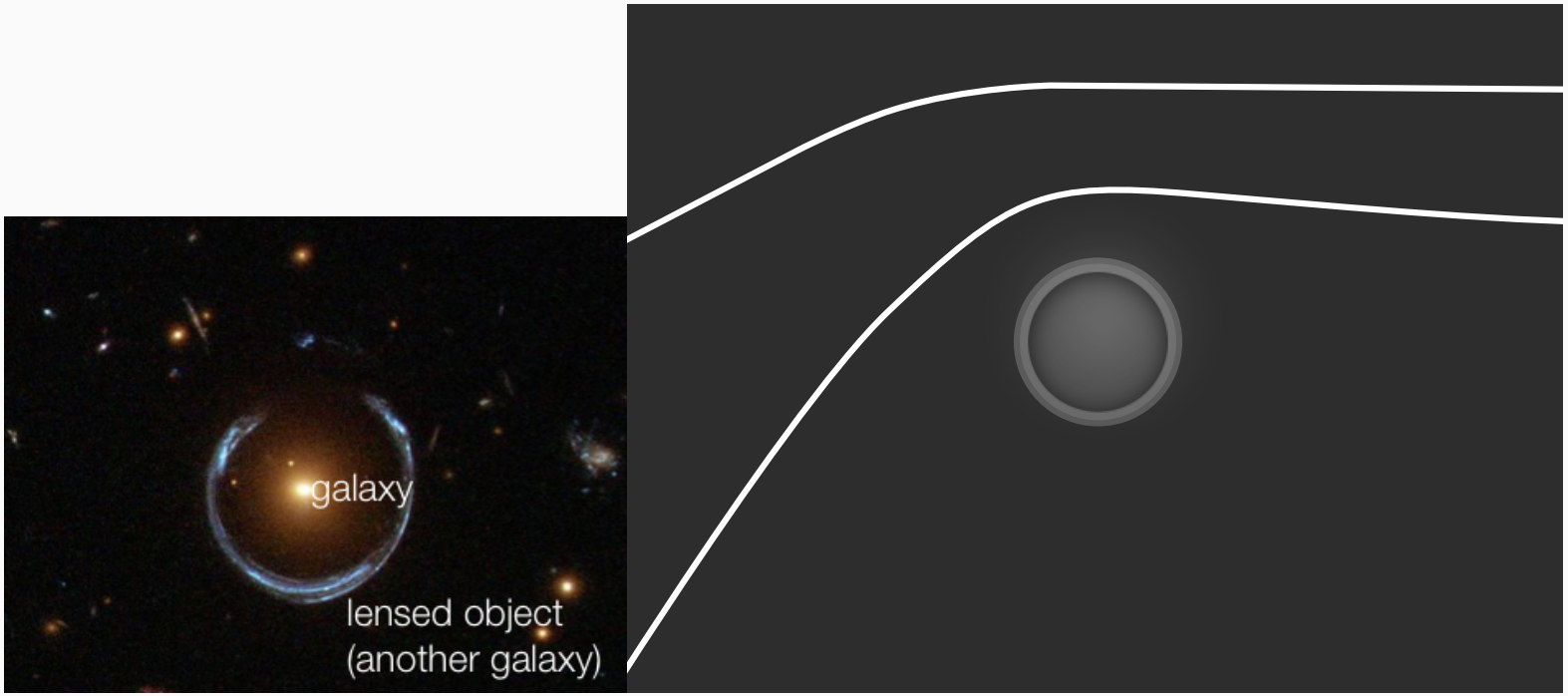
February 26, 2016

(Dir. thesis: Dr. rer. nat. Francisco Frutos)
School of Physics
University of Costa Rica

1. Description of the problem
2. Structure of the program
3. Example inputs and results

# Description of the problem

# Compact/massive objects and bending of the light

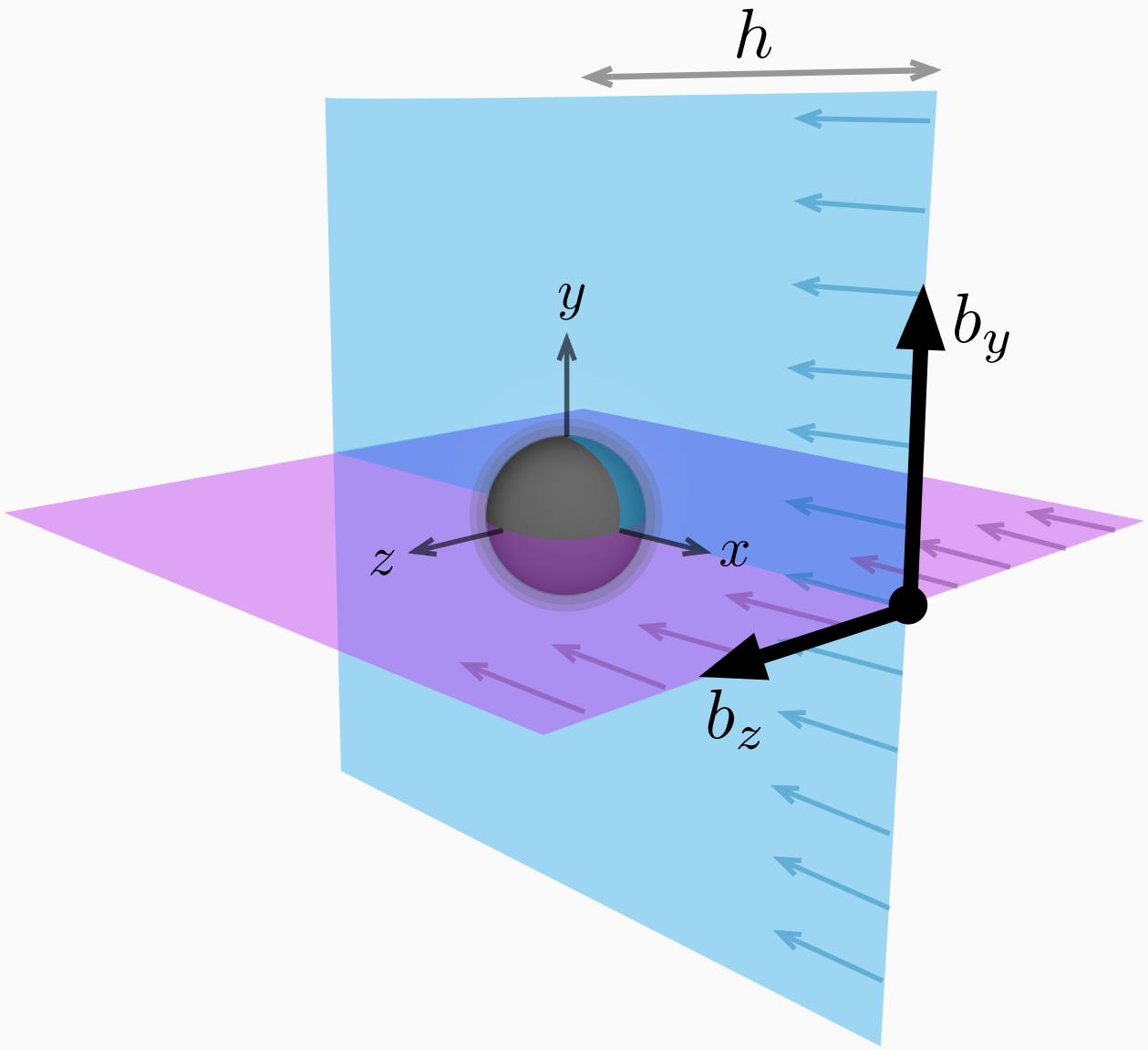



- Gravitational lenses
- Emission from accretion disks

# Mathematical setup

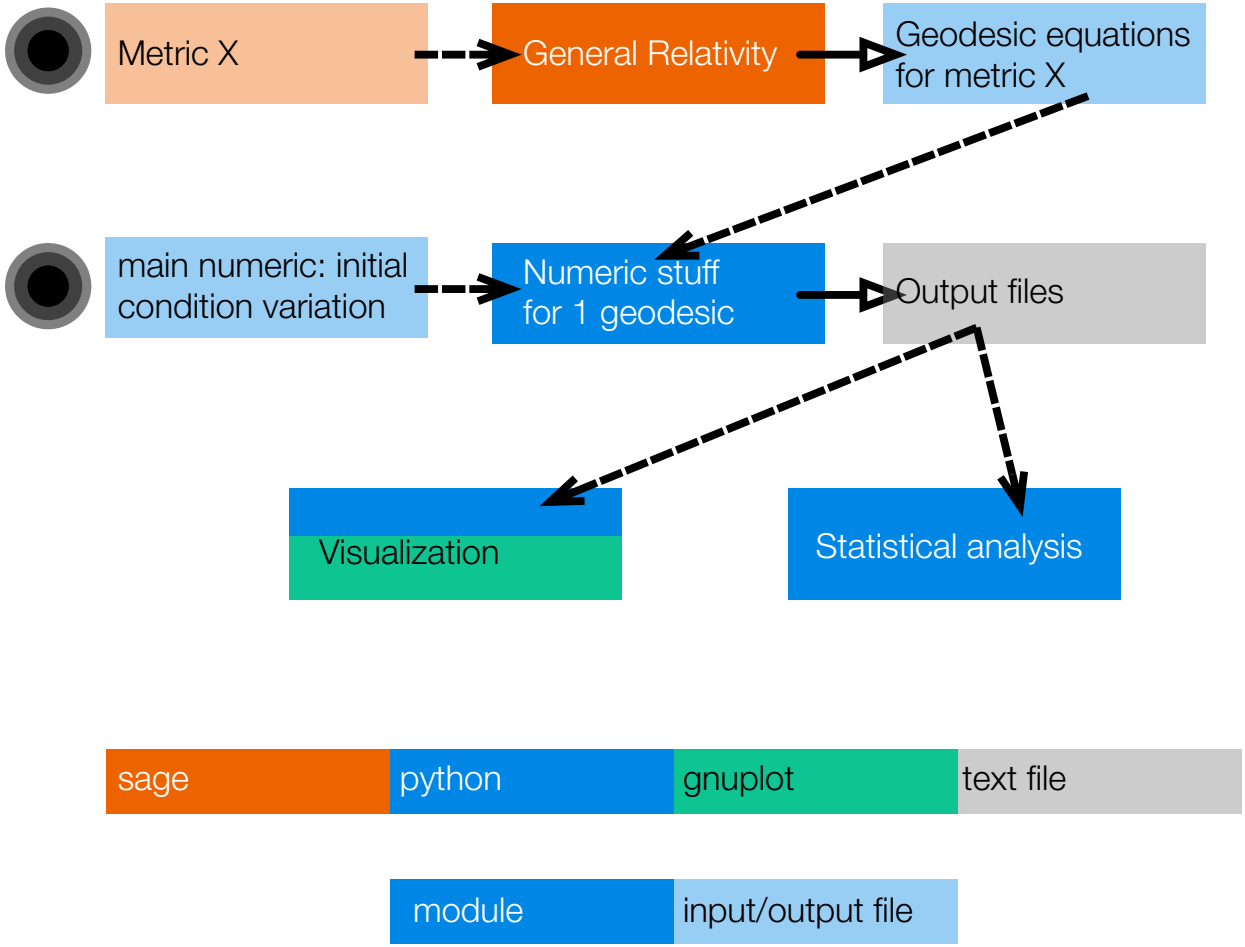
- Metrics
  - Describe the spacetime for a given configuration of matter–energy
  - Solutions of Einstein Field equations
  - Hard to find!
- Geodesics
  - Trajectories of light and massive particles
  - We are concerned with light only
  - Calculated with geodesic equations
- Geodesic equations
  - Second order differential equation
  - Contains derivatives of the metric
  - Initial conditions determine the type of geodesic

# Initial conditions



# Structure of the program

# General scheme



# Decisions I had to make

- Languages
  - Sage: symbolic calculations
  - Python: numeric / statistical / plotting / gluing
  - Gnuplot: some plotting
- Input / output
  - Library for GR calculations $\Longrightarrow$ input is a script
  - Library for geodesic calculations $\Longrightarrow$ input is a script
  - Easy extension of the program to solve similar problems
  - Output are plots
- Validation
  - Used known metrics (Minkowski, Schwarzschild)

# Examples of input and results

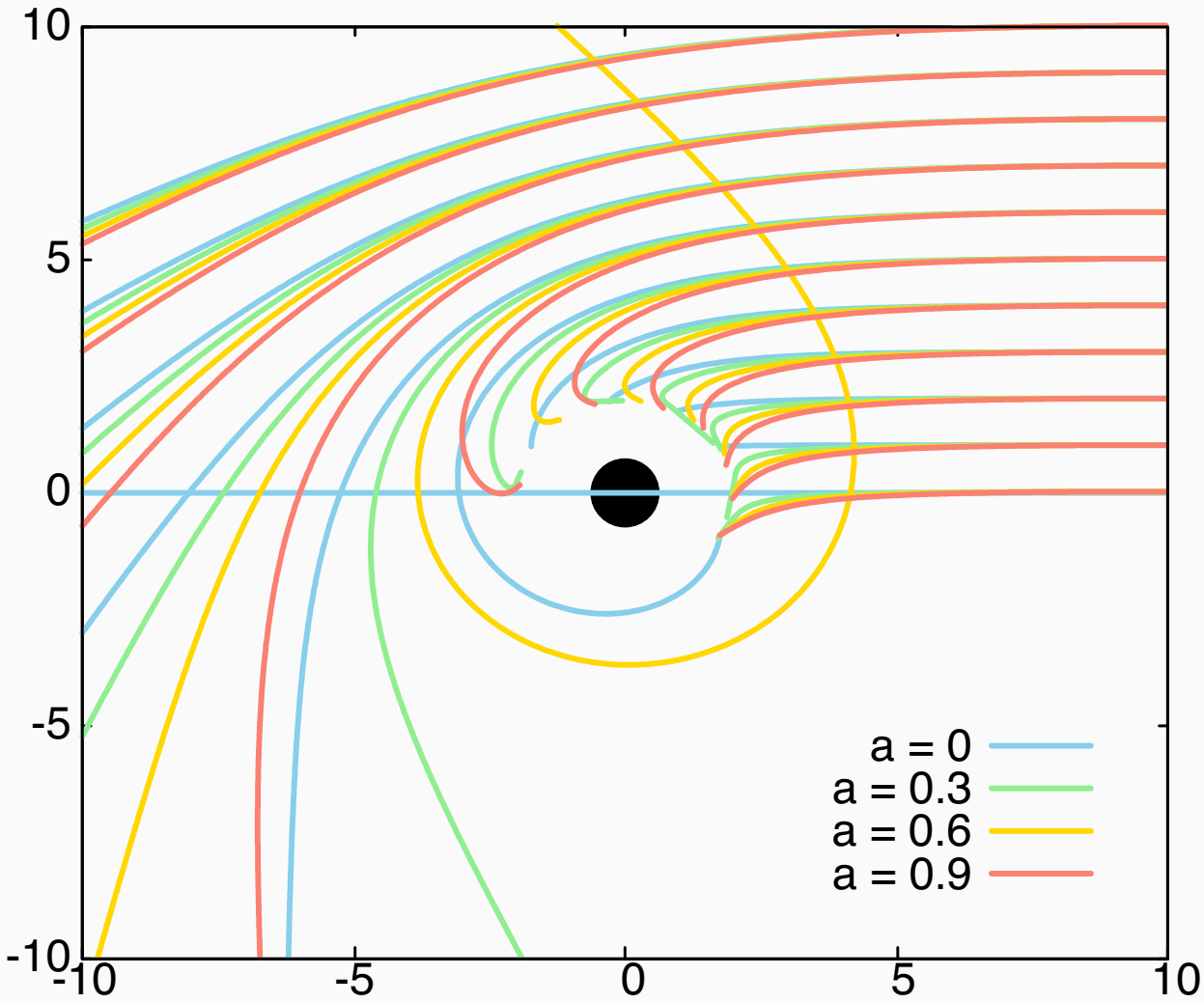
# Input for the symbolic part I

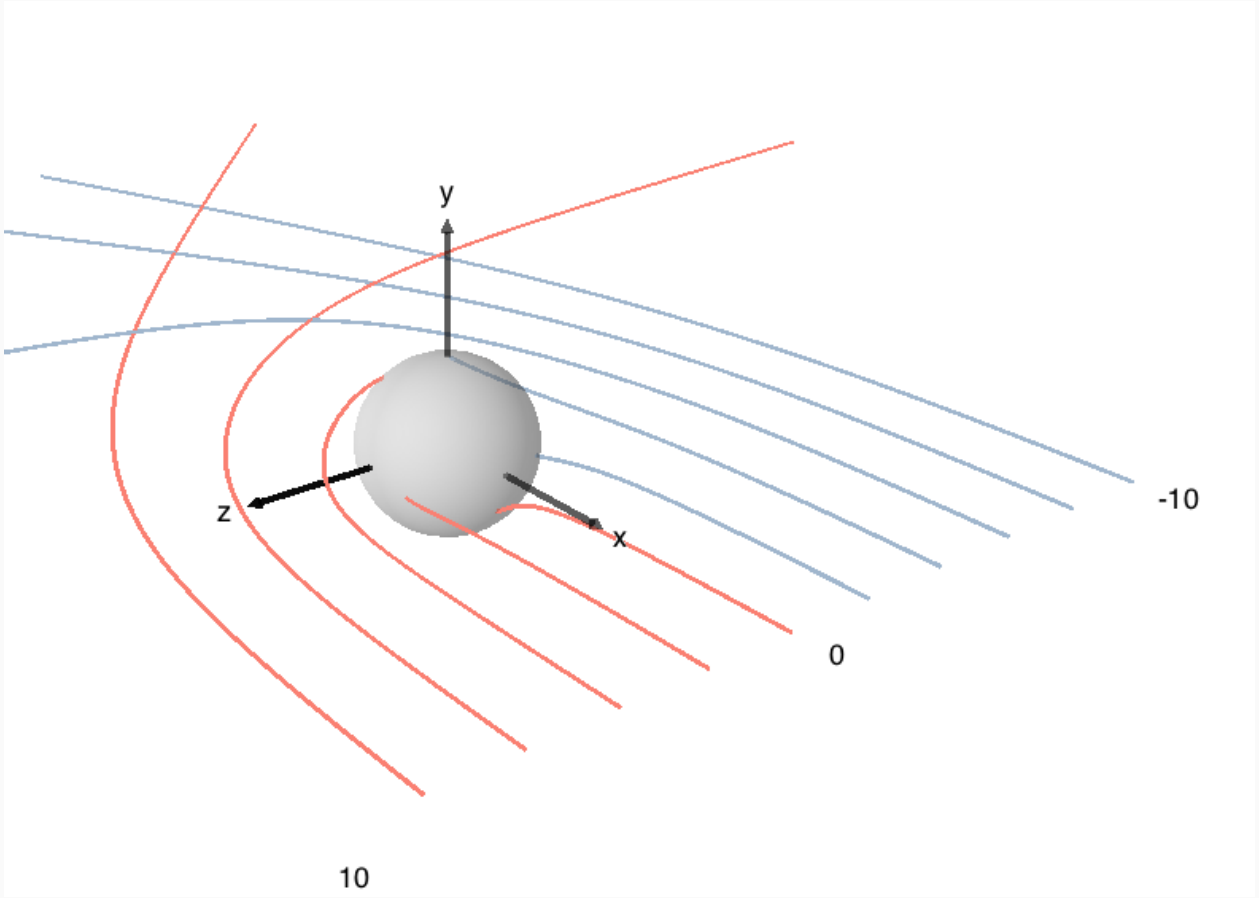
```
load "equations.sage"

var('x0 x1 x2 x3 v0 v1 v2 v3 M a q')
params=[M,a,q]

. . .   (some parameter definitions)

frutos = SpaceTime()
frutos.metric = Tensor([oneForm(),oneForm()],4)
frutos.metric[0,0] = exp(-2*psi)*(a^2*(sin(x2))^2 - delta)/rh^2
frutos.metric[1,1] = rh^2*exp(2*ji)/delta
frutos.metric[2,2] = rh^2*exp(2*ji)
frutos.metric[3,3] = exp(2*psi)*sin(x2)^2*(( x1^2 + a^2 )^2 - a^2*delta*(sin(x2))^2)/rh^2
frutos.metric[0,3] = -2*M*a*x1*(sin(x2))^2/rh^2
frutos.metric[3,0] = frutos.metric[0,3]

frutos.conn = Tensor([tangent(),oneForm(),oneForm()],4)

christoffel(frutos.conn,frutos.metric,[x0,x1,x2,x3])
eqs = generate_equations(frutos.conn,[x0,x1,x2,x3],[v0,v1,v2,v3])

Write(frutos)
```

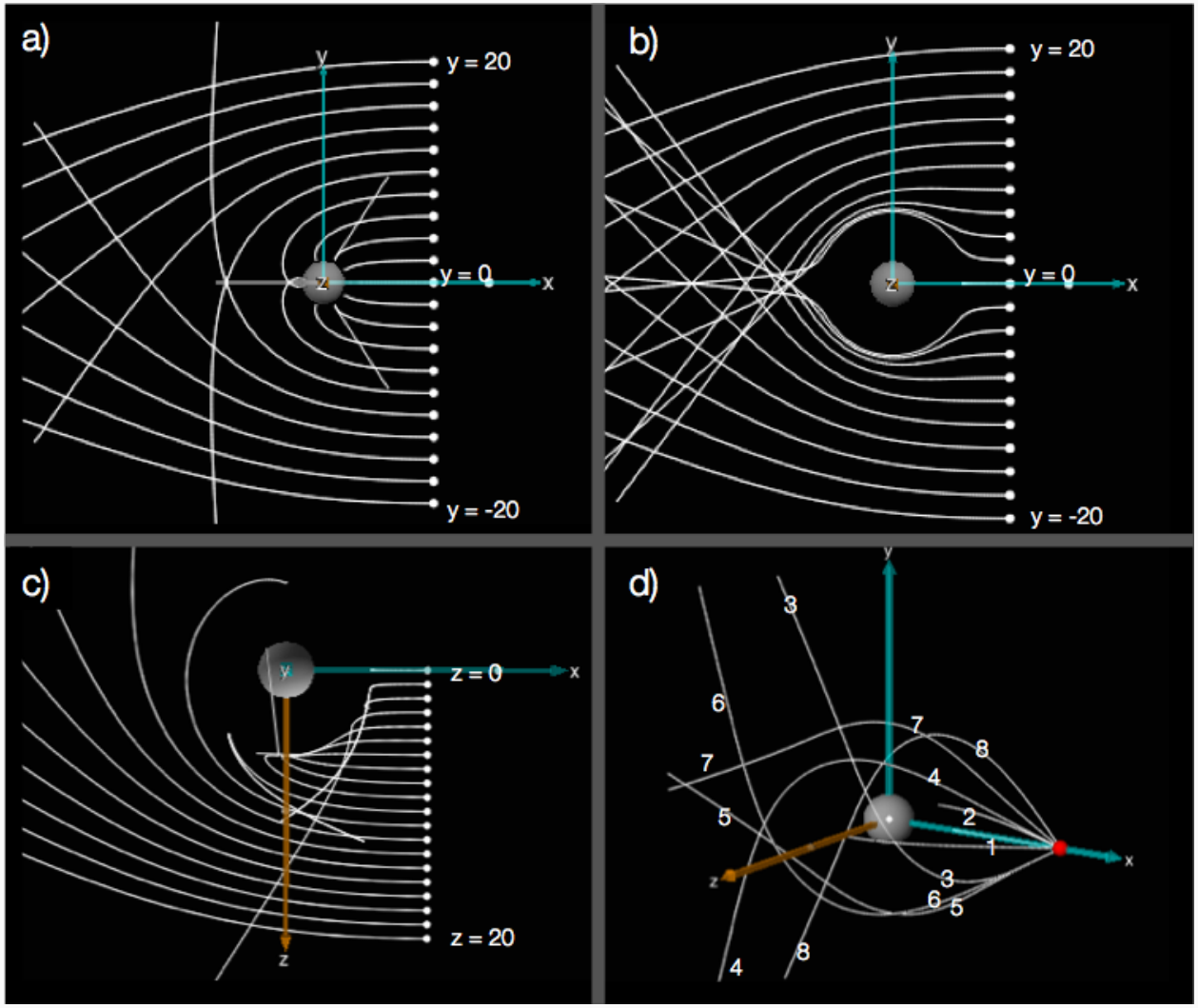
# Input for the numerical part

```
from geodesics import *


for k in arange(-12,12,0.5):
    pos_cart = (10,0,k)
    r,theta,phi = to_spherical(*pos_cart)
    vel = (-sin(theta)*cos(phi),-cos(theta)*cos(phi)/r,sin(phi)/(r*sin(theta)))
    state = [0,r,theta,phi,0,vel[0],vel[1],vel[2]]
    state[4] = calculate_v0(state)
    generate_geodesics(lam=0,state,dlam=0.05,limits=[40,2,20])
```

# Example plots




y
z
x
-10
0
10


a)
y
y = 20
y = 0
x
z
y = -20
b)
y
y = 20
y = 0
x
z
y = -20
c)
y
z = 0
x
z = 20
z
d)
y
3
6
7
8
7
4
5
2
1
x
z
3
6
5
4
8

# References


- Oliva, A., Frutos, F., Bonatti, J., González, K. (2016) . A numerical study of a Kerr-like metric. `http://gandreoliva.org/papers/simmac16-a.pdf` (not yet published).
- Oliva, A., Bonatti, J., Cordero, I. & Frutos, F. (2015) . A Visualization of Null Geodesics for the Bonnor Massive Dipole. *Revista de Matemática: Teoría y Aplicaciones* 22(2), 255-264.
- Frutos, F. (2015). New approximate Kerr-like metric with quadrupole. ArXiv:1509.03698v1 (not yet published)