REVISTA DE MATEMÁTICA: TEORÍA Y APLICACIONES 2015 **22**(2) : 255–264
CIMPA – UCR ISSN: 1409-2433 (PRINT), 2215-3373 (ONLINE)

A VISUALIZATION OF NULL GEODESICS FOR THE BONNOR MASSIVE DIPOLE

VISUALIZACIÓN DE GEODÉSICAS NULAS PARA EL DIPOLO MASIVO DE BONNOR

G. ANDREE OLIVA–MERCADO* JAVIER BONATTI–GONZÁLEZ†
IVÁN CORDERO–GARCÍA‡ FRANCISCO FRUTOS–ALFARO§

Received: 25/Feb/2014; Revised: 19/Apr/2015;
Accepted: 11/May/2015

*Escuela de Física, Universidad de Costa Rica. San José, Costa Rica. E-Mail: gandreoliva@gmail.com

†Misma dirección que/*Same address as*: G.A. Oliva-Mercado, & Centro de Investigación en Ciencias Atómicas, Nucleares y Moleculares (CICANUM). E-Mail: jbonatti2011@gmail.com

‡Misma dirección que/*Same address as*: G.A. Oliva-Mercado. E-Mail: ivancg@msn.com

§Centro de Investigaciones Espaciales (CINESPA), Universidad de Costa Rica. San José, Costa Rica. E-Mail: frutos62@gmail.com

Abstract

In this work we simulate null geodesics for the Bonnor massive dipole metric by implementing a symbolic-numerical algorithm in Sage and Python. This program is also capable of visualizing in 3D, in principle, the geodesics for any given metric. Geodesics are launched from a common point, collectively forming a cone of light beams, simulating a solid-angle section of a point source in front of a massive object with a magnetic field. Parallel light beams also were considered, and their bending due to the curvature of the space-time was simulated.

Keywords: general relativity; geodesics; differential geometry; free software.

Resumen

En este trabajo se simulan geodésicas nulas para la métrica de dipolo masivo de Bonnor, implementando un algoritmo simbólico-numérico en Sage y Python. Este programa es capaz, en principio, de visualizar en 3D las geodésicas de cualquier métrica. Estas geodésicas inician en un punto común, formando colectivamente un cono de rayos de luz, simulando una sección de ángulo sólido de una fuente puntual frente a un objeto masivo con campo magnético. Se consideran también rayos de luz paralelos, y se simuló su cambio de trayectoria debida a la curvatura del espacio-tiempo.

Palabras clave: relatividad general; geodésicas; geometría diferencial; software libre.

Mathematics Subject Classification: 83C10.

1 Introduction

In General Relativity, Einstein field equations are a set of non-linear partial differential equations that describe the curvature of space-time due to the presence of mass-energy-momentum. One of the exact solutions for the Einstein field equations in the vacuum was proposed in 1965 by W.B. Bonnor[2]; it describes a massive spherical stationary object with a dipole magnetic field. This solution, expressed using the metric line element $ds^2 = g_{\mu\nu}dx^\mu dx^\nu$ ([1]), in which $g_{\mu\nu}(\mathbf{x})$ is the metric describing the manifold solution to these equations is:

$$ds^2 = -\frac{Y^2P^2}{Q^3Z}(dr^2 + Zd\theta^2) - Z\left(\frac{Y}{P}\right)^2 \sin^2\theta d\phi^2 + \left(\frac{P}{Y}\right)^2 dt^2 \qquad (1)$$

[1] Einstein summation convention applies.

where

$$P = r^2 - 2mr - b^2cos^2\theta \tag{2a}$$

$$Q = (r-m)^2 - (b^2+m^2)cos^2\theta \tag{2b}$$

$$Y = r^2 - b^2cos^2\theta \tag{2c}$$

$$Z = r^2 - 2mr - b^2. \tag{2d}$$

t, r, θ, ϕ are the standard spherical coordinates. The geometric system of units is used along this paper ($c = G = 1$). The parameters m and b are defined, according to Bonnor, as b being such that if r becomes large, the magnetic field is that of a magnetic dipole along $\theta = 0$ of moment $mb\pi^{-\frac{1}{2}}$ and the gravitational field becomes that of a mass $2m$ at the origin [2].

Trajectories on this metric must be found by using the geodesic equations, which can be thought as lines of minimum length (result from $\delta\int ds = 0$):

$$\frac{d^2x^\kappa}{d\lambda^2} = -\Gamma^\kappa_{\mu\nu}\frac{dx^\mu}{d\lambda}\frac{dx^\nu}{d\lambda} \tag{3}$$

subject to initial conditions that determine the kind of geodesics[2]: time-like ($ds^2 > 0$, trajectories of particles with mass), light-like or null ($ds^2 = 0$, trajectories of photons), or space-like ($ds^2 < 0$, trajectories of hypothetical faster-than-light particles). λ is an affine parameter, and $\Gamma^\kappa_{\mu\nu}$ are the components of the Christoffel symbol.

In this paper null geodesics for the Bonnor metric are calculated in order to simulate the effect of the curvature of the space-time on a point source of electromagnetic radiation located at a certain distance from the origin, where the massive object is located. To achieve this purpose, a program in Python and Sage was developed to calculate analytically the Christoffel symbols from the metric and use them to build the geodesic equations. Once this is done, the equations are solved numerically using user-specified initial conditions, and these conditions are then varied to simulate a cone of light beams that represents a solid-angle section of the source.

Other efforts have been made in this topic. For example, Müler and Frauendiener (2010, [4]) developed Geodesic Viewer, which implements a selection of metrics which does not include the one studied here; and unlike this program, a local frame (tetrads) is required to specify the initial direction of the light beam. The justification for not using a local frame is that our scope is to study the

[2] The sign of ds^2 that determines kind of geodesic depends on the signature of the metric. In this paper, the signature of the metric is $(1, -1, -1, -1)$

collective behavior of geodesics, and high precision knowledge of the local directions is not required, as it would be for calculating gravitational lensing for such a point source. The Bonnor geodesics were also previously studied in 2D by Sanabria and Valenzuela [5], for charged test particles, as well as by Kovář et al. [3], for neutral particles, from the point of view of orbit stability.

Sage is an integrated environment consisting of several open-source packages (last count 89) with a common Python-based interface. It is capable of handling algebraic expressions (through Maxima and Sympy), as well as numerical calculations (through Scipy and Numpy), and many other tasks.

Vpython ("visual Python") is a 3D modeling library for Python, that uses OpenGL for rendering graphics. Its main features are its simplicity, and real-time visualization capabilities [6].

2 Equations of motion for the Bonnor metric

Expressions for the Bonnor geodesic equations were also calculated by means of the variational principle, to contrast them with the results generated with Sage. These expressions are:

$$\begin{aligned}\ddot{r} = &-\frac{1}{2}\frac{\partial}{\partial r}\ln\left(\frac{Y^2P^2}{Q^3Z}\right)\dot{r}^2 - \frac{\partial}{\partial\theta}\ln\left(\frac{Y^2P^2}{Q^3Z}\right)\dot{\theta}\dot{r} + \frac{Z}{2}\frac{\partial}{\partial r}\ln\left(\frac{Y^2P^2}{Q^3Z}\right)\dot{\theta}^2\\ &-\frac{Q^3Z}{2}\left[\frac{\varepsilon^2}{P^4}\frac{\partial}{\partial r}\ln\left(\frac{P^2}{Y^2}\right) + \frac{l^2}{Y^4Z\sin^2\theta}\frac{\partial}{\partial r}\ln\left(\frac{Y^2Z\sin^2\theta}{P^2}\right)\right]\end{aligned} \tag{4}$$

$$\begin{aligned}\ddot{\theta} = &-\frac{1}{2}\frac{\partial}{\partial\theta}\ln\left(\frac{Y^2P^2}{Q^3}\right)\dot{\theta}^2 - \frac{\partial}{\partial r}\ln\left(\frac{Y^2P^2}{Q^3}\right)\dot{\theta}\dot{r} + \frac{1}{2Z}\frac{\partial}{\partial\theta}\ln\left(\frac{Y^2P^2}{Q^3}\right)\dot{r}^2\\ &-\frac{Q^3}{2Y^4}\left[\varepsilon^2\frac{\partial}{\partial\theta}\ln\left(\frac{P}{Y^2}\right) - \frac{l^2}{Z\sin^2\theta}\frac{\partial}{\partial\theta}\ln\left(\frac{Y^2\sin^2\theta}{P^2}\right)\right]\end{aligned} \tag{5}$$

where the dots mean derivative with respect to λ, and ε and l are integration constants.

3 Our program

3.1 Analytical parts of the program

As sketched in the figure 1, the program is formed by several modules. In principle, the program is capable of taking any metric written in any coordi-

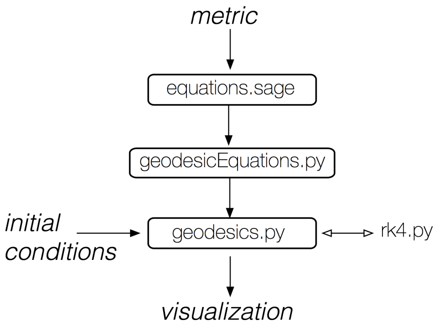

Figure 1: Structure of the program.

nate system and plotting geodesics with very little changes. The first module, `equations.sage`, computes the Christoffel symbols for the given metric analytically using Sage, and stores the result as a Python script that can be later used to do calculations using only Python. This Python script, named by default `geodesicEquations.py`, contains the right-hand side of the Eq. 3 defined as a function, as well as the metric.

3.2 Numerical calculations

The script `geodesicEquations.py` is used in the second module, `geodesics.py` to calculate and visualize the geodesics. The 4th-order Runge-Kutta method is used to solve the set of differential equations, and it is programmed in `rk4.py`. The numeric calculations may also be handled using Numpy, a Python library with a variety of optimized functions written for this purpose; or a specific numerical method can also be integrated to the code this way.

The initial conditions are given to this second module, and a function is called to vary them in order to form the light beam cone. Also, these initial conditions are varied in order to get a parallel set of beams. The given initial conditions are: the initial four-vector for the position, and the three spatial initial directions `v[1], v[2], v[3]` for the light beam; the other one is calculated by a function named `calculate_v0()` that uses the condition that $ds^2 = 0$ for null geodesics. Since we are not using local frames, these spatial directions are really specified respect to a distant observer, assuming an asymptotically flat metric. Using a local frame and the condition of null geodesics, the spatial initial directions can be specified using only two angles, as done in [4].

Users can give input to the program by creating scripts that call the files `equations.sage` and `geodesics.py`. In this way, multiple metrics and initial conditions can be tested with no changes to the central code, and more functionality can be added in the future.

The number of iterations and the step size can be specified, if not, default values are used. To avoid the effects of singularities on the behavior of geodesics, a stopping point based on distance to the origin is implemented and also can be specified in the main script.

3.3 Visualization

Vpython is used not only to plot the data, but also to manipulate the view in 3D in real time. Objects type `curve()` are used for the geodesics, and two small spheres mark the positions of both the source and the origin, where the massive object is located. A distinct arrow also points in the direction of $+z$ to indicate the direction of the dipole in the case of the Bonnor metric. While it is true that the code works for any coordinate system, a transformation to the Cartesian coordinate system is needed in order to represent positions in $\mathbb{R}^3$. So far, transformations from spherical and cylindrical coordinates to Cartesian and vice-versa are already implemented.

4 Results

4.1 Test runs

The program was tested first with Minkowski and Schwarzschild metrics. In Fig. 2a a light beam cone is drawn in a Minkowski space-time (in spherical coordinates), presented here to compare with the results obtained with other metrics. As it can be seen, each geodesic starts at the same point, marked by a small red sphere in the figure. As expected, geodesics are straight lines. Results for Schwarzchild metric are shown in Fig. 2b. Fewer geodesics are displayed this time to better show how they are curved by the space-time. As expected, the light beams even cross each other when passing very close to the event horizon, and they are just slightly deviated if they pass farther away.

Now, parallel beams are simulated for both metrics, results are shown in Fig. 3. In both cases, the direction of the beams is the negative x axis, and all the geodesics remain in the x-y plane. The mass for the Schwarzschild metric is set to 0.5. The main scope for running these test metrics, where geodesics are known, was to detect possible errors and select more carefully the initial conditions for later runs.

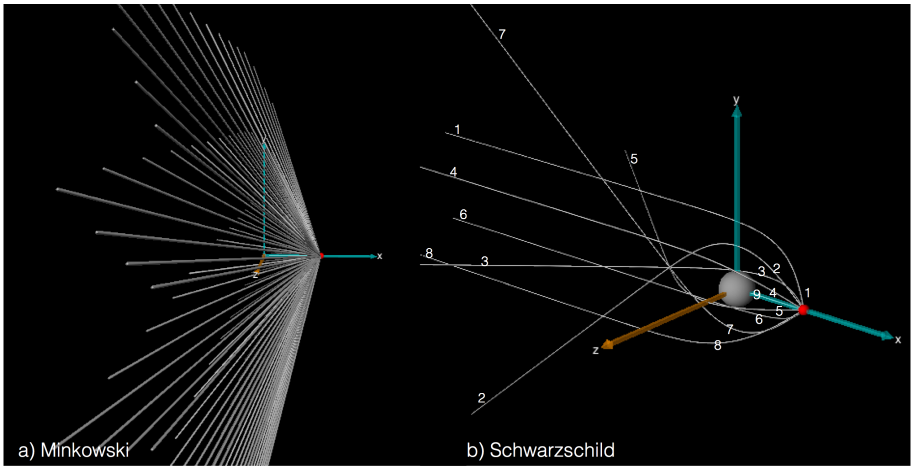

Figure 2: Test runs with point sources.

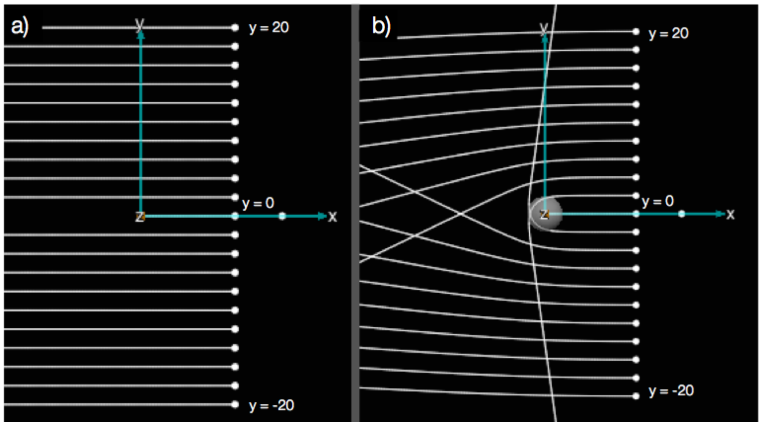

Figure 3: Test runs with parallel beams.

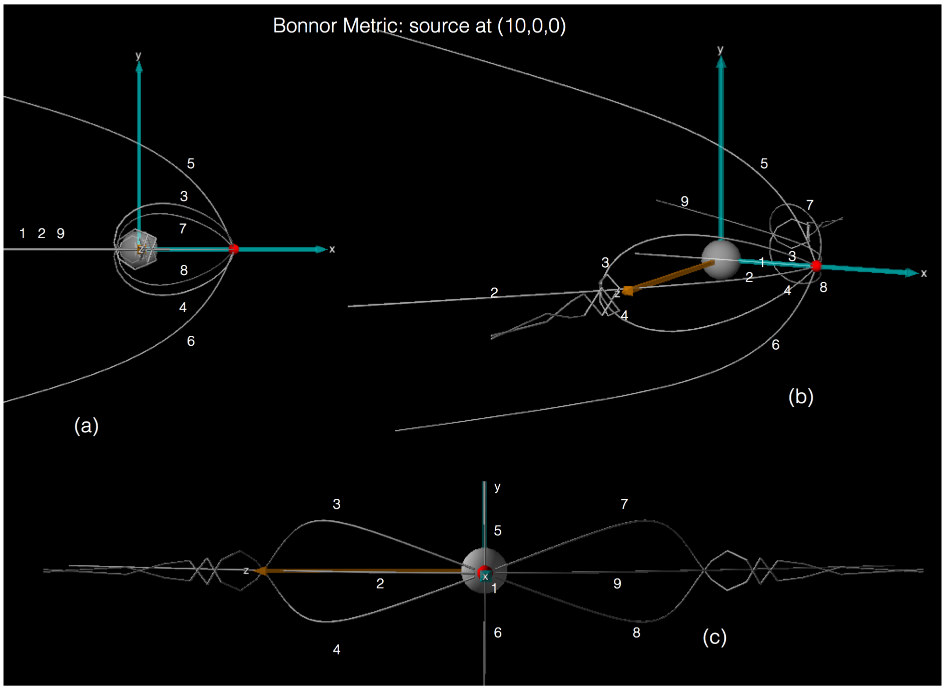

Figure 4: Geodesics for the Bonnor metric with $b = 0.001$.

4.2 Point source with $b < 1$

Now, to the Bonnor metric. In Fig. 4, nine geodesics have been simulated and viewed from different angles, to give a sense of spatial orientation. The source is located[3] at $(10, 0, 0)$, and the direction of the central geodesic is radially inwards, varied then in $-0.8 \leq$ `v[2]` ≤ 0.8 and $-0.3 \leq$ `v[3]` ≤ 0.3, corresponding to the θ and ϕ directions. 20 iterations with `dlam = 0.1` (step size) have been carried out. The metric parameters are $m = 1$ and $b = 0.001$.

The presence of the magnetic field makes the photons move in an helical motion around the z axis, that is, around the poles of the magnetic dipole, forming a jet-like structure. This behavior is consistent to the behavior of null geodesics for the Melvin metric, that is, locally, the magnetic field is dominant (see [1]).

[3] These position vectors are given in Cartesian coordinates; units are in the geometric system.

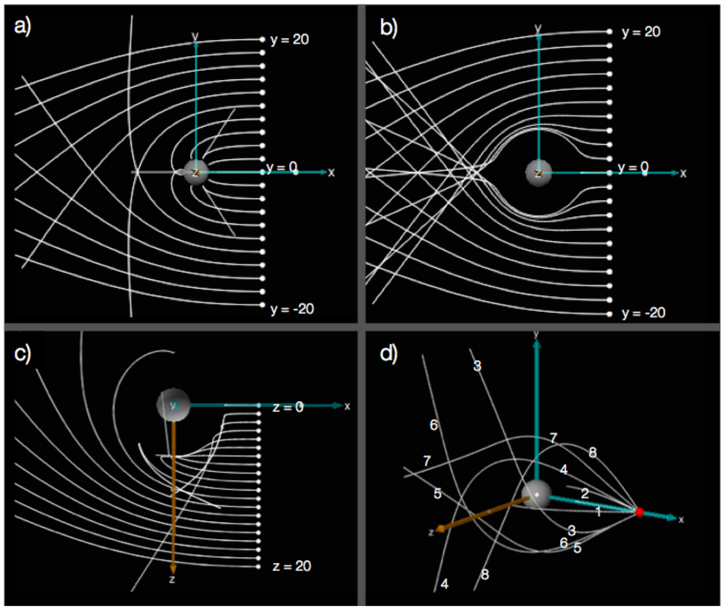

Figure 5: Geodesics for the Bonnor metric.

4.3 The values of b and m

Various values of b and m were tested. The behavior obtained in the preceding section corresponds to tested values $1 \leq b \leq 10^{-30}$. With values of b in the interval $2 \leq b \leq 10$ at least, the swirling seems to disappear. As predicted by Bonnor, with a value of $m = 0$ the space-time becomes flat and the result is similar to Fig. 3a. With a value of $b = 0$, also pointed out by Bonnor, the metric does not tend to the Schwarszchild one, but can be interpreted as a monopole of mass $2m$ together with higher mass multipoles dependent on m. Parallel beams of light on both metrics show the difference (compare Fig. 5a and Fig 3b).

4.4 Light beams with $b \sim 5$

For the tested values of $b = 2, 5, 10$ and $m = 1$ the results were similar. In Fig. 5b, the same x-y view is shown, for comparison with Figs. 5a and 3b. The magnetic field bends the light in such a way that a big opening is created around the object, and the light seems not to collide with it. Now, in Fig. 5c, variation in the z axis is tested to see the effect on the distance for the magnetic dipole. As expected, the effect decreases with distance, even in the z axis where the dipole points to. It is important to notice that for Figs. 5b and 5c, the curves shown are all in the same plane.

A point source located at $(15, 0, 0)$ shows a different story (Fig. 5d). Geodes̄ics are no longer necessarily contained in a plane, and bend creating an "envelope" around the object that is less evident with the distance.

5 Physical remarks

Parallel beams of light simulate distant sources and therefore can be used for gravitational lensing, helping to diagnose which kind of object causes a particular deformation on an image. Neutron stars can have very strong magnetic fields, up to the point of not being ignorable by General Relativity, but also have very rapid rotation rates, usually dominant. Light that behaves as shown in this paper, then, alerts of the presence of a massive object with a very strong magnetic field, but with negligible rotation. A point source, in the other hand, is also important to take into account, since several of these massive objects (black holes or neutron stars) also have companions, that is, nearby stars that act like point sources of radiation.

References

[1] Bonatti, J.; Semionova, L.; Páez, J. (1985) “Motion in Melvin metric”, *Uniciencia* **2**(1): 13–19.

[2] Bonnor, W.B. (1965) “An exact solution of the Einstein-Maxwell equations referring to a magnetic dipole”, *Zeitschrift für Physik* **190**: 444–445.

[3] Kovář, J.; Kopáček, O.; Karas, V.; Kojima, Y. (2013) “Regular and chaotic orbits near a massive magnetic dipole”, *Classical and Quantum Gravity* **30**(2), 025010.

[4] Müller, T.; Frauendiener, J. (2011) “Studying null and time-like geodesics in the classroom”, *Eur. J. Phys.* **32** 747 doi:10.1088/0143-0807/32/3/011

[5] Sanabria V., J.D.; Valenzuela, C.A. (2011) “Motion of charged test particles in a static spacetime endowed with magnetic field”, *Revista Colombiana de Física* **43**(1), 4 pages.

[6] Scherer, D.; Dubois, P.; Sherwood, B. (2000) “VPython: 3D interactive scientific graphics for students”, *Computing in Science and Engineering*, Sept./Oct. 2000: 82–88.