Gen Relativ Gravit (2018) 50:75
https://doi.org/10.1007/s10714-018-2387-z

RESEARCH ARTICLE

MALBEC: a new CUDA-C ray-tracer in general relativity

G. D. Quiroga[1]

Received: 24 January 2018 / Accepted: 5 April 2018

Abstract A new CUDA-C code for tracing orbits around non-charged black holes is presented. This code, named MALBEC, take advantage of the graphic processing units and the CUDA platform for tracking null and timelike test particles in Schwarzschild and Kerr. Also, a new general set of equations that describe the closed circular orbits of any timelike test particle in the equatorial plane is derived. These equations are extremely important in order to compare the analytical behavior of the orbits with the numerical results and verify the correct implementation of the Runge–Kutta algorithm in MALBEC. Finally, other numerical tests are performed, demonstrating that MALBEC is able to reproduce some well-known results in these metrics in a faster and more efficient way than a conventional CPU implementation.

Keywords CUDA-C · GPU · Timelike test particles · Geodesics around black holes

1 Introduction

The study and characterization of the geodesic motion around compact objects is extremely important in astrophysics. The orbits followed by test particles in the presence of a massive object can give information about the nature of the astrophysical source, also the dynamics around the compact sources like Neutron Stars (NS) or Black Holes (BHs) may be helpful to the gravitational wave observatories such LIGO [1]. For instance, during a binary black hole coalescence, in which one member of the binary is much more massive than the other. In this kind of system, it is possible to

✉ G. D. Quiroga
gonzalo.quiroga1@correo.uis.edu.co

[1] Grupo de Investigación en Relatividad y Gravitación, Escuela de Física, Universidad Industrial de Santander, A. A. 678, Bucaramanga, Colombia

Published online: 02 June 2018

assume that the small body moves on a geodesic of the background Kerr spacetime [2], at least during the early-stages. This is, in the test-mass limit, the full nonlinear relativistic force law corresponds to a geodesic motion in a Kerr spacetime. Thus, one way to investigate the properties of such astrophysical sources is through the study of the geodesic motion in these backgrounds.

In the particular case of BHs, the geodesic structure around these compact sources was studied during decades. A detailed study of the geodesic motion around Kerr and the Schwarzschild metric is summarized in the Chandrasekhar book [3], where timelike and null particles are analyzed, the stability of the trajectories is studied, and also the innermost stable circular orbits (ISCOs) are computed. In a recent work, the motion of classical spinning test particles in Schwarzschild and Kerr metrics was considered, and the ISCO was investigated [4]. In the cited work, small-spin corrections for the ISCO parameters were found analytically. On the other hand, the circular motion of charged test particles in the gravitational field of a charged BH described by the Reissner–Nordström spacetime was studied. This makes it possible to develop techniques that allow to distinguish between black holes and naked singularities [5].

Over the years, several analytical treatments have been carried out to track orbits around compact objects. For example, in the following Ref. [6], a system based on the Poincare approach is presented to classify orbits on the equatorial plane of rotating BH. In the mentioned article, the authors define a complete taxonomy of orbits that allows to establish a correspondence between periodic orbits and a set of rational numbers. On the other hand, the analytical solutions of bound timelike geodesic in Kerr were introduced in [7]. These solutions are given in terms of elliptic integrals using the Mino's method [8], which consist in a perturbative approach to track an orbital evolution around a super massive black hole. Bound geodesic orbits around a Kerr black hole can be parametrized by three constants of motion which can be associated with three frequencies related to the radial, longitudinal and azimuthal motion. However, it was shown recently that these two ways to characterize bound geodesics are not in a one-to-one correspondence in the strong field regime [9]. That is, there is a region of the parameter space in which pairs of physically distinct orbits can have the same three frequencies.

The geodesic motion of test particles is not restricted only to theoretical analysis, the nonlinearity of the system of equations requires in many cases the use of numerical technics. Several computational codes can simulate and track the trajectories followed by test particles around BHs. For instance, the open source ray-tracer GYOTO developed by Vincent et al. [10], or the GPU ray-tracer recently published by Daniel Kuchelmeister [11]. Both have a great handling of images and can integrate trajectories of test particles in vacuum solutions like Schwarzschild and Kerr, and in the case of GYOTO in numerical metrics as well. However, since our point of view, there several limitations in both implementations. For example, GYOTO has been developed using the standard CPU programming instead using the high performance of the actual graphics processing units (GPUs), which take advantage of the benefits offered by the CUDA platform developed by NVidia to accelerate the calculation of the orbits [12]. On the other hand, the GPU ray-tracer, only can launch null geodesics in the Kerr and/or Schwarzschild metric, and it does not use the $3 + 1$ formulation, which hinders the implementation of numerical metrics.

In this context arrived MALBEC, MALBEC is a new GPU code developed in CUDA-C for tracing orbits around compact objects in General Relativity (GR) that seeks to overcome the limitations of the previously mentioned implementations. Thus, in this work, we will introduce the first version of MALBEC, and we perform several validation tests. For this, we will analyze the motion of timelike particles around Schwarzschild and Kerr black holes. Now, from the mathematical aspect, MALBEC has been written using the $3+1$ formulation of the GR and is able to handle timelike and null geodesic in Schwarzschild and Kerr spacetime. The use of the $3+1$ formulation is important in order to bring future support to numerical metrics. Furthermore, in a future update, we will include extra analytical metrics as well as numerically computed, so the results presented in this article are not a limitation in our GPU implementation. From the technical issue, MALBEC implements the Runge–Kutta method, labeled as RK4, to find the solution of the system of ordinary differential equations (ODEs) that describe the geodesic motion around the compact source. MALBEC has been written in the programming language C and uses the CUDA platform [12] with a double precision arithmetic. Additionally, MALBEC was tested on Linux and Windows, and can work with a NVidia CUDA enabled GPU of computing capability 1.3 or higher, making use of the official NVidia NVCC compiler.

Since there are many technical details involved, it is better to outline the main ideas followed to perform the test of our code. In this way, the reader can have a broad picture without the technical complications. In order to perform the first validation of MALBEC, we will use the following procedure: first, we will derive the equations of motion for timelike and null particles by using the $3+1$ formulation. These equations are written as a system of eight coupled ordinary differential equations. So, we obtain a general set of equations that allow to calculate the initial conditions of any circular orbits in the equatorial plane of Kerr and Schwarzschild at any position outside of the event horizon. From this approach, a simple set of equations for the orbital energy and angular momentum is obtained, which are used to construct the set of initial conditions that are evolved by MALBEC, thus, the closed orbits and the stability regions can be verified numerically showing the correct implementation of the code. In other words, we will verify that the numerical solution reflects the expected theoretical behavior. Then, we focus on the convergence and speed tests, where we show that the numerical method implemented converges to the expected numerical order, and also prove that our code is capable of running a bigger number of geodesics in a shorter time than an analogous CPU implementation.

This article is organized as follows. In the Sect. 2, some mathematical foundations are given, particularly the $3+1$ decomposition of the GR is introduced, and the Kerr metric written in this formulation. In Sect. 3, the closed circular orbits in the equatorial plane of Kerr are derived, and the main equations of this work are presented. The special case of the circular orbits around a Schwarzschild black hole is treated in the Sect. 4. Then, the numerical simulations are performed in the Sect. 6. Also, in this section, we perform several tests in order to validate the efficient and correct execution of MALBEC. Finally, we closed the work given some final remarks and conclusions.

2 Foundations

In this section, we will introduce some mathematical foundations needed to develop our formalism. In particular, we will introduce the 3 + 1 decomposition of the GR which allows us to obtain the equations of motion followed by test particles around BHs, these equations of motion will be given as a system of ODEs.

In the 3 + 1 decomposition of the GR [13], the spacetime manifold $\mathcal{M}$ is assumed to be globally hyperbolic and to admit a foliation with spacelike hypersurfaces Σ_t, which are parametrized by the parameter $t \in \mathcal{R} : \mathcal{M} = \mathcal{R} \times \Sigma_t$ [14]. In this approach, the line element can be written as

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt), \tag{1}$$

where α defines the lapse of proper time $d\tau$, measured for those observers moving along the normal direction to the hypersurfaces $\mathbf{n} = -\alpha \nabla t$, where $\mathbf{n}$ is a timelike future pointing 4-vector, β is a spatial vector called the *shift* vector and describes how the spatial coordinates shift when moving from the slice Σ_t to another one $\Sigma_{t+\delta t}$, and γ_{ij} are the components of the induced metric over each hypersurface, such that γ is a projector orthogonal to the timelike vector $\mathbf{n}$, *i.e.*, $\gamma \cdot \mathbf{n} = 0$.

Now, the equations of motion for a free particle in the generalized spacetime (1), can be obtained from the following Lagrangian [13],

$$\begin{aligned}\mathcal{L} = &-\frac{1}{2}\left(\alpha^2 - \sum_{i=1}^{3}\sum_{j=1}^{3}\gamma_{ij}\beta^i\beta^j\right)\dot{t}^2 \\ &+ \left(\sum_{i=1}^{3}\sum_{j=1}^{3}\gamma_{ij}\beta^j\dot{x}^i\right)\dot{t} + \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\gamma_{ij}\dot{x}^i\dot{x}^j,\end{aligned} \tag{2}$$

here the dot denotes differentiation with respect to an affine parameter τ. Thus, following the canonical formulation for the geodesics [15], the equations of motion can be obtained by using the following expressions

$$\dot{x}^\mu = \frac{\partial \mathcal{H}}{\partial p_\mu}, \quad \dot{p}_\mu = -\frac{\partial \mathcal{H}}{\partial x^\mu}, \tag{3}$$

where

$$\mathcal{H} = \frac{1}{2} g^{\mu\nu} p_\mu p_\nu, \quad p_\mu = \frac{\partial \mathcal{L}}{\partial \dot{x}^\mu}, \tag{4}$$

are the Hamiltonian and the four momentum, respectively. It is worth mentioning that the energy is only derived from the kinetic energy in such way that $\mathcal{H} = \mathcal{L}$. In the case of charged particles this relation is not satisfied, that is $\mathcal{H} \neq \mathcal{L}$. Furthermore, the geodesic equation for a timelike particle, wrote in the 3 + 1 formulation, obey the following constraint,

$$2\mathcal{L} = -1 \tag{5}$$

Throughout this paper, we assume geometrized units, that is, $G = c = 1$. Also, we introduce the Kerr line element in Boyer–Lindquist (BL) coordinates $x^\alpha = (t, r, \theta, \phi)$ as [3],

$$
\begin{aligned}
ds^2 = &-\left(\frac{\Delta - a^2 \sin^2\theta}{\rho^2}\right) dt^2 - 4\frac{aMr\sin^2\theta}{\rho^2} dt d\phi \\
&+ \frac{\rho^2}{\Delta} dr^2 + \rho^2 d\theta^2 + \frac{\Sigma}{\rho^2} \sin^2\theta d\phi^2,
\end{aligned} \tag{6}
$$

with the usual definitions

$$
\begin{aligned}
\rho^2 &= r^2 + a^2\cos^2\theta, \\
\Delta &= r^2 - 2Mr + a^2, \\
\Sigma &= \left(r^2 + a^2\right)^2 - a^2\Delta\sin^2\theta.
\end{aligned}
$$

Here M is the mass of the BH, and a the black hole angular momentum per unit mass (J/M), note that in the case of the Schwarzschild metric the parameter $a = 0$. Additionally, we will investigate only the case $a \leq M$, since for $a > M$ the Kerr metric describes a naked singularity. Furthermore, the coordinate transformation between BL coordinates (r, θ, ϕ) and the Cartesian (x, y, z) is given by

$$
\begin{aligned}
x &= \sqrt{r^2 + a^2}\sin\theta\cos\phi, \\
y &= \sqrt{r^2 + a^2}\sin\theta\sin\phi, \\
z &= r\cos\theta.
\end{aligned} \tag{7}
$$

Now, by comparing Eqs. (1) and (6), we get

$$
\alpha = \sqrt{\frac{\rho^2\Delta}{\Sigma}}, \quad \beta^i = \left(0, 0, -\frac{2Mar}{\Sigma}\right), \quad \gamma_{ij} = \mathrm{diag}\left[\frac{\rho^2}{\Delta}, \rho^2, \frac{\Sigma}{\rho^2}\sin^2\theta\right], \tag{8}
$$

in this way the Kerr metric is expressed in terms of the lapse function α, the shift vector β^i, and the components of the induced metric γ_{ij}.

3 Closed circular orbits it the equatorial plane

In this section, we will obtain the equations of the closed circular orbits needed to test MALBEC. These orbits correspond to timelike test particles moving in closed circular trajectories in the equatorial plane of a non-charged black hole. The equatorial plane is the plane defined by the condition $z = 0$, or in BL coordinates $\theta = \pi/2$. Now, the restriction $\theta = \pi/2 = const.$ imposes the following condition $\dot{\theta} = 0$, thus, the ODEs system coming from the Eq. (7) are reduced to the following,

$$\dot{t} = \frac{-(2Ma^2 + a^2 r + r^3)p_t + 2aMp_\phi}{r(2Mr - a^2 - r^2)}, \tag{9}$$

$$\dot{r} = -\frac{(2Mr - a^2 - r^2)}{r^2} p_r, \tag{10}$$

$$\dot{\phi} = \frac{-2Map_t + (2M - r)p_\phi}{r(2Mr - a^2 - r^2)}, \tag{11}$$

$$\begin{aligned} \dot{p}_r = &-\frac{(Mr - a^2)}{r^3} p_r^2 + \frac{M(4Ma^2 r - a^4 - 2a^2 r^2 - r^4)}{r^2(2Mr - a^2 - r^2)^2} p_t^2 \\ &- \frac{2Ma(4Mr - a^2 - 3r^2)}{r^2(2Mr - a^2 - r^2)^2} p_t p_\phi + \frac{(4M^2 r - Ma^2 - 4Mr^2 + r^3)}{r^2(2Mr - a^2 - r^2)^2} p_\phi^2, \end{aligned} \tag{12}$$

$$\dot{\theta} = \dot{p}_t = \dot{p}_\theta = \dot{p}_\phi = 0, \tag{13}$$

In order to get these circular closed trajectories around Kerr, it is necessary to keep the radial coordinate constant, and also the radial canonical momentum, i. e.,

$$\dot{r} = 0, \quad p_r = 0. \tag{14}$$

Note that these two conditions are compatible with Eq. (10), also the last condition implies that $\dot{p}_r = 0$. Now, to guarantee that $p_r = 0$ in Eq. (12), we will set to zero the l.h.s of (12), and we will solve for p_r^2, from where we can write,

$$\begin{aligned} p_r^2 = &\frac{r(4M^2 r - Ma^2 - 4Mr^2 + r^3)}{(Mr - a^2)(2Mr - a^2 - r^2)^2} p_\phi^2 \\ &- \frac{2raM(4Mr - a^2 - 3r^2)}{(Mr - a^2)(2Mr - a^2 - r^2)^2} p_t p_\phi \\ &+ \frac{rM(4Ma^2 r - a^4 - 2a^2 r^2 - r^4)}{(Mr - a^2)(2Mr - a^2 - r^2)^2} p_t^2 \end{aligned} \tag{15}$$

Then, we introduce the last equation into the geodesic equation coming from (5) to get the following,

$$\begin{aligned} -1 = &-\frac{(M - r)(2Mr + a^2 - r^2)}{r(Mr - a^2)(2Mr - a^2 - r^2)} p_\phi^2 \\ &+ \frac{2Ma(2Mr + a^2 - 3r^2)}{r(Mr - a^2)(2Mr - a^2 - r^2)} p_t p_\phi \\ &- \frac{(2M^2 a^2 r + Ma^4 - 3Ma^2 r^2 - 2Mr^4 + a^4 r + a^2 r^3)}{r(Mr - a^2)(2Mr - a^2 - r^2)} p_t^2 \end{aligned} \tag{16}$$

Finally, we have obtained a system of two algebraic equations given by (16), and by the r.h.s of (15) equal to zero, i.e. $p_r^2 = 0$. On the other hand, it is very usual to associate the orbital energy E and the angular momentum about the normal axis to the equatorial plane L, to the canonical momenta p_t and p_ϕ respectively. Now, solving the system given by Eqs. (15) and (16) for positive energies $E > 0$, and writing the equations in terms of E and L, we find the following,

$$E^{\pm} = \left| \frac{4M^2r - Ma^2 - 4Mr^2 + r^3}{\sqrt{R \pm 2M^{3/2}r^{3/2}(2Mr - a^2 - r^2)a}} \right| \tag{17}$$

$$L^{\pm} = \pm \left| \frac{Ma(4Mr - a^2 - 3r^2) + r^{3/2}(2Mr - a^2 - r^2)M^{1/2}}{\sqrt{R \pm 2M^{3/2}r^{3/2}(2Mr - a^2 - r^2)a}} \right| \tag{18}$$

Here the superscript "+" represent the co-rotating orbit, while "−" the counter-rotating, also the symbol R is introduced for shorts, and it is given by,

$$R = r^2(-12M^3r + 5M^2a^2 + 16M^2r^2 - 3Ma^2r - 7Mr^3 + r^4). \tag{19}$$

Note the circular trajectories exist only if r takes values such that the factor inside of the square root be positive. On the other hand, the minimal in the energy for the co-rotating, or the counter-rotating orbit, can be found from the first derivative of Eq. (17), in the following way,

$$\frac{E^{\pm}}{dr} = 0, \tag{20}$$

then, we solve (20) for r, the roots of interest must meet simultaneously the following conditions: r must be a real value, should be located outside of the event horizon r_+ of the BH, that is $r \geq r_+$ with $r_+ = M + \sqrt{M^2 - a^2}$ [16]. Also, the test particle for this radius should have a positive energy and real angular momentum. Thus, this root will correspond to the "Innermost Stable Circular Orbits" (ISCO), which is the smallest orbit where the test particle can stably stable circular orbits around a BH. So, in our approach, the ISCO is given by,

$$\begin{aligned} r^{\pm}_{isco} &= 3M + \frac{1}{2}\sqrt{A + B + C} \\ &\mp \sqrt{-A - B + 2C + \frac{224a^2M + 1728M(a^2 - 6M^2)}{4\sqrt{A + B + C}}}, \end{aligned} \tag{21}$$

where

$$\begin{aligned} A &= \frac{4(a^4 - 10a^2M^2 + 9M^4)}{(a^6 + 17a^4M^2 - 45a^2M^4 + 27M^6 + 8\sqrt{a^{10}M^2 - 2a^8M^4 + a^6M^6})^{1/3}}, \\ B &= 4(a^6 + 17a^4M^2 - 45a^2M^4 + 27M^6 + 8\sqrt{a^{10}M^2 - 2a^8M^4 + a^6M^6})^{1/3}, \\ C &= 36M^2 + 4(a^2 - 6M^2), \end{aligned}$$

with $0 \leq a < M$. Furthermore, the ISCO for the extreme Kerr can be found taking the limit $a \rightarrow M$ in Eq. (21), doing this we recover the well know results [17],

$$r^{+}_{isco} = M, \quad L^{+}_{isco} = \frac{2}{\sqrt{3}}M, \quad E^{+}_{isco} = \frac{1}{3}, \tag{22}$$

$$r^{-}_{isco} = 9M, \quad L^{-}_{isco} = -\frac{22}{3\sqrt{3}}M, \quad E^{-}_{isco} = \frac{5}{3\sqrt{3}}, \tag{23}$$

where $L^{\pm}$ and $E^{\pm}$ are the energy and the angular momentum associated to the ISCO radius, which is computed from the Eqs. (17) and (18). In order to find the energy and the angular momentum associate with the co-rotating orbit in the extreme Kerr, L^{+}_{isco}, E^{+}_{isco}, must be taken the right-hand limit $r \rightarrow M^{+}$ since the radius of the orbit coincides with the event horizon. Finally, in the Sect. 6, we will use these equations to compute several ISCOs for Kerr.

On the other hand, in a closed circular orbit, the geodesics will describe a harmonic motion around the black hole. For those trajectories, the cartesian coordinates (x, y) will obey,

$$x = r\cos(\omega\tau + \alpha), \tag{24}$$

$$y = r\sin(\omega\tau + \alpha), \tag{25}$$

where τ is the proper time of the test particle, α is an initial phase, and r is the constant radius of the circular orbit. Now, the angular frequency ω can be found comparing Eqs. (24) with (7) assuming $\theta = \pi/2$, thus we can find the following equation,

$$\omega\tau + \alpha = \phi. \tag{26}$$

Now, by taking one derivative and using (11) we get,

$$\omega = \dot{\phi} = \frac{-2Map_t + (2M - r)p_\phi}{r(2Mr - a^2 - r^2)}, \tag{27}$$

so, the period of the motion will be given by,

$$T = \frac{2\pi}{\omega} = \frac{2\pi r(2Mr - a^2 - r^2)}{-2Map_t + (2M - r)p_\phi} \tag{28}$$

thus, the time taken by the particle to complete a lap around the Kerr BH depends on the radius, the energy, and the angular momentum.

4 Circular orbits around Schwarzschild black hole

In this section, we will apply our previous results in a spherically symmetric static black hole, which is given by the Schwarzschild metric. The Schwarzschild metric is a vacuum solution to the Einstein field equations that describes the gravitational field outside of a non-rotating massive spherically-symmetric object. It is possible to reduce the Kerr solution to the Schwarzschild ones by setting the parameter $a = 0$, thus the equations of motion for a timelike test particle are simpler than Kerr, and also can be obtained from (3). These equations can be written as follows [3],

$$\dot{t} = \frac{p_t}{1 - \frac{2M}{r}} \quad \dot{r} = \left(1 - \frac{2M}{r}\right) p_r \tag{29}$$

$$\dot{\theta} = \frac{p_\theta}{r^2} \quad \dot{\phi} = \frac{p_\phi}{r^2 \sin^2\theta} \tag{30}$$

$$\dot{p}_r = -\frac{M p_t^2}{(-r + 2M)^2} - \frac{M p_r^2}{r^2} + \frac{p_\theta^2}{r^3} + \frac{p_\phi^2}{r^3 \sin^2\theta} \tag{31}$$

$$\dot{p}_t = 0 \quad \dot{p}_\theta = \frac{p_\phi^2 \cos\theta}{r^2 \sin^3\theta} \quad \dot{p}_\phi = 0 \tag{32}$$

In the equatorial plane, the circular orbits around the BH are determined by the conditions,

$$\theta = \pi/2, \quad r = const., \tag{33}$$

which implies that $\dot{\theta} = 0 \rightarrow p_\theta = 0$, $\dot{r} = 0 \rightarrow p_r = 0$, and $\dot{p}_r = 0$. Now, the equations for the energy and the angular momentum can be obtained from the Eqs. (17) and (18) just imposing the condition $a = 0$, from where we can write,

$$E^2 = \frac{4M^2 - 4Mr + r^2}{r(r - 3M)}, \tag{34}$$

$$L^2 = \frac{Mr^2}{r - 3M}. \tag{35}$$

Note the energy must be a positive quantity, so one can observe that circular orbits for a timelike particle could not exist for $r \leq 3M$. Now, the minimal in the energy is found by taking a derivative to Eq. (34) and solving for r as follows,

$$\frac{dE}{dr} = \frac{M(6M - r)}{2r^{3/2}(r - 3M)^{3/2}} = 0. \tag{36}$$

Thus, for $r = 6M$ we get the critical point which corresponds to the Schwarzschild ISCO. Now, evaluating the Eqs. (34) and (35) in $r = 6M$ we get

$$r = 6M \quad E = \sqrt{\frac{8}{9}} \quad L = \sqrt{12}M. \tag{37}$$

The values $r = 3M$ and $r = 6M$ defines two ranges for the radius where the orbits will have different behaviors and stabilities [3],

$$6M < r(\text{stable}) < \infty, \quad \text{and} \quad 3M \leq r(\text{unstable}) \leq 6M. \tag{38}$$

The circular orbits of the larger radius, where the minimum of the energy is located, will be stable in contrast to the circular orbit of the smallest radius where the energy is undefined [3].

Now, in several applications, it is quite convenient to assume the radius of the orbit as proportional to the mass, i.e. $r = kM$ with $k > 0$. In general, if we write $r = kM$ for Schwarzschild the constant must satisfy $k > 3$ in order to get closed orbits, thus we can write the following equations,

$$r = kM \quad p_t = \sqrt{\frac{k^2 - 4k + 4}{k(k-3)}} \quad p_\phi = \frac{kM}{\sqrt{k-3}} \tag{39}$$

Finally, putting $a = 0$ in Eq. 28 we get the period of the circular motion,

$$T = \frac{2\pi}{\omega} = \frac{2\pi r^2}{p_\phi}, \tag{40}$$

thus, for a non-rotating BH, the time to complete a lap around the singularity do not depends on energy, just depends on the radius and the angular momentum.

5 CUDA-C implementation for computing orbits in general relativity

In 2006, Compute Unified Device Architecture (CUDA) was introduced by NVidia, this new platform has opened the possibility of using NVidia GPUs to parallel computing. CUDA leverages the parallel compute engine in video cards to solve many complex computational problems in a more efficient way than on a CPU. Also, the Nvidia tool-kit comes with a software environment that allows to use C as a high-level programming language [18].

The CUDA platform has many benefits for paralleling computing, benefits such as the existence of a shared memory (i.e. a memory area to be shared between threads), faster loads of data between GPU and CPU, full support for integer, bitwise operations and scattered reads. Despite CUDA has many advantages over other types of computing systems, there are some limitations that must be taken into account during its implementation. For example, in single precision NaNs are not supported, you can not use pointers to functions or functions with variable parameters. Also, for reasons of efficiency, the threads must be launched in groups of at least 32 with thousands of threads in total, but the biggest drawback, is a bottleneck between the CPU and the GPU by bandwidths and latencies buses. This bottleneck may affect the rate of data transfer CPU-GPU and thus the efficiency in the execution of the kernels.

Malbec, is a new CUDA-C code for tracing orbits around compact sources, this code takes advantage of the graphic processing units and the CUDA platform in order to track the geodesic motion of timelike and null test particles in curved spacetimes. The code is divided into several source files and headers located in different folders as one can see in the following diagram (Fig. 1).

The main.cu is the main file of the code, this introduces some spacetime parameters such as the BH rotation, the number of initial conditions, and controls all the necessary

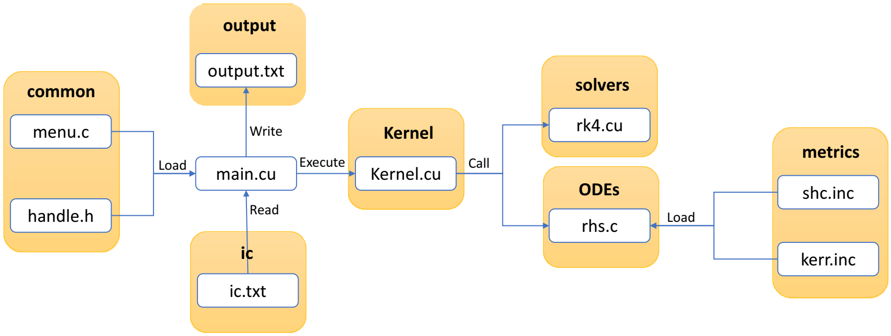

Fig. 1 MALBEC code structure. The orange blocks represent the folders where the files are located, for simplicity some headers are not included in the diagram

processes during the execution. The code can be compiled with the standard nvcc, e.g. in a UNIX system like Ubuntu, the command line will be `nvcc main.cu -o malbec.out`. In the main file, the host and device input/output are defined, also, following the standard CUDA programming, the GPU is denoted as device and the CPU as host. The memory for each vector on the GPU and CPU is allocated, and the initial conditions are read. Moreover, the initial conditions are stored in a text file as an array of $8 \times N$, where N is the desired number of independent initial conditions or systems.

t	r	θ	ϕ	p_t	p_r	p_θ	p_ϕ
0.0000	6.0000	1.5708	0.0000	0.9428	0.0000	0.0000	3.4641
0.0000	3.5000	1.5708	0.0000	1.1339	0.0000	0.0000	4.9497
0.0000	4.0000	1.5708	0.0000	1.0000	0.0000	0.0000	4.0000
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Each row corresponds to the initial conditions of a test particle ordered as shown in the previous diagram. Note that each initial condition is a double precision number, however, for simplicity we show four digits.

Additionally, the ODEs system is composed by eight differential equations which are hosted in the ODEs folder, the rhs.c handles the sets of equations. In the current version, MALBEC includes the geodesic equations in Kerr and Schwarzschild spacetime. However, in a future update, other spacetimes will be included and also the code capabilities will be extended to handle numerical metrics. MALBEC use a GPU implementation of the traditional fourth order Runge–Kutta (RK4) method, others method of the RK family will be included soon. The use of the Runge–Kutta family is a well extended practice in order to solve a system of ODEs, a complete description of the implicit and explicit RK solvers can be found in Ref. [19]. Additionally, the output file is an ASCII text, where all the generated data during the integration is written, this file will be stored in the output folder in the following format,

τ	t	x	y	z	
0.0000	0.0000	6.0000	0.0000	0.0000	→ i.c. for the 1st test particle
0.0000	0.0000	3.5000	0.0000	0.0000	→ i.c. for the 2nd test particle
0.1000	0.1414	5.9997	0.0577	0.0000	→ 1st test particle in $\tau = h$
0.1000	0.2646	3.4971	0.1414	0.0000	→ 2nd test particle in $\tau = h$
0.2000	0.2828	5.9989	0.1155	0.0000	→ 1st test particle in $\tau = 2h$
0.2000	0.5292	3.4886	0.2825	0.0000	→ 2nd test particle in $\tau = 2h$
⋮	⋮	⋮	⋮	⋮	⋮

The previous example shows the output file for two geodesics in Schwarzschild. The output.txt file is printed following the same particle order in the ic.txt file, in addition, the numerical solutions are stored in Cartesian coordinates.

In CUDA, the threads are grouped into blocks, and the blocks are grouped into grids. All threads running in the same block share the fast shared memory, these threads can exchange data using this shared memory. It is convenient to understand the restrictions on the kernel and the GPU in which the different threads are running, because the proper choice of the block size can affect the code performance. One of the keys to improve the performance is to keep the multiprocessors on the device as busy as possible. To implement this idea, the maximum occupancy criterion is used in MALBEC. The occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of warps that can be active on the multiprocessor at once. Higher occupancy does not always translates into higher performance, there is a point above which additional occupancy does not improve performance. However, low occupancy always interferes with the ability to hide memory latency resulting in performance degradation, according to the CUDA-C best practices guide [20].

The kernel.cu file is responsible for calling the integrator method hosted in the file rk4.cu and the rhs.c file, which loads the system of differential equations. Now, the integration goes on until one of the following stop conditions are fulfilled.

- The final evolution time is reached.
- The particle approaches to the event horizon, i.e. the radial coordinate $r \leq 1 + \sqrt{1 - a^2}$.

The first stop condition controls the global iterations, that is, the main loop of all threads in the main file. This condition stops the evolution of all initial conditions when final time set by the user is reached. On the other hand, the second stop condition is checked individually for each thread. This verification is written inside of the rk4.cu file and stops the evolution of the particular thread when the test particle reach to the event horizon.

Finally, there are an auxiliary header and an extra file located in the common folder, these files contains some functions definition needed to check the right data transfer to the device, and also to display the MALBEC option menu. Finally, the numerical implementation made in MALBEC allows to solve a large set of system independently. Currently, MALBEC continues under development, however, this first release v0.5.0 is available and free to download from the following link: https://github.com/GonzaQuiro/MALBEC.

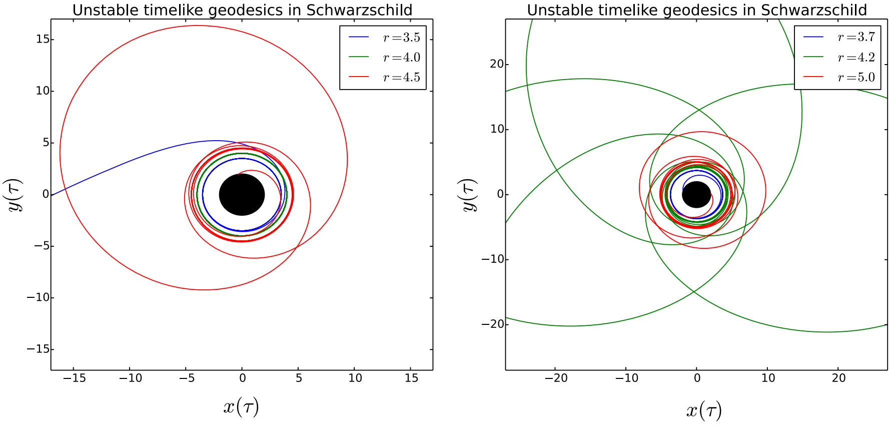

Fig. 2 These figures show some unstable orbits around Schwarzschild. Since these orbits are unstable, will not keep in its circular closed trajectories

6 Numerical analysis

In this section, we will perform several simulations of timelike orbits, these trajectories are numerically evolved using MALBEC. As mentioned above, the goal in this work is to introduce MALBEC into the community and perform several tests to ensure proper execution of our GPU implementation, and also measures its performance. For this, we start by studying the circular motion of timelike test particles around Schwarzschild and Kerr BH using the standard RK4 method with a step-size $h = 0.001$. The initial conditions are computed by Eqs. (17) and (18) assuming a unit mass as is usual in the numerical practices. Then, at the end of this section, we focus on the calculation of the convergence rates, we also make a performance test comparing our GPU scheme against an analogous implementation in CPU.

6.1 Schwarzschild black hole

Consider first, the closed orbits around a Schwarzschild black hole in the equatorial plane. Now, using MALEBC, we compute some unstable orbits in the equatorial plane with a radius between the region $3 \leq r \leq 6$. The Fig. 2 show the numerical results of these simulations,

Since these orbits are unstable, they are expected to be sensitive to small fluctuations in the energy and momentum presumably caused by the accumulated error inherent in the numerical method. In other words, these orbits cannot remain in their circular trajectories indefinitely, instead all the unstable trajectories will give a certain number of laps around the compact object before to be expelled or reach the event horizon. On the other hand, the orbits which are located at radius greater and equal to the ISCO, can be kept in their closed circular orbits, as one can see in the Fig. 3.

A different behavior can be observed between the orbits initiated in the unstable region against the stable ones. As mentioned above, small fluctuations in energy and

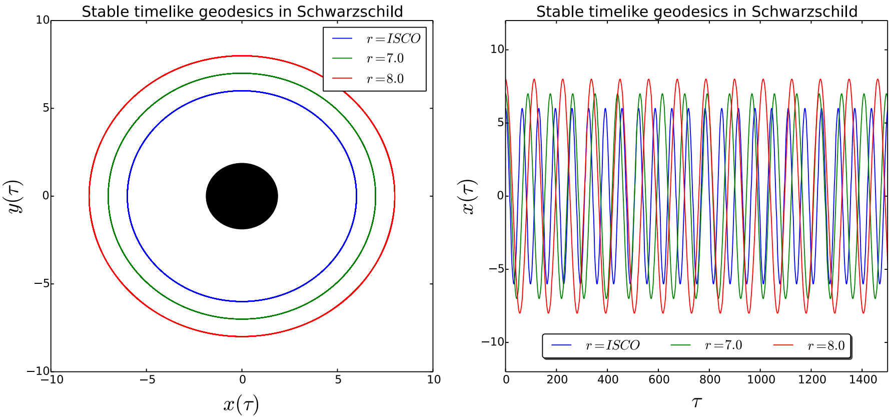

Fig. 3 Some stable trajectories around the black hole, in the right figure each oscillation of the x coordinate represent a lap around the compact source

momentum produce that the test particle leaves its circular closed trajectory. This situation can be observed in MALBEC, just running the same initial condition using different integration steps. The Fig. 4 show how the step-size h, and therefore, the numerical precision (or the numerical resolution) may affect the final state of the test particle.

Note that the blue horizontal line in the figure of the lower right corner represents that the particle reaches the event horizon of the black hole, so the time evolution of the geodesic finished. On the other hand, stable orbits are more "robust" to these differences and remain unchanged regardless of the integration step, as we can see in the Fig. 5.

Finally, in these numerical simulations it is possible to evidence the behaviour of the unstable and stable orbits. As one can see in this subsection, the code is able to evolve the timelike test particles by keeping the stable orbits at the same radius around the black hole. However, the geodesics located in the unstable region show the expected variations in the radial coordinate due to the unstable nature of the orbits.

6.2 Kerr black hole

The Kerr spacetime describes the geometry of a rotating uncharged axially-symmetric black hole, so it is possible to find two types of closed trajectories, one co-rotating to the black hole and the other counter-rotating. All the simulations performed by MALBEC in this subsection will be assuming $M = 1$ and Kerr parameter $a = 0.5$. However, the results presented here are analogous to those obtained for any a. For a Kerr BH of $a = 0.5$, the co-rotating and counter-rotating ISCOs are given by the following initial conditions,

$$r^{+}_{ISCO} = 4.233002529530 \quad p^{+}_{t} = 0.9178820066607 \quad p^{+}_{\phi} = 2.902866153235,$$
$$r^{-}_{ISCO} = 7.554584714512 \quad p^{-}_{t} = 0.9548577730472 \quad p^{-}_{\phi} = -3.884212632015,$$

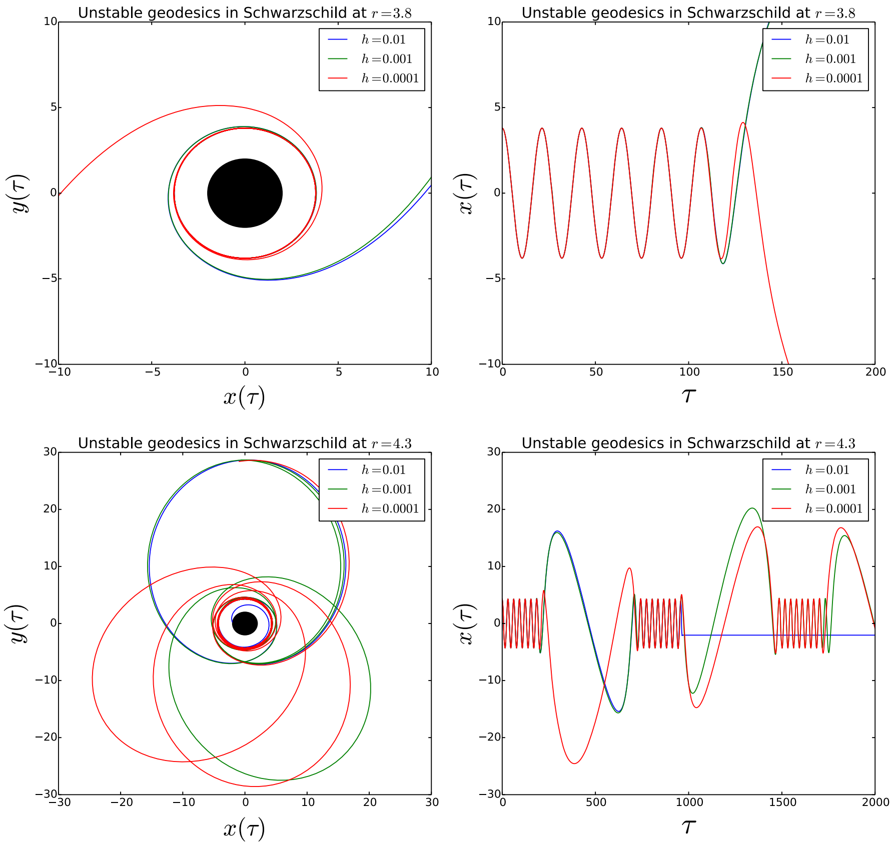

Fig. 4 These figures show how the numerical resumption affects the trajectory and the number of oscillations of two unstable geodesics

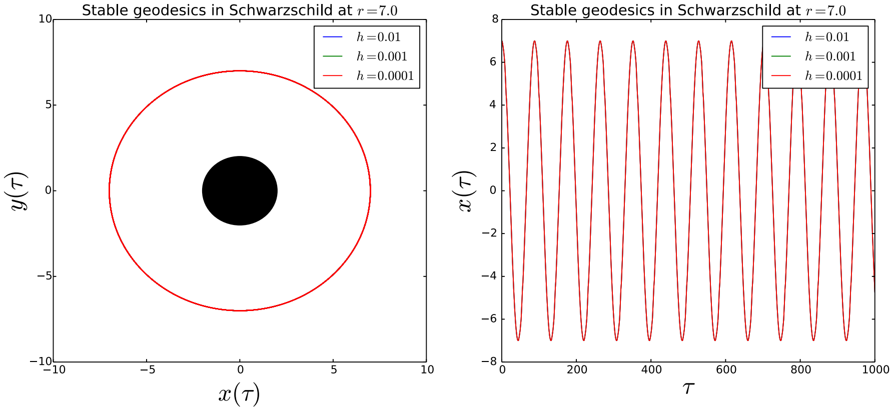

Fig. 5 These figures show the trajectory followed by a timelike particle in a stable orbit, the integration was performed using different numerical resolutions

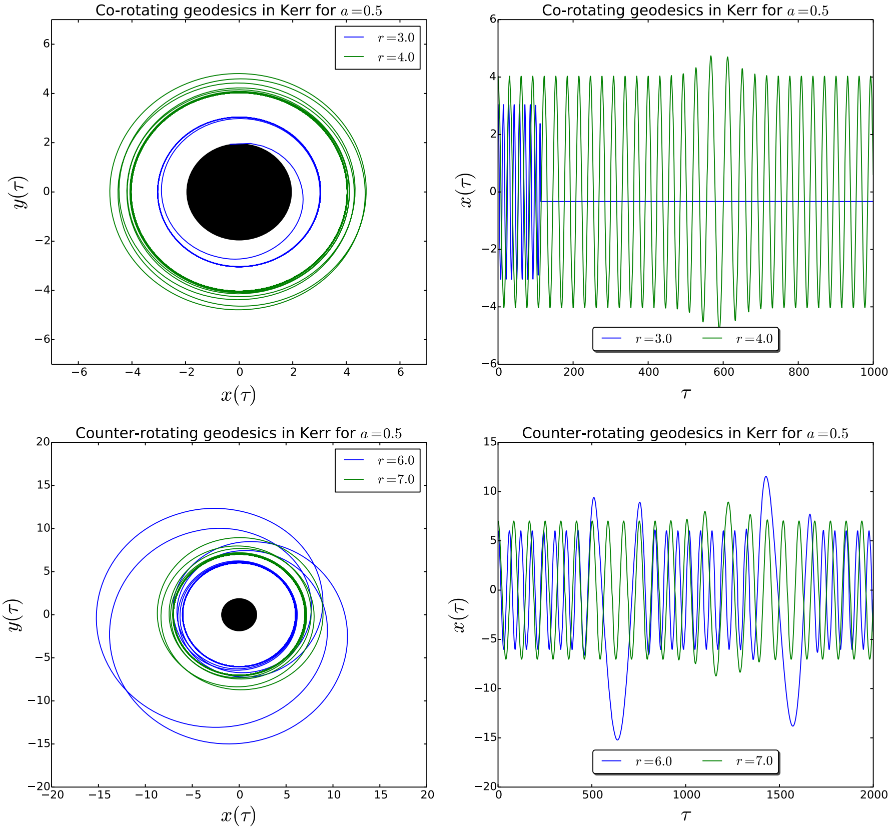

Fig. 6 Unstable circular orbits for a Kerr spacetime with $a = 0.5$ and $M = 1$. The upper figures correspond to co-rotating geodesics inside of the co-rotating ISCO while the inferior are counter-rotating orbits

where the ISCO radius can be calculated from Eq. (21) together with Eqs. (17) and (18). We start by showing some trajectories inside of the ISCO, see Fig. 6,

It can be seen from the figures on the right that the test particles oscillate around the compact source with a non-constant radius, also in some cases they can reach the horizon of the black hole. As we can appreciate in the Fig. 6, these particles deviate from their circular trajectories. This kind of behavior is a typical feature of unstable orbits as we discussed in the Schwarzschild subsection. Thus, we can conclude that the co-rotating orbits inside of the co-rotating ISCO, and analogously the counter-rotating orbits inside of the counter-rotating ISCO, are unstable. On the other hand, the geodesics located outside of ISCO are perfectly closed and they keep their radius constant, as we can see in the following figures (Fig. 7).

As in the Schwarzschild case, the orbits initiated in the region $r \geq r_{ISCO}$ are stable trajectories and therefore will not be affected by the level of resolution chosen by the user. Thus, we can highlight two regions, one unstable between the horizon and r_{ISCO}, and another stable which is outside of the ISCO. Finally, and for completeness, we compute from Eq. (21) the ISCO for several angular momentum-mass ratio for a

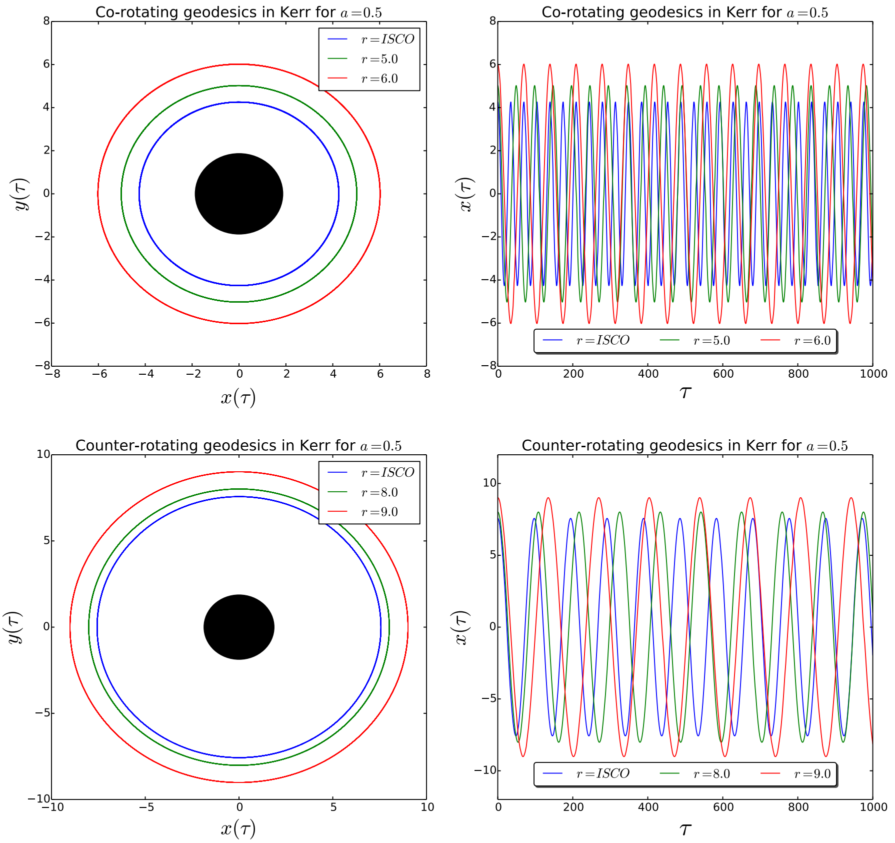

Fig. 7 These figures show the ISCO and two stable circular orbits for a Kerr BH of $a = 0.5$. The upper figures correspond to the co-rotating trajectories while the lower are the counter-rotating

Kerr BH of unit mass (Tables 1, 2). Firstly, we show the co-rotating ISCOs with their associated Kerr parameter, orbital energy, and angular momentum, and then, the counter rotating orbits.

These values are useful for testing numerical codes because they delimit the regions of stability of the orbits. Also, these values coincide with those obtained from Ref. [4] for non-spinning particles. However, it should be noted that our equations are more general in the sense it allows to compute any circular orbit around the compact source, not just the ISCO as in that reference. Finally, based on the numerical results, we can assure that the code has a good error handle, and evolves the system of equations correctly.

6.3 Numerical convergence and error analysis

Now, we are ready to investigate about the numerical accuracy of MALBEC by computing the convergence rates using the self-convergence test. For that, we introduce

Table 1 Co-rotating ISCOs in Kerr for $M = 1$ and $0.1 \leq a \leq 1$

a	r	p_t	p_ϕ
0.1	5.669302571208	0.9393655520586	3.367109920810
0.2	5.329443296434	0.9353655759711	3.264028637021
0.3	4.978616830575	0.9306417139417	3.153598281507
0.4	4.614335370564	0.9249447067862	3.034065651040
0.5	4.233002529530	0.9178820066607	2.902866153235
0.6	3.829069418814	0.9087867149047	2.755986288640
0.7	3.393128470181	0.8963952722476	2.586500326227
0.8	2.906643854506	0.8778612656718	2.380440624350
0.9	2.320883041784	0.8442470080056	2.099784756124
1	1	0.5773502691896	1.154700538379

Table 2 Counter-rotating ISCOs in Kerr for $M = 1$ and $0.1 \leq a \leq 1$

a	r	p_t	p_ϕ
0.1	6.322894723789	0.9458134386776	– 3.555943626969
0.2	6.639040203567	0.9484639574909	– 3.643358857192
0.3	6.949272527004	0.9508242198443	– 3.726917098184
0.4	7.254268411283	0.9529428538771	– 3.807076678477
0.5	7.554584714512	0.9548577730472	– 3.884212632015
0.6	7.850686185306	0.9565990449850	– 3.958636355418
0.7	8.142965464834	0.9581908703729	– 4.030609693862
0.8	8.431757830806	0.9596529818543	– 4.100355267874
0.9	8.717352279606	0.9610016543547	– 4.168064196332
1	9	0.9622504486493	– 4.233901974057

a coefficient Q, which is used to measure the convergence order of our RK4 implementation. This coefficient is found by computing the ratio of the differences between y_h, $y_{h/2}$, $y_{h/4}$, which are the numerical solution with step-sizes h, $h/2$, and $h/4$ respectively. Then, the coefficient Q is defined as,

$$Q = \frac{\parallel y_h - y_{h/2} \parallel}{\parallel y_{h/2} - y_{h/4} \parallel} = 2^m, \tag{41}$$

where m is the order of the numerical method, $\parallel y_h - y_{h/2} \parallel$ corresponds to subtraction between the numerical solution y using the step h, and $h/2$ respectively. We apply the Euclidean norm to find the differences between y_h and $y_{h/2}$, and also between $y_{h/2}$ and $y_{h/4}$ as follows,

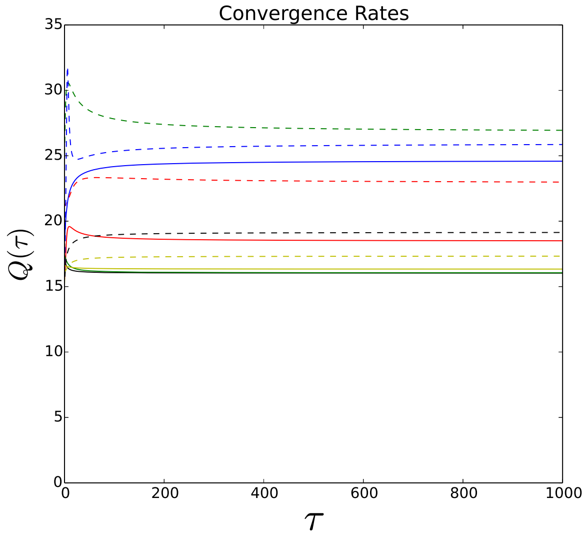

Fig. 8 Precision coefficient for 10 timelike geodesics solved numerically in parallel using the RK4 implementation of MALBEC

$$\parallel y_h - y_{h/2} \parallel = \sqrt{\sum_{i=1}^{8} |y_h^i - y_{h/2}^i|^2}, \tag{42}$$

$$\parallel y_{h/2} - y_{h/4} \parallel = \sqrt{\sum_{i=1}^{8} |y_{h/2}^i - y_{h/4}^i|^2}, \tag{43}$$

here y_h^i, $y_{h/2}^i$, and $y_{h/4}^i$ are the i-th components of the numerical solution. The sum ranges from 1 to 8 since we are solving a system of eight ordinary differential equations (see Sect. 2).

In order to compute the precision coefficient Q, we evolve 10 random timelike geodesic in the Kerr spacetime with parameter $a = 0.5$. The results of this test is shown in the following Fig. 8,

Since the RK4 is a fourth order method, the coefficient Q given by Eq. (41) must be greater or closer to $2^4 = 16$ for all time τ in a well implemented method. Also is expected that the coefficient Q has fluctuations around 16 or tend to this value as time increases as we can see in the last figure. So, based on the curves plotted in Fig. 8, where the $Q \geq 16$, we can confirm that our numerical implementation works adequately.

6.4 Measuring MALBEC performance

Finally, we focus on evaluating the performance of the Runge–Kutta solver implemented in CUDA. The speed-up of MALBEC will be compared with and equivalent algorithm written in a standard sequential C. All the numerical runs in this paper were performed on the Intel Core i7 CPU with 8 cores running at 2.6 GHz, 8 Gb of

Table 3 Runtime between GPU and CPU for the RK4 implementation

Initial conditions	Runtime (min)		
	t_{GPU}	t_{CPU}	$\frac{t_{CPU}}{t_{GPU}}$
10	0.0545956	0.04732473	0.9
100	0.11868735	0.49806882	4.2
500	0.3829293	2.43829992	6.4
1000	0.66657395	4.84753887	7.3
5000	3.543259	24.8818684	7.0
10000	7.14927418	48.8400202	6.8
50000	35.6253648	241.872849	6.8

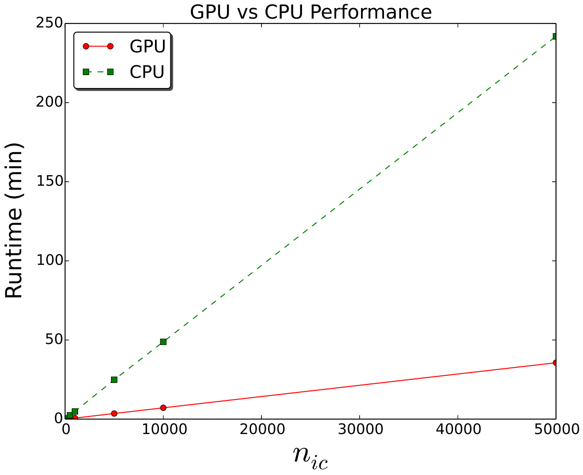

Fig. 9 The MALBEC runtime when 50 thousand random timelike geodesics in a Kerr spacetime are evolved. The red line correspond to the RK4 GPU implementation while the green line to the CPU

RAM, a Nvidia Geforce GTX 960M with CUDA Toolkit 7.5.17, and GNU/gcc 4.8.4 installed on the 64-bit GNU/Linux Ubuntu 14.04. For this performance test, we will measure the runtime needed to solve a set of maximum 50 thousand timelike random geodesics. These geodesics will describe orbits around a Kerr BH with rotation parameter $a = 0.5$, the final time for those running will be $t_f = 1000$ and the step-size used will be $h = 0.1$. We decided to use the Kerr BH instead of Schwarzschild since the system of ODEs is more complex. The temporal metric used to compute the time during this test is the C function **clock_t** located in the **time.h** header. The time measured in these running are shown in Table 3.

Now by plotting these points, it is possible to evidence a linear behavior in the execution times. Therefore, making a linear regression, it is possible to estimate the time necessary to evolve an arbitrary number of initial conditions n_{ic} in CPU and GPU (Fig. 9).

Table 4 Runtime between GPU and CPU without data saving

Initial conditions	Runtime (s)		
	t_{GPU}	t_{CPU}	$\frac{t_{CPU}}{t_{GPU}}$
10	2.16987	2.496072	1.2
100	2.301033	26.063659	11.3
500	4.628749	128.347269	27.7
1000	4.63499	254.726977	55.0
5000	13.598218	1270.015683	93.4
10000	27.066675	2550.618381	94.2

These equations can we written as,

$$t_{GPU} = 0.0007 n_{ic} + 0.0146, \tag{44}$$

$$t_{CPU} = 0.0048 n_{ic} + 0.1876. \tag{45}$$

Thus, the time needed to run 1 million of initial conditions can be estimated in $t_{CPU} \approx$ 3.3 days versus $t_{GPU} \approx 11.7$ h, showing an approximate speed-up factor of 7 in our GPU implication.

The speed-up ratio depends mainly on two factors, the GPU-CPU data transfer, and the data write rate to disk. Thus, this factor may increase if the computer used have higher specifications, particularly if the PCI-e bus speed between the device and the host is higher. On the other hand, the data saving can be optimized by using unformatted output, which is the most efficient way to store data, such as a portable binary. In order to illustrate both situations, we will run again a set of initial conditions, but disabling the data saving in MALBEC and the memory copy from device to host. Finally, the time measured is shown in the Table 4.

As one can see in this table, the performance of the GPU code increases notably while the CPU has a small variation (see Table 3). MALBEC uses a formatted output, however, the formatted output is more computationally expensive because of the need to convert between internal binary data to ASCII text, besides, the data is written from an array which have to iterate through the array elements. On the other hand, unformatted data is not directly human readable, therefore, special care must be taken when the file is written, since it could happen a mixture between the data of the different orbits evolved simultaneously. We are evaluating to implement the binary outputs, in our next update, in order to improve even more the speed-up of the code.

7 Final remarks and conclusions

We introduce and validate a new general relativistic ray-tracer code named MALBEC, this code uses a GPU implementation of the Runge–Kutta solver to integrate null and timelike geodesics around compact sources. The system of ODEs solved by MALBEC is obtained from the $3+1$ formulation of the GR, this describe the geodesic motion for test particles moving around Schwarzschild and Kerr black hole.

In order to validate our code, we derive a general set of equations that describe any closed circular timelike orbits around Schwarzschild or Kerr. These equations represent the orbital energy and the angular momentum of the test particle outside the event horizon in the equatorial plane of the source. They are obtained by applying several restrictions, like to assume a constant radius and set the radial momentum to zero in the differential equations that govern the geodesic motion.

These equations are critical to show that MALBEC performs the correct integration of the ODEs system, since the numerical simulation can be contrasted with the theoretical orbits in order to verify the correct numerical evolution. Several orbits, in different stability regions, were subjected to numerical simulations. These simulations show that the test particles fulfil the expected behavior of the theoretical orbits, validating the numerical solutions obtained by MALBEC.

On the other hand, other tests were performed to our CUDA RK4 implementation. First, we check the convergence rates of MALBEC, and then we do a performance test where a large number of simulations were launched in parallel, obtaining a speed-up factor of seven compared with a usual CPU implementation. Finally, based on the excellent results obtained in each test, we can conclude that the first version of MALBEC is ready to be released as a public open source. The code can be downloaded from the following repository: https://github.com/GonzaQuiro/MALBEC. In a next update, we will include extras metrics and astrophysical objects, we develop a new algorithm to handle numerically computed metrics, and also we investigate several performance technics for increase the speed of MALBEC.

Acknowledgements The author would like to thank to Dr. Omar Ortiz for his suggestions concerning the convergence test.

References

1. The LSC: Advanced ligo. Class. Quantum Grav. **32**(7), 074001 (2015)
2. Sundararajan, P.A., Khanna, G., Hughes, S.: Binary black hole merger gravitational waves and recoil in the large mass ratio limit. Phys. Rev. D **81**(10), 104009 (2010)
3. Chandrasekhar, S.: The Mathematical Theory of Black Holes, vol. 69. Oxford University Press, Oxford (1998)
4. Jefremov, P.I., Yu Tsupko, O., Bisnovatyi-Kogan, G.S.: Innermost stable circular orbits of spinning test particles in schwarzschild and kerr space-times. Phys. Rev. D **91**(12), 124030 (2015)
5. Pugliese, D., Quevedo, H., Ruffini, R.: Motion of charged test particles in Reissner–Nordström spacetime. Phys. Rev. D **83**(10), 104052 (2011)
6. Levin, J., Perez-Giz, G.: A periodic table for black hole orbits. Phys. Rev. D **77**(10), 103005 (2008)
7. Fujita, R., Hikida, W.: Analytical solutions of bound timelike geodesic orbits in kerr spacetime. Class. Quantum Grav. **26**(13), 135002 (2009)
8. Mino, Y.: Perturbative approach to an orbital evolution around a supermassive black hole. Phys. Rev. D **67**(8), 084027 (2003)
9. Warburton, N., Barack, L., Sago, N.: Isofrequency pairing of geodesic orbits in kerr geometry. Phys. Rev. D **87**(8), 084012 (2013)
10. Vincent, F.H., Paumard, T., Gourgoulhon, E., Perrin, G.: Gyoto: a new general relativistic ray-tracing code. Class. Quantum Grav **28**(22), 225011 (2011)
11. Kuchelmeister, D., Müller, T., Ament, M., Wunner, G., Weiskopf, D.: Gpu-based four-dimensional general-relativistic ray tracing. Comput. Phys. Commun. **183**(10), 2282–2290 (2012)
12. The Compute Unified Device Architecture CUDA. NVIDIA Corporation 2018. http://developer.nvidia.com

13. Gourgoulhon, E.: 3 + 1 Formalism in General Relativity: Bases of Numerical Relativity, vol. 846. Springer, Berlin (2012)
14. York Jr., J.W.: Kinematics and dynamics of general relativity. In: Smarr, L.L. (ed.) Sources of Gravitational Radiation, pp. 83–126. Cambridge University Press, Cambridge (1979)
15. van Holten, J.W.: Gravitational waves and black holes. Fortschr. Phys. **45**, 439–516 (1997)
16. Poisson, E.: A Relativist's Toolkit: The Mathematics of Black-hole Mechanics. Cambridge University Press, Cambridge (2004)
17. Bardeen, J.M., Press, W.H., Teukolsky, S.A.: Rotating black holes: locally nonrotating frames, energy extraction, and scalar synchrotron radiation. Astrophys. J. **178**, 347–370 (1972)
18. Nvidia, C.U.D.A.: Nvidia cuda c programming guide. Nvidia Corp. **120**(18), 8 (2011)
19. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 2nd edn. Wiley, Chichester (2008)
20. CUDA-C Best practice guide. NVIDIA Corporation 2018. http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html