

# PyGRO: A Python integrator for General Relativistic Orbits

Riccardo Della Monica<sup>\*</sup> 

Departamento de Física Fundamental y Matemáticas, Universidad de Salamanca, Patio de Escuelas 1, Salamanca 37008, Spain

Received 27 February 2025 / Accepted 25 April 2025

## ABSTRACT

The advancement in recent years in the field of experimental gravitation has allowed for testing of the equivalence principle in regimes that were previously unexplored and for unprecedented verifications of general relativity while also enabling tests of alternative theories of gravity. We introduce a new computational tool, PyGRO, with the aim of numerically integrating the geodesic equations for the trajectories of massive and massless test particles in any analytic four-dimensional space-time. The result of our work is a fast, modern, highly customizable, and user-friendly open-source Python package that can perform numerical integration of the geodesic equations. Combining symbolic and numerical calculations, PyGRO offers a variety of methods to obtain fully relativistic orbits with minimal intervention by the user, and it is able to work in full generality with any user-given symbolic expression of the space-time metric tensor. We tested PyGRO in an array of scenarios, and we validated the methodology by successfully reproducing classical results from general relativity, which we report in this article.

**Key words.** astrometry – celestial mechanics

## 1. Introduction

The equivalence principle lies at the heart of general relativity and has been pivotal in shaping our understanding of gravity (Dicke 1964). Historically, it provided the conceptual foundation for the formulation of Einstein's theory (Einstein 1915a) and has served as the basis for classical tests of gravity. These classical tests – namely, the deflection of light by the Sun (Dyson et al. 1920), the precession of Mercury's perihelion (Einstein 1915b), and the gravitational redshift (Pound & Rebka 1960) – stand as cornerstones of experimental gravitation, providing early confirmations of general relativity. A direct consequence of the equivalence principle is the geodesic nature of motion, which dictates that freely falling test particles, whether massive or massless, follow trajectories determined solely by the underlying geometry of space-time.

This fundamental aspect of relativistic dynamics has allowed experimental gravitation to advance significantly over recent decades, leading to numerous groundbreaking discoveries and increasingly precise tests of the equivalence principle at different scales (Will 2014). Notable modern developments include missions such as Gravity Probe A (Vessot et al. 1980) and Gravity Probe B (Everitt et al. 2011), which measured gravitational time dilation and the frame-dragging effect with unprecedented precision, along with the detection of strong gravitational lensing (Bernstein et al. 1993) and the confirmation that black holes are actual astrophysical entities (Bolton 1972; Webster & Murdin 1972). Additionally, the recent advent of gravitational wave astronomy (Abbott et al. 2016) marked the birth of a completely new avenue for experimental gravitation.

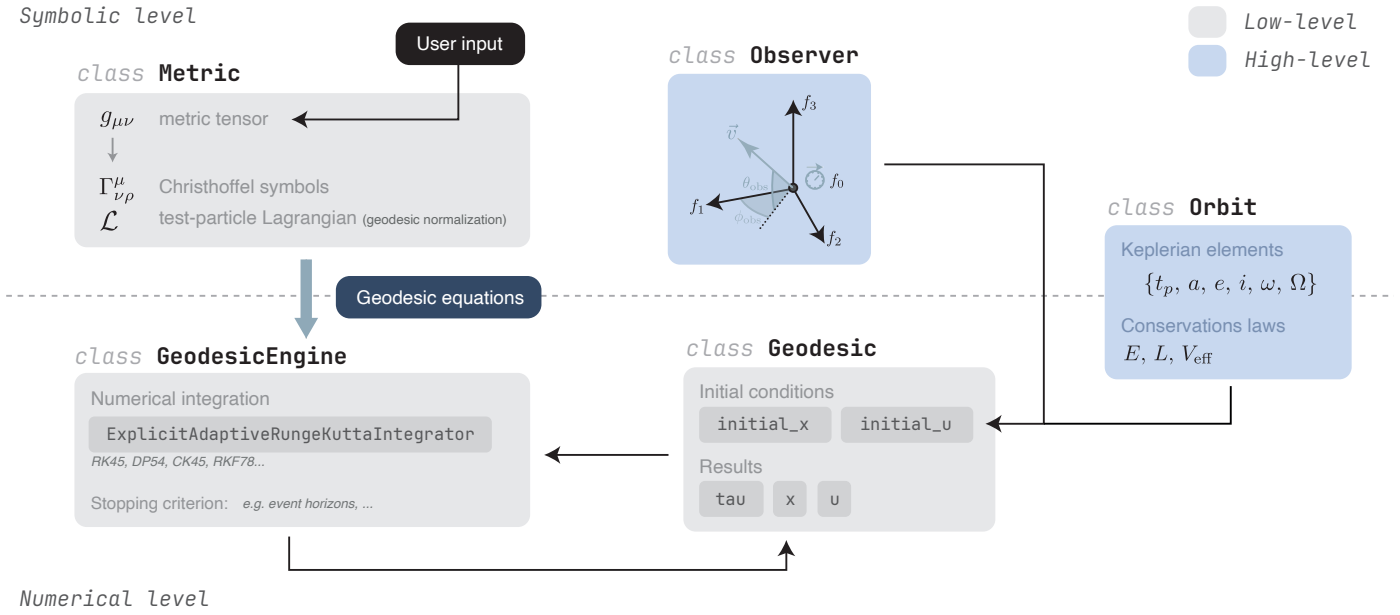
One of the most exciting developments in the past thirty years is the detailed study of the Galactic Center and its supermassive black hole, Sagittarius A\* (Sgr A\*) (De Laurentis et al. 2023). Observations have revealed the existence of a large population

of young stars, called the S-stars, moving on tight, highly eccentric orbits around Sgr A\* (Eckart & Genzel 1996; Ghez et al. 1998). These stars provided the first definitive evidence of the presence of a supermassive black hole at the Galactic Center (Genzel et al. 2010). Over time, precise measurements of their orbits have enabled direct tests of relativistic effects in a strong gravitational field (Hees et al. 2017). The star S2, in particular, has been crucial in this effort, with its periapsis passage in 2018 confirming the predicted gravitational redshift and orbital precession as described by general relativity (Do et al. 2019; Gravity Collaboration 2018, 2020).

The successful imaging of the supermassive black holes M87\* (Event Horizon Telescope Collaboration 2019) and Sgr A\* (Event Horizon Telescope Collaboration 2022) by the Event Horizon Telescope (EHT) represents another milestone in experimental gravitation, as it offered a direct detection of the shadow of a black hole, thus providing a strong-field test of gravitational lensing by a compact object (Psaltis et al. 2020). Moreover, the recent detection of polarized emission on the event-horizon scales of both M87\* and Sgr A\* (Event Horizon Telescope Collaboration 2021a, 2024) allowed the structure of magnetic fields and the plasma properties near supermassive black holes to be probed (Event Horizon Telescope Collaboration 2021b).

Looking ahead, future experiments promise to push the boundaries of experimental gravitation even further. The next-generation EHT (Johnson et al. 2023, ngEHT) will enable the direct observation of black hole photon rings, while instruments such as GRAVITY+ (GRAVITY+ Collaboration 2022) at the Very Large Telescope (VLT), MICADO (Sturm et al. 2024) at the Extremely Large Telescope (ELT), and the Thirty Meter Telescope (Skidmore et al. 2015) in Hawaii will allow even closer stars in the Galactic Center to be resolved, thus enabling the community to dive deeper into the relativistic regime. Moreover, the potential discovery of pulsars orbiting a supermassive black hole by facilities such as the Square Kilometre Array (Keane et al. 2015, SKA), the Five-hundred-meter Aperture Spherical

\* Corresponding author: [rdellamonica@usal.es](mailto:rdellamonica@usal.es)



**Fig. 1.** Schematic illustration of the code structure of PyGRO. The `Metric` object serves as the primary symbolic tool in PyGRO, computing all tensorial quantities required for deriving geodesic equations. The `GeodesicEngine`, on the other hand, is responsible for the numerical integration of these equations, acting as a worker that processes `Geodesic` objects. These objects encapsulate the geodesic type (e.g., null or time-like) and properly normalized initial conditions (initial coordinates and initial tangent four-vector to the geodesic), and they receive the integration results. Higher-level APIs in PyGRO, such as the `Observer` or the `Orbit` class, comprising both symbolic and numerical calculations, provide a more physically intuitive approach to initializing geodesics.

Telescope (Nan et al. 2011, FAST;), and the next-generation Very Large Array (Di Francesco et al. 2019, ngVLA;) would open new avenues for precision tests of the equivalence principle and alternative theories of gravity (Della Monica et al. 2023a).

In this context, we have developed PyGRO, a Python integrator for General Relativistic Orbits. Building on numerous studies in the past years (De Martino et al. 2021; Della Monica & de Martino 2022; Della Monica et al. 2022; Della Monica & de Martino 2023b; Della Monica et al. 2023b,a; Della Monica & de Martino 2023a; Cadoni et al. 2023; Fernández Fernández et al. 2023; de Mora Losada et al. 2025), in which we have exploited observations and simulations of the S-stars at the Galactic Center to constrain several alternative theories of gravity, black hole mimicker models, and extended distribution of dark matter around Sgr A\*, the code provides a computational tool designed to numerically integrate the geodesic equations for both massive and massless test particles in any analytic four-dimensional space-time. PyGRO is a fast, modern, highly customizable, and user-friendly open-source framework that combines symbolic and numerical methods to obtain fully relativistic orbits with minimal user intervention. In this article, we detail the structure and functionality of PyGRO (Section 2), we demonstrate its capabilities with various examples and benchmarks (Section 3), and we highlight its potential as a powerful resource for studying relativistic dynamics in a wide range of astrophysically relevant scenarios.

## 2. The code structure

Distributed as a Python package, PyGRO is an open-source project<sup>1</sup>, and its documentation is publicly available on Github<sup>2</sup>.

<sup>1</sup> <https://pypi.org/project/pygro/>

<sup>2</sup> <https://rdellamonica.github.io/pygro/index.html>

The basic structure of PyGRO is shown in Fig. 1. PyGRO operates on both symbolic and numerical levels. The `Metric` engine (Section 2.1) represents the main symbolic tool within the code, dealing with the analytical computation of all the tensorial quantities required to compute the geodesic equations. The `GeodesicEngine` (Section 2.2), on the other hand, is the main numerical class in PyGRO, performing the numerical integration of the geodesic equations. It acts as a worker to which integration jobs can be submitted in the form of `Geodesic` objects (Section 2.3), which store the geodesic type (e.g., null or time-like), the appropriately normalized initial conditions (initial coordinates and initial tangent four-vector to the geodesic) and receives the results of the integration. Instead of assigning the initial conditions directly from the initial coordinates and initial tangent four-vector one can use higher-level APIs in PyGRO (Section 2.4), namely the `Observer` or the `Orbit` class in spherically symmetric space-times, which allow for a more physically intuitive assignment of initial data to geodesics.

### 2.1. The Metric engine

The `Metric` object serves as a fundamental tool for defining four-dimensional space-time metrics in PyGRO. Unlike other computational tools, like GYOTO (Vincent et al. 2011), that are tailored specifically for particular analytical solutions such as the Schwarzschild or Kerr metrics (or numerical metrics in the 3+1 formalism), the `Metric` class in PyGRO is designed with generality in mind. It allows users to specify an arbitrary coordinate system,  $\{x^\mu\}_{\mu=0\dots3}$ , and a symbolic expression of the metric tensor,  $g_{\mu\nu}$  in the chosen chart, making it possible to study a wide range of space-time geometries. Conventionally, PyGRO adopts the space-time signature  $(-, +, +, +)$ .

The `Metric` class is built upon `sympy` (Meurer et al. 2017), and it allows one, with minimal manual intervention, to derive all

the essential quantities required (from the user-input symbolic expression of the metric tensor) to later integrate geodesics in that space-time. In fact, upon initialization, the `Metric` object computes the inverse metric  $g^{\mu\nu}$ , the Christoffel symbols of the metric, which in the Levi-Civita connection<sup>3</sup> take the form

$$\Gamma_{\nu\rho}^{\mu} = \frac{1}{2}g^{\mu\sigma}(\partial_{\nu}g_{\sigma\rho} + \partial_{\rho}g_{\nu\sigma} - \partial_{\sigma}g_{\nu\rho}), \quad (1)$$

from which the geodesic equations that govern the motion of free-falling particles in the given space-time are derived<sup>4</sup>:

$$\ddot{x}^{\mu} + \Gamma_{\nu\rho}^{\mu}\dot{x}^{\nu}\dot{x}^{\rho} = 0. \quad (2)$$

Here, the dots represent derivatives with respect to an affine parameter  $\tau$  on the geodesic (coinciding with the proper time for the time-like case) and the functions  $\{x^{\mu}(\tau)\}_{\mu=0\dots3}$  are its unknown space-time coordinates as a function of  $\tau$ . Moreover, PyGRO symbolically derives helper functions that allow one to normalize the initial tangent four-vector to the geodesic, thus defining its causal character:

$$g_{\mu\nu}\dot{x}^{\mu}\dot{x}^{\nu} = \begin{cases} -1 & \text{time-like} \\ 0 & \text{null} \end{cases}. \quad (3)$$

This allows one to study the motion of both massive particles and light rays in PyGRO. More specifically, these helper functions allow one of the components of the initial tangent four-vector to be computed from given values of the other three components, thus solving Eq. (3) for the unknown one once the causal character of the geodesic has been defined. The `Metric` object also allows one to compute the test-particle Lagrangian:

$$\mathcal{L} = \frac{1}{2}g_{\mu\nu}\dot{x}^{\mu}\dot{x}^{\nu}, \quad (4)$$

which is particularly useful to derive conserved quantities that are needed for higher level APIs, such the `Orbit` class.

A key advantage of the symbolic approach presented in this section is its versatility: users can define arbitrary metrics without requiring explicit derivations of Christoffel symbols or other geometric quantities. However, this flexibility comes at a computational cost compared to hardcoded implementations in lower-level languages such as C, where the metric and its derived quantities are pre-computed and optimized for specific cases. In PyGRO, especially for algebraically complicated metric tensors, symbolic differentiation and matrix operations introduce an overhead. While PyGRO prioritizes user-friendliness and generality over raw computational speed, its performance remains suitable for a broad range of applications. These aspects are all quantified in Section 3.7.

## 2.2. The GeodesicEngine

The `GeodesicEngine` class in PyGRO is the primary numerical tool designed to integrate geodesic equations within a specified space-time metric. By linking to an initialized `Metric` object, the `GeodesicEngine` utilizes the pre-computed symbolic quantities to perform numerical integrations of geodesics.

<sup>3</sup> While the Levi-Civita connection is built into PyGRO, it is fairly easy to implement any metric connection, from its symbolic expression.

<sup>4</sup> In PyGRO, it is fairly easy to introduce modifications to the geodesic equations, such as when considering interactions of test particles with a vector field that appear as a non-zero term at the right-hand-side of Eq. (2), as done in Della Monica et al. (2022).

Users can specify the symbolic backend used for the integration. The default, *autowrap*, converts the symbolic expressions of the geodesic equations into pre-compiled C functions, which gives a  $\sim 2\times$  boost in performance during integrations. Alternatively, the *lambdify* backend converts expressions into Python callables, which may be preferable in environments lacking a C compiler or when the metric is defined upon auxiliary Python functions (lacking a closed-form analytical expression). Additionally, users can select from various explicit embedded adaptive-step-size ordinary differential equation integrators of the Runge-Kutta family, such as Runge-Kutta-Fehlberg4(5), Dormand-Prince5(4), Cash-Karp4(5), and Runge-Kutta-Fehlberg7(8) (Press et al. 1992), each offering different orders of accuracy and truncation error estimation methods. Users can configure parameters such as maximum and minimum step sizes and the accuracy (AG) and precision (PG) goal by which the absolute and relative tolerances

$$\text{tol}_a = 10^{-\text{AG}} \quad \text{tol}_r = 10^{-\text{PG}} \quad (5)$$

are defined. At each step, the integrator adapts the step size in order to maintain the truncation error  $\varepsilon$ , estimated from the embedded Runge-Kutta formulas, below a tolerance threshold defined as

$$\text{tol} = \text{tol}_a + \text{tol}_r\|y\|, \quad (6)$$

where  $\|y\|$  is the norm of the current state vector. In the future, we also plan to implement implicit schemes of integration and symplectic integrators.

An important feature of the `GeodesicEngine` is the possibility to handle stopping criteria during integration. Users can define conditions under which the integration should halt, such as when a particle reaches a specific region of space-time. This is particularly useful, for instance, in the Schwarzschild metric, in which one might set a stopping criterion at the event horizon ( $r = 2M$ ).

## 2.3. The low level geodesic API: Geodesic

The `Geodesic` class in PyGRO represents the core data structure for describing geodesics in a given space-time. Each `Geodesic` instance stores essential information, such as the type of geodesic (time-like or null), the initial position and tangent four-vector to the geodesic, normalized to ensure consistency with the geodesic causal character, and stores the results of the integration from the `GeodesicEngine`.

The `Geodesic` class represents a low-level API for the geodesic integration. Initial data, in fact, must be assigned directly in terms of the initial coordinates in the given chart and the components of the geodesic's tangent four-vector in the coordinate basis (which, in general, lack a direct physical interpretation). For this specific purpose, the `Geodesic` class makes use of the helper symbolic functions computed by the `Metric` object to compute one of the components of the initial tangent four-vector starting from the values assigned to the other three, solving Eq. (3) for the unknown component.

## 2.4. High level APIs: Observer and Orbit

Assigning initial conditions that have a meaningful physical meaning is important to compute geodesic trajectories that are relevant for astrophysical applications of PyGRO. Directly assigning the initial values of the space-time coordinates and

initial tangent four-vector to Geodesic objects can be avoided by making use of higher-level APIs in PyGRO that allow one to specify positions and velocities tied to the reference frame of a given physical observer in the space-time (Section 2.4.1) or that have a direct connection with the classical Keplerian parametrization of orbits in Newtonian mechanics (Section 2.4.2).

### 2.4.1. Defining physical observers in PyGRO

The `Observer` class in PyGRO provides a high-level interface for defining observers in any given space-time and assigning initial data to geodesics in a physically meaningful way. This abstraction is particularly useful in astrophysical settings, where one wants to compute trajectories of freely falling objects and photons as observed by a specific observer (e.g., in a ray-tracing scenario to image a black hole shadow, to compute the astrometric positions of stars around a massive black hole as seen by a given observer, or to perform pulsar timing).

The approach to general relativity that relies on observer frames is known as the tetrad formalism (or Cartan formalism) (Straumann 2013). In this framework, one constructs frames – also referred to as vierbeins – at each point on the curved manifold, i.e., constructing a frame bundle over the manifold. These frames consist of four orthonormal vectors denoted as  $\{f_a^\alpha\}$ , where Greek indices refer to the components of the frame vectors in the chart coordinates on the manifold, while Roman indices label each frame vector. The set  $\{f_a^\alpha\}$  contains one time-like vector ( $f_0^\alpha$ ) and three space-like vectors ( $f_i^\alpha$ ,  $i = 1, 2, 3$ ), representing, respectively, the observer’s time axis (which coincides with the physical observer’s four-velocity) and a set of orthonormal spatial axes forming a spatial tetrad at that point. The observer uses these axes to define tangent vectors to the manifold and project them conveniently into its own frame. In this sense, the tetrad provides a local basis for the tangent space at the observer’s location, enabling the projection of physical quantities from the global manifold coordinates into the local reference frame of the observer.

Similarly, one can define a co-frame  $\{f_b^\beta\}$ , which forms the basis for tangent 1-forms at the observer’s location. This can be generalized, allowing one to use the defined frame and co-frame basis to represent any tensorial quantity in the tangent or co-tangent bundle (or any product of these). For instance, the metric tensor can be expressed in the co-frame as

$$g = -f^0 \otimes f^0 + \sum_i f^i \otimes f^i, \quad (7)$$

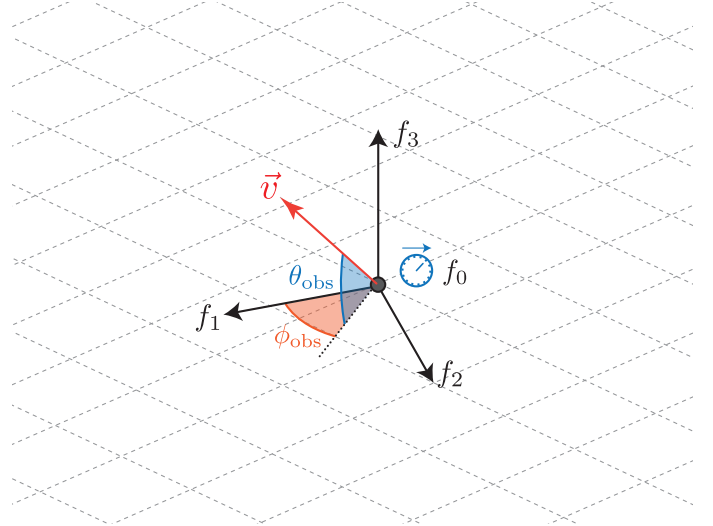
which allows one to obtain relations between the local frame metric and the space-time metric:

$$f_a^\alpha f_b^\beta \eta_{ab} = g_{\alpha\beta}, \quad (8)$$

$$f_a^\alpha f_b^\beta g_{\alpha\beta} = \eta_{ab}. \quad (9)$$

In other words, in the local observer’s frame, the metric reduces to the ordinary Minkowski metric, which reflects the local flatness property of general relativity.

In PyGRO the `Observer` class is built starting from a symbolic expression of the frame (co-frame) vectors in the coordinate basis of the tangent (co-tangent) space to the manifold. Upon definition, the code autonomously computes symbolic expressions of the transformation matrices from the observer’s reference frame to the space-time coordinate basis and vice-versa.



**Fig. 2.** Illustration of the reference frame carried by a physical observer in PyGRO. Angles of observations in the observer’s reference frame are defined as standard longitude,  $\phi_{\text{obs}}$ , and latitude,  $\theta_{\text{obs}}$ , with the origin on the  $f_1$  direction and reference plane identified by  $f_1$  and  $f_2$ . Helper functions built into the `Observer` class allow one to fire geodesics with any given pair of angles, defined in this way, from any of the frame vectors.

This allows to place an observer at a specific event in space-time, carrying a local system of orthonormal axes on which one can project physical quantities, e.g. four-velocities of test particles. More specifically, one can define a spatial vector  $\boldsymbol{v}$  in this system of axes, whose direction is identified by polar coordinates  $(\theta_{\text{obs}}, \phi_{\text{obs}})$  and with a given magnitude  $|\boldsymbol{v}|$  (see Fig. 2). By projecting this vector on the observer’s frame, one can derive in the coordinate basis the components of the corresponding four-vector  $v^\mu$  whose spatial part coincides with  $\boldsymbol{v}$ , once a given causal character for  $v^\mu$  has been chosen (the temporal component,  $v^0$ , is fixed by the normalization condition in Eq. (3) with  $\dot{x}^\mu$  replaced by  $v^\mu$ ). In case the desired four-vector is null, the magnitude of  $\boldsymbol{v}$  is automatically normalized to unity. The four-vector obtained with this procedure can then be used as initial tangent four-vector for a geodesic starting at the observer’s location, which will correspond to a test particle, which according to the observer, has a physical velocity  $\boldsymbol{v}$  in its reference frame.

### 2.4.2. Quasi-elliptic orbits in spherically symmetric space-times

When the considered space-time is spherically symmetric (Chandrasekhar 1985), assigning initial conditions to freely falling massive test particles can be greatly simplified, enabling a parametrization of the orbit which is familiar to classical Keplerian parametrization of elliptical orbits in Newtonian celestial mechanics (Taff 1985).

In this section, we assume that a chart of coordinates  $(t, r, \theta, \phi)$  has been chosen to express the metric tensor, with  $t$  a coordinate-time,  $r$  a radial coordinate (for example, but not necessarily, the aerial radius of the Schwarzschild coordinate system), and  $\theta$  and  $\phi$  are the usual angular coordinates on the 2-spheres. In this setting, the geodesic equations in Eq. (2) are a system of four second-order differential equations for the unknown functions  $(t(\tau), r(\tau), \theta(\tau), \phi(\tau))$  that describe the space-time trajectory of a test particle. A solution is uniquely determined once second-order initial conditions are assigned.

Namely, one has to specify at a given proper time  $\tau_0$  the initial space-time position of the test particle ( $t(\tau_0)$ ,  $r(\tau_0)$ ,  $\theta(\tau_0)$ ,  $\phi(\tau_0)$ ) and space-time four-velocity ( $\dot{t}(\tau_0)$ ,  $\dot{r}(\tau_0)$ ,  $\dot{\theta}(\tau_0)$ ,  $\dot{\phi}(\tau_0)$ ). Importantly, due to the spherical symmetry, one can always perform appropriate rotations to make the trajectory lie on the equatorial plane for the whole duration of the motion. This reduces the number of free parameters, since  $\theta = \pi/2$  and  $\dot{\theta} = 0$  are assumed implicitly. Second, since the metric components (and thus the geodesic equations) do not explicitly depend on the time coordinate  $t$ , the value  $t(\tau_0)$  can be chosen arbitrarily. Finally, as mentioned above, the normalization condition of the four-velocity for massive test particles, in Eq. (3), represents a constraint on the initial data, thus allowing one of the components of the four-velocity to be derived as a function of the others. A complete set of initial data that uniquely identifies a trajectory on the equatorial plane of a spherically symmetric space-time can thus be  $(r(\tau_0), \phi(\tau_0), \dot{t}(\tau_0), \dot{\phi}(\tau_0))$ , with  $\dot{r}(\tau_0)$  fixed by the choice of  $\dot{t}(\tau_0)$  and  $\dot{\phi}(\tau_0)$  through the normalization condition. Instead of fixing the initial conditions directly in terms of these components of the four-velocity, we introduce a different parameterization in terms of constants of motion. In particular, in spherically symmetric space-times from the relativistic test-particle Lagrangian in Eq. (4) (which is by itself a constant of motion since, modulo the factor  $1/2$ , it coincides with the norm of the particle's four-velocity), one can always define two additional constants of motion,

$$E = -\frac{\partial \mathcal{L}}{\partial \dot{t}}, \quad (10)$$

$$L = \frac{\partial \mathcal{L}}{\partial \dot{\phi}}, \quad (11)$$

that correspond to the specific energy and angular momentum of the test particle. These constants allow one to reduce the problem of describing the radial motion on the equatorial plane to the equation

$$\dot{r}^2 = E - V_{\text{eff}}(r; L), \quad (12)$$

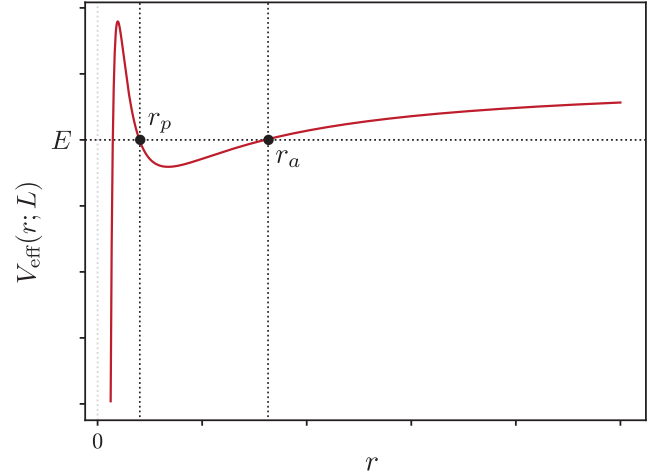
where we have introduced the effective potential  $V_{\text{eff}}$ , parametrized by the orbital angular momentum  $L$ , which only depends functionally on the radial coordinate  $r$ . A choice of  $E$  and  $L$  uniquely fixes the evolution of the radial coordinate and, more specifically, defines the radial turning points, that is, points where  $\dot{r} = 0$ . In the case of quasi-elliptic orbits, we call the two radial turning points the pericenter (identified by the radial coordinate  $r_p$ ) and the apocenter (identified by  $r_a$ ). These are shown in Fig. 3 for an example orbit in the Schwarzschild space-time.

More conveniently, one can introduce two orbital parameters, namely, the eccentricity  $e$  and the semi-major axis  $a$ , defined implicitly by

$$r_p = a(1 - e) \quad (13)$$

$$r_a = a(1 + e), \quad (14)$$

whose values are uniquely identified by a choice of  $E$  and  $L$ . We can invert this mapping so that unique values for  $E$  and  $L$  are assigned upon fixing a value for  $a$  and  $e$ . This inverse map can be derived analytically for the Schwarzschild space-time. However, to preserve the generality of the approach to any spherically symmetric space-time, in PyGRO the mapping from pairs  $(a, e)$  to pairs  $(E, L)$  is computed numerically within the `Orbit` class. Once the values of  $E$  and  $L$  are known, one can invert Eqs. (10) and (11) to obtain  $\dot{t}$  and  $\dot{\phi}$ .



**Fig. 3.** Effective potential for given values of  $E$  and  $L$  in the Schwarzschild space-time. A choice of  $E$  and  $L$  uniquely identifies radial turning points, that is, values of the radial coordinates at which the radial velocity,  $\dot{r}$ , is zero. These points are identified by the pericenter,  $r_p$ , and the apocenter,  $r_a$ .

In the flat space-time limit, the orbital parameters that we have introduced perfectly match the concept of semi-major axis and eccentricity classically defined in Newtonian celestial mechanics for Keplerian elliptical orbits. In the general relativistic case, these parameters depend on the particular choice of coordinates used (for example, in the harmonic gauge of post-Newtonian mechanics, these parameters have a slightly different definition and physical interpretation, e.g., Damour & Deruelle, 1985), and their physical meaning is not uniquely defined in the strong-field regime. As such, they do not have a direct physical meaning but only serve as a familiar and useful parametrization of initial data.

At this point, we could choose whether to start the integration at the apocenter or pericenter and to fix the rest of the initial conditions from there. For example, if we call  $t_p$  the time of pericenter passage, we have the following set of initial data:

$$t(\tau_0) = t_p, \quad (15)$$

$$r(\tau_0) = a(1 - e), \quad (16)$$

$$\phi(\tau_0) = 0, \quad (17)$$

$$\dot{r}(\tau_0) = 0, \quad (18)$$

and  $\dot{t}(\tau_0)$  and  $\dot{\phi}(\tau_0)$  are derived using the procedure that we have just explained. Alternatively, one can fix initial conditions at the apocenter, and in this case, one would have

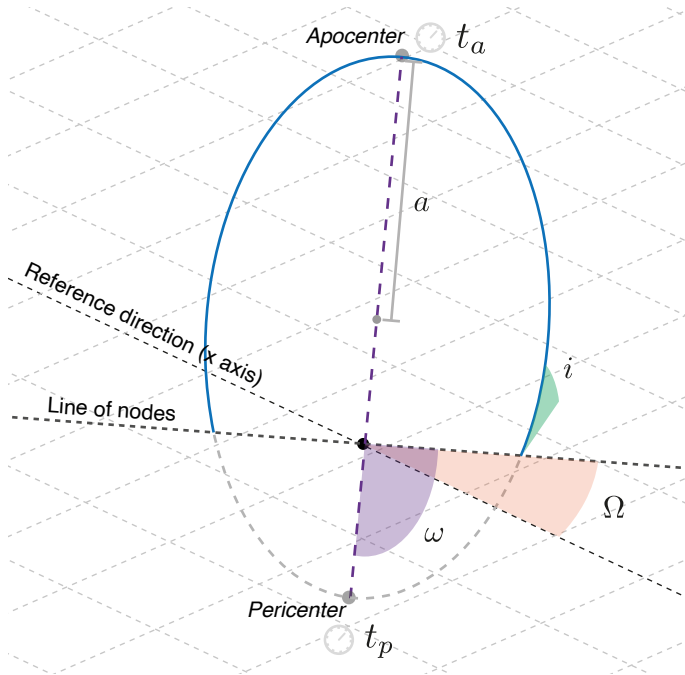
$$t(\tau_0) = t_a, \quad (19)$$

$$r(\tau_0) = a(1 + e), \quad (20)$$

$$\phi(\tau_0) = \pi, \quad (21)$$

$$\dot{r}(\tau_0) = 0, \quad (22)$$

where we call  $t_a$  the time of passage at the apocenter. We note that the choice of  $\phi(\tau_0) = 0$  at the pericenter or  $\phi(\tau_0) = \pi$  at the apocenter comes from the analogy between the azimuthal coordinate  $\phi$  and the true anomaly of the orbit. This of course is only valid at the initial time since the relativistic orbital precession (which is described in more detail in Section 3.1) causes the two quantities to eventually deviate from one another with a constant rate.



**Fig. 4.** Illustration of the angular orbital elements for a Keplerian orbit. The orbital inclination,  $i$ , represents the inclination of the orbital plane with respect to a reference plane (e.g., the plane of sky for a distant observer), and the longitude of the ascending node,  $\Omega$ , corresponds to the angle between a reference direction (in our case the  $\phi = 0$  line on the equatorial plane) and the line of nodes, namely, the line in which the orbital plane intersects the reference plane. The argument of the pericenter,  $\omega$ , is the angle along the orbital plane between the line of nodes and the orbit's pericenter.

The procedure described above completely fixes the initial conditions on the equatorial plane and allows one to integrate the geodesic equations numerically. To bring the orbit outside the equatorial plane, one can introduce the standard angular Keplerian parameters (see Fig. 4). In particular: the inclination ( $i$ ) measures how much the orbital plane is inclined with respect to the equatorial plane of the spherically symmetric space-time considered. The two planes intersect on a line known as the line of nodes. The orbiting object crosses this line twice over one period. The point where it cuts the line from below the plane is known as the ascending node, while the point where it cuts it from above is known as the descending node; the longitude of the ascending node ( $\Omega$ ) is the angle between the  $x$ -axis (i.e., the one identified by  $\theta = \pi/2$  and  $\phi = 0$ ) and the ascending node; the argument of the pericenter ( $\omega$ ) is the angle between the line of nodes and the semi-major axis of the orbit, specifically its pericenter, over the orbital plane. These angles correspond to three subsequent rotations, that are applied to both the initial position and velocity before integrating the geodesic, that bring the orbit into the desired reference frame: (i) a rotation around the  $z$ -axis by an angle  $-\omega$  (the minus sign is due to the fact that by definition,  $\omega$  points towards  $x$  and not from it); (ii) a rotation by an angle  $-i$  around the new  $x$ -axis, corresponding to the line of nodes; (iii) a rotation around the  $z$ -axis by an angle  $-\Omega$ . It is important to note that in spherically symmetric spacetimes, the existence of an  $SO(3)$  isometry subgroup ensures that rotations can always be defined in a covariant manner. This allows one to orient a given geodesic into a preferred plane, often the equatorial plane, without loss of generality, as done for the covariant

definition of the equatorial plane motion. However, the angular orbital elements such as  $i$ ,  $\omega$  and  $\Omega$ , introduced above, are defined with respect to a chosen reference plane and thus depend on the coordinate system. While useful for comparisons with Newtonian orbital elements, they are not intrinsic to the geodesic itself.

The methodology that we have explained in this section, requires a combination of symbolic and numerical calculations that are implemented in the `Orbit` class. This acts as a wrapper to the `Geodesic` class, for the time-like case, and it allows one to integrate an orbit once values for the Keplerian orbital parameter ( $t_p$ ,  $a$ ,  $e$ ,  $i$ ,  $\Omega$ ,  $\omega$ )<sup>5</sup> are assigned.

### 3. Examples and benchmarks

In this section, we analyze a series of example applications of PyGRO that serve as a benchmark of the methodology implemented in the code. Most of these examples test a series of theoretical predictions for the Schwarzschild space-time, here written assuming  $G = c = 1$ ,

$$ds^2 = -\left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 d\Omega^2, \quad (23)$$

where  $M$  is the mass of the central object,  $r$  is the aerial radius, and  $d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2$  is the solid angle element. The correct recovery of theoretical predictions for the Schwarzschild metric effectively validates PyGRO and works as a benchmark for the methodology that we have discussed. The code to reproduce all the examples shown in this section can be found in the [example notebooks](#) in PyGRO's official documentation.

#### 3.1. Orbital precession of massive particles

Here, we illustrate how in PyGRO one can accurately reproduce the expected orbital motion for bound time-like orbits and also numerically determine the rate of orbital precession in the strong-field regime.

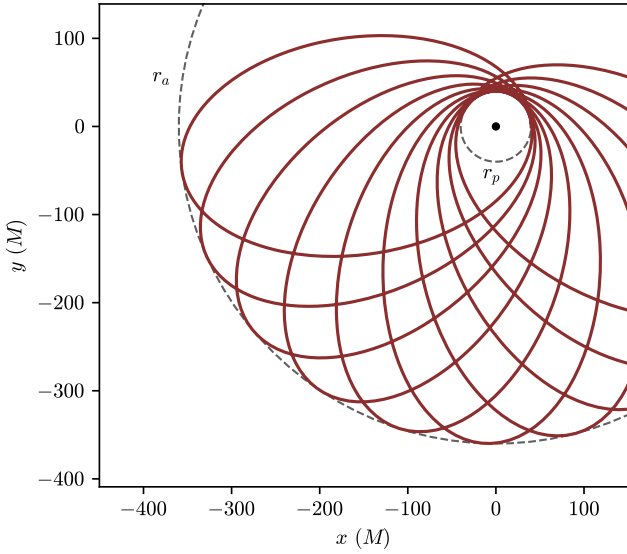
Let us consider eccentric orbits on the equatorial plane of the Schwarzschild black hole in Eq. (23). An orbit with semi-major axis  $a$  and eccentricity  $e$  is known to have a rate of orbital precession which, at first post-Newtonian order, can be quantified by (Poisson & Will 2014)

$$\Delta\phi = \frac{6\pi GM}{ac^2(1-e^2)} \quad (24)$$

per orbital period. This means that each complete oscillation on the radial direction between  $r_a$  and  $r_p$  corresponds to a complete turn in the azimuthal direction plus a small angle  $\Delta\phi$ , which thus corresponds to an advance of the periastron of the orbit.

In PyGRO, one can compute the rate of orbital precession by directly estimating the angle spanned by an orbit over one full cycle in the radial direction. To do so, first we made use of the `Orbit` class to define orbits with a specific semi-major axis and eccentricity. An example orbit retrieved in the Schwarzschild metric with this methodology is shown in Fig. 5. Since we are fixing initial conditions like in Eqs. (15)–(18), a full azimuthal cycle corresponds to an angle  $\phi = 2\pi$ . Due to the orbital precession, however, this full cycle does not correspond to a

<sup>5</sup> Alternatively, ( $t_a$ ,  $a$ ,  $e$ ,  $i$ ,  $\Omega$ ,  $\omega$ ), with  $t_a$  as the time of apocenter passage. In this case, the initial conditions are assigned at  $t(\tau_0) = t_a$ ,  $r(\tau_0) = a(1+e)$ , and  $\phi(\tau_0) = \pi$ .



**Fig. 5.** Example orbit with  $e = 0.8$  and  $a = 200M$  integrated with PyGRO. The integration has been carried out for ten full revolutions around the central object. The trajectory is not a closed ellipse. Orbital precession makes the orbit’s pericenter advance, while the radial coordinate periodically oscillates between the pericenter and apocenter, Eq. (14), here indicated by the dashed circles. This example orbit is labeled as (1) in Table 1.

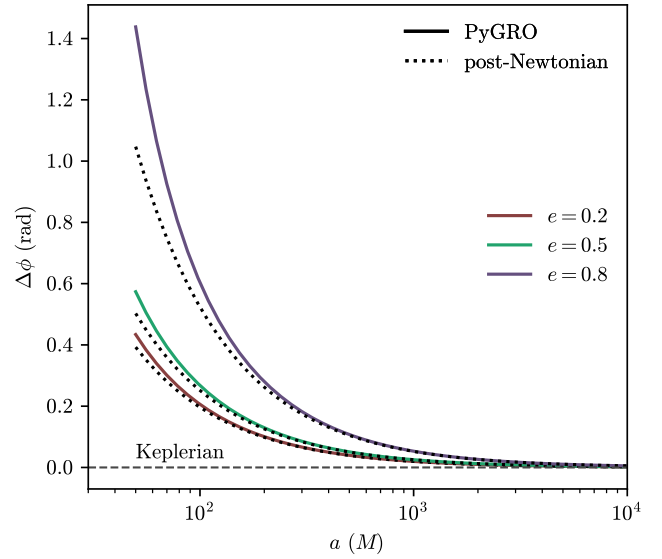
full period in the radial direction and is thus not a return to the pericenter radius  $r_p$ . This occurs at an angle  $\phi = 2\pi + \Delta\phi$ , with  $\Delta\phi$  as the rate of orbital precession that we want to compute. On a practical level, this can be done by considering an interpolating function  $\dot{r} = \dot{r}(\phi)$  built from the integrated geodesic and the subsequent application of a root finding algorithm, to look for the radial turning points (roots of  $\dot{r}$ ) around  $2\pi$ . The difference of the angle obtained and  $2\pi$  is the desired value of  $\Delta\phi$ .

In Fig. 6, we show the values of  $\Delta\phi$  obtained with PyGRO for a large range of semi-major axes and for three eccentricities  $e = 0.2, 0.5$  and  $0.8$  in the case of a Schwarzschild black hole. We also compare our results with the post-Newtonian formula in Eq. (24). As it appears, the post-Newtonian values is correctly recovered for orbits in a weak gravitational field (e.g.,  $a \gtrsim 100M$  in the low-eccentricity cases or  $a \gtrsim 1000M$  for higher eccentricities), validating the accuracy of the relativistic orbital integration in PyGRO. For orbits in the strong field regime, on the other hand, the formula in Eq. (24) fails to describe accurately the rate of periastron advance, underestimating the actual value of  $\Delta\phi$  that the code is able to compute. In this sense, not only does PyGRO offer the advantage of being able to compute the relativistic rate of orbital precession directly on the fully relativistic integrated geodesic, and thus one does not have to resort to approximate formulas, but it also offers the complete generality of the approach, which can be applied to any spherically symmetric space-time by simply defining a different Metric object.

### 3.2. Circular orbits

In this section, we study a different example of motion for massive particles on time-like geodesics in the Schwarzschild space-time: circular orbits.

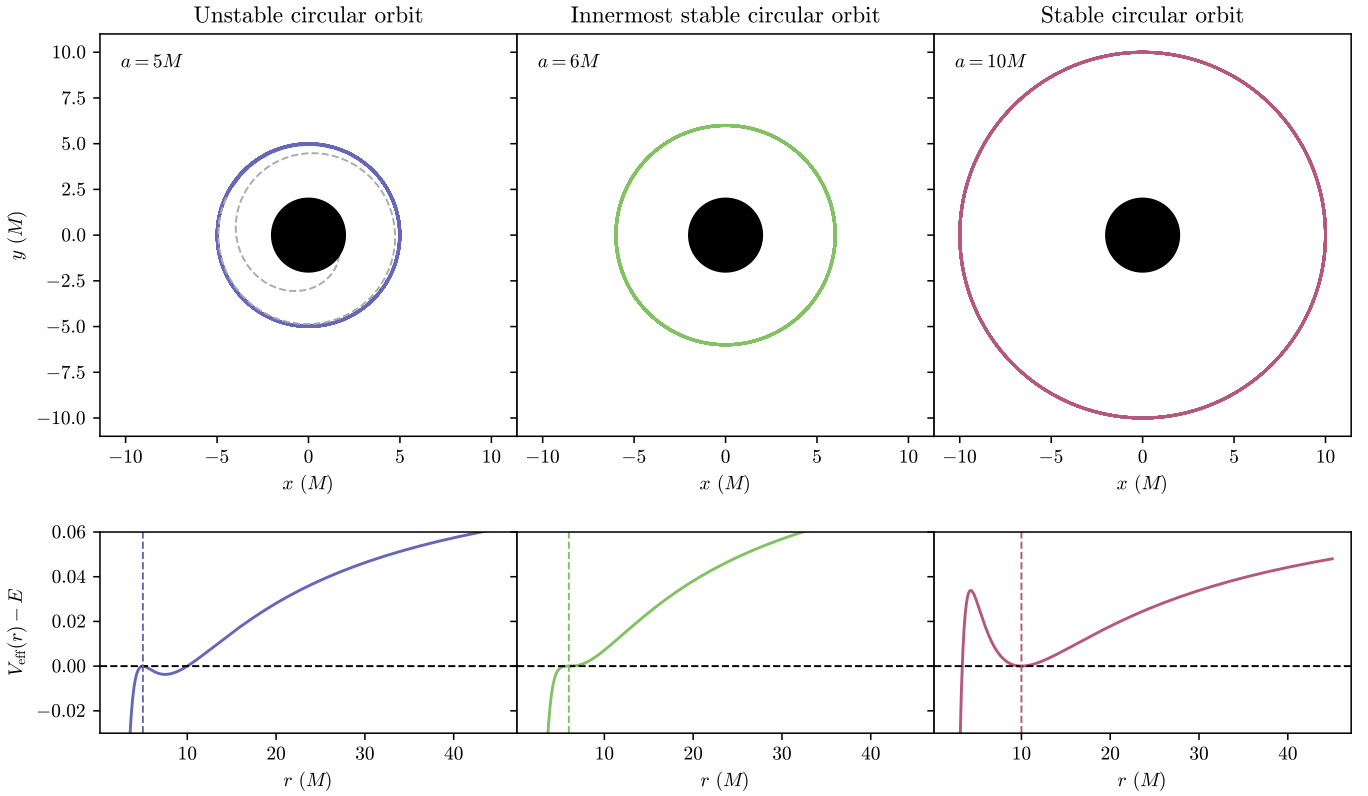
Orbits with zero eccentricity are characterized by  $r_p = r_a = a$ . For this reason, the initial condition search algorithm has to be slightly modified compared to the non-zero eccentricity case. We must now look for those values of specific energy  $E$  and



**Fig. 6.** Rate of orbital precession for orbits with  $a \in [50, 10\,000]$  and for three values of the eccentricity,  $e = 0.2, 0.5$ , and  $0.8$ . The solid lines correspond to the rate of precession computed with PyGRO, while the dotted lines represent the post-Newtonian prediction for the same orbits, computed with the formula in Eq. (24). PyGRO correctly recovers the values of the predicted rate of precession for systems in the weak field regime. Also, for systems that scan increasingly strong gravitational regimes (e.g.,  $a \lesssim 100M$  in the low-eccentricity cases or even  $a \lesssim 1000M$  for higher eccentricities), the fully relativistic computations in PyGRO allow one to go beyond the first-order post-Newtonian prediction and correctly account for an additional contribution to the periastron advance.

angular momentum  $L$ , Eqs. (10)–(11), that realize the conditions  $\dot{r}^2(a) = 0$ , and  $V'_{\text{eff}}(a) = 0$ , with a prime denoting a derivative with respect to the radial coordinate. In other terms, we are looking for those value of energy and angular momentum that make so that the given value of the semi-major axis  $a$  is a radial turning point, i.e. a root for Eq. (12), and at the same time a stationary point for the effective potential. This procedure is built-into the Orbit class in PyGRO and no additional operation on the user’s side is required to deal with circular orbits.

To test the validity of this approach, we can study peculiar circular orbits that characterize the dynamical motion of massive test particles around a Schwarzschild black hole which present distinctive features not present in a Newtonian scenario. While in classical celestial mechanics it is possible to realize a circular orbital configuration for any value of  $a$ , around a point-like massive particle, in general relativity, and in particular in the Schwarzschild space-time, this is only possible for values of  $a > 3M$  (Chandrasekhar 1985). No stable orbital configurations exist very close to the event horizon. Even in the regime where circular orbits do exist, they are not necessarily stable circular orbits. In fact, another known results from general relativity is that stable circular orbits only exist up to a minimum radius of  $a = 6M$ , denoted as the innermost stable circular orbit (ISCO). Orbits with  $a > 6M$  can be circular and are stable, meaning that they realize a minimum for the effective potential. Orbits with  $3M < a < 6M$  can still be circular but they are unstable, sitting on a maximum of the effective potential. Finally, the specific realization with  $a = 6M$  corresponds to a saddle point of the effective potential, thus representing the separation between the two classes. In Fig. 7, we display examples of circular orbital configurations obtained with PyGRO for all the cases



**Fig. 7.** Circular orbits around a Schwarzschild black hole integrated with PyGRO. The top panels display the resulting orbits on the equatorial plane, with the black circle representing the central black hole. The bottom panels show the effective potential in Eq. (12) computed with PyGRO for the distinct orbital configurations. We analyze: (left panels) an unstable circular orbit with  $a = 5M$ , thus lying on a maximum of the effective potential. The instability of the orbit manifests with the orbit falling into the black hole (gray dashed line) due to the accumulation of numerical errors when integrating for a long time interval. This example orbit is labeled as (2) in Table 1; (central panels) the case with  $a = 6M$  corresponding to the ISCO and thus a saddle point in the effective potential; (right panels) a normal stable circular orbit which can be realized for all  $a > 6M$ , corresponding to local minimums of the effective potential. This example orbit is labeled as (3) in Table 1.

listed above, along with the corresponding effective potential. These reproduce the results expected from the theory. In particular, along with a stable orbital configuration with  $a = 10M$  and the innermost stable circular orbit with  $a = 6M$ , both of which are perfectly circular based on the integration, we show an unstable circular orbit with  $a = 5M$ , which, after a large number of orbits falls into the central black hole due to the accumulation of numerical errors<sup>6</sup>.

### 3.3. Strong deflection of massless particles

In this section, we analyze the case of massless particles (e.g., photons) moving on null geodesics of the Schwarzschild space-time and being strongly deflected by the massive central object.

Our goal is to recover in PyGRO one of the most striking predictions of general relativity, namely the existence of a circular (unstable) photon orbit for  $r = 3M$  (the so called photons sphere or last photon orbit) around a Schwarzschild black hole. This photon trajectory separates the classes of null geodesics that escape to infinity after being deflected by the central object and those captured by the black hole. This peculiar feature is what ultimately leads to the emergence of the shadow and photon rings in black hole imaging (Luminet 1979).

<sup>6</sup> Clearly, one can always set the precision tolerances of the integrator to make the orbit stick to the circular configuration for the entire duration of the integration.

In this case, to assign initial conditions to the null geodesics, we made use of the `Observer` class in PyGRO, introduced in Section 2.4.1. In particular, we considered stationary observers in the Schwarzschild space-time, Eq. (23), with frame axes oriented along the coordinate lines of the Schwarzschild coordinate system. It is straightforward to see that such a system of observers can be defined by the following system of frame co-vectors:

$$f^0 = \sqrt{1 - \frac{2M}{r}} dt, \quad f^1 = -\frac{1}{\sqrt{1 - \frac{2M}{r}}} dr, \quad (25)$$

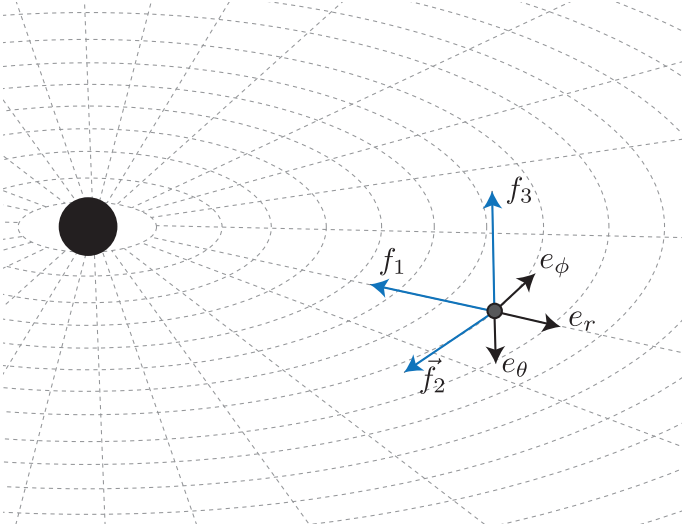
$$f^2 = -r \sin \theta d\phi, \quad f^3 = -r d\theta, \quad (26)$$

which satisfies the relation in Eq. (7). This choice of frame co-vectors results in the frame vectors

$$f_0 = \frac{1}{\sqrt{1 - \frac{2M}{r}}} \partial_t, \quad f_1 = -\sqrt{1 - \frac{2M}{r}} \partial_r, \quad (27)$$

$$f_2 = -\frac{1}{r \sin \theta} \partial_\phi, \quad f_3 = -\frac{1}{r} \partial_\theta. \quad (28)$$

Essentially, the frame that we have defined describes an observer that is stationary (its time axis,  $f_0$  is future-pointing and only directed in the temporal direction) and carries a system of  $f_1, f_2, f_3$  left-handed axes with  $f_1$  on the radial direction pointing inward



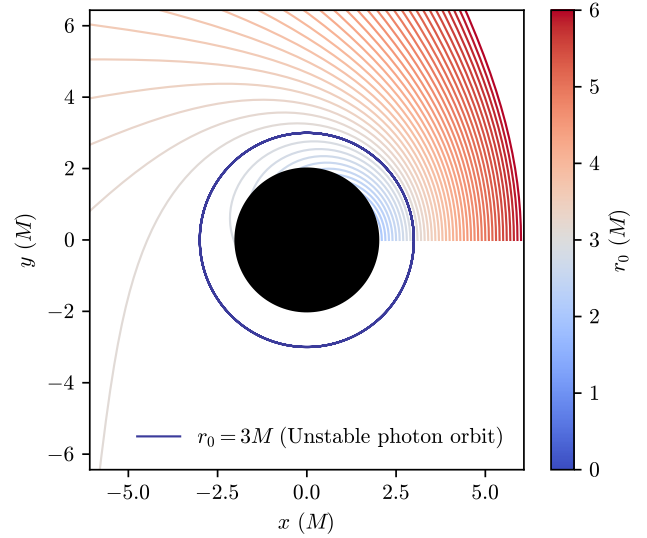
**Fig. 8.** Illustration of reference frame for a stationary observer in the Schwarzschild case, defined as in Eqs. (27)–(28). The observer (black dot) carries a left-handed system of axes,  $f_1 f_2 f_3$ , with  $f_1$  pointing radially towards the central black hole,  $f_2$  pointing tangentially to  $r = \text{const}$ . Two spheres are opposite to the vector  $e_\phi$  of the local coordinate basis, and  $f_3$  orthonormal is to both  $f_1$  and  $f_2$  (opposite to the vector  $e_\theta$  of the local coordinate basis).

towards the black hole,  $f_2$  pointing tangentially in the longitudinal direction (opposite to the natural counter-clockwise direction of the angle  $\phi$ ), and  $f_3$  points “upward” in the latitudinal direction, opposite to the natural direction of the angle  $\theta$ . The geometrical configuration of the frame vectors for our observer in the Schwarzschild space-time is illustrated in Fig. 8.

We now consider several observers defined by Eqs. (27)–(28) positioned at coordinates  $r = r_0$ ,  $\theta = \pi/2$  (i.e., lying on the equatorial plane) and  $\phi = 0$  with  $r_0$  spanning from  $2M$  (right on the horizon) to  $6M$ , and for each of them we fire a null geodesic from the  $f_2$  axis, integrating it backward in time. Hence, we are considering photons reaching these observers from the direction of  $f_2$  and tracing them back in time, as one usually does in raytracing applications. We obtain the photon paths depicted in Fig. 9. As one would expect from theoretical insights, photons reaching tangentially the observers at  $r_0 < 3M$  hit the horizon. Photons reaching the observers at  $r_0 > 3M$ , on the other hand come from infinity, while at  $r = 3M$ , the photon moves on a perfect circumference, revolving indefinitely around the central black hole.

### 3.4. Reconstructing astronomical observables: The S2 star in the Galactic Center

In this example, we studied how one can use the `Orbit` class in PyGRO and the results of the numerical integration of the geodesic equations to obtain astronomical observables for a test particle around a Schwarzschild (or any spherically symmetric) black hole. Specifically, we particularized the computation for the case of the S2 star in the Galactic Center De Laurentis et al. (2023). The high ratio between the star’s mass ( $m \lesssim 10 M_\odot$ ) and that of the supermassive black hole Sgr A\* ( $M \sim 4 \times 10^6 M_\odot$ ) allows the dynamics of the star to be described as a freely falling test particle in the gravitational field of the central object. For this reason, we assumed that S2 moves on a time-like geodesic and used the `Orbit` class to obtain its trajectory in PyGRO. The set of classical Keplerian orbital parameters, ( $t_p$ ,  $a$ ,  $e$ ,  $i$ ,  $\omega$ ,  $\Omega$ ), as



**Fig. 9.** Photon trajectories obtained with PyGRO, fired by observers at positions  $r_0 \in [2, 6]M$ , around a Schwarzschild black hole with an initially tangential direction on the equatorial plane. Photons fired from below  $r_0 < 3M$  end up in the horizon. Photons fired from  $r_0 > 3M$  escape to infinity, after being strongly deflected by the central object (trajectories grazing closer to the  $r_0 = 3M$  circumference experience a stronger lensing). Finally, for  $r_0 = 3M$ , the photon describes a circumference around the central black hole corresponding to the unstable photon orbit. In Table 1, we label with (4), (5), and (6) the photons with  $r_0 = 2.1M$ ,  $3M$ , and  $6M$ , respectively.

obtained from astrometric and spectroscopic observations for S2 is reported in the Table 3 of Gillessen et al. (2017), assuming a mass of Sgr A\* of  $M = (4.35 \pm 0.13) \times 10^6 M_\odot$  and a distance of Earth from the Galactic Center of  $D = 8.33 \pm 0.12$  kpc. In our case we can use the values of these Keplerian elements with the same physical meaning that we have provided in Section 2.4.2. This results in an integrated four-dimensional trajectory for S2, ( $t(\tau)$ ,  $r(\tau)$ ,  $\theta(\tau)$  and  $\phi(\tau)$ ) in the Schwarzschild coordinate system as a function of the rest frame proper time  $\tau$  of the star. To make this trajectory comparable with astronomical observations, it is necessary to reconstruct physically observable quantities. The available data for the S-stars in the Galactic Center (Gillessen et al. 2017) consist of sky-projected astrometric positions for an Earth-based observer and spectroscopically reconstructed line of sight velocities. The information on both observables is thus carried by photons emitted by the star along its orbit and collected by a distant observer. If one wants to approach the problem of reconstructing the astrometric and spectroscopic observables in a fully relativistic way, photon paths represented by null geodesics must be integrated after an appropriate initial-condition-search has been carried out to identify the right null trajectory connecting emitter and observer at each epoch, the so-called “emitter-observer problem”. The advantage of this approach is that one would be able to take into account at once all the relativistic effects on the star and the photons. This technique, which has been developed in full generality for spherically symmetric space-times in Della Monica et al. (2023a) and is built upon PyGRO, is quite costly in terms of computational resources. On the other hand, the current observational sensitivity in the Galactic Center only allows the leading-order relativistic effects to be detected (see, e.g., the supplementary materials of Do et al. 2019). For this reason, we can neglect the fully relativistic treatment of light rays and use a semi-classical approach to reconstruct the two observables. A useful way to

understand this approach is to consider it as a two-stage process. First, the fully relativistic timelike geodesic is integrated in a chosen coordinate system (e.g., the usual spherical coordinate) within the curved spacetime. Then, these coordinates are identified with those of a Newtonian spacetime, allowing for the projection of light rays in a flat spacetime using standard Newtonian formulas. This corresponds to using the aerial coordinate  $r$  as representative of the Euclidean concept of distance between the test particle and the central source, which of course implies attributing a physical meaning to a space-time coordinate that *per se* does not have one. While breaking the general covariance of general relativity, this assumption allows one to work with the integrated trajectory as one would do in Newtonian mechanics, by constructing a Cartesian frame  $(x, y, z)$  of which  $(r, \theta, \phi)$  constitute the usual spherical coordinates. From the projection of the star's position and velocity in this Cartesian frame one can retrieve the relative positions between Sgr A\* and the star projected on the sky-plane (which we consider to be the  $xy$  plane) and the kinematic velocity along the line-of-sight,  $v_z$ . At this point, it is fairly easy to derive the angular separations between Sgr A\* and S2 (i.e., the experimentally observable astrometric positions),

$$\alpha = \frac{y(\tau)}{D}, \quad \delta = \frac{x(\tau)}{D}, \quad (29)$$

which respectively correspond to right ascension and declination.

Positions and velocities of the test particle are computed in terms of proper time. However, the zeroth component of the integrated space-time trajectory of the particle,  $t(\tau)$ , represents a map between this proper time  $\tau$  and the coordinate time  $t$ , namely (in Schwarzschild coordinates) the time measured by a clock at infinity. This quantity is very useful when reconstructing astronomical observables because we can effectively identify it with the time measured by clocks on Earth, very distant from the gravitational field of Sgr A\*, that is used to label the observation epochs. We can thus use the coordinate time as a bookkeeper coordinate to label both the moments of emission of the photons,  $t_{\text{em}}$ , and of arrival at the observer's location,  $t_{\text{obs}}$ . The relation between the two quantities is given by the photon travel time between emission and reception, which is a combination of classical effects and relativistic delays (Damour & Deruelle 1986). For what concerns the S-stars observations, the only effect that gives a non-negligible contribution is the classical Rømer delay. This effects simply accounts for the light-ray propagation time assuming straight-line propagation and can thus be expressed as (we restore here and for the rest of the paragraph the speed of light  $c$  for a clearer notation)

$$\Delta t_{\text{Rømer}} \equiv t_{\text{em}} - t_{\text{obs}} = \frac{z(t_{\text{em}})}{c}. \quad (30)$$

For inclined orbits, the time-varying character of  $z(t_{\text{em}})$  induces a periodic modulation on the time of arrival of the photons that one has to take into account. Solving Eq. (30) for  $t_{\text{obs}}$  gives out a map between observation times and emission times. The zeroth component of the integrated geodesic,  $t(\tau)$  relates coordinate times of emission to proper times. Thus, we have an invertible chain of maps that links the proper times with the times of observation and vice versa,

$$\tau \leftrightarrow t_{\text{em}}(\tau) \leftrightarrow t_{\text{obs}}(t_{\text{em}}), \quad (31)$$

that one can invert numerically to express the proper time as a function of the observation time:

$$\tau = \tau(t_{\text{obs}}). \quad (32)$$

This allows one to re-parametrize the reconstructed observables as a function of the observation time,

$$\alpha(\tau) \rightarrow \alpha(t_{\text{obs}}), \quad (33)$$

$$\delta(\tau) \rightarrow \delta(t_{\text{obs}}), \quad (34)$$

$$v_z(\tau) \rightarrow v_z(t_{\text{obs}}), \quad (35)$$

which is a fundamental step for the comparison of synthetic orbits with actual observational data. Due to its high inclination, observations for S2 are drastically affected by Rømer delay. The photons emitted at the apocenter will arrive on Earth almost a week later as compared to a case with no inclination, while at the pericenter photons arrive  $\sim 13$  hours earlier.

Here, we move on to the line-of-sight velocity. This observable was reconstructed from spectroscopic observations. The apparent redshift  $\zeta$  in the star's spectrum was measured and converted into an apparent line-of-sight velocity,  $v_{\text{LOS}}$ , by considering that this shift is produced only due to the longitudinal Doppler effect:

$$\zeta = \frac{\Delta\lambda}{\lambda} - 1 \equiv \frac{v_{\text{LOS}}}{c}. \quad (36)$$

For non-relativistic orbits, the line-of-sight velocity reconstructed with this method matches the kinematic velocity  $v_z$  of the celestial body along the line of sight. While  $v_z$  accounts for most of the apparent redshift,

$$\zeta_k(t_{\text{obs}}) = \frac{v_z(t_{\text{obs}})}{c}, \quad (37)$$

the combination of the intense gravitational field produced by Sgr A\* and the subsequent high orbital speed brings the S2 star orbit into the relativistic regime. More specifically, the gravitational time dilation due to the proximity to the supermassive source and the transverse Doppler effect (i.e., Lorentz time dilation) due to the high velocity of the star, especially near the pericenter, produce a non-negligible increase in the apparent redshift that must be taken into account to properly reconstruct the spectroscopic observable. The combination of the two effects is named the Einstein delay. From our fully relativistic integration of the star's motion, we can directly estimate the coordinate-to-proper-time shift, which corresponds to the integrated zeroth component of the four-velocity:

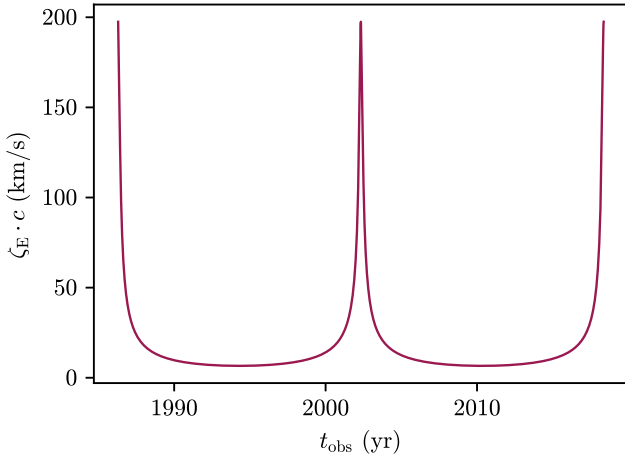
$$\zeta_E(t_{\text{obs}}) + 1 = i(t_{\text{obs}}). \quad (38)$$

We can then combine this redshift component to the kinematic term in Eq. (37), giving the total apparent redshift

$$\frac{v_{\text{LOS}}(t_{\text{obs}})}{c} + 1 = (\zeta_k(t_{\text{obs}}) + 1)(\zeta_E(t_{\text{obs}}) + 1). \quad (39)$$

In Fig. 10, we report the relativistic redshift reconstructed from the integrated geodesic for the S2 star. The relativistic contribution at the pericenter accounts for an additional  $\sim 200$  km/s kick in the apparent line-of-sight velocity that has been successfully measured by high-resolutions spectroscopic observations in the Galactic Center (Do et al. 2019; Gravity Collaboration 2018).

When the whole procedure explained in this section is taken into account, a fully relativistic set of observables  $(\alpha(t_{\text{obs}}), \delta(t_{\text{obs}}), v_{\text{LOS}}(t_{\text{obs}}))$  for a massive test particle around a black hole is obtained. We show in Fig. 11 the observables reconstructed using PyGRO for the S2 star and compare them to the available sets of astrometric and spectroscopic data for this object



**Fig. 10.** Additional contribution to the apparent line-of-sight velocity given by the combination of the special relativistic Doppler effect and gravitational time dilation for the S2 star in the Galactic Center. The relativistic redshift component is maximized around the pericenter (at  $t \sim 2002$  and  $t \sim 2018$ ), reaching an additional contribution of  $\sim 200$  km/s (Do et al. 2019; Gravity Collaboration 2018), where the star goes deeper in the gravitational potential well of Sgr A\* and is accelerated to its maximum orbital velocity.

(Gillessen et al. 2017). The agreement between the reconstructed observables and the observational data for S2 is striking and showcases the potentiality of PyGRO to link theoretical models for space-time geometries to astrophysically relevant measurable quantities. This is especially useful to perform model fitting (e.g., by applying Bayesian parameter estimation techniques) of orbital models based on extensions to the Schwarzschild metric, which account for modifications to general relativity or the existence of black hole mimickers, thus leveraging the generality of the orbital integration in PyGRO (De Martino et al. 2021; Della Monica & de Martino 2022; Della Monica et al. 2022; Della Monica & de Martino 2023b; Della Monica et al. 2023b,a; Della Monica & de Martino 2023a; Cadoni et al. 2023; Fernández Fernández et al. 2023; de Mora Losada et al. 2025).

### 3.5. Photon trajectories around a rotating black hole

In this section, we show how PyGRO can also work outside the family of spherically symmetric space-times. Even though the Orbit class can only work in spherical symmetry, the low level APIs in PyGRO and the Observer class can work in full generality. Here, for example, we study the photon trajectories in the Kerr space-time, whose metric in Boyer-Lindquist coordinates  $(t, r, \theta, \phi)$  takes the form (Bardeen et al. 1972)

$$ds^2 = -\left(1 - \frac{2Mr}{\Sigma}\right) dt^2 - \frac{4Mar \sin^2 \theta}{\Sigma} dt d\phi + \frac{\Sigma}{\Delta} dr^2 + \Sigma d\theta^2 + \left(r^2 + a^2 + \frac{2Ma^2 r \sin^2 \theta}{\Sigma}\right) \sin^2 \theta d\phi^2, \quad (40)$$

with  $\Sigma \equiv r^2 + a^2 \cos^2 \theta$  and  $\Delta \equiv r^2 - 2Mr + a$ . Eq. (40) describes the space-time around a rotating black hole of mass  $M$  with angular momentum per unit mass  $a$ . In the limit  $M \rightarrow 0$ , the Kerr metric reduces to flat Minkowski space-time in oblate spheroidal coordinates, that are related to standard flat-space Cartesian coordinates via the relations  $x = \sqrt{r^2 + a^2} \sin \theta \cos \phi$ ,

$y = \sqrt{r^2 + a^2} \sin \theta \sin \phi$  and  $z = r \cos \theta$ . We used these flat-space limit Cartesian coordinates for the purpose of visualization of the geodesics resulting from our calculations.

For simplicity, we restricted our consideration to the equatorial plane<sup>7</sup>  $\theta = \pi/2$ . On this plane, one can uniquely assign initial conditions to null geodesics at a given point by defining an impact parameter  $b \equiv L/E$  where  $L$  and  $E$  are the azimuthal angular momentum and specific energy as defined in Eqs. (10)–(11). A choice of  $b$  uniquely fixes<sup>8</sup> values of  $\dot{t}$  and  $\dot{\phi}$  and the conservation of the Lagrangian itself, due to the normalization of null geodesic in Eq. (3), fixes the remaining component  $\dot{r}$ . We can compute all these quantities symbolically in PyGRO and then invert the relation between  $b$  and  $(\dot{t}, \dot{r}, \dot{\phi})$  numerically. This allows one to assign initial conditions in PyGRO once an initial position and a value for the impact parameter  $b$  have been assigned. An interesting classical result of geodesic motion in the Kerr space-time that we want to recover in PyGRO is the existence of critical impact parameters  $b_c$  on the equatorial plane for which the null geodesic results in an unstable photon orbit. It is a known result (see, e.g., Chandrasekhar 1985) that  $b_c$  satisfies the equation

$$(b_c + a)^3 = 27M^2(b_c - a) \quad (41)$$

for any given value of  $a$ . For  $a = 0$  (non-rotating case), the only solution is  $b_c = \sqrt{27}M$ , which corresponds to the circular photon orbit with  $r = 3M$  that we obtained in Section 3.3 for the Schwarzschild case. For  $0 < a < 1$ , due to frame dragging, Eq. (41) has two solutions:  $b_{c,\pm}$  with opposite signs. One ( $b_{c,+}$ ) is for prograde (null geodesics with  $L$  having the same sign of  $a$ ), and the other ( $b_{c,-}$ ) is for retrograde (with  $L$  and  $a$  having opposite signs) null geodesics. This means that there exist two different circular photon orbits on the equatorial plane at radii

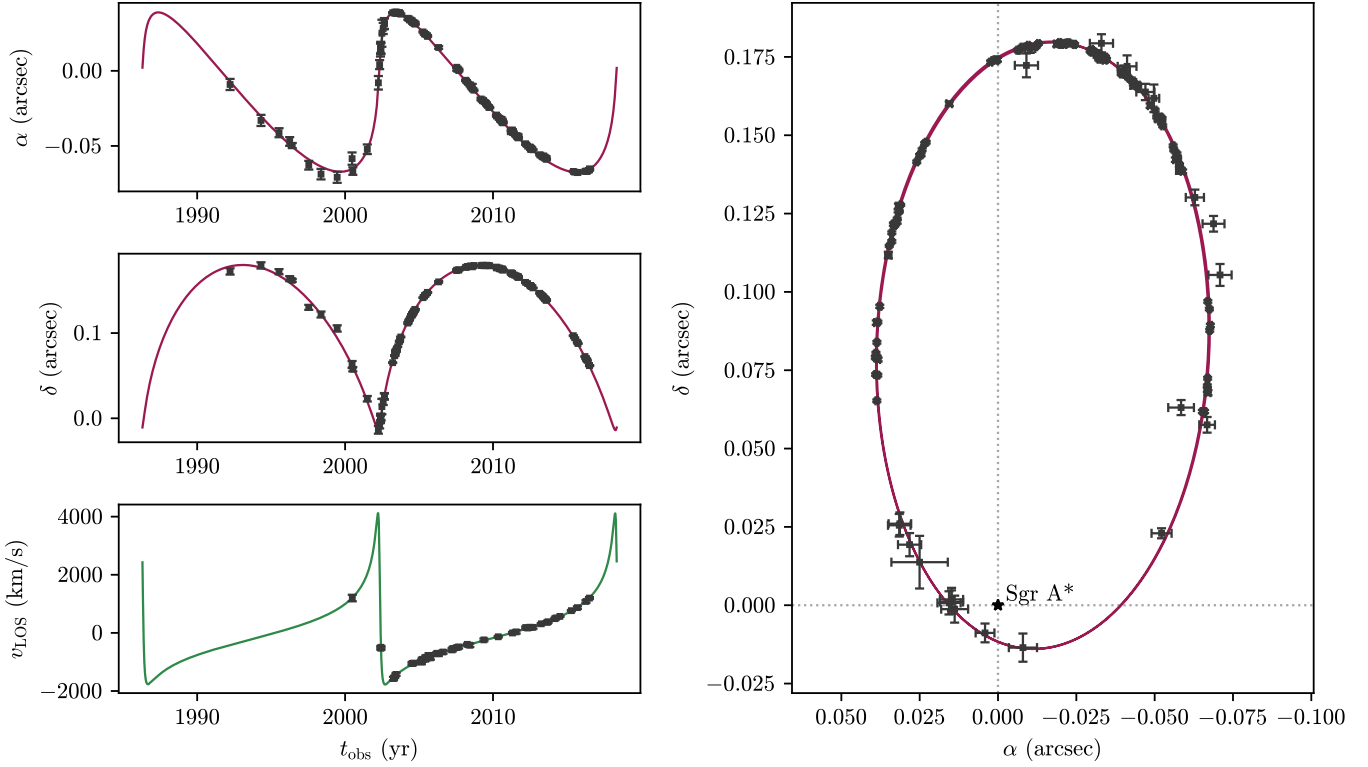
$$r_{\pm} = 2M \left[ 1 + \cos \left( \frac{2}{3} \arccos \frac{\pm a}{M} \right) \right]. \quad (42)$$

Null geodesics with  $b_{c,-} < b < b_{c,+}$  will be bound to the central black hole and cross the event horizon. Null geodesics with impact parameter outside this interval will escape to infinity. Null geodesics with  $b_{c,\pm}$  will approach asymptotically the unstable circular orbits in Eq. (42).

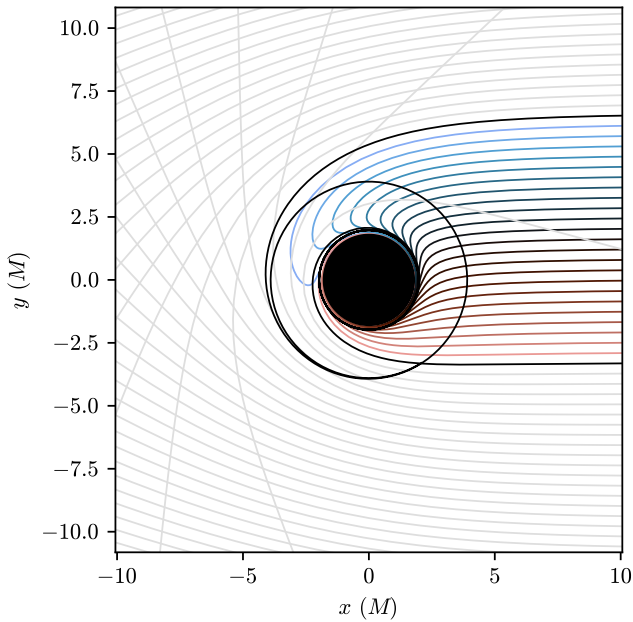
In PyGRO, we can reproduce this result by integrating numerically null geodesics with different impact parameters in the space-time in Eq. (40). In particular, in Fig. 12 we show the results for a rapidly rotating black hole with  $a = 0.8$ . We consider an equally spaced set of impact parameters, spanning positive and negative values, and including the two critical impact parameters  $b_{c,\pm}$  that we compute by numerically solving Eq. (41). We fix the starting position of the photons at  $r = 10^4 M$  on the equatorial plane, so to mimic photons received by a distant observer who performs ray-tracing for the Kerr black hole. For each value of  $b$  we assign initial conditions as described above and we integrate the geodesic equations backward in time. As expected theoretically, photons with  $b_{c,-} < b < b_{c,+}$  (colored lines) cross the horizon, while the others (gray lines) escape to

<sup>7</sup> Due to breaking of the spherical symmetry in rotating space-times, not all planes are equal and the equatorial one benefits from additional symmetries, e.g. the fact that geodesic trajectories on the equatorial plane will stick to it for the full duration of the motion.

<sup>8</sup> More precisely, a choice of  $b$  fixes the ratio  $\dot{\phi}/\dot{t}$  and one can freely assign  $\dot{t}$  at the initial time, given the independence of the geodesic equations on the choice of the affine parameterization.



**Fig. 11.** Full set of observables for S2 in the Galactic Center reconstructed from the geodesic integration with PyGRO. In particular, in the left panels we show the right ascension (top panel) and declination (middle panel) relative to Sgr A\* of S2 over two full orbital periods and the fully relativistic line-of-sight velocity (bottom panel). In the right panel, we display the full apparent trajectory in the sky relative to Sgr A\*. The black dots correspond to the publicly available astrometric and spectroscopic data for S2 from [Gillessen et al. \(2017\)](#) with their corresponding error.



**Fig. 12.** Photon trajectories on the equatorial plane of the Kerr space-time with  $a = 0.8$ . We obtained the trajectories by integrating backward in time from a distant observer on the  $x$  axis. The solid black lines correspond to the prograde and retrograde circular photon orbits, obtained for the critical values,  $b_{c,\pm}$ , of the impact parameter. Photons with impact parameters between  $b_{c,+}$  and  $b_{c,-}$  are bound to the black hole and cross the horizon (colored solid lines). Outside this interval photons escape to infinity (gray lines). Due to frame dragging the two critical impact parameters are not symmetrical with respect to the central black hole, which ultimately causes the asymmetry of the Kerr shadow.

infinity. Geodesics with impact parameter matching the critical values (black lines) approach the unstable circular orbits for the prograde and retrograde cases. The asymmetry of the region corresponding to bound photons with respect to a photon initially aligned with the central black hole ( $b = 0$ ) ultimately causes the asymmetry of the Kerr shadow ([Cunningham & Bardeen 1973](#); [Luminet 1979](#)).

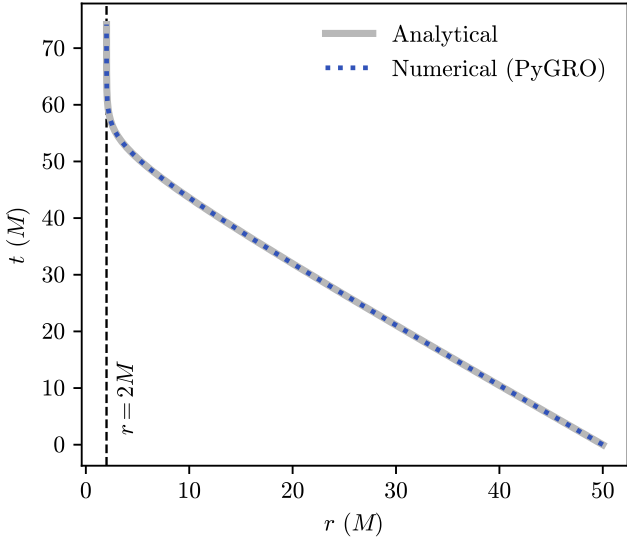
### 3.6. Benchmark of integration accuracy: The radial infall case

In this last example, we go back to Schwarzschild metric, Eq. (23), and consider the case of a radially infalling null geodesic, for which an analytical solution is known. We compare the results obtained with PyGRO with those from the analytical formula and check the accuracy of the integration of the geodesic equation within PyGRO. Also, we perform some convergence check based on the conserved norm of the tangent four-vector to the integrated geodesic.

Radial null geodesics are the simplest case of trajectories that one can consider in the space-time described by Eq. (23). In this case, due to the spherical symmetry of the problem, the geodesic equations reduce only to the temporal and radial components, enabling a complete analytical solution to be obtained, which has the form ([Chandrasekhar 1985](#))

$$t(r) = \mp(r + 2M \log(r - 2M)) \pm (r_0 + 2M \log(r_0 - 2M)), \quad (43)$$

where  $r_0$  is the initial radial coordinate, and the solution corresponding to the first sign that is  $-$  ( $+$ ) corresponds to an inward (outward) pointing geodesic. The appearance of a logarithm exhibits one of the most peculiar behaviors of the dynamics of particles in the near-horizon zone. The coordinate time of an



**Fig. 13.** Analytical (gray solid line) and numerical (blue dotted line, obtained with PyGRO) solutions for a radially infalling null geodesic in the Schwarzschild space-time when starting from an initial radial coordinate of  $r_0 = 50M$ . The two solutions perfectly agree. The dashed vertical line corresponds to the Schwarzschild radius where the coordinate time exhibits a divergence.

infalling geodesic tends to diverge as it approaches the horizon at  $r = 2M$ . This divergence means that a very distant physical observer will never see the infalling particle cross the horizon. We are going to reproduce the same configuration in PyGRO. To do so, we considered an Observer defined as in Eqs. (27)–(28), at a radial coordinate  $r_0$  on the equatorial plane of the Schwarzschild space-time. We then consider a null geodesic fired at the observer’s location along the  $f_1$  axis, so that it points radially inward. We carried out the integration up to a given stopping criterion (e.g., in this case we stop the integration when the geodesic reaches radial coordinate which is right above the event horizon radius  $r = 2M$ ). We then apply the formula in Eq. (43) to obtain the analytical solution and we compare the two results at the same radii.

In Fig. 13, we show the  $t(r)$  profiles obtained in the two cases for an initial radius of  $r_0 = 50M$ . The two curves perfectly agree, with the numerical result obtained in PyGRO reproducing the divergence of the coordinate time at  $r = 2M$ . To better quantify the agreement between the two estimations, we perform a convergence check, whose results we report in Fig. 14. Our analysis shows that: (i) by reducing the integration tolerance the discrepancy between the result of the numerical integration and the analytical formula in Eq. (13) converges to zero and always matches, on average, the order of magnitude of the integration tolerance; (ii) the amount of constraint violation (we focus specifically on the normalization of the tangent four-vector to the geodesic reported in Eq. (3)) tends to zero as well, as the integration tolerance is improved. These results show not only that the numerical routine implemented in PyGRO allows for accurate integration of the geodesic equations, but also that the amount of error and constraint violation in the integration can be controlled effectively by setting appropriate precision and accuracy goals in the integrator. Additionally, we have checked that the best results, in terms of integration accuracy and computational time, are obtained employing a Dormand-Prince5(4) integrator, which is thus set as the default choice in PyGRO.

### 3.7. Integration performances

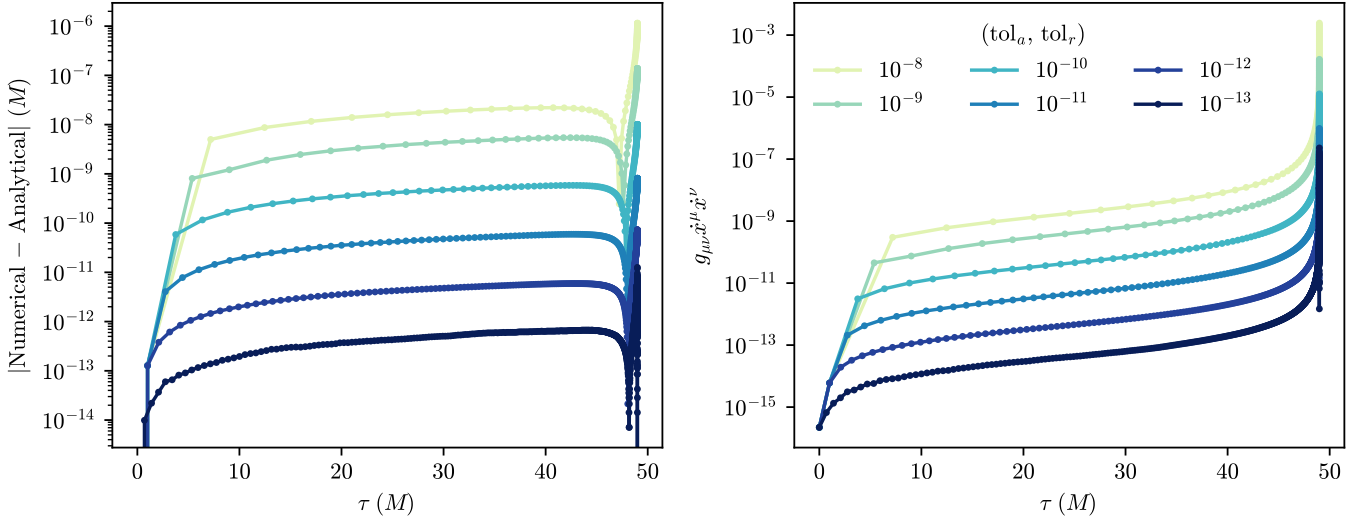
In this section, we report the performances of PyGRO, showcasing the runtimes for all the examples described in the previous sections. PyGRO is light-weight and well optimized to run on any machine. All the examples and tutorials on the official documentation can be run on a normal laptop or even on mobile platforms (e.g., Juno for the iOS ecosystem or Pydroid for Android platforms). All the runtimes reported here have been obtained on a M1 MacBook Pro with 8GBs of RAM.

The symbolic differentiation and matrix operations required for the initialization of the Metric object, upon which all the symbolic manipulations are done to compute the Christoffel symbols and the geodesic equations (see Section 2.1), while introducing an overhead compared to other codes (e.g., Vincent et al. 2011) where all derived quantities are hardcoded, is usually fast, with the operation of computing the helper functions to normalize the initial tangent four-vector being the most computationally expensive. In general, initializing a Metric object takes about  $\sim O(1s)$ . The initialization of a GeodesicEngine can require a slightly longer time, normally  $\sim O(10s)$ , due to the pre-compilation of the C callables for the geodesic equations. This overhead, represents the trade-off between flexibility and performance that allows PyGRO to work with arbitrary space-time geometries without requiring manual derivations. If the *lambdify* wrapper is used instead (see Section 2.2), the overhead is much shorter, usually  $\sim O(0.1s)$ , but will bring to a loss in performances on integration.

Integrating a Geodesic in PyGRO can be really fast depending on the specific configuration of the trajectory (e.g., how close it gets to an horizon, where a smaller step size is required to attain a given precision tolerance). From all the examples that we have shown throughout Section 3 for the Schwarzschild metric, we have focused on seven specific cases for which we systematically assess the integration performances of PyGRO. The cases taken into considerations are as follows:

1. The precessing time-like orbit in the Schwarzschild space-time shown in Fig. 5 with  $e = 0.8$  and  $a = 200M$ , integrated over ten full revolutions;
2. The unstable circular orbit for a time-like geodesic with  $a = 5M$  shown in the left panel of Fig. 7. As in the last case, we carried out the integration for ten full revolutions;
3. A stable circular orbit for a time-like geodesic with  $a = 10M$  shown in the left panel of Fig. 7. Again, we considered ten full revolutions;
4. One of the plunging null geodesics shown in Fig. 9. In particular, we considered the geodesic with  $r_0 = 2.1M$ , and we used as a stopping criterion the condition  $r > 2.001M$  (to stop the integration when the geodesic is right above the horizon);
5. The unstable photon orbit with  $r_0 = 3M$  shown in Fig. 9. In this case, we stopped the integration after ten full revolutions;
6. One of the escaping photon trajectories from Fig. 9. We focused in particular on the geodesic starting at  $r_0 = 6M$ . The integration was stopped at  $s = 1000M$ , with  $s$  as the affine parameter on the null geodesic;
7. The radially infalling geodesic in Fig. 13 starting at  $r_0 = 50M$ . As in case (4), we considered  $r > 2.001M$  as a condition for the stopping criterion.

In Table 1, we report the runtimes obtained for all the cases described above, for three different values of the integration tolerance in Eq. (6), using the DormandPrince5(4) integrator. The values reported in the table correspond to the mean runtime



**Fig. 14.** Convergence check of the numerical integration routine in PyGRO. In particular, for the same case illustrated in Fig. 13, with  $r_0 = 50M$ , we compute the difference (left panel) between the known analytical results, Eq. (43) and the results obtained in PyGRO, for different values of the precision and accuracy tolerances (Eq. (6)) that we vary over six orders of magnitude. The plots shows the convergence of the numerical result to the analytical one as the integration tolerances are reduced. Moreover, we compute on the integrated geodesic the norm of the tangent four-vector (right panel) which, following Eq. (3), should be identically zero. The plot reports the amount of constraint violation, which shows convergence to zero as the integration precision is improved, thus validating the numerical routines implemented in PyGRO.

**Table 1.** Integration performances for all the examples in Section 3.

Example	Tolerance	Runtime (ms)
(1)	$10^{-8}$	$77.5 \pm 3.6$
	$10^{-11}$	$202 \pm 22$
	$10^{-13}$	$405 \pm 28$
(2)	$10^{-8}$	$2.41 \pm 0.23$
	$10^{-11}$	$5.28 \pm 0.49$
	$10^{-13}$	$9.3 \pm 2.5$
(3)	$10^{-8}$	$1.97 \pm 0.04$
	$10^{-11}$	$2.96 \pm 0.17$
	$10^{-13}$	$4.19 \pm 0.08$
(4)	$10^{-8}$	$8.6 \pm 0.5$
	$10^{-11}$	$31.1 \pm 1.9$
	$10^{-13}$	$74.8 \pm 0.5$
(5)	$10^{-8}$	$17.5 \pm 0.47$
	$10^{-11}$	$39.1 \pm 3.1$
	$10^{-13}$	$97.8 \pm 1.6$
(6)	$10^{-8}$	$7.75 \pm 0.40$
	$10^{-11}$	$26.3 \pm 3.2$
	$10^{-13}$	$61.4 \pm 1.6$
(7)	$10^{-8}$	$18.5 \pm 0.7$
	$10^{-11}$	$50.5 \pm 4.1$
	$10^{-13}$	$110.0 \pm 3.4$

**Notes.** The examples considered correspond to: (1) Precessing orbit shown in Fig. 5; (2) Unstable circular orbit in Fig. 7 (left panel); (3) Stable circular orbit in Fig. 7 (right panel); (4) Plunging null geodesic with  $r_0 = 2.1M$  in Fig. 9; (5) Unstable photon orbit with  $r_0 = 3M$  in Fig. 9; (6) Escaping null geodesic with  $r_0 = 6M$  in Fig. 9; (7) radially infalling geodesic in Fig. 13.

over  $10^2$  repetitions of the geodesic integration, with the error computed as their standard deviation.

#### 4. Discussion and conclusions

The study of relativistic orbits plays a central role in our understanding of the nature of gravity and the ability to test predictions of general relativity. Over the past century, the equivalence principle and classical tests of gravity have evolved from basic confirmations of Einstein’s theory to precision measurements in extreme astrophysical environments (Will 2014). Recent advances in experimental gravitation – such as the detailed monitoring of the S-stars at the Galactic Center (De Laurentis et al. 2023), the groundbreaking imaging of black holes by EHT (Event Horizon Telescope Collaboration 2019, 2022), and the potential discovery of pulsars orbiting a supermassive black hole (Della Monica & de Martino 2025) – have ushered in a new era of strong-field gravitational physics.

These advancements call for modern computational tools that can bridge theoretical predictions with observations in increasingly complex and general settings. We developed PyGRO with this goal in mind and to provide a fast and highly customizable open-source framework for the numerical integration of geodesic equations in any analytic four-dimensional space-time. By combining symbolic and numerical methods, PyGRO allows users to explore relativistic dynamics in a wide range of scenarios, from classical tests in weak gravitational fields to orbits around compact objects in the strong-field regime. The examples presented in this article and those in the official documentation demonstrate the robustness of PyGRO. The code’s methodology has been validated through successful reproduction of key results of general relativity and by its extensive use in past scientific studies (De Martino et al. 2021; Della Monica & de Martino 2022; Della Monica et al. 2022; Della Monica & de Martino 2023b; Della Monica et al. 2023b,a

Della Monica & de Martino 2023a; Cadoni et al. 2023 Fernández Fernández et al. 2023; de Mora Losada et al. 2025) that demonstrate its great flexibility. Beyond these benchmarks, the modular and open-source nature of PyGRO enables future extensions and extra features to further improve its range of applications. For example, we plan in the near future to extend the capabilities of integrating general relativistic orbits with a parameterization based on the Keplerian orbital elements of classical celestial mechanics and to the case of axisymmetric space-times, namely rotating black holes. While we have shown in this article that the low-level APIs in PyGRO offer sufficient generality to be easily applied to axisymmetric space-times, such as the Kerr solution, the mapping between Keplerian elements and initial conditions in this case requires further theoretical modeling, especially for orbits outside the equatorial plane. Notably, the implementation of polarized general relativistic ray-tracing techniques (like the ones presented in Aimar et al. 2024) could significantly extend the scope of PyGRO and enable tests of the properties of relativistic astrophysical plasmas. PyGRO opens new possibilities for the broader astrophysics community by offering a powerful tool for both theoretical studies and the analysis of observational data in preparation of next-generation experiments (GRAVITY+ Collaboration 2022; Keane et al. 2015).

*Acknowledgements.* PyGRO is the culmination of the work carried out during my PhD studies at the University of Salamanca. I am sincerely indebted to my supervisor Ivan de Martino for his guidance. I also thank Frédéric Vincent for providing valuable insights and suggesting relevant improvements on the final manuscript. Moreover, I thank Gabriel Sánchez Pérez and Diego Martín González for testing my code and suggesting some relevant implementations. Finally, I acknowledge financial support from the grant PID2021-122938NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, from Consejería de Educación de la Junta de Castilla y León and the European Social Fund +.

## References

- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016, *Phys. Rev. Lett.*, **116**, 061102
- Aimar, N., Paumard, T., Vincent, F. H., Gourgoulhon, E., & Perrin, G. 2024, *Class. Quant. Grav.*, **41**, 095010
- Bardeen, J. M., Press, W. H., & Teukolsky, S. A. 1972, *ApJ*, **178**, 347
- Bernstein, G. M., Tyson, J. A., & Kochanek, C. S. 1993, *AJ*, **105**, 816
- Bolton, C. T. 1972, *Nature*, **235**, 271
- Cadoni, M., De Laurentis, M., De Martino, I., et al. 2023, *Phys. Rev. D*, **107**, 044038
- Chandrasekhar, S. 1985, *The Mathematical Theory of Black Holes* (Oxford: Clarendon Press)
- Cunningham, C. T., & Bardeen, J. M. 1973, *ApJ*, **183**, 237
- Damour, T., & Deruelle, N. 1985, *Ann. Inst. Henri Poincaré Sect. A Phys. Theor.*, **43**, 107
- Damour, T., & Deruelle, N. 1986, *Ann. Inst. Henri Poincaré Sect. A Phys. Theor.*, **44**, 263
- De Laurentis, M., de Martino, I., & Della Monica, R. 2023, *Rep. Progr. Phys.*, **86**, 104901
- Della Monica, R., & de Martino, I. 2022, *J. Cosmology Astropart. Phys.*, **2022**, 007
- Della Monica, R., & de Martino, I. 2023a, *Phys. Rev. D*, **108**, L101303
- Della Monica, R., & de Martino, I. 2023b, *A&A*, **670**, L4
- Della Monica, R., & de Martino, I. 2025, *Phys. Rev. D*, submitted [arXiv:2501.03912]
- Della Monica, R., de Martino, I., & de Laurentis, M. 2022, *MNRAS*, **510**, 4757
- Della Monica, R., De Martino, I., & De Laurentis, M. 2023a, *MNRAS*, **524**, 3782
- Della Monica, R., de Martino, I., Vernieri, D., & de Laurentis, M. 2023b, *MNRAS*, **519**, 1981
- De Martino, I., della Monica, R., & De Laurentis, M. 2021, *Phys. Rev. D*, **104**, L101502
- de Mora Losada, V., Della Monica, R., de Martino, I., & De Laurentis, M. 2025, *A&A*, **694**, A280
- Dicke, R. H. 1964, in *Les Houches Summer School of Theoretical Physics: Relativity, Groups and Topology*, 165
- Di Francesco, J., Chalmers, D., Denman, N., et al. 2019, in *Canadian Long Range Plan for Astronomy and Astrophysics White Papers*, 2020, 32
- Do, T., Hees, A., Ghez, A., et al. 2019, *Science*, **365**, 664
- Dyson, F. W., Eddington, A. S., & Davidson, C. 1920, *Philos. Trans. Roy. Soc. Lond. Ser. A*, **220**, 291
- Eckart, A., & Genzel, R. 1996, *Nature*, **383**, 415
- Einstein, A. 1915a, *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften* (Berlin), 844
- Einstein, A. 1915b, *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)*, **1915**, 831
- Event Horizon Telescope Collaboration (Akiyama, K., et al.) 2019, *ApJ*, **875**, L1
- Event Horizon Telescope Collaboration (Akiyama, K., et al.) 2021a, *ApJ*, **910**, L12
- Event Horizon Telescope Collaboration (Akiyama, K., et al.) 2021b, *ApJ*, **910**, L13
- Event Horizon Telescope Collaboration (Akiyama, K., et al.) 2022, *ApJ*, **930**, L12
- Event Horizon Telescope Collaboration (Akiyama, K., et al.) 2024, *ApJ*, **964**, L25
- Everitt, C. W. F., Debra, D. B., Parkinson, B. W., et al. 2011, *Phys. Rev. Lett.*, **106**, 221101
- Fernández Fernández, R., Della Monica, R., & de Martino, I. 2023, *J. Cosmology Astropart. Phys.*, **2023**, 039
- Genzel, R., Eisenhauer, F., & Gillessen, S. 2010, *Rev. Mod. Phys.*, **82**, 3121
- Ghez, A. M., Klein, B. L., Morris, M., & Becklin, E. E. 1998, *ApJ*, **509**, 678
- Gillessen, S., Plewa, P. M., Eisenhauer, F., et al. 2017, *ApJ*, **837**, 30
- GRAVITY Collaboration (Abuter, R., et al.) 2018, *A&A*, **615**, L15
- GRAVITY Collaboration (Abuter, R., et al.) 2020, *A&A*, **636**, L5
- GRAVITY+ Collaboration (Abuter, R., et al.) 2022, *The Messenger*, **189**, 17
- Hees, A., Do, T., Ghez, A. M., et al. 2017, *Phys. Rev. Lett.*, **118**, 211101
- Johnson, M. D., Akiyama, K., Blackburn, L., et al. 2023, *Galaxies*, **11**, 61
- Keane, E., Bhattacharyya, B., Kramer, M., et al. 2015, in *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, 40
- Luminet, J. P. 1979, *A&A*, **75**, 228
- Meurer, A., Smith, C. P., Paprocki, M., et al. 2017, *PeerJ Comput. Sci.*, **3**, e103
- Nan, R., Li, D., Jin, C., et al. 2011, *Int. J. Mod. Phys. D*, **20**, 989
- Poisson, E., & Will, C. M. 2014, *Gravity* (Cambridge University Press)
- Pound, R. V., & Rebka, G. A. 1960, *Phys. Rev. Lett.*, **4**, 274
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical recipes in C. The art of scientific computing* (Cambridge University Press)
- Psaltis, D., Medeiros, L., Christian, P., et al. 2020, *Phys. Rev. Lett.*, **125**, 141104
- Skidmore, W., TMT International Science Development Teams, & Science Advisory Committee, T. 2015, *Res. Astron. Astrophys.*, **15**, 1945
- Straumann, N. 2013, *General Relativity*, Graduate Texts in Physics (Dordrecht: Springer)
- Sturm, E., Davies, R., Alves, J., et al. 2024, in *Ground-based and Airborne Instrumentation for Astronomy X*, 13096, eds. J. J. Bryant, K. Motohara, & J. R. D. Vernet, International Society for Optics and Photonics (SPIE), 1309611
- Taff, L. G. 1985, *Celestial Mechanics: A Computational Guide for the Practitioner*
- Vessot, R. F. C., Levine, M. W., Mattison, E. M., et al. 1980, *Phys. Rev. Lett.*, **45**, 2081
- Vincent, F. H., Paumard, T., Gourgoulhon, E., & Perrin, G. 2011, *Class. Quant. Grav.*, **28**, 225011
- Webster, B. L., & Murdin, P. 1972, *Nature*, **235**, 37
- Will, C. M. 2014, *Liv. Rev. Relativ.*, **17**, 4