



ODYSSEY: A PUBLIC GPU-BASED CODE FOR GENERAL RELATIVISTIC RADIATIVE TRANSFER IN KERR SPACETIME

HUNG-YI PU^{1,5}, KIYUN YUN^{2,5}, ZIRI YOUNSI^{3,4}, AND SUK-JIN YOON²

¹ Institute of Astronomy & Astrophysics, Academia Sinica, 11F of Astronomy-Mathematics Building, AS/NTU No. 1, Taipei 10617, Taiwan

² Department of Astronomy & Center for Galaxy Evolution Research, Yonsei University, Seoul 120-749, Korea

³ Institut für Theoretische Physik, Max-von-Laue-Straße 1, D-60438 Frankfurt am Main, Germany

⁴ Mullard Space Science Laboratory, University College London, Holmbury St. Mary, Dorking, Surrey, RH5 6NT, UK

Received 2015 November 3; accepted 2016 January 6; published 2016 March 28

ABSTRACT

General relativistic radiative transfer calculations coupled with the calculation of geodesics in the Kerr spacetime are an essential tool for determining the images, spectra, and light curves from matter in the vicinity of black holes. Such studies are especially important for ongoing and upcoming millimeter/submillimeter very long baseline interferometry observations of the supermassive black holes at the centers of Sgr A* and M87. To this end we introduce *Odyssey*, a graphics processing unit (GPU) based code for ray tracing and radiative transfer in the Kerr spacetime. On a single GPU, the performance of *Odyssey* can exceed 1 ns per photon, per Runge–Kutta integration step. *Odyssey* is publicly available, fast, accurate, and flexible enough to be modified to suit the specific needs of new users. Along with a Graphical User Interface powered by a video-accelerated display architecture, we also present an educational software tool, *Odyssey_Edu*, for showing in real time how null geodesics around a Kerr black hole vary as a function of black hole spin and angle of incidence onto the black hole.

Key words: black hole physics – methods: numerical – radiative transfer

1. INTRODUCTION

Theoretical studies of observable features in the strong gravity environment around black holes, such as emission-line profiles (e.g., Fuerst & Wu 2004, 2007; Müller & Camenzind 2004; Younsi et al. 2012), reverberation (e.g., Reynolds et al. 1999), light curves of a hot spot (e.g., Schnittman & Bertschinger 2004; Broderick & Loeb 2005), quasi-periodic oscillations (QPOs; e.g., Schnittman & Bertschinger 2004; Schnittman & Rezzolla 2006; Fukumura & Kazanas 2008), and black hole shadow images (e.g., Falcke et al. 2000; Takahashi 2004), provide useful insights into understanding the nature of these systems. Recent millimeter very long baseline interferometry (VLBI) observations of Sgr A* and M87 by the Event Horizon Telescope (EHT) helped to place constraints on the core size to within a few Schwarzschild radii and demonstrated the potential capability of the EHT to directly image the shadow of a black hole (Doeleman et al. 2008, 2012). In the near future, images of black hole shadows are expected to be observed by millimeter/submillimeter VLBI observations, e.g., the EHT, BlackHoleCam, and Greenland Telescope (GLT) projects (Inoue et al. 2014).

As the resulting image and spectrum from black hole accretion and jets can vary owing to the dynamics of the accretion flow (e.g., free-fall versus Keplerian sphere of plasma; Falcke et al. 2000; Broderick & Loeb 2006) and are also strongly dependent on the distribution of (both thermal and nonthermal) electrons (e.g., Akiyama et al. 2015; Chan et al. 2015a), a systematic study of the observational predictions of many different models throughout physically relevant parameter space is essential. As such, the rapid, efficient and accurate computation of both geodesics and the solution of the radiative transfer equations along these geodesics in the Kerr spacetime is a very important topic.

The integration of geodesics in the Kerr spacetime can be performed either by using the transfer function method (Cunningham 1975; Fabian et al. 2000), by using the elliptic function method (e.g., Rauch & Blandford 1994; Agol 1997; Dexter & Agol 2009; Yang & Wang 2013), or by direct numerical integration of the geodesic equations of motion (e.g., Fuerst & Wu 2004; Levin & Perez-Giz 2008; Psaltis & Johannsen 2012; Younsi et al. 2012). In the absence of scattering, the integration of the radiation transfer equation can be performed by dividing each ray into a series of small steps. While the elliptic function method, being semianalytic in form, is efficient for the calculation of emission from axisymmetric and optically thick objects like a geometrically thin accretion disk, it does not fare so well for systems that do not possess the necessary symmetry, such as the highly turbulent, nonsymmetric, and magnetized flows found in GRMHD simulations of accretion onto black holes. Direct numerical integration of the geodesic equations of motion by the Runge–Kutta method is more suitable for general relativistic radiative transfer (GRRT) computations as it makes no assumptions of the underlying geometry or thermodynamics of the accretion flow or spacetime being considered. Additionally, because a complex change of variables is not needed, the direct integration method is more straightforward to implement numerically, and its incorporation into more sophisticated and physically realistic models is transparent.

The calculation of geodesics and radiative transfer along each ray (geodesic) can be efficiently boosted through parallel computation. In capitalizing on the advantages offered by parallel programming, much attention has been paid in recent years to the graphics processing unit (GPU). Being somewhat analogous to the messaging passing interface (MPI) model composed of multiple CPUs, the GPU model is built around the concept of multiple streaming multiprocessors (SMs) containing several hundred threads. These threads work as a CPU does, i.e., concurrently, with a single-instruction, multiple-

⁵ Equal first authors.

thread (SIMT) model on a single graphics card. However, a graphics card may comprise several thousand processors. In order to take advantage of the architecture of the GPU, one of the major graphics card manufacturers, NVIDIA, released the Compute Unified Device Architecture (CUDA) platform for General-Purpose computing on Graphics Processing Units (GPGPU). Given that ray tracing, in the absence of scattering, is a trivially parallelizable problem (each ray may be treated as independent from all other rays), no communication between threads is necessary during the calculation of each ray. This makes the GPU model particularly appealing.

In this article we present a public, GPU-based GRRT code, *Odyssey*, based on the ray-tracing algorithm presented in Fuerst & Wu (2004) and the radiative transfer formulation described in Younsi et al. (2012), implemented in CUDA C/C++. The performance on a single NVIDIA GPU graphics card exceeds 1 ns per Runge–Kutta step per geodesic, similar to or slightly better than that reported in another GPU-based ray-tracing code, *GRay* (Chan et al. 2013), in which a different ray-tracing algorithm is adopted. One of the direct applications of *Odyssey* is for studying the spectra and images from black holes at horizon scales. This is an important observational goal of current and future millimeter/submillimeter VLBI observations of the accretion flows onto supermassive black holes.

The article is organized as follows. In Section 2 the ray tracing and radiative transfer formulation are derived and the numerical solution of this formulation is discussed. In Section 3 we introduce and discuss the GPU scheme on which *Odyssey* is based. In Section 4 we present the results of several different benchmarking calculations for *Odyssey*, including the calculation of images, spectra, and light curves necessary for astrophysical calculations and comparisons with future observations. In Section 5 we assess the performance of *Odyssey*, presenting timing benchmarks and comparisons with other ray-tracing codes. Section 6 is devoted to the summary and discussion.

2. FORMULATION

In this article we adopt the natural unit convention ($c = G = 1$), wherein the gravitational radius of a black hole of mass M is given by $r_g = M$. For a rotating (Kerr) black hole the spacetime metric may be written in Boyer–Lindquist coordinates as

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

where $\Sigma \equiv r^2 + a^2 \cos^2 \theta$ and $\Delta \equiv r^2 - 2Mr + a^2$. Hereafter the black hole mass is set to unity, which is equivalent to normalizing the length scale to r_g and the timescale to r_g/c . From the separability of the Hamilton–Jacobi equations for a Kerr black hole, the geodesic equations of motion may be reduced to a problem of quadratures (four constants of motion [see Section 2.1] and four ordinary differential equations) with the variables $(\dot{t}, \dot{r}^2, \dot{\theta}^2, \dot{\phi})$, where an overdot denotes differentiation with respect to the affine parameter. Complications related to the uncertainty in the signs of r and θ at turning points in the geodesic motion arise from this approach, and so to circumvent this issue, we instead consider the second

derivatives of r and θ and integrate six differential equations for $(\dot{r}, \dot{\theta}, \dot{\phi}, \dot{t}, \dot{p}_r, \dot{p}_\theta)$, where (p_r, p_θ) are components of the covariant four-momentum of the geodesic (Fuerst & Wu 2004).

Such an algorithm is sufficiently straightforward to implement into GRRT calculations. The covariant four-momentum p_α is computed at each integration step and can be directly used to compute the relative energy shift, γ , between radiation emitted from material circulating around the black hole with four-velocity u^α and radiation received by a distant observer:

$$\gamma \equiv \frac{\nu}{\nu_0} = \frac{p_\alpha u^\alpha|_\infty}{p_\alpha u^\alpha|_\lambda}. \quad (2)$$

The scalar product $p_\alpha u^\alpha$ is a frame-invariant quantity that may be calculated in any desired frame at that position in spacetime. For simplicity, we choose to evaluate $p_\alpha u^\alpha$ in the observer reference frame at that particular point along the ray (indicated by the affine parameter, λ). Subscripts “0” and “ ∞ ” denote quantities evaluated in the local fluid rest frame and in the reference frame of a distant (stationary) observer, respectively. For completeness, in this section we summarize the governing differential equations (Fuerst & Wu 2004), initial conditions, and radiative transfer formulation (Younsi et al. 2012) used in *Odyssey*.

2.1. Ray-tracing Algorithm

The Kerr spacetime is of Petrov type D and, being independent of both t and ϕ coordinates, possesses two Killing vectors. These give rise to the conservation of energy, E , and conservation of angular momentum, L_z , where L_z is the projection of the particle angular momentum along the black hole spin axis. The rest mass, μ , of the particle (0 for photons and massless particles and -1 for particles with mass) and the Carter constant, Q , are also conserved along each geodesic. As such, the Kerr black hole possesses four constants of motion. From the Lagrangian, the covariant four-momenta components of a geodesic may be written as

$$p_t = -E, \quad (3)$$

$$p_r = \frac{\Sigma}{\Delta} \dot{r}, \quad (4)$$

$$p_\theta = \Sigma \dot{\theta}, \quad (5)$$

$$p_\phi = L_z. \quad (6)$$

In addition, E and L_z may be derived from the initial conditions of the ray via the following formulae:

$$E^2 = \left(\frac{\Sigma - 2r}{\Sigma \Delta}\right) (\Sigma \dot{r}^2 + \Sigma \Delta \dot{\theta}^2 - \Delta \mu) + \Delta \dot{\phi}^2 \sin^2 \theta, \quad (7)$$

$$L_z = \frac{(\Sigma \Delta \dot{\phi} - 2arE) \sin^2 \theta}{\Sigma - 2r}. \quad (8)$$

The corresponding geodesic equations of motion may then be written as

$$\dot{t} = E + \frac{2r(r^2 + a^2)E - 2ar L_z}{\Sigma \Delta}, \quad (9)$$

$$\dot{r}^2 = \frac{\Delta}{\Sigma} (\mu + E\dot{t} - L_z \dot{\phi} - \Sigma \dot{\theta}^2), \quad (10)$$

$$\dot{\theta}^2 = \frac{1}{\Sigma^2} [Q + (E^2 + \mu)a^2 \cos^2 \theta - L_z^2 \cot^2 \theta], \quad (11)$$

$$\dot{\phi} = \frac{2arE + (\Sigma - 2r)L_z \csc^2 \theta}{\Sigma \Delta}, \quad (12)$$

where

$$Q \equiv p_\theta^2 + [L_z^2 \csc^2 \theta - a^2(E^2 + \mu)] \cos^2 \theta. \quad (13)$$

Equations (10) and (11) may be replaced with two equations for the covariant four-momenta (for details, see Fuerst & Wu 2004) as follows:

$$\begin{aligned} \dot{p}_r = \frac{1}{\Sigma \Delta} \{ & [(r^2 + a^2)\mu - \kappa](r - 1) + r\Delta\mu \\ & + 2r(r^2 + a^2)E^2 - 2aEL_z \} - \frac{2p_r^2(r - 1)}{\Sigma}, \end{aligned} \quad (14)$$

$$\dot{p}_\theta = \frac{\sin \theta \cos \theta}{\Sigma} \left[\frac{L_z^2}{\sin^4 \theta} - a^2(E^2 + \mu) \right], \quad (15)$$

where

$\kappa \equiv Q + L_z^2 + a^2(E^2 + \mu) = p_\theta^2 + L_z^2 \csc^2 \theta + a^2(E^2 \sin^2 \theta + \mu)$. The final equations of motion for the six parameters ($r, \theta, \phi, t, p_r, p_\theta$) are then given by Equations (3), (4), (9), (12), (14), and (15).

In *Odyssey*, a fifth-order Runge–Kutta scheme with adaptive step size (Press et al. 1992) is used to integrate these equations. In addition, to avoid numerical problems when a photon passes too close to the pole ($\theta = 0, \pi$), $\sin \theta$ is set to be 10^{-8} when $\sin \theta < 10^{-8}$. As shown later in Section 4, such a consideration is acceptable in practical ray-tracing and GRRT calculations.

2.2. Initial Conditions

For an observer who receives the ray at an inclination angle θ_{obs} and radial position $r_{\text{obs}} = \infty$, the celestial coordinates in the observer's image frame, (α, β) , are calculated as (Chandrasekhar 1983)

$$\alpha = L_z \csc \theta_{\text{obs}}, \quad (16)$$

$$\beta = \sqrt{Q/E^2 + a^2 \cos^2 \theta_{\text{obs}} - L_z^2 \cot^2 \theta_{\text{obs}}}, \quad (17)$$

where β is also equal to the initial value of $-p_\theta$. However, we wish to define an observer at some arbitrary position (r, θ, ϕ) in space (not necessarily infinitely removed from the black hole). Such an approach has several advantages, including reducing the time needed to integrate the geodesic from the observer to the black hole. The initial conditions of a ray arriving in this observer's image plane are calculated as follows.

The observer grid is constructed as a left-handed rectangular coordinate system with the z -axis oriented toward the black hole center. The observer's axes are denoted by $\bar{\mathbf{x}} \equiv (x, y, z)^T$. The black hole coordinate system is right-handed, rectangular, and denoted by $\bar{\mathbf{x}}' \equiv (x', y', z')^T$. The observer is located at a distance r_{obs} from the black hole center, at an angle θ_{obs} from the positive black hole z' -axis (coinciding with the spin axis) and at an angle ϕ_{obs} with respect to the black hole's x' -axis. While the value of ϕ_{obs} is arbitrary, since the Kerr metric does not depend on ϕ , there are situations where specifying the observer's azimuthal position is important. For instance, in

time-dependent radiation transfer calculations or imaging fully three-dimensional anisotropic accretion flows such as those found in state-of-the-art GRMHD simulations (e.g., McKinney et al. 2013), one may wish to resolve particular local and perhaps transient features, e.g., outflows, magnetic reconnection events, and shocks.

It is assumed that the observer's image plane is a two-dimensional grid with zero curvature and that all rays received by the observer arrive perpendicular to this grid. Close to the black hole, spacetime curvature becomes significant and the observer image plane must possess some curvature-dependent distortion, which would in turn distort the calculated image. Rays would no longer arrive perpendicular to the image plane. To image a black hole closer to the event horizon would require defining an appropriate orthonormal tetrad basis in which to place the observer (e.g., Bardeen et al. 1972; Marck 1996). Since we concern ourselves only with calculating what a distant observer would actually observe, we place our observer at a distance of $10^3 r_g$ from the black hole. At this distance the deviation of geodesics from Minkowski spacetime is smaller than the numerical precision used to integrate the geodesic itself. Therefore, the spacetime may be taken as Euclidean, and we can safely employ the reverse ray-tracing method under the aforementioned assumptions.

In order to determine the initial conditions of rays starting on the observer's grid, the observer coordinate system $\bar{\mathbf{x}}$ must be transformed into the black hole coordinate system $\bar{\mathbf{x}}'$. This may be accomplished through the following series of transformations: (i) rotate clockwise by $(\pi - \theta_{\text{obs}})$ about the x -axis (\mathbf{R}_x), (ii) rotate clockwise by $(2\pi - \phi_{\text{obs}})$ about the z -axis (\mathbf{R}_z), (iii) reflect in the plane $y = x$ ($\mathbf{A}_{y=x}$), (iv) translate $\bar{\mathbf{x}}$ so that the origins of both coordinate systems coincide ($\mathbf{T}_{\mathbf{x} \rightarrow \mathbf{x}'}$). This may be calculated as follows:

$$\begin{aligned} \mathbf{x}' &= \mathbf{A}_{y=x} \mathbf{R}_z (2\pi - \phi_{\text{obs}}) \mathbf{R}_x (\pi - \theta_{\text{obs}}) \mathbf{x} + \mathbf{T}_{\mathbf{x} \rightarrow \mathbf{x}'} \\ &= \begin{pmatrix} \mathcal{D}(y, z) \cos \phi_{\text{obs}} - x \sin \phi_{\text{obs}} \\ \mathcal{D}(y, z) \sin \phi_{\text{obs}} + x \cos \phi_{\text{obs}} \\ (r_{\text{obs}} - z) \cos \theta_{\text{obs}} + y \sin \theta_{\text{obs}} \end{pmatrix}, \end{aligned} \quad (18)$$

where

$$\mathcal{D}(y, z) \equiv (\sqrt{r_{\text{obs}}^2 + a^2} - z) \sin \theta_{\text{obs}} - y \cos \theta_{\text{obs}}. \quad (19)$$

The transformation from Cartesian coordinates to Boyer–Lindquist coordinates is given by

$$r = \sqrt{\frac{w + \sqrt{w^2 + 4a^2 z'^2}}{2}}, \quad (20)$$

$$\theta = \arccos\left(\frac{z'}{r}\right), \quad (21)$$

$$\phi = \text{atan2}(y', x'), \quad (22)$$

where $w \equiv x'^2 + y'^2 + z'^2 - a^2$. Substituting the components of Equation (18) into Equations (20)–(22) gives the initial (r, θ, ϕ) conditions for a photon on the observer grid.

Next, the initial velocities of the ray must be determined. Each ray arrives perpendicular to the image plane, moving parallel to the z -axis; hence, we set $(\dot{x}, \dot{y}, \dot{z}) = (0, 0, 1)$. Subsequent differentiation of Equation (18) yields the

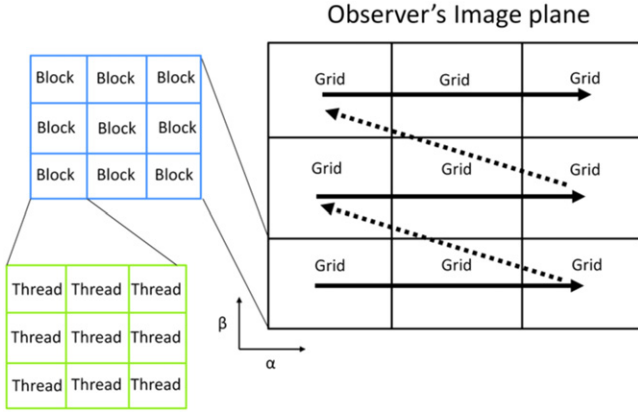


Figure 1. Mapping the threads to pixels of the image, as discussed in Section 3.

Cartesian components of the ray's velocity in black hole coordinates:

$$\dot{x}' = \begin{pmatrix} -\sin \theta_{\text{obs}} \cos \phi_{\text{obs}} \\ -\sin \theta_{\text{obs}} \sin \phi_{\text{obs}} \\ -\cos \theta_{\text{obs}} \end{pmatrix}. \quad (23)$$

Finally, to obtain the ray's velocity components in Boyer–Lindquist coordinates, we differentiate Equations (20)–(22) with respect to affine parameter, solve for $(\dot{r}, \dot{\theta}, \dot{\phi})$, and substitute for Equation (23). Upon simplification this yields

$$\dot{r} = -\frac{r\mathcal{R} \sin \theta \sin \theta_{\text{obs}} \cos \Phi + \mathcal{R}^2 \cos \theta \cos \theta_{\text{obs}}}{\Sigma}, \quad (24)$$

$$\dot{\theta} = \frac{r \sin \theta \cos \theta_{\text{obs}} - \mathcal{R} \cos \theta \sin \theta_{\text{obs}} \cos \Phi}{\Sigma}, \quad (25)$$

$$\dot{\phi} = \frac{\sin \theta_{\text{obs}} \sin \Phi}{\mathcal{R} \sin \theta}, \quad (26)$$

where $\mathcal{R} \equiv \sqrt{r^2 + a^2}$ and $\Phi \equiv (\phi - \phi_{\text{obs}})$. We now have initial conditions for $(r, \theta, \phi, t, p_r, p_\theta)$.

2.3. Radiative Transfer

To compute the jet emission and image in the observers' reference frame, one can use the ray-tracing scheme outlined in Section 2.1 to trace the received ray backward in time with initial conditions as described in Section 2.2. The frequency shift is related to the plasma motion and varies from point to point along the ray; see Equation (2). Consequently, a frequency correction from the observed frequency to the local frequency is required at every point along the ray, because physical processes take place in the local comoving frame.

Along each ray the covariant GRRT equation may be written as

$$\frac{d\mathcal{I}}{d\tau_\nu} = -\mathcal{I} + \frac{\eta}{\chi}, \quad (27)$$

where the Lorentz-invariant intensity (\mathcal{I}) is related to the specific intensity (I) as $\mathcal{I} = I_\nu/\nu^3 = I_{\nu_0}/\nu_0^3$, where ν is the frequency of radiation. The invariant absorption coefficient (χ) and invariant emission coefficient (η) are given by $\chi = \nu\alpha_\nu$ and $\eta = j_\nu/\nu^2$, respectively, where α_ν and j_ν are, respectively,

the specific emission and absorption coefficient evaluated at a frequency ν . The optical depth of the medium at a given frequency ν is denoted by τ_ν .

Defining the source function $\mathcal{S} \equiv \eta/\chi$, Equation (27) may be directly integrated, yielding

$$\mathcal{I}(\tau_\nu) = \mathcal{I}_0 e^{-\tau_\nu} + \int_{\tau_0}^{\tau_\nu} \mathcal{S}(\tau'_\nu) e^{-(\tau_\nu - \tau'_\nu)} d\tau'_\nu, \quad (28)$$

where the optical depth τ_ν is given by

$$\tau_\nu(\lambda) = -\int_{\lambda_0}^{\lambda} \alpha_{0,\nu}(\lambda') k_\alpha u^\alpha|_\lambda d\lambda', \quad (29)$$

and λ is the affine parameter.

By combining Equations (28) and (29), the solution of the radiative transfer equation can be reduced to two decoupled differential equations (Younsi et al. 2012)

$$\frac{d\tau}{d\lambda} = \gamma^{-1} \alpha_{0,\nu}, \quad (30)$$

$$\frac{d\mathcal{I}}{d\lambda} = \gamma^{-1} \left(\frac{j_{0,\nu}}{\nu^3} \right) \exp(-\tau), \quad (31)$$

which are easily integrated along with the geodesic as the ray is propagated backward in time.

3. GPU SCHEME

GPUs enable high multithreading and are designed for massively parallel computation.⁶ Using a mapping of one CUDA thread to one pixel, the method of parallel computation done by CUDA can be illustrated in Figure 1.

In this analogy, the observer's image plane is composed of $N_\alpha \times N_\beta$ (hereafter assume $N_\alpha = N_\beta = N$) pixels in the α and β directions and is decomposed into multiple *Grids*. Each *Grid* is subdivided into a two-dimensional hierarchy of blocks and threads, and a CUDA kernel function run on each GPU is performed, *Grid* by *Grid*, until the entire image plane is covered. To ensure that there are enough threads to cover an image composed of N^2 pixels, the number of *Grids* in each direction (α, β) must satisfy the following condition:

$$k_i = \left\lceil \frac{N}{\text{Dim}_i^{\text{Block}} \times \text{Dim}_i^{\text{Grid}}} \right\rceil. \quad (32)$$

Note again that Dim^{Grid} represents the number of blocks, $\text{Dim}^{\text{Block}}$ represents the number of threads, and $i = (\alpha, \beta)$ is the dimension index. The notation $\lceil k \rceil$ is the smallest integer not less than k (the “ceiling” function).

The global memory on each GPU is allocated by the CUDA command `cudaMalloc()`, which is composed of two parts. First, for each working thread, some global memory is required for saving specific variables until each thread finishes its computation. Let P be the corresponding memory for each thread; then an amount $P \times T$ of memory is required, where T is the number of working threads. Second, part of the global memory must be assigned for saving the computed data that will later be returned to the CPU. For each pixel, if Q is the amount of information for each pixel to be later transported

⁶ Physically, a GPU is built in an array of streaming multiprocessors (SMs). Each SM has N cores, or streaming processors (SP), and each SP is massively threaded. Typical CUDA-capable GPUs consist of several hundred to several thousand cores.

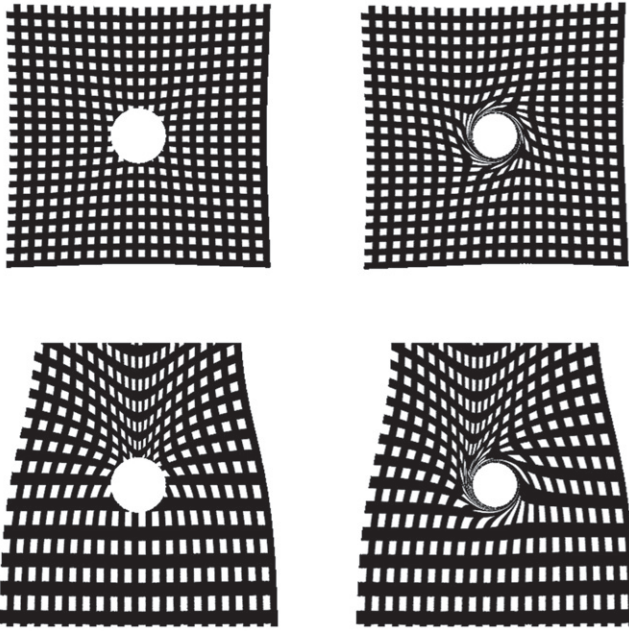


Figure 2. Projection of uniform grids in the observer’s frame (α, β) onto the equatorial plane (x, y) of a nonrotating ($a = 0$; left panels) and rapidly rotating ($a = 0.95$; right panels) black hole. The observer inclination angle is set to be $i \simeq 0^\circ$ (top panels) and $i = 60^\circ$ (bottom panels). This figure recovers the results of previous studies, e.g., Figure 2 of Schnittman & Bertschinger (2004) and Figure 3 of Dexter & Agol (2009). The horizontal and vertical axes correspond to the $-\alpha$ and $-\beta$ direction, respectively, as adopted by the aforementioned authors.

from CUDA to the host CPU, then $Q \times N^2$ of memory is required for the entire image. By manipulating the value of P and Q , the user can easily design their own task and assign multiple outputs (e.g., $\alpha, \beta, \gamma, l_p, \dots$) to each pixel of the image for subsequent calculations.

4. BENCHMARK ASTROPHYSICAL CALCULATIONS

4.1. Grid Projection

In this section we perform several standard benchmark tests of the performance of *Odyssey*. First, we calculate the projection of an evenly spaced rectangular grid in the observer’s image plane (α, β) onto the equatorial plane of a Kerr black hole (see Schnittman & Bertschinger 2004; Dexter & Agol 2009; Chan et al. 2013) via the following coordinate transformation:

$$x = \sqrt{r^2 + a^2} \cos \phi, \quad (33)$$

$$y = \sqrt{r^2 + a^2} \sin \phi. \quad (34)$$

This is illustrated in Figure 2, which recovers the results of the aforementioned previous studies, e.g., Figure 2 of Schnittman & Bertschinger (2004) and Figure 3 of Dexter & Agol (2009). Note that the horizontal and vertical axis in Figure 2 corresponds to the $-\beta$ and $-\alpha$ direction, as adopted by those authors.

In Figure 3 we also consider a more illustrative example by projecting an evenly spaced grid in the equatorial plane (x, y) of the black hole onto the observer’s image plane (α, β) for nonspinning ($a = 0$, left) and spinning ($a = 0.95$, right) black holes. The observer’s image frame is shown in Figure 3;

therefore, the horizontal and vertical axes simply correspond to the α and β directions, respectively. The bending of rays emitted around the black hole results in an expansion of the grid on the equatorial plane, as can be clearly seen in the face-on case (top panels). Compared with the grid in front of the black hole, the thickness of the grid behind is magnified by gravitational lensing (bottom panels). When the black hole is rotating (right panels), the rays are dragged by the rotation of the spacetime around the black hole (the frame-dragging effect). This effect results in the grid being distorted in an anticlockwise direction (i.e., in the same direction as the rotation of the black hole).

4.2. Black Hole Shadows

The appearance of the black hole shadow, which is the shadow cast by the photon capture surface of the black hole (and not the event horizon itself), can be determined by simply plotting all rays that are captured by the black hole. The black hole shadow profile is associated with the black hole spin and inclination angle of the observer, as studied in, e.g., Falcke et al. (2000), Takahashi (2004), and Chan et al. (2013). In Figure 4 we plot the black hole shadow for cases of a Schwarzschild ($a = 0$) and a Kerr ($a = 0.998$) black hole, when the observer is in the $x-y$ plane, i.e., $i = 90^\circ$. For $a = 0$, the shape of the black hole shadow, or more accurately the shadow of the photon capture region, follows the analytic description $\alpha^2 + \beta^2 = 27$. The analytic solution for the general case, $a \neq 0$, is discussed in, e.g., Grenzebach et al. (2014) and Abdujabbarov et al. (2015). The analytic solutions (dashed red curves) are plotted in Figure 4 for comparison.

4.3. Keplerian Thin Disk

We now consider an infinitesimally thin, Keplerian disk around a rotating black hole. The inner edge r_{in} of the disk is located at the marginally stable orbit (Bardeen et al. 1972), r_{ISCO} (innermost stable circular orbit). The outer edge, r_{out} , is assumed to be located at $50r_g$. For an observer inclination angle satisfying $\cos \theta_{\text{obs}} = 0.25$, the relative energy/frequency shift, γ , is shown in Figure 5 as a series of solid blue (blueshift) and solid red (redshift) contours. The approaching (left) side is blueshifted ($\gamma > 1$), and the receding (right) side is redshifted ($\gamma < 1$). The dotted line denotes the region of zero redshift. The radial contours ($r = \text{constant}$; see the grid profile in the bottom left panel of Figure 3) are also shown as a series of concentric solid rings.

We next compute spectra from a multitemperature disk as described in Novikov & Thorne (1973) and Page & Thorne (1974). The relation between the disk flux, F , and its effective temperature, T_{eff} , may be written as

$$F(r) = \frac{3\dot{M}}{8\pi r^2} \frac{M}{r} f(r) = \sigma T_{\text{eff}}^4(r), \quad (35)$$

where M is the mass of the black hole, \dot{M} is the accretion rate, $f(r)$ is a correction factor related to the inner boundary of the disk and relativistic effects, and σ is the Stefan–Boltzmann constant. Under the assumption that the viscous stress vanishes at r_{ISCO} , the value of $f(r)$ (and hence $F(r)$) also vanishes there.

The spectrum from a disk with $a = 0.999$, $i = 85^\circ$, $M = 10 M_\odot$, $\dot{M} = 10^{19} \text{ g s}^{-1}$, and distance $D = 10 \text{ kpc}$ is shown in Figure 6. The spectrum is plotted in terms of photon number flux density, f_E (photons $\text{cm}^{-2} \text{ s}^{-1} \text{ keV}^{-1}$), instead of

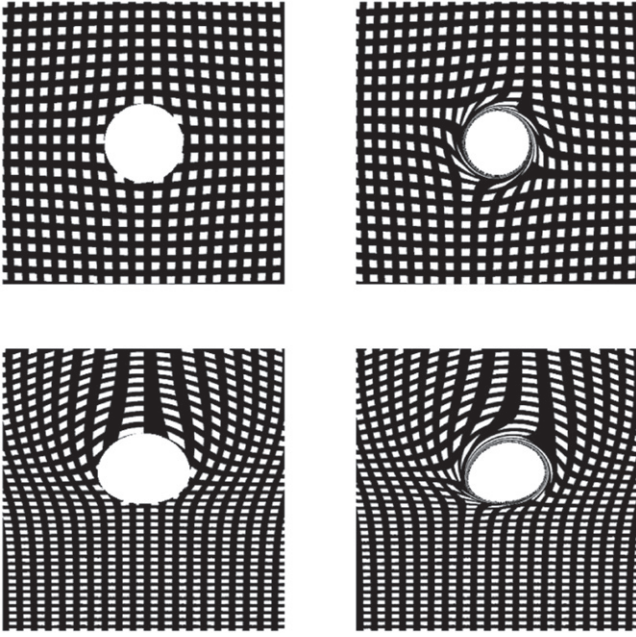


Figure 3. Projection of uniform grids in the equatorial plane (x, y) onto the observer's image plane (α, β), with the same parameters as used in Figure 2. The size of the field of view is $10 r_g \times 10 r_g$. The horizontal and vertical axes correspond to the α and β directions, respectively (cf. Figure 2).

energy flux density, F_ν ($\text{erg cm}^{-2} \text{s}^{-1} \text{Hz}^{-1}$).⁷ We fix the disk inner radius at $r_{\text{in}} = r_{\text{ISCO}}$ and vary the outer radius of the disk as $r_{\text{out}} = 50 r_g$ and $500 r_g$ to demonstrate how the turnover at lower energies depends on the disk's outer radius. Conversely, the high-energy part of the spectrum manifests predominantly from regions with higher T_{eff} (i.e., from the inner edge of the disk) and therefore remains almost completely unchanged.

Incorporated into the X-ray spectral fitting package XSPEC, the KERRBB model is a multitemperature model of a geometrically thin, steady accretion disk (see Li et al. 2005). Disk self-irradiation, torque at the inner boundary of the disk, and limb darkening may be included within KERRBB by adjusting the appropriate model parameters. For comparison, the result of the KERRBB model calculation with similar parameters is plotted as a dashed line in Figure 6 (for details, see the figure caption).

4.4. Keplerian Hot Spot

A photon received by a distant observer at an observer time t_{obs} is emitted at a coordinate time, $t_{\text{em}} = t_{\text{obs}} - \Delta t$, where Δt is the time taken for the photon to travel from its point of emission to its point of reception on the observer image plane (see Schnittman & Bertschinger 2004).

The top panel of Figure 7 shows the spectrogram (time-dependent spectrum) of the direct emission from an orbiting hot spot rotating in a clockwise direction around the central black hole. The center of the hot spot, $\mathbf{x}_{\text{spot}}(t)$, orbits on the equatorial plane of a nonrotating black hole at r_{ISCO} , observed at $i = 60^\circ$. The emissivity is assumed to be a function of the distance (d) from the hot spot center, where $d = |\mathbf{x} - \mathbf{x}_{\text{spot}}(t)|$. Within $z > 0$ and $d < 4 r_{\text{spot}}$, $r_{\text{spot}} = 0.5 r_g$ and the emission is modeled as Gaussian in profile, in pseudo-Cartesian

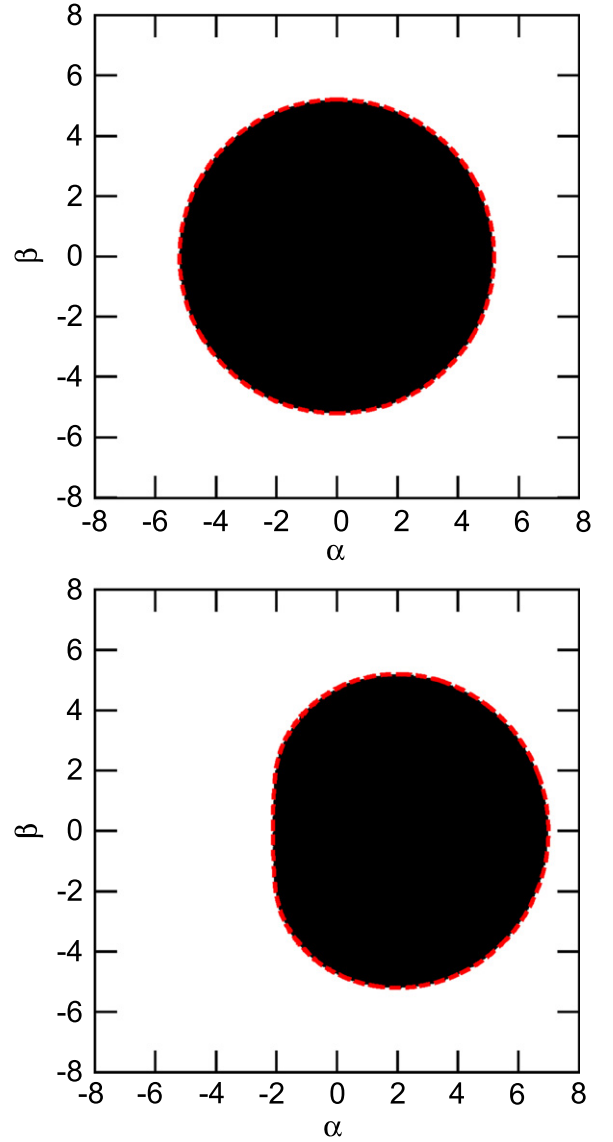


Figure 4. Black hole shadow images for black holes with spin parameters of $a = 0$ (top panel) and $a = 0.998$ (bottom panel), as viewed along the equatorial plane. The dashed red curves indicate the analytic solution for the photon ring. The image resolution is 512×512 pixels.

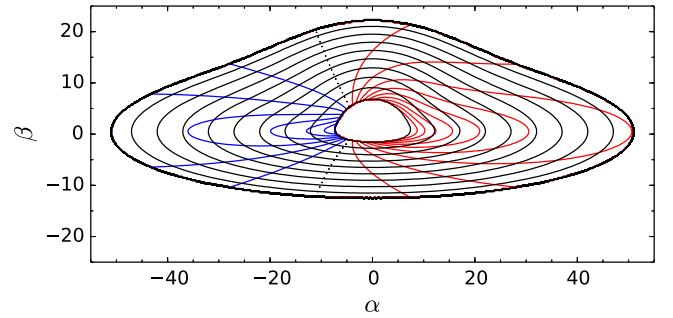


Figure 5. Energy shift of a geometrically thin, Keplerian disk around a nonrotating black hole, with $\cos \theta_{\text{obs}} = 0.25$. The disk is rotating anticlockwise (i.e., in the ϕ -direction). Only rays emitted directly from the disk are considered. The same contour values used in Figure 5.2 of Agol (1997) are adopted: the radial contour plot shows $r/r_g = 6.1, 11, 16, \dots, 41, 46, 50$, and the redshift contour plot shows $\gamma = 0.55, 0.6, \dots, 0.95$ (red solid lines), 1.0 (black dotted line), 1.05, ..., 1.25, 1.3 (blue solid lines). This figure recovers Figure 5.2 of Agol (1997).

⁷ $f_E = 1.51 \times 10^{26} F_\nu / E$, where E is in units of keV.

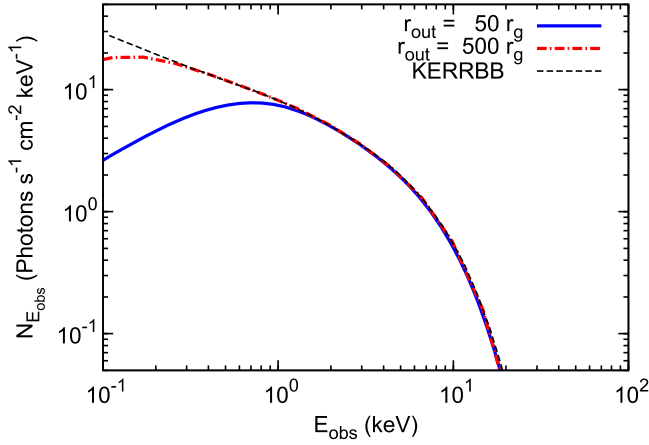


Figure 6. Spectra from a relativistic multitemperature disk described by Novikov & Thorne (1973), with $a = 0.999$, $i = 85^\circ$, and varying outer disk radius r_{out} . The KERRBB model profile (dashed line) with similar parameters is obtained by using the KERRBB model in XSPEC with parameters [par1, par2, ..., par9] = [0, 0.999, 85, 10, 10, 10, 1, -1, -1], where the effects of torque at the inner edge of the disk, self-irradiation, and limb darkening are omitted (see <http://heasarc.nasa.gov/xanadu/xspec/manual/XSmodelKerrbb.html> for definitions of the parameters).

coordinates (Equations (33), (34), and $z = r \cos \theta$)

$$j(\mathbf{x}) \propto \exp\left(-\frac{d^2}{2r_{\text{spot}}^2}\right). \quad (36)$$

The hot spot is assumed to be optically thick, and therefore the computation is terminated once the ray intersects the hot spot surface. The shape and corresponding energy shift of the hot spot at specific orbital phases are shown in the bottom panel of Figure 7. The contribution of rays emitted from different locations of the hot spot at different local coordinate times, coupled with relativistic (e.g., length contraction and time dilation) and general relativistic (e.g., gravitational lensing) effects, results in a distorted hot spot image (see Younsi & Wu 2015). Again, the emission region is magnified when the hot spot is “behind” the black hole, farthest away from the observer. The light curve of the hot spot emission can be obtained by integrating the intensity over all frequency bins for each image, as shown in Figure 8. The spectrogram and light curve in Figure 7 recover those found in Figures 4 and 6 of Schnittman & Bertschinger (2004).

4.5. Keplerian Shell

In order to test the radiative transfer formulation, we now consider the emission from a Keplerian shell of plasma in rotation around a black hole (see Broderick & Loeb 2006). By setting $u^\theta = 0$, the remaining components of the 4-velocity of the flow follow the same description as that of a Keplerian disk as given by, e.g., Cunningham (1975). As a demonstrative calculation, we consider thermal synchrotron radiation from a distribution of relativistic electrons with an underlying relativistic Maxwellian profile. The angle-averaged emissivity coefficient for relativistic thermal synchrotron radiation may be written as (Mahadevan et al. 1996)

$$j_\nu(T) = \nu \frac{4\pi n e^2}{\sqrt{3} c K_2(1/\Theta_e)} M(x_M), \quad (37)$$

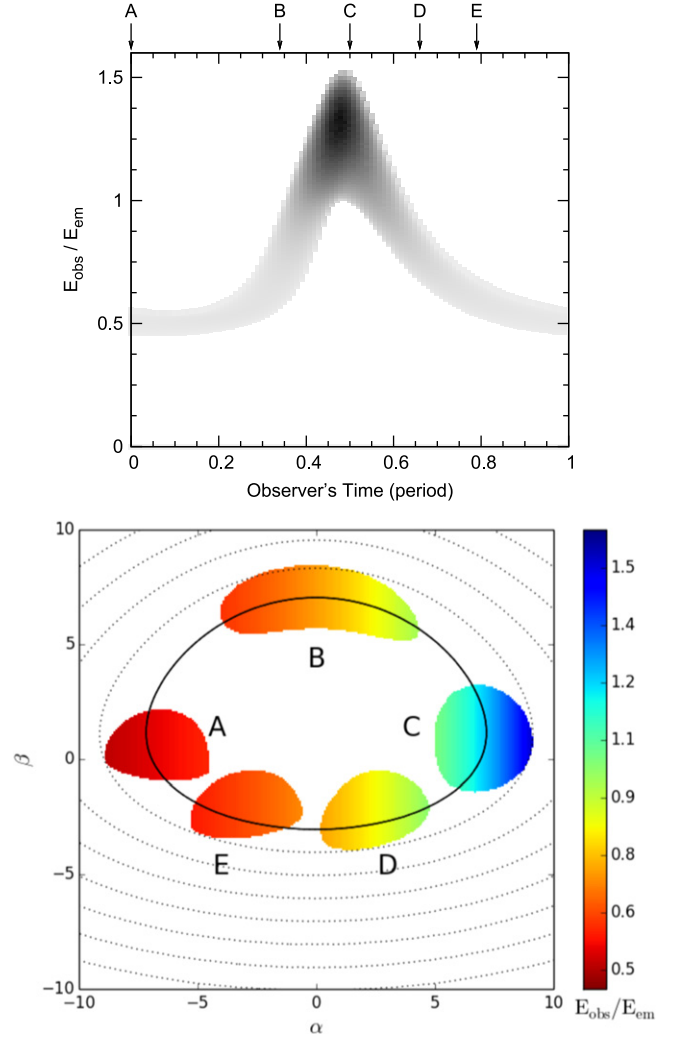


Figure 7. Top panel: spectrogram of a circular hot spot with $r_{\text{spot}} = 0.5 r_g$, orbiting a nonrotating black hole at r_{ISCO} . The observer inclination angle is $i = 60^\circ$. This result recovers Figure 4 of Schnittman & Bertschinger (2004). Bottom panel: shape and energy shift of the hot spot at specific observer times, colored by energy shift, and also indicated by arrows A–E in the top panel.

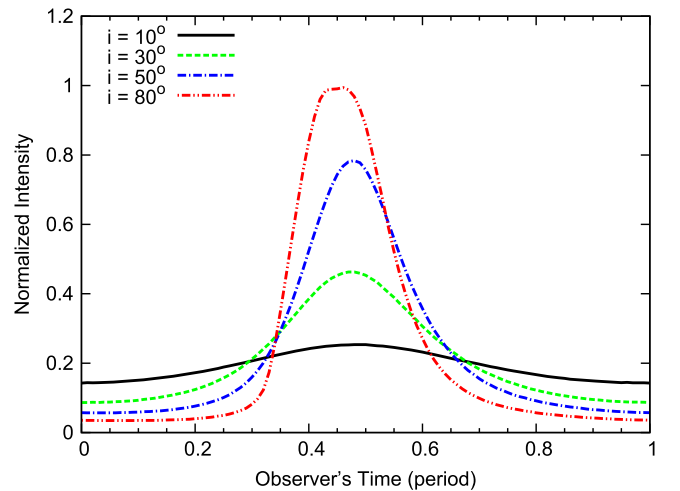


Figure 8. Light curves of the hot spot described in Figure 7 for different observer inclination angles. The intensity is normalized to the sum of total intensity over one orbital period. This result is in agreement with Figure 6 of Schnittman & Bertschinger (2004).

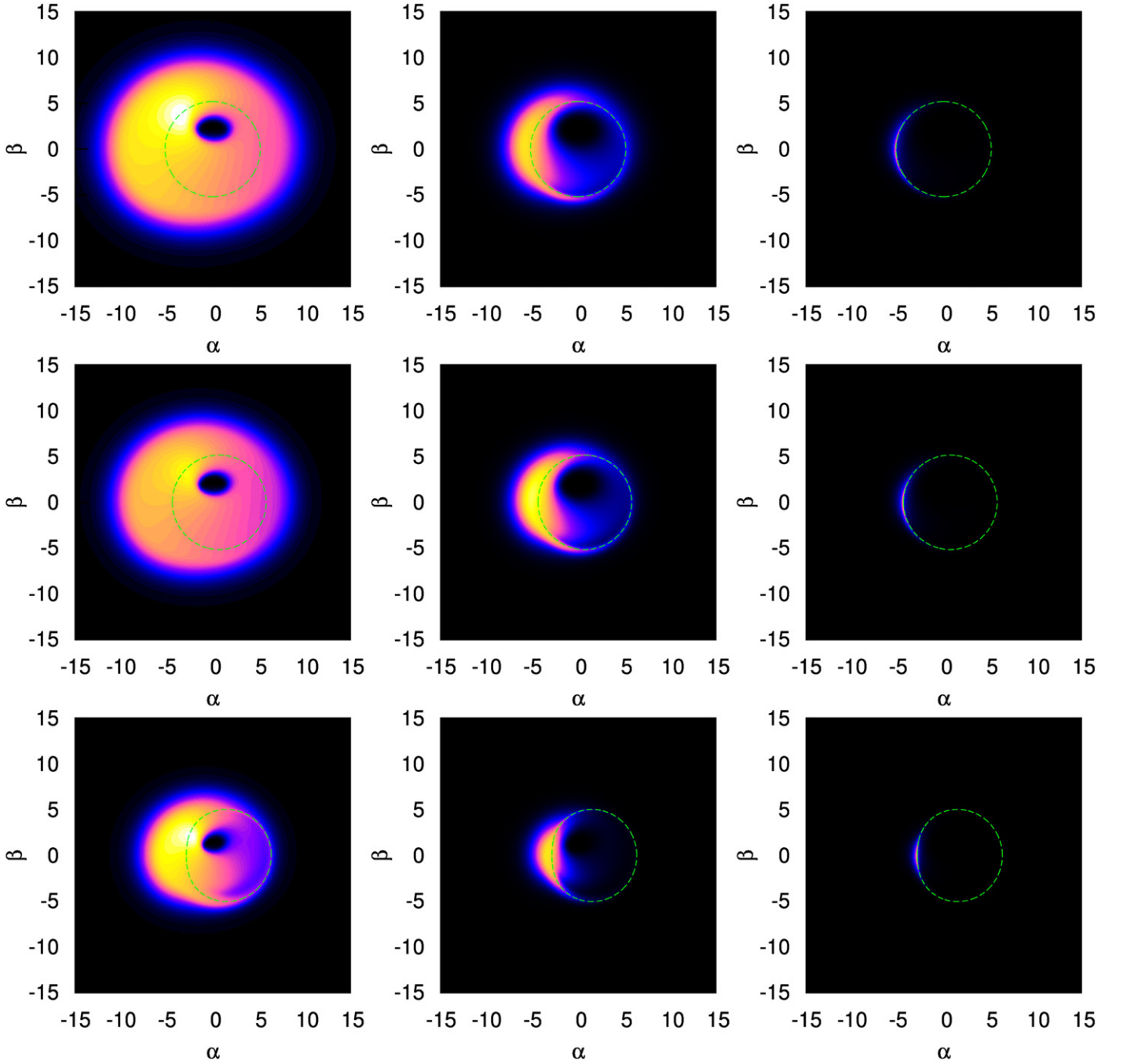


Figure 9. Thermal synchrotron radiation image of a Keplerian shell around a black hole at 150 GHz (left column), 340 GHz (central column), and 1000 GHz (right column), as viewed at an observer inclination angle of 45° . From top to bottom, $a = 0, 0.5$, and 0.998 , respectively. The analytic solutions for the photon rings are shown by dashed green curves for reference. The image intensity is plotted on a linear scale.

where

$$M(x_M) = \frac{4.0505}{x_M^{1/6}} \left(1 + \frac{0.40}{x_M^{1/4}} + \frac{0.5316}{x_M^{1/2}} \right) \exp(-1.8899x_M^{1/3}). \quad (38)$$

Here $x_M \equiv \nu/\nu_c$ and

$$\nu_c = \left(\frac{3eB}{4\pi m_e c} \right) \Theta_e^2, \quad (39)$$

where $\Theta_e \equiv k_B T/m_e c^2$ is the dimensionless electron temperature and K_n is a modified Bessel function of the second kind of order n . Since thermal synchrotron radiation is the only radiation source being considered, the absorption coefficient may be calculated via Kirchoff's law.

The Keplerian shell model has previously been used to simulate the image and spectrum of Sgr A* (e.g., Broderick & Loeb 2006; Broderick et al. 2011). We adopt the same self-similar electron number density profile and temperature profile given in Equation (1) and Table 1 of Broderick & Loeb (2006), for the cases of black hole spin

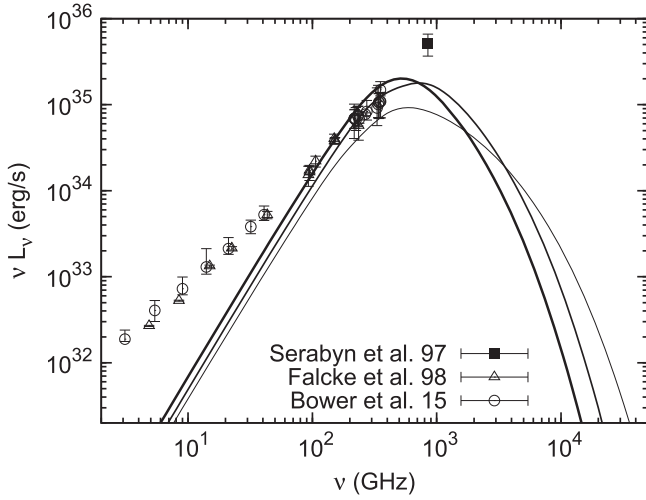


Figure 10. Corresponding spectrum of the emission from a Keplerian shell around a nonrotating black hole as illustrated in Figure 9. The cases $a = 0, 0.5$, and 0.998 are plotted as thick, medium, and thin solid lines, respectively. Observational data from Sgr A* (Serabyn et al. 1997; Falcke et al. 1998; Bower et al. 2015) are overlaid for comparison.

Table 1
Timing Performance of *Odyssey* Code

Image Size (Number of Geodesics)	Run Time ^a (ms)	Total R-K Steps	Average Time (ns/step/ photon)
2^2	56.921631	5.718000e+003	2488.703693
4^2	113.811523	2.484900e+004	286.257806
8^2	116.409309	9.876600e+004	18.416211
16^2	114.792221	3.929330e+005	1.141180
32^2	117.429955	1.548845e+006	0.074041
64^2	238.210617	6.183181e+006	0.009406
128^2	605.967957	2.468902e+007	0.001498
256^2	1873.401367	9.866115e+007	0.000290
512^2	7090.669434	3.944880e+008	0.000069
1024^2	25732.214844	1.577673e+009	0.000016

Note.

^a Time required to compute the bottom right panel of Figure 2 by one NVIDIA GeForce GTX 780 Ti graphics card with double-precision floating-point arithmetic. The GeForce GTX 780 Ti graphics card has 15 SMs and 192 cores per SM, giving 2880 CUDA cores in total.

parameters of $a = 0, 0.5$, and 0.998 . The GRRT computation is performed within a shell of outer radius $500 r_g$, with a central black hole mass of $4 \times 10^6 M_\odot$. The resulting images at 150, 340, and 1000 GHz are plotted in Figure 9. The spectrum is shown in Figure 10. It is clear that, besides the blackbody-like spectral component contributed by the thermal synchrotron emission, an additional power-law component is further needed to explain the observed data of Sgr A*. On the other hand, because thermal synchrotron dominates the total emission of the image at the frequencies of interest, Figure 9 in general shows good agreement with Figure 1 of Broderick & Loeb (2006), in which the contribution from a nonthermal electron population is also included.

5. TIMING BENCHMARK

To test the speed of *Odyssey*, we perform multiple runs on different image sizes for the same parameters as used to calculate the bottom right panel of Figure 2, a benchmark also

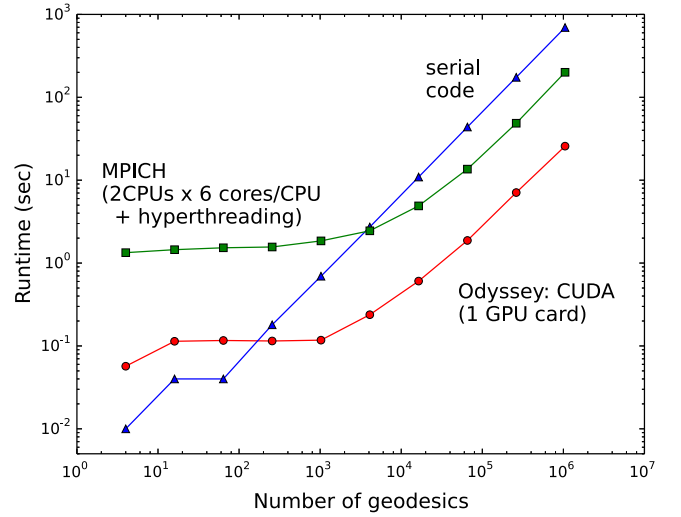


Figure 11. Comparison of the computational time required to obtain the result in the bottom right panel of Figure 2, for an image size of $2^n \times 2^n$ pixels ($n = 1, 2, \dots, 10$), using three different programming architectures: serial code (blue triangles), parallelized by MPICH (green squares), and CUDA (red circles). For small numbers of geodesics, the computational time is dominated by the time to launch the CUDA or MPICH kernel. The serial code is run on an Intel(R) Core(TM) i7-4940 K 4.00 GHz CPU. The MPICH code was run on two Intel(R) Xeon(R) E5-2620 2.00 GHz CPUs (each CPU has six cores), with hyperthreading enabled. Our GPU code (*Odyssey*) was run on a single NVIDIA GeForce GTX 780 Ti graphics card, as summarized in Table 1.

used in the GPU-based ray-tracing code, *G_{RAY}* (Chan et al. 2013). The timing benchmark was performed with a single NVIDIA GeForce GTX 780 Ti graphics card (with 2880 CUDA cores and 3 GB of GDDR5 RAM) and computed in double-precision floating-point arithmetic. In Table 1, we record the run time and total number of steps used for the Runge–Kutta method. Although the run time and number of Runge–Kutta steps depend on the accuracy required by the adaptive step size (Press et al. 1992), the average time per Runge–Kutta step, per photon, remains roughly uniform with increasing numerical precision. Because the radiative transfer integration is also performed piecewise according to these steps, the average time per integration step, per photon provides a better standard for comparison with other ray-tracing codes with different underlying numerical schemes. The value for the averaged run time per Runge–Kutta step, per photon as a function of image size is given in the last column of Table 1.

We are also interested in how GPUs can boost the computation speed compared to conventional serial CPU and parallel CPU cluster codes with the same numerical algorithm. Therefore, we apply the same algorithm described in Section 2 to a serial code and a parallel code in MPICH, which is a high-performance implementation of the MPI standard. These two codes are both written in C, using double-precision floating-point arithmetic. We compare the run-time result in Figure 11. The profile for the case of *Odyssey* and MPICH flattens when integrating a small number of geodesics because the run time is dominated by the time taken to invoke the CUDA or MPICH kernels. For small numbers of photons, the serial code outperforms the parallelized codes. However, parallel computations efficiently reduce the run time for larger numbers of geodesics. Comparing the unit price of a CPU cluster that can deliver computational results in an equal amount of time

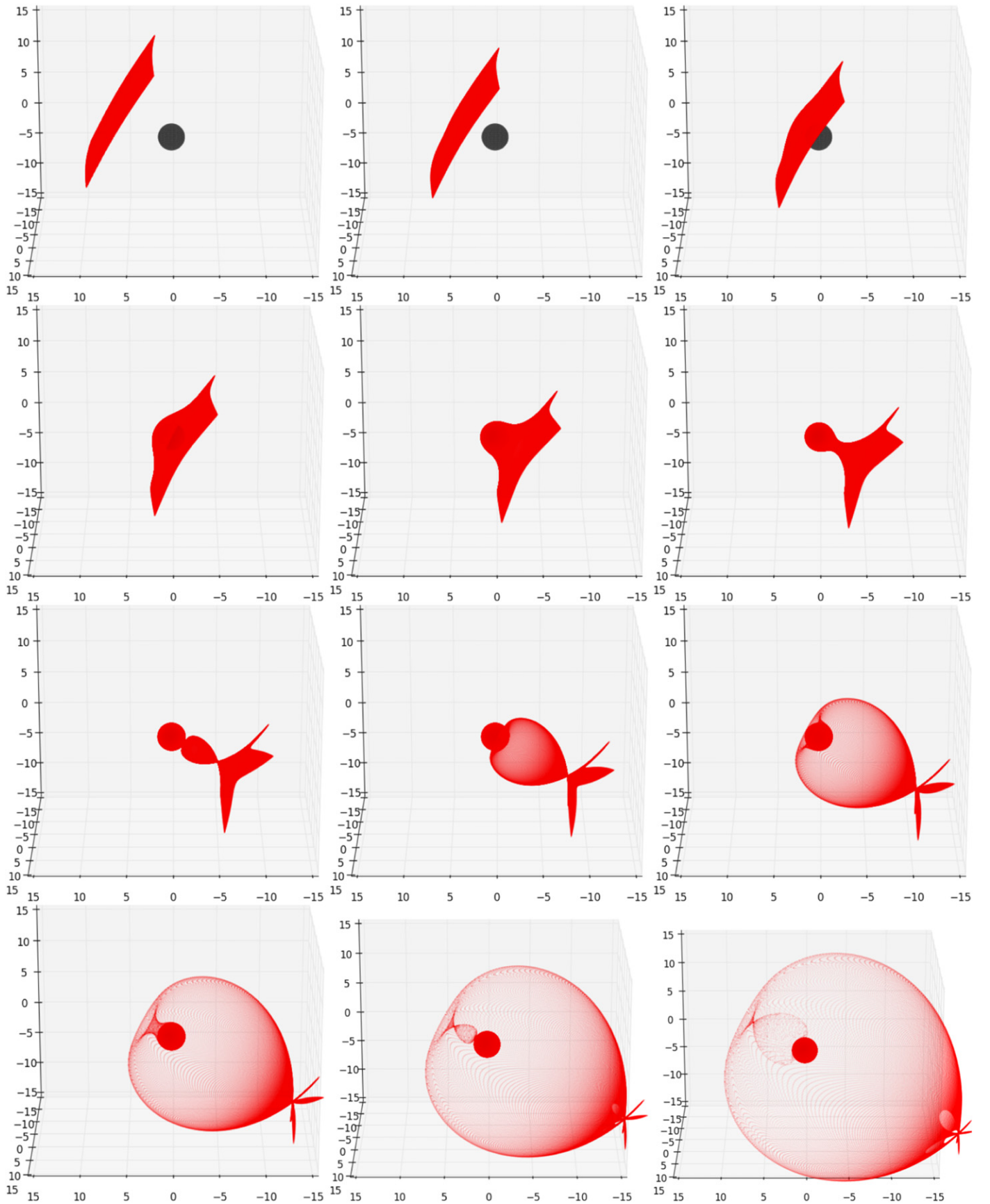


Figure 12. Evolution of photon plane (backward in time) near a nonrotating black hole at various coordinate times t , at which the emitted photons all reach the observer simultaneously. The time sequence is from left to right, top to bottom, and the time interval between snapshots is $dt = 4$.

compared to one GPU card, it is clear that for the same unit price GPUs deliver results faster.

G_{Ray} is a GPU-based ray-tracing code, based on the ray-tracing algorithm of Psaltis & Johannsen (2012) and Bauböck et al. (2012). With the same benchmarks (bottom right panel of Figure 2), *Odyssey* and G_{Ray} reveal similar profiles in run-time measures (comparing Figure 11 with Figure 4 in Chan et al. 2013). *Odyssey* also reaches similar levels of performance to those reported by G_{Ray} (~ 1 ns per photon, per time step). The average run time for *Odyssey* can be less than 1 ns when the image size is larger than 32^2 pixels.⁸

6. SUMMARY AND DISCUSSION

Odyssey is an accurate, flexible, and efficient GPU-based code for ray tracing and radiative transfer in the Kerr spacetime. Compared with G_{Ray} (Chan et al. 2013), *Odyssey* adopts a different ray-tracing algorithm and CUDA code structure, with similar performance for the ray tracing ($\lesssim 1$ ns per step, per photon). The source code for *Odyssey* is freely available (Pu et al. 2016)⁹, with two default tasks for computing Figure 5 and the top middle panel of Figure 9. Users can easily modify the source code to suit their needs and efficiently perform GRRT calculations on their computer with a CUDA-capable GPU graphics card.

An immediate and important application of *Odyssey* is the rapid computation of images, spectra, and light curves of different black hole accretion and/or jet models, provided either semi-analytically (e.g., Broderick & Loeb 2006; Broderick & Loeb 2009; Pu et al. 2015) or numerically, e.g., from simulations of current state-of-the-art general relativistic magnetohydrodynamic (GRMHD) simulation codes, such as HARM3D (Noble et al. 2009; Mościbrodzka et al. 2014, 2016) and RAISHIN (Mizuno et al. 2006). Post-processing GRMHD simulation data for GRRT calculations has been considered in several studies to calculate the simulated spectrum and VLBI images from Sgr A* and M87 (e.g., Mościbrodzka et al. 2009; Dexter et al. 2012; Mościbrodzka & Falcke 2013). The works of Chan et al. (2015a, 2015b) provide a good example of how GPUs can help to accelerate these computations significantly, one important application of which is to a time series of GRMHD data, in particular to extract the observed spectra and light curves from many different accretion models both accurately and efficiently. Such calculations are important for calculating the time-dependent emission from accretion onto black holes and warrant a detailed separate study.

Time-dependent GRRT involving a full consideration of the light-crossing time of each ray is needed when the light-crossing timescale is comparable with the dynamical timescale of the system, for example, near the black hole event horizon or strong shock regions. Starting from the observer’s frame, the photon plane moving backward in time with the same increment Δt resembles a plane on which all emitted photons will reach the observer simultaneously. We therefore term this plane the *frozen photon plane*. At large distances, the *frozen photon plane* resembles a conventional Euclidean plane because length-contraction and time-dilation effects are negligible far from the black hole. However, closer to the black hole, the *frozen photon plane* becomes distorted, as

demonstrated in Figure 12. Although the GRRT integration along each ray between successive *frozen photon planes* requires only the data within two planes, near the black hole every photon on the same plane requires different numbers of Runge–Kutta steps to reach the final photon plane (i.e., the distant observer). This is essentially the “fast light” approximation, whereby it is assumed that all rays from everywhere within the computational time, at that particular observer time, arrive simultaneously. This enables ray tracing on static time slices of GRMHD simulation data and is trivially parallelizable.

Relaxing this approximation and considering the dynamical evolution of the medium as the ray propagates through it, along with the arrival time delays between neighboring rays, is a significant computational challenge. Owing to the limited amount of memory on board each GPU and the size of even modest 3D GRMHD data, post-processing GRMHD simulation results with time-dependent GRRT is not currently feasible with GPUs. However, recent progress in parallel computing, most notably with hybrid CPU and GPU programming architectures like OpenCL and CUDA-Aware MPI, offers several possibilities to approach this problem. Through sufficient load balancing of the computation between CPUs and GPUs, one can in principle access the large amounts of RAM required for this task. We leave this to a future update of *Odyssey*.

Finally, we note that *Odyssey* is developed in the Microsoft Visual Studio environment, and as such it is possible to combine the *Odyssey* algorithm with DirectX for visualizing the propagation of rays in the Kerr spacetime. Together with the Graphical User Interface, powered by DirectX, we also present a public software package and educational tool *Odyssey-Edu*, for demonstrating null geodesics around a Kerr black hole.¹⁰

We thank the anonymous referee for useful comments and suggestions that helped improve the manuscript. We thank Steven V. Fuerst for sharing his code, Hsi-Yu Schive for helpful discussions about GPU programming, and K. Akiyama for discussions on spectral data from Sgr A*. H.-Y.P. and Z.Y. are grateful for numerous helpful and stimulating discussions with Kinwah Wu. H.-Y.P. also thanks members of the GLT team for their support and encouragement. H.-Y.P. is supported by the Ministry of Science and Technology (MOST) of Taiwan under the grant MOST 103-2112-M-001-038-MY2. Z.Y. is supported by an Alexander von Humboldt Fellowship and acknowledges support from the ERC Synergy Grant “BlackHoleCam—Imaging the Event Horizon of Black Holes” (Grant 610058). S.-J.Y. acknowledges support from the National Research Foundation of Korea to the Centre for Galaxy Evolution Research (No. 2010-0027910), from the Mid-Career Researcher Program (No. 2015-008049) through the National Research Foundation (NRF) of Korea, and from the Yonsei University Future-leading Research Initiative of 2014–2015. This research has made use of NASA’s Astrophysics Data System.

REFERENCES

- Abdujabbarov, A. A., Rezzolla, L., & Ahmedov, B. J. 2015, *MNRAS*, **454**, 2423
 Agol, E. 1997, PhD thesis, Univ. California
 Akiyama, K., Lu, R.-S., Fish, V. L., et al. 2015, *ApJ*, **807**, 150

⁸ The run time varies from GPU to GPU according to the number of CUDA cores, as well as the number of GPUs.

⁹ <https://github.com/hungyip/odyssey>

¹⁰ Available at <https://odysseyedu.wordpress.com/>.

- Bardeen, J. M., Press, W. H., & Teukolsky, S. A. 1972, *ApJ*, **178**, 347
- Bauböck, M., Psaltis, D., Özel, F., & Johannsen, T. 2012, *ApJ*, **753**, 175
- Bower, G. C., Markoff, S., Dexter, J., et al. 2015, *ApJ*, **802**, 69
- Broderick, A. E., & Loeb, A. 2006, *ApJL*, **636**, L109
- Broderick, A. E., Fish, V. L., Doeleman, S. S., & Loeb, A. 2011, *ApJ*, **735**, 110
- Broderick, A. E., & Loeb, A. 2005, *MNRAS*, **363**, 353
- Broderick, A. E., & Loeb, A. 2009, *ApJ*, **697**, 1164
- Chan, C.-K., Psaltis, D., & Oumlzel, F. 2013, *ApJ*, **777**, 13
- Chan, C.-K., Psaltis, D., Özel, F., Narayan, R., & Sadowski, A. 2015a, *ApJ*, **799**, 1
- Chan, C.-K., Psaltis, D., Özel, F., et al. 2015b, *ApJ*, **812**, 103
- Chandrasekhar, S. 1983, *The Mathematical Theory of Black Holes* (Oxford: Oxford Univ. Press)
- Cunningham, C. T. 1975, *ApJ*, **202**, 788
- Dexter, J., & Agol, E. 2009, *ApJ*, **696**, 1616
- Dexter, J., McKinney, J. C., & Agol, E. 2012, *MNRAS*, **421**, 1517
- Doeleman, S. S., Fish, V. L., Schenck, D. E., et al. 2012, *Sci*, **338**, 355
- Doeleman, S. S., Weintraub, J., Rogers, A. E. E., et al. 2008, *Natur*, **455**, 78
- Fabian, A. C., Iwasawa, K., Reynolds, C. S., & Young, A. J. 2000, *PASP*, **112**, 1145
- Falcke, H., Goss, W. M., Matsuo, H., et al. 1998, *ApJ*, **499**, 731
- Falcke, H., Melia, F., & Agol, E. 2000, *ApJL*, **528**, L13
- Fuerst, S. V., & Wu, K. 2004, *A&A*, **424**, 733
- Fuerst, S. V., & Wu, K. 2007, *A&A*, **474**, 55
- Fukumura, K., & Kazanas, D. 2008, *ApJ*, **679**, 1413
- Grenzebach, A., Perlick, V., & Lämmerzahl, C. 2014, *PhRvD*, **89**, 124004
- Inoue, M., Algaba-Marcos, J. C., Asada, K., et al. 2014, *RaSc*, **49**, 564
- Levin, J., & Perez-Giz, G. 2008, *PhRvD*, **77**, 103005
- Li, L.-X., Zimmerman, E. R., Narayan, R., & McClintock, J. E. 2005, *ApJS*, **157**, 335
- Mahadevan, R., Narayan, R., & Yi, I. 1996, *ApJ*, **465**, 327
- Marck, J.-A. 1996, *CQGra*, **13**, 393
- McKinney, J. C., Tchekhovskoy, A., & Blandford, R. D. 2013, *Sci*, **339**, 49
- Mizuno, Y., Nishikawa, K.-I., Koide, S., Hardee, P., & Fishman, G. J. 2006, arXiv:astro-ph/0609004
- Mościbrodzka, M., & Falcke, H. 2013, *A&A*, **559**, L3
- Mościbrodzka, M., Falcke, H., & Shiokawa, H. 2016, *A&A*, **586**, A38
- Mościbrodzka, M., Falcke, H., Shiokawa, H., & Gammie, C. F. 2014, *A&A*, **570**, A7
- Mościbrodzka, M., Gammie, C. F., Dolence, J. C., Shiokawa, H., & Leung, P. K. 2009, *ApJ*, **706**, 497
- Müller, A., & Camenzind, M. 2004, *A&A*, **413**, 861
- Noble, S. C., Krolik, J. H., & Hawley, J. F. 2009, *ApJ*, **692**, 411
- Novikov, I. D., & Thorne, K. S. 1973, in *Black Holes (Les Astres Occlus)*, ed. C. Dewitt & B. S. Dewitt (New York: Gordon and Breach), 343
- Page, D. N., & Thorne, K. S. 1974, *ApJ*, **191**, 499
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical Recipes in C. The Art of Scientific Computing* (2nd ed.; Cambridge: Cambridge Univ. Press)
- Psaltis, D., & Johannsen, T. 2012, *ApJ*, **745**, 1
- Pu, H.-Y., Nakamura, M., Hirotani, K., et al. 2015, *ApJ*, **801**, 56
- Pu, H.-Y., Yun, K., Younsi, Z., & Yoon, S.-J. 2016, *Odyssey*, v1.0, doi:10.5281/zenodo.45186
- Rauch, K. P., & Blandford, R. D. 1994, *ApJ*, **421**, 46
- Reynolds, C. S., Young, A. J., Begelman, M. C., & Fabian, A. C. 1999, *ApJ*, **514**, 164
- Schnittman, J. D., & Bertschinger, E. 2004, *ApJ*, **606**, 1098
- Schnittman, J. D., & Rezzolla, L. 2006, *ApJL*, **637**, L113
- Serabyn, E., Carlstrom, J., Lay, O., et al. 1997, *ApJL*, **490**, L77
- Takahashi, R. 2004, *ApJ*, **611**, 996
- Yang, X., & Wang, J. 2013, *ApJS*, **207**, 6
- Younsi, Z., & Wu, K. 2015, *MNRAS*, **454**, 3283
- Younsi, Z., Wu, K., & Fuerst, S. V. 2012, *A&A*, **545**, AA13