ALGORITHMS AND PROGRAMS FOR STRONG GRAVITATIONAL LENSING IN KERR SPACE-TIME INCLUDING POLARIZATION

Bin Chen1,2, Ronald Kantowski2, Xinyu Dai2, Eddie Baron2, and Prasad Maddumage1
1 Research Computing Center, Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA; bchen3@fsu.edu
2 Homer L. Dodge Department of Physics and Astronomy, University of Oklahoma, Norman, OK 73019, USA

Received 2014 August 15; accepted 2015 March 18; published 2015 May 1

ABSTRACT

Active galactic nuclei (AGNs) and quasars are important astrophysical objects to understand. Recently, microlensing observations have constrained the size of the quasar X-ray emission region to be of the order of 10 gravitational radii of the central supermassive black hole. For distances within a few gravitational radii, light paths are strongly bent by the strong gravity field of the central black hole. If the central black hole has nonzero angular momentum (spin), then a photon’s polarization plane will be rotated by the gravitational Faraday effect. The observed X-ray flux and polarization will then be influenced significantly by the strong gravity field near the source. Consequently, linear gravitational lensing theory is inadequate for such extreme circumstances. We present simple algorithms computing the strong lensing effects of Kerr black holes, including the effects on polarization. Our algorithms are realized in a program “KERTAP” in two versions: MATLAB and Python. The key ingredients of KERTAP are a graphic user interface, a backward ray-tracing algorithm, a polarization propagator dealing with gravitational Faraday rotation, and algorithms computing observables such as flux magnification and polarization angles. Our algorithms can be easily realized in other programming languages such as FORTRAN, C, and C++. The MATLAB version of KERTAP is parallelized using the MATLAB Parallel Computing Toolbox and the Distributed Computing Server. The Python code was sped up using Cython and supports full implementation of MPI using the “mpi4py” package. As an example, we investigate the inclination angle dependence of the observed polarization and the strong lensing magnification of AGN X-ray emission. We conclude that it is possible to perform complex numerical-relativity related computations using interpreted languages such as MATLAB and Python.

Key words: accretion, accretion disks – gravitational lensing: strong – methods: numerical – polarization – quasars: supermassive black holes

1. INTRODUCTION

Accretion disks are one of the most spectacular phenomena in modern astrophysics. The large luminosity of active galactic nuclei (AGNs) and quasars (as large as 1047 erg s−1) is believed to be the result of gas accreted by central supermassive black holes. Up to ~10% of the accreting mass can be converted to radiated energy during the accretion process (much higher than in nuclear fusion processes, about ~0.5%). Besides emission in the infrared, optical, and UV bands, most AGNs emit X-rays. Unlike optical emission, which can be explained by standard accretion disk theory (Shakura & Sunyaev 1973), the physical mechanism of AGN X-ray emission remains an enigma, despite decades of research (Krolik 2008). According to standard theory, the temperature of an AGN accretion disk is not hot enough to emit X-rays. The X-ray emission is thought to be generated through inverse Compton scattering of disk photons with hot electrons in the so-called X-ray “corona.” However, both the physics and geometry of this mysterious X-ray corona are not well understood. Observations have revealed that photons of different frequencies correspond to emission regions of different sizes: the higher the frequency, the smaller the emission size, and the closer it is to the central black hole. Very recently, observations in quasar microlensing (Kochanek 2004; Blackburne et al. 2006, 2014, 2015; Morgan et al. 2008, 2012; Chartas et al. 2009; Dai et al. 2010; Chen et al. 2011, 2012; Mosquera et al. 2013; MacLeod et al. 2015) have constrained the AGN X-ray emission size to be of the order of 10 gravitational radii (rG ≡ GM/c2, where M is the mass of the central black hole). Within distances of a few rG, Einstein’s general relativistic gravity theory plays a fundamental role in understanding accreting phenomena. In particular, the X-ray emission is strongly lensed by the central black hole before it escapes the gravity field and arrives at a distant observer. It is well known that photons follow null geodesics in curved spacetimes. The spacetime around a rotating black hole is described by the Kerr metric (Kerr 1963). The strong field environment of AGN X-ray emission invalidates the linear approximation as made in standard gravitational lensing theory (Schneider et al. 1992). Consequently, numerical integration of the geodesic equations is necessary for many problems involving the Kerr metric. Another well-known fact is that a photon’s polarization vector is parallel transported along the photon’s geodesic and its plane of polarization is rotated as the photon passes the rotating black hole, i.e., the so-called gravitational Faraday rotation (Balazs 1958; Plebanski 1960; Ishihara et al. 1988; Agol 1997; Agol & Krolik 2000; Frolov & Shoom 2012; Yoo 2012). Since X-rays are emitted in regions very close to the black hole, the observed X-ray polarization will be influenced by the strong gravity field. Following the pioneering work of Stark, Connors, and Piran (Connors & Stark 1977; Stark & Connors 1977; Connors et al. 1980), X-ray polarization has now become a very promising probe of important parameters such as black hole spin and the inclination angle of the accretion disk (Agol 1997; Agol & Krolik 2000; Dovčiak et al. 2004; Li et al. 2009; Schnittman et al. 2013). It is therefore essential to include the effects of strong gravity when studying AGN X-ray emission.
Numerical ray-tracing in the Kerr metric is not new (Cunningham 1975; Luminet 1979; Rauch & Blandford 1994; Bromley et al. 1997; Beckwith & Done 2004; Fuerst & Wu 2004; Schnittman & Bertschinger 2004; Broderick 2006; Cadeau et al. 2007; Dolence et al. 2009; Psaltis & Johannsen 2012; Chan et al. 2013; Chen et al. 2013a, 2013b; Schnittman & Krolik 2013). However, public codes are relatively rare. Exceptions are Dexter & Agol (2009; a FORTRAN code, fast but mathematically convoluted; see also Yang & Wang 2013), Vincent et al. (2011); a C++ code, not parallelized, Kuchmeister et al. (2012), and Chan et al. (2013; both are GPU-based ray-tracing code solving second-order geodesic equations using CUDA). Polarization and gravitational Faraday rotation have been studied extensively in the literature (Agol 1997; Agol & Krolik 2000; Schnittman & Krolik 2009; however, there seems to be no public code with a focus on X-ray polarizations. Furthermore, there seems to be no ray-tracing code written in interpreted high-level languages such as MATLAB or Python. We present simple algorithms for computing the effect of strong lensing of AGN accretion disks by a Kerr black hole, including polarization analysis of AGN X-ray emission. Our algorithms are realized in the high-level languages MATLAB and Python, and they are fully parallelized and can be easily realized in other languages such as FORTRAN and C++.

The plan of this paper is as follows. An introduction to the Kerr metric, the tetrad formalism, the strong lensing formalism, and the polarization formalism is given in Section 2. The graphic user interface (GUI) and numerical algorithms are discussed in detail in Section 3. In Section 4, we give an example of the use of our MATLAB/Python program KERTAP (Kerr Ray-Tracing And Polarization). In Section 5, we conclude.

2. THEORETICAL FRAMEWORK

Since the gas accreted by black holes carries angular momentum, the black holes powering the accretion processes are expected to possess angular momentum themselves. The proper metric describing the spacetime around a rotating black hole was discovered by Kerr in 1963 (Kerr 1963) and was named after him. We introduce the Kerr metric in this section and discuss the other formalism needed in the strong lensing and polarization analysis of AGN X-ray emission.

2.1. Kerr Metric

In Boyer–Lindquist coordinates, the Kerr metric can be written as

\[ ds^2 = -\alpha^2 dt^2 + \frac{\Sigma^2}{\rho^2} \sin^2 \theta (d\phi - \omega dt)^2 + \frac{\rho^2}{\Delta} dr^2 + \rho^2 d\theta^2, \]

where \((t, r, \theta, \phi)\) are the four independent coordinate variables, and we have defined

\[ \alpha^2 \equiv \frac{\rho^2 \Delta}{\Sigma^2}, \quad \omega \equiv \frac{2a M \rho}{\Sigma^2} \]

with the constants \(M\) and \(a\) being the mass and angular momentum (i.e., \(J / M\)) of the Kerr black hole (we have taken \(c = G = 1\)). The geodesic motion of a photon (its 4 momentum satisfies \(p^a \rho_a = 0\), index \(a\) runs from 1 to 4, with 1 being the time component) can be described by a group of eight first-order Hamilton equations (Arnold 1978):

\[ \frac{dx^a}{d\tau} = \frac{\partial H}{\partial p^a}, \quad \frac{dp^a}{d\tau} = -\frac{\partial H}{\partial x^a}, \]

where \(\xi\) is the affine parameter and \(H(x^a, p_\rho) \equiv \frac{1}{2} g^{ab}(x) p_a p_b\) is the Hamiltonian. The number of equations to integrate is reduced to six since \(p_t\) and \(p_\phi\) are motion constants (the Hamiltonian \(H\) does not depend on \(t\) or \(\phi\)). Another motion constant was discovered by Carter (1968):

\[ C = p_\theta^2 + \cos^2 \theta \left( \frac{p_\phi^2}{\sin^2 \theta} - a^2 p_t^2 \right). \]

The number of ordinary differential equations (ODEs) used to solve the geodesic equations could have been reduced by one using this constant; however, we use Carter’s constant \(C\) as an independent check of our ray-tracing code. A Kerr black hole has two horizons (outer and inner) defined as \(r = M \pm \sqrt{M^2 - a^2}\). The maximum angular momentum of a Kerr black hole was found in Thorne (1974) to be \(a_{\text{max}} = 0.998\). We stop the ray-tracing when the geodesic is within 0.1% of the outer horizon \(r_o\). For accretion disks (normally assumed in the equatorial plane, \(\theta = \pi / 2\)), an important quantity is the so-called innermost-stable-circular-orbit (ISCO) in the equatorial plane (Chandrasekhar 1983). For example, \(r_{\text{ISCO}} = 6 r_o\) and 1.24 \(r_o\), respectively, for a Schwarzschild black hole (\(a = 0\)) and an extreme Kerr black hole. We use \(r_{\text{ISCO}}\) as the inner cutoff of the accretion disk in this paper.

2.2. Tetrad Formalism

In curved spacetime, the frequency (or energy) of a photon depends on the spacetime point where the frequency is measured, and also on the 4 velocity \(u^a\) of the observer measuring the frequency \(u^a u_a = -1\). If \(p_a\) is the 4 momentum of the photon, then the observed photon frequency \(\nu = -u^a p_a\). Let \(u_o^a\) and \(u_e^a\) be the photon’s frequency measured by a distant observer and an observer co-moving with the light source (emitter). We define the redshift factor

\[ g = \frac{\nu_o}{\nu_e} = \frac{(-u^a p_a)_o}{(-u^a p_a)_e} \]

where \(u_o^a\) and \(u_e^a\) are the respective 4 velocities of the distant observer and the source.
For the distant observer, we choose \( u^a = \frac{1}{\gamma}(1, 0, 0, \omega) \), the so-called zero-angular-momentum observer (\citep[ZAMO;][]{Bardeen1972}), where \( \omega \) and \( \alpha \) are defined in Equation (2). A photon’s momentum can be specified using the orthonormal tetrad of ZAMO instead of the coordinate basis, i.e.,

\[
\begin{pmatrix}
\hat{p}^\mu \\
\hat{p}^r \\
\hat{p}^\theta \\
\hat{p}^\phi
\end{pmatrix} = \begin{pmatrix}
\alpha & 0 & 0 & 0 \\
0 & \rho & 0 & 0 \\
-\Sigma \sin \theta / \rho & 0 & \rho & 0 \\
\rho \Omega & 0 & 0 & \rho \Omega
\end{pmatrix} \begin{pmatrix}
p^\mu \\
p^r \\
p^\theta \\
p^\phi
\end{pmatrix}.
\]

We will use Equation (6) to compute the polarization measured by a distant observer (see Section 2.5). The 4 velocity of the source is model-dependent. For accretion disks, we assume a Keplerian flow for the accreting gas (Bardeen et al. 1972), i.e.,

\[ u_\gamma^a = \gamma(1, 0, 0, \Omega_K), \]

where

\[
\Omega_K \equiv \frac{M^{1/2}}{\epsilon^{3/2} + aM^{1/2}}
\]

\[
\gamma \equiv \left( -g_{11} - 2g_{14}\Omega_K - g_{44}\Omega_K^2 \right)^{-1/2}.
\]

We similarly define an orthonormal tetrad in the comoving frame of the Keplerian flow, and to change between tetrad and coordinate components we use

\[
\begin{pmatrix}
p^\gamma \\
p^r \\
p^\theta \\
p^\phi
\end{pmatrix} = \begin{pmatrix}
\gamma & 0 & 0 & \tau \lambda \\
0 & \sqrt{\Delta} / \rho & 0 & 0 \\
0 & 0 & -1 / \rho & 0 \\
\gamma \Omega & 0 & 0 & \tau
\end{pmatrix} \begin{pmatrix}
p^\gamma \\
p^r \\
p^\theta \\
p^\phi
\end{pmatrix},
\]

where

\[
\lambda \equiv \frac{g_{14} + g_{44}\Omega_K}{g_{11} + g_{14}\Omega_K} \\
\tau \equiv \left( g_{11}\lambda^2 + 2g_{14}\lambda + g_{44} \right)^{-1/2}.
\]

The above equations are useful since both the intensity and the polarization profile of the source emission are often given in the local rest frame of the source. In Section 2.5, we use Equations (9) to initialize the polarization propagator at the end of backward ray-tracing.

### 2.3. Strong Lensing by Kerr Black Holes

X-ray emission from an accretion disk is strongly lensed by the central Kerr black hole as it leaves the source to arrive at a distant observer. A lensed accretion disk appears very different from an unlensed one. The frequency of the source emission is gravitationally and Doppler blueshifted depending on the position of the emitter, the 4 velocity of the source, and the distance and inclination angle of the observer. Consequently, the intensity profile of a lensed accretion disk can be very different from an unlensed one, and the observed flux can also be different from the unlensed case.

By definition, the monochromatic flux \( F_{\nu} \) measured by an observer \( O \) very far from the black hole, is (Mihalas 1978)

\[
F_{\nu} \equiv \oint I_{\nu,\text{obs}} d\Omega_o = \int I_{\nu,\text{source}} d\Omega_o,
\]

where \( d\Omega_o \) is the differential solid angle measured at the observer, and \( \mu \) is the angle cosine between the light ray and the normal of the detector window. We can safely drop the factor \( \mu \) for sources at cosmological redshifts because these sources are observed with tiny solid angles. If we ignore Kerr strong lensing and Doppler shifts, photons travel along straight lines and there is no frequency redshift. Consequently, \( I_{\nu,\text{source}} \) is a conserved quantity along the light path,

\[
F_{\nu,\text{unlensed}} \equiv \oint I_{\nu,\text{obs}} (\hat{n}) d\Omega_o = \frac{\cos \theta}{r_{\text{obs}}^2} \oint I_{\nu,\text{source}} (\hat{n}) dA,
\]

where \( \hat{n} \) is the 3D photon direction at the observer, \( dA \) is the differential area element of the accretion disk, and we have changed the solid angle integration at the observer into a 2D surface integral over the accretion disk. On the other hand, for Kerr lensing, we have

\[
F_{\nu,\text{lensed}} \equiv \oint g^3 (\hat{n}) I_{\nu,\text{source}} (x^\nu (\hat{n}), \theta_\nu (\hat{n})) d\Omega_o,
\]

where \( \nu_c \) is the source frequency, \( g \) is the redshift factor of Equation (5), and \( (x^\nu, \theta_\nu) \) is the photon’s position and momentum at the emitter found by backward ray-tracing (see Section 2.4). To obtain Equation (13), we simply used the fact that \( I_{\nu,\text{source}} / \nu^3 \) is conserved along each geodesic. The strong lensing magnification of the specific flux is then

\[
\mu \equiv \frac{F_{\nu,\text{lensed}}}{F_{\nu,\text{unlensed}}}.
\]

#### 2.4. Backward Ray-tracing

Since the emission region can be as close as a few gravitational radii from the black hole, the standard linearized gravitational lensing theory is not valid. Our strong lensing analysis in Kerr spacetime is based on backward ray-tracing. Given a source profile \( I_s (x^\nu, \theta^\nu) \), the flux integral in Equation (12) can be easily computed either analytically or numerically. However, it is not a trivial job to evaluate the lensed flux integral Equation (13) in which the integral is over the solid angle at the observer but the integrand is evaluated at the source. We do this through backward ray-tracing. We choose a thin pencil beam (a sharp pentahedron) focused on the observer (the pigment core toward the black hole), and divide the solid angle space of this beam (large enough to contain all disk emission arriving at the observer) into a uniform grid of pixels. Through each pixel, we shoot one ray backward from the observer to the source. To compute the image area, we need only count the number of pixels whose light rays end at the accretion disk. To compute the observed flux, we weight pixels by the integrands in Equation (13). In this way, our backward ray-tracing algorithm takes into account gravitational light deflection, Doppler or gravitational redshift, and area distortion simultaneously.
2.5. Polarization and Gravitational Faraday Rotation

Because the polarization at the source is known and needs to be computed at the observer, most current polarization propagators are based on forward ray-tracing (Schnittman & Krolik 2009). Our polarization integrator is based on backward ray-tracing. We assume the polarization and limb darkening of the source emission are given as in the classical work of Chandrasekhar (1960) where the radiation transfer equation with polarization was solved assuming a scattering-dominated semi-infinite atmosphere. Other initial polarization profiles such as an inverse Compton scattering-dominated optically thin atmosphere can also be implemented.

At the end of the backward ray-tracing of each ray, we know the landing point on the accretion disk of a geodesic and its 4 momentum $p^\nu$. First, we compute the photon’s 4 momentum components $\hat{p}^\alpha$ in the comoving frame. Then we compute the angle cosine, $\mu$, between the photon’s momentum and the disk normal measured by the comoving observer. The angular-dependent part of the intensity profile, $w(\mu)$ (see Equation (21) in Section 4), and the degree of polarization, $\delta_{\text{source}}(\mu)$, can be computed by interpolating Table XXIV of Chandrasekhar (1960). We then compute the photon’s polarization vector $E$ in the local rest frame using the fact that $E$ is parallel to the disk plane and normal to the photon’s momentum vector ($E \cdot p = 0$), i.e.,

$$\hat{E}^\alpha = \frac{1}{\sqrt{(\hat{p}^\alpha)^2 + (\hat{p}^\beta)^2}}(0, -\hat{p}^1, 0, \hat{p}^2).$$

Next, we use Equation (9) to compute the coordinate components $E^a$ of the polarization vector $E$. We then solve the parallel propagation equation of the polarization vector forward along the photon geodesic toward the distant observer using

$$\frac{dE^a}{d\xi} = -\Gamma^a_{bc} p^b E^c,$$

where $\Gamma^a_{bc}$ are the Christoffel symbols. Since the geodesic equations have already been solved backward from the observer to the source, and because we have dropped those rays that missed the accretion disk, the integral from the source forward to the observer is straightforward and fast.

At the end of the forward polarization propagating process, we are back to the observer with rotated polarization vector $E_{\text{obs}}$. Here, we first compute the tetrad components of $E_{\text{obs}}$ using Equation (6). We then work out the Stokes parameters corresponding to this ray,

$$2\chi = \arctan\left(\frac{-\hat{E}^\theta}{\hat{E}^\phi}\right),$$

$$Q = \delta L_{\nu_o} \cos 2\chi,$$

$$U = \delta L_{\nu_o} \sin 2\chi,$$

where $\chi$ is the polarization angle, $\delta = \delta_{\text{source}}$ is the degree of polarization conserved along a photon geodesic, and $L_{\nu_o}$ is the observed specific intensity at frequency $\nu_o$, which is related to the source intensity by $I_{\nu_o} = g L_{\nu}$. To compute the integrated degree of polarization and the polarization angle, we add $Q, U$, and $I_{\nu_o}$ from each image pixel, obtaining $Q_{\text{total}}, U_{\text{total}},$ and $I_{\nu_o}^{\text{total}}$.

Finally, we use Equation (17) again to compute $\delta_{\text{total}}$ and $\chi_{\text{total}}$. Detailed algorithms are given in the next section.

3. GUI AND NUMERICAL ALGORITHMS OF KERTAP

3.1. Graphic User Interface

The MATLAB version of KERTAP contains a GUI (see Figure 1). It asks the user for the black hole parameters, the resolution desired, and the number of parallel workers for parallel computing. The GUI contains a status panel indicating the status of the code: waiting for input, running, or finished. If the code is run successfully, the GUI will generate the redshifted image of the accretion disk, and output important parameters such as the flux magnification and the degree and angle of the observed polarization. The performance of the code, such as the wall clock time used for ray-tracing, will also be recorded.

3.2. Initialization Algorithm For Backward Ray-tracing

Given an observer distance $r_{\text{obs}}$, and the dimensions of the image rectangle (orthogonal to the line of sight at the source redshift), $x_{\text{size}}$ and $y_{\text{size}}$, the angular size of the cone focusing at the observer can be inferred. Given the resolutions in the x and y directions, the pixel size can be computed as $dx_{\text{grid}} = x_{\text{size}}/n_x$ and $dy_{\text{grid}} = y_{\text{size}}/n_y$. For a pixel labeled by $(ix, iy)$, the photon’s starting 8D phase-space coordinates are computed in Algorithm 1.

Algorithm 1. Initialization of one pixel $(ix, iy)$ of the grid. $\hat{p}^{r,\theta,\phi}$ are elements of the 3D photon momentum vector with respect to the ZAMO. The function $p3\_to\_p4(\cdot)$ will normalize it assuming $\hat{p}^r = 1$, and then convert the ZAMO tetrad components to covariant coordinate components. We choose $t_{\text{obs}} = 0$ and $\theta_{\text{obs}} = 0$ without a loss of generality.

| Parameter | Description |
|-----------|-------------|
| $X_0$     | $(0, \theta_{\text{obs}}, 0)$ |
| $\beta^r$ | $r_{\text{obs}}$ |
| $\beta^\theta$ | $dx_{\text{grid}} \times ix$ |
| $\beta^\phi$ | $dy_{\text{grid}} \times iy$ |
| $p^4$     | $p3\_to\_p4(\beta^r, \beta^\theta, \beta^\phi)$ |
| $Y_0$     | $(x_{\text{obs}}, p^4)$ |

3.3. Ray-tracing Using MATLAB ODE45

For each ray arriving at the observer, we solve the geodesic equations using MATLAB ODE solver ode45. The syntax of ode45 is

$$[\xi, Y, \xi_{\text{range}}, Y_0, \text{options}, \ldots] = \text{ode45}(\text{ode45}, \text{range}, Y_0, \text{options}, \ldots),$$

where $Y(\xi, Y)$ is an 8D column vector containing the right-hand side of Equation (3), $Y_0$ is the 8D initial condition for ode45, and $\xi_{\text{range}}$ is the range of the independent variable (affine parameter) where we want to solve the ODEs. A subtle point of backward ray-tracing is that we do not shoot rays toward the accretion disk (say, take $\hat{p}^r < 0$), instead we must have $\hat{p}^r > 0$ (toward the observer). This point is important.
because the Kerr metric is not symmetric with respect to the reflection of the time coordinate (time reversal will change the direction of the black hole angular momentum). The backward ray-tracing is realized by properly choosing $\xi$, i.e., we define $\xi \in [\xi_0, \xi_1]$, where $Y(\xi_0) = Y_0$, and $Y(\xi_1)$ corresponds to the light emitting event ($\xi_1 < \xi_0$, and $t_1 < t_0$). In Equation (18), $\text{options}$ is a structure containing optional parameters that change the default integration properties, such as the relative and absolute error tolerance, and events to stop the integration. For example, suppose we want to stop the ray-tracer when the geodesic is very close to the event horizon, say, $r < (1 + \delta)r_e$ for some small $\delta$, we define a black hole hitting event as

$$\text{flag}_{\text{BH}} = (r < (1 + \delta)r_e) - 1. \quad (19)$$

Since we want to know when the geodesic hits the equatorial plane, we define another event

$$\text{flag}_{\text{Equator}} = \cos \theta. \quad (20)$$

Then, $\text{ode}45$ will record the point where $\text{flag}_{\text{BH}} = 0$ or $\text{flag}_{\text{Equator}} = 0$ and stop the integration if the user requests to (as we did). These two events are defined in $\text{disk}_{-}\text{events.m}$. In the left-hand side of Equation (18), the index $I$, tells you which event happens, $(\xi_e, Y_e)$ stores the point where the event happens, and $(\xi, Y)$ stores the results of the integration up to the event point (i.e., the photon geodesic). Based on $\text{ode}45$, the ray-tracing part of KERTAP is easily realized in Algorithm 2.

### Algorithm 2. Ray-tracing Algorithm. MATLAB $\text{ode}45$ is Based on Runge–Kutta45.

```
for iy = ny:n do
  for ix = nx:n do
    Initialize pixel $(ix, iy)$ giving $Y_0$
    $\xi_0 \ldots = \text{ode}45(@Y(x), Y_0, \text{options}, \ldots)$
    if ray goes to infinity then
      goes to next ray
    else if hit on the horizon then
      go to next ray
    else if hit on the equatorial plane then
      if hit on the accretion disk then
        compute redshift, write $(x_e, y_e)$, ...
        go to next ray
      end if
    end if
  end for
end for
```

#### 3.4. Polarization Propagator

At the end of the backward ray-tracing of each ray, the polarization propagator is initialized and propagated forward to the observer, see Algorithm 3.

### Algorithm 3. Polarization Propagating Algorithm. In the following, $\text{polarization}(\cdot)$ computes the polarization degree $\delta_{\text{source}}$ and polarization vector $E^p_{\text{source}}$ on the accretion disk.
using \((x^a, p^a)_{\text{source}}\) obtained from backward ray-tracing. \(\delta_{\text{source}}\) is interpolated from Table XXIV of Chandrasekhar (1960). The polarization vector \(E^a\) is propagated forward from the source to the observer using Equation (16). \(\text{polar}_\text{obs}()\) computes the polarization angle \(\chi\) at the observer. \(\delta_{\text{obs}} = \delta_{\text{source}}\) since the degree of polarization is conserved along geodesics.

\[
\begin{align*}
\text{for } i_x = -nx: nx & \text{ do} \\
\text{for } i_y = -ny: ny & \text{ do} \\
\{i_x, Y, j_y\} & = \text{ode45}(\text{Y}’, \ldots) \\
\text{if a good shot} & \text{ then} \\
[\delta_{\text{source}}, E_{\text{source}}] & = \text{polarization}(Y, M, a) \\
E_{\text{curr}} & = E_{\text{source}} \\
n_{\text{point}} & = \text{length}(\xi) \\
\text{for } j = 0, n_{\text{point}} - 2 & \text{ do} \\
E_{\text{source}} & \rightarrow E_{\text{obs}} \\
d^2 & = \xi_{j+1} - \xi_j \\
dE^a & = -\sum_j (T_{\xi} E_{\text{curr}}^c) d^2 \\
e^{\xi}_{\text{next}} & = E_{\text{curr}} + dE^a \\
\text{end for} \\
E_{\text{curr}} & = E_{\text{next}} \\
E_{\text{obs}} & = E_{\text{curr}} \\
\chi & = \text{polar}_\text{obs}(E_{\text{obs}}) \\
\text{end if} \\
\text{end for} \\
\end{align*}
\]

### 3.5. Post Processing

The results of ray-tracing are stored in different data structures. For example, \(\text{Image}_G\) and \(\text{Image}_Y\) store the redshifts \((g = 0 \text{ for a missed shot})\) and 8D phase-space coordinates \((x^a, p_b)\) of the target points on the accretion disk for all successful shots. \(\text{Image}_P\) stores the polarization data \((w, \delta, \chi)\), where \(w\) is the limb darkening factor from Table XXIV of Chandrasekhar (1960). The strong lensing amplifications of the observed flux and image area are computed in Algorithm 4. The observed degree and angle of polarization are computed using Algorithm 5. Thanks to the data visualization tools intrinsic to MATLAB and Python, the image of the lensed disk can be easily generated using functions such as \text{imagesc()}\) for MATLAB and \text{imshow()}\) for Python, and the gravitational Faraday rotated polarization vector field can be conveniently visualized using the function \text{quiver()}, which is similarly defined for the two languages. The redshifted image of an accretion disk will be plotted in the GUI after a successful run, and another two image windows (one for the same redshift image, the other for the intensity profile and polarization visualization) will pop up for the user to edit or save the figures in their desired formats (eps, jpg, etc.).

#### Algorithm 4. Strong lensing algorithm. \(dA\) and \(d\Omega\) are the differential image area and solid angle for one pixel. The unlensed image area and flux are easy to compute, and are assumed to be known.

\[
\begin{align*}
f_{\text{lensed}} & = 0; A_{\text{lensed}} = 0 \quad \triangleright \text{Initialization} \\
\text{for } i_x = -nx: nx & \text{ do} \\
\text{for } i_y = -ny: ny & \text{ do} \\
\text{if a good shot} & \text{ then} \\
A_{\text{lensed}} & = A_{\text{lensed}} + 1 \\
\end{align*}
\]

#### Algorithm 5. Polarization synthesizing algorithm. \(I_{\text{total}}\) and \(\chi_{\text{total}}\) are the integrated degree and angle of polarization.

\[
\begin{align*}
I_{\text{total}} & = 0; Q_{\text{total}} = 0; U_{\text{total}} = 0 \quad \triangleright \text{Initialization} \\
\text{for } i_x = -nx: nx & \text{ do} \\
\text{for } i_y = -ny: ny & \text{ do} \\
\text{if IsImagePoint} & \text{ then} \\
g & = \text{Image}_G(i_x, i_y) \\
r & = \text{Image}_Y(i_x, i_y) \\
[w, \delta, \chi] & = \text{Image}_P(i_x, i_y) \\
dl & = w^2 \delta^2 + \mu_n \\
dQ & = \delta \times dl \times \cos(2\chi) \\
dU & = \delta \times dl \times \sin(2\chi) \\
Q_{\text{total}} & = Q_{\text{total}} + dQ \\
U_{\text{total}} & = U_{\text{total}} + dU \\
I_{\text{total}} & = I_{\text{total}} + dl \\
\text{end if} \\
\text{end for} \\
\chi_{\text{total}} & = 0.5 \tan^{-1}(U_{\text{total}}/Q_{\text{total}}) \\
\delta_{\text{total}} & = U_{\text{total}}/I_{\text{total}} \sin(2\chi_{\text{total}}) \\
\end{align*}
\]

### 3.6. Python Version of KERTAP

For users who do not have access to a MATLAB cluster, we provide a Python version of KERTAP. The Python code uses exactly the same algorithms as the MATLAB code except that it does not have a GUI. For the MATLAB code, we used the ODE solver, \text{ode}45 provided by MATLAB (already highly optimized and with stringent error control), whereas for the Python code we created a fifth-order Runge–Kutta solver using the Cash–Karp methods. The MATLAB \text{ode}45 provides a very convenient “event” capturing mechanism to stop the ray-tracing at user-defined events (such as when a ray crosses the event horizon, or punches through the accretion disk, see Section 3.3), whereas for the Python code, these events are coded into the ODE solver by hand. Consequently, to extend the code to include other events (e.g., higher-order Kerr-lensing images) is slightly more difficult for the Python code than for MATLAB.

The MATLAB version of KERTAP was parallelized using the Parallel Computing Toolbox and the MATLAB Distributed
Computing Server (MDCS). A very useful feature of MDCS is that it allows parallel jobs to be submitted from the user’s desktop directly to distant parallel computing clusters, i.e., the user provides the desired parameters through the GUI of KERTAP, and KERTAP automatically creates a job submission script and submits the job to the remote cluster dealing with popular job schedulers such as MOAB, SLURM, etc. The Python version of KERTAP was parallelized using the free mpi4py package which supports a full implementation of the Message Passing Interface (MPI). The pure Python version of KERTAP is slower than the MATLAB version by a factor of $\sim 2$. We speed up the Python code by adding C-type static type declarations to the pure Python code using Cython, which results in a $\sim 100$ times performance gain, see Figure 7. We provide our users with a pure Python version of KERTAP, a Cython version, and a parallel Cython version. To run the Cython version of KERTAP, the Cython source file “mod.pyx” which contains most of the routines needed to be compiled into a Python extension module “cymod.so” before it can be imported as a Python package. To run the Cython code on a cluster in parallel, an MPI compiler (e.g., gnu-openmpi) is needed. The details of implementation are included in the documentation to KERTAP.

4. AN EXAMPLE APPLICATION

We assume that the X-ray corona is a thin layer immediately above the AGN accretion disk, and moves with Keplerian flows (see Equations (7) and (8)). We take $r_{\text{disk}} = 20 r_g$ according to the constraints obtained from quasar microlensing observations (Kochanek 2004; Blackburne et al. 2006, 2014, 2015; Morgan et al. 2008, 2012; Chartas et al. 2009; Dai et al. 2010; Mosquera et al. 2013; MacLeod et al. 2015), and we use $\Gamma_{\text{ISCO}}$ as the inner cutoff. The image window for backward ray-tracing is $50 r_g \times 50 r_g$. X-ray coronae of this type are motivated by the so-called sandwich model (Haardt & Maraschi 1991, 1993) and are often used in the literature (Ruszkowski & Fabian 2000). For a corona geometry such as a sphere or a disk above the accretion disk, see Chen et al. (2013a). Because quasar X-ray emission is observed to follow a power law, we assume the following form for the source intensity profile,

$$I_{\nu}(\nu, \mu, r) \propto \frac{1}{r^n} \frac{w(\mu)}{r^{n-1}}$$

(21)

where $\mu$ is the cosine between the photon’s 4 momentum and the upward disk normal measured by the comoving observer, and $w(\mu)$ is the angular-dependence of the intensity profile (we have assumed azimuthal symmetry for simplicity). To be more specific, we take $w(\mu)$ from Chandrasekhar (1960; classical results from solving the radiation transfer equation with polarization assuming a scattering-dominated semi-infinite atmosphere). The parameter $\Gamma$ is referred to as the photon index, and $n$ specifies the radial steepness of the profile. We take $\Gamma = 2.0$ (e.g., Chen et al. 2012), and $n = 3$ (radially steep) or 0 (radially flat). We test for two black hole spins, $a = 0.998$ (Thorne 1974) or 0 (Schwarzschild black hole). We take the observer distance as $r_{\text{obs}} = 10^9 r_g$ and test for different inclination angles (from nearly face on to nearly edge on).

For the assumed simple geometry and intensity profile, the unlensed specific flux at frequency $\nu_o$ can be analytically integrated using Equation (12)

$$F_{\nu} = \frac{2\pi \theta_{\text{obs}}^2 \nu^2}{r_{\text{obs}}^2} \frac{1}{\nu_o^{1-1/n}} \int_{r_{\text{ISCO}}}^{r_{\text{disk}}} dr \frac{1}{r^{n-1}} = \frac{2\pi \theta_{\text{obs}}^2 \nu^2}{(n-2)r_{\text{obs}}^2} \frac{1}{\nu_o^{1-1/n}} \left[ \frac{1}{r_{\text{ISCO}}^{n-2}} - \frac{1}{r_{\text{disk}}^{n-2}} \right]$$

(22)

where $\theta_{\text{obs}} = \cos(\theta_{\text{obs}})$. We have dropped the unimportant constant in the definition of $L_{\nu}$. For $n = 2$, the $(n-2)$ factor in the denominator is absent and the expression within the square brackets is replaced by $\log r_{\text{disk}}/r_{\text{ISCO}}$. From Equation (13), the lensed flux is

$$F_{\nu} = \frac{1}{\nu_o^{1-1/n}} \int_0^\Lambda g^{\nu+2} w(\mu) \frac{d\Omega_o}{r^n}$$

$$= \frac{1}{\nu_o^{1-1/n}} \sum_{ij} \frac{g^{ij}}{x_{ij}} \frac{dA_{\text{grid}}}{r^{i-1}}$$

(23)

where the summation is over the accretion disk image, and $dA_{\text{grid}} = dx_{\text{grid}} \times dy_{\text{grid}}$ is the area of one image pixel. Note that $w(\mu)$ varies from pixel to pixel because of the general relativistic light bending and the aberration effect. The magnification $\mu = F_{\nu_o}^{\text{lensed}}/F_{\nu_o}^{\text{unlensed}}$ can be computed using Equations (22) and (23) immediately (note that the term $1/(r_{\text{ISCO}}^{n-1})$ cancels out).

In Figure 2, we plot the lensed intensity profile and distribution of polarization for the X-ray disk observed at inclination angle $75^\circ$. Figure 3 is similar to Figure 2, but is for spin $a = 0.5$ and we have included up to fourth-order Kerr-lensing images. Kerr strong lensing changes both the shape and the intensity profile of the disk significantly. The intensity is strongly concentrated in a small region (in dark red) near the black hole and to the left of the accretion disk (where the rotating source is approaching the observer). As for polarization, the degree of polarization to the left of the black hole is on average smaller than the classical value ($\sim 4.6\%$ for $\theta = 75^\circ$; Chandrasekhar 1960) because the angle between the photon’s 4
momentum and the disk’s normal measured by the comoving observer is smaller than the inclination angle. Therefore, $\delta(\mu)$ is smaller, see the black lines in the second row of Figure 4 for the inclination dependence of $\delta$. The contrary holds for the right-hand side of the accretion disk. On the other hand, the direction of polarization can change significantly because of the gravitational Faraday rotation. This effect is more significant for source regions closer to the black hole where the observed polarization angle $\chi$ can be positive or negative (depending on the actual source location) instead of horizontally aligned as in Chandrasekhar (1960).

The strong lensing magnification of the image area and specific flux, and the integrated (or averaged) polarization degree and angle are plotted in Figure 4. We did not consider higher-order images here since the accretion disk is believed by many to be optically thick. We plot the inclination dependence of $\mu$, $\mu_{\text{max}}$, $\mu_{\text{flat}}$, $\delta$, and $\chi$ for spin $a = 0.998$ and 0, and for a steep and flat radial profile ($n = 3$ or 0). The magnification increases with inclination angles and is more significant for steeper profiles and for larger spins (smaller $r_{\text{ISCO}}$), both of which amount to more concentrated emission closer to the black hole. The fact that for small inclination angles (nearly face on), the observed degree of polarization is greater than the classical result is mainly caused by gravitational light bending ($\delta = 0$ for the face on case, see Chandrasekhar 1960). For moderate to high inclination angles, the emission is strongly focused in a small region with a low degree of polarization, see Figure 2. Consequently, special and general relativity effects tend to reduce the observed degree of polarization, confirming earlier results (Connors et al. 1980).

Another interesting point is that for a fixed emission size (e.g., $r_{\text{disk}} = 20 r_g$ as used in this paper) with the intensity profile Equation (21), the effects on polarization, i.e., the polarization profile, the integrated degree and angle of polarization, are achromatic. This is because both the inputs of the polarization propagator, i.e., $\delta(\mu)$ and $w(\mu)$ from Chandrasekhar (1960), and gravitational Faraday rotation are wavelength independent, and the $1/\nu$ factor cancels out when averaging over the image of

Figure 4. Inclination dependence of Kerr strong lensing and polarization. We test for two black hole spins $a = 0.998$ and 0 (the first and second column), and two radial profiles $n = 3$ (steep) and $n = 0$ (flat). The first row plots the strong lensing magnification of the image area and total flux. The second and third rows plot the observed degree and angle of polarization, respectively. The black lines in the second and the third rows are classical Chandrasekhar (1960) results.
Table 1
Convergence and Performance of KERTAP. We Run KERTAP on a Single Computer with Eight MATLAB Parallel Workers

| Resolution | $h_{\text{min}}/h_{\text{max}}$ | $r_{\text{min}}/r_{\text{max}}$ | $\delta$ | $\gamma$ (deg) | Wall time (hr) | Speed (ray s\(^{-1}\)) |
|------------|-----------------|-----------------|-------|----------|--------------|------------------|
| 20 \times 20 | 1.563/1.742 | 0.257/1.280 | 0.0230 | ~8.473 | 0.004 | 27.33 |
| 50 \times 50 | 1.568/1.587 | 0.051/1.334 | 0.0157 | ~6.728 | 0.011 | 66.15 |
| 100 \times 100 | 1.567/1.712 | 0.039/1.356 | 0.0159 | ~8.017 | 0.034 | 83.65 |
| 200 \times 200 | 1.563/1.723 | 0.037/1.357 | 0.0158 | ~8.052 | 0.120 | 93.07 |
| 500 \times 500 | 1.563/1.726 | 0.038/1.360 | 0.0158 | ~8.005 | 0.713 | 97.75 |
| 1000 \times 1000 | 1.563/1.726 | 0.036/1.360 | 0.0158 | ~7.989 | 2.881 | 96.62 |

Note. Accretion disk size $r_{\text{disk}} = 20 r_g$. The size of the image window is $50 r_g \times 50 r_g$. Black hole spin $a = 0.998$. Photon index $\Gamma = 2.0$ and radial profile $n = 3$. Inclination angle $\theta = 75^\circ$. Observer distance $r_{\text{obs}} = 10^6 r_g$.

Figure 5. Accuracy of KERTAP. A typical ray is shot backward toward the black hole and hits the accretion disk, after which the polarization vector is forward propagated to the observer. The dotted-dashed blue curve shows the relative error in Carter’s constant. The dashed magenta curve shows the norm of the photon 4 momentum vector. The solid red line shows error in the norm of the forward parallel-propagated (space-like) photon polarization vector.

The disk when computing $\chi_{\text{total}}$ and $\delta_{\text{total}}$, see Equations (17), (22), and (23). This apparent discrepancy with Schnittman & Krolik (2009) is caused by the fact that Schnittman & Krolik (2009) have assumed thermal emission for X-rays emitted by the accretion disk around a stellar mass black hole, and therefore the X-ray emission of higher frequencies are emitted by regions closer to the black hole (the effective temperature of the disk is higher closer to the black hole), whereas we have assumed the same power-law $E \propto r^{-1}$ at each radius for X-ray emission by AGNs. If we assume that hard X-rays are emitted by a region smaller than that of the soft X-rays and closer to the black hole as suggested by Chen et al. (2011), say, $r_{\text{disk}} = 10 r_g$, then, even under the same power-law assumption, the integrated degree and angle of polarization should differ from current results, i.e., the results will become chromatic.

The accuracy of KERTAP is very good. We have used Carter’s constant as an independent check of accuracy. At the end of the long ray-tracing ($r_{\text{obs}} = 10^6 r_g$), the relative error of Carter’s constant is never worse than $\sim 10^{-7}$ for all cases tried by the authors. We also checked the accuracy of KERTAP using the norm of the photon’s 4 momentum vector and the polarization vector $E^\theta$; see Figure 5 for the results of a typical ray. The results of the MATLAB code are highly consistent with the Python code, e.g., the norm of the difference between the redshift images generated using the Python and MATLAB code is of the order of $10^{-6}$, and there were no rays that hit the accretion disk or entered the horizon in one case that did not hit the disk or enter the horizon in the other. A convergence test is shown in Table 1 and Figure 6. The convergence of KERTAP is pretty good, e.g., the amplification to image area and specific flux, and the polarization degree and angle all converge at a grid of moderate size 500 x 500. In Figure 7, we test the scalability of KERTAP. Both the MATLAB (parallelized using MCDS) and Python (parallelized using mpi4py) codes scale roughly linearly with the number of CPUs. Consequently, the performance of KERTAP can be significantly improved by parallel computing. However, the performance improvement of the MATLAB code through parallel computing was restricted by the number of MDCS licenses available in a user’s MATLAB cluster, whereas the Python code is only restricted by the size of the cluster (number of CPUs available for the MPI job). It is important to note that since MATLAB2014b, the Parallel Computing Toolbox allows up to 512 parallel workers (this number is not constrained by the number of MDCS licenses available). Consequently, significant speedup can still be achieved by users without a MATLAB cluster but with a single node with many cores.

5. DISCUSSION

We have developed KERTAP, the first ray-tracing code using the interpreted languages MATLAB and Python, for studying strong gravitational lensing in Kerr spacetime including polarization. The key ingredients of KERTAP are a GUI (for the MATLAB version), backward ray-tracing realized by a fifth-order Runge–Kutta method, a polarization propagator and synthesizer, and the formalism and algorithms for computing strong lensing effects of a Kerr black hole centered on an accretion disk. The backward ray-tracing part of KERTAP most resembles that of Schnittman & Bertschinger (2004) and Vincent et al. (2011). Vincent et al. (2011) focuses on radiation transfer while this paper focuses on the observable effects of Kerr strong lensing and polarization. The independence of the geodesic equations from the polarization propagation equations makes it possible to solve the geodesic equations (backward) first, then propagate the polarization vector forward. The idea of backward ray-tracing followed by forward polarization propagating is new, to the best of our knowledge.

8 The code is available at https://bitbucket.org/binchen14/kertap.
knowledge. Polarization and gravitational Faraday rotation have been studied extensively in the literature (e.g., Agol 1997; Agol & Krolik 2000; Schnittman & Krolik 2009). However, public code explicitly dealing with polarization is not yet available. Published papers investigating the X-ray polarization of accretion disks are basically all doing forward ray-tracing (with or without Monte-Carlo radiation transfer; see, e.g., Schnittman & Krolik 2009). Since X-rays are emitted by sources very close to the central black hole and moving with highly relativistic flows, in order to compute the observable property measured by a distant observer at a given inclination angle, at each point on the accretion disk and for each direction, many forward rays have to be traced (because no one knows which ray will go where). This increases the forward ray-tracing time significantly, and the results are usually noisier than those from backward ray-tracing (e.g., compare Figure 1 of Schnittman & Krolik 2009 with Figure 2 of this paper).

The idea of performing very computationally intensive jobs such as ray-tracing in Kerr spacetime using interpreted languages such as MATLAB or Python might sound odd at first. It is hard to expect a dynamic language such as Python to significantly over the past decade, in particular, through the development of application programming interfaces (APIs) with low-level languages such as C/FORTRAN (e.g., the CPython API), multi-cpu parallel computing (mpi4py for Python; MDCS for MATLAB), and by supporting GPUs. For example, both Python and MATLAB support GPU programming, e.g., PyCUDA for Python. The performance of the pure Python version of KERTAP was improved by two orders of magnitude using Cython. The disadvantage in speed is compensated by parallel computing. In MATLAB, this is achieved by using the Parallel Computing Toolbox on local machines with multiple cores, or by using the Distributed Computing Server over a cluster. For Python, the Cython version of KERTAP supports full implementation of MPI. For example, in this paper, we have shown that it is feasible to study strong lensing of X-ray emission of accretion disks around a Kerr black hole using MATLAB/Python. The strong lensing analysis of an X-ray emitting disk of radius $r_g$ can be done with high accuracy in ~40 minutes by running the MATLAB version of KERTAP on a desktop computer with 8 MATLAB workers.

---

4 This does not imply that the back and forth ray-tracing algorithms presented in this paper are intrinsically slow.

5 http://mathema.tician.de/software/pycuda/
desired form of interactive high-performance computing.\textsuperscript{6} If the reader desires, the strong lensing and polarization algorithms can be easily realized in FORTRAN or C++. The advantage of coding using high-level languages such as Python and MATLAB is manifold. First, the object-oriented nature of MATLAB makes it very easy to build GUIs. For example, the GUI as shown in Figure 1 is written using the MATLAB GUI building tool “guide.” Second, it is much easier to write code using MATLAB or Python than FORTRAN or C++. Programing in MATLAB/Python greatly shortens the whole cycle of a scientific project, i.e., from prototyping and debugging the code, to production runs, to visualization of the scientific results and generating high quality plots. For example, this whole paper can be done using a single language (either MATLAB or Python). Our code is very short. The kernel of the MATLAB version of KERTAP is only about 600 lines. The kernel of the Python version of KERTAP is about 200 lines longer (including the Runge–Kutta solver). The reader should experience no difficulty in understanding our code after reading this paper. Third, since MATLAB was designed to be a data analysis and visualization tool (and thanks to free Python packages such as “matplotlib”), the post-processing of ray-tracing data using MATLAB or Python is almost trivial. For example, the piece of code generating the redshifted image of the accretion disk (see Figure 1) is only a few lines. Users do not have to write separate codes visualizing their results using software such as IDL or Mathematica. Our code would be very useful for scientists new to the Kerr geometry or in need of polarization calculations, or to those people who need a second code to compare with their own. KERTAP is the first public code specifically dealing with strong lensing of X-ray polarization. For less complicated computing jobs, our MATLAB/Python code can be used directly or after some modification. For complicated computing jobs, KERTAP might still be useful if the user’s institute has a parallel computing cluster since KERTAP is fully parallelized. We conclude that it is possible to perform complex numerical-relativity-related computations using interpreted languages such as MATLAB and Python.

NSF AST-0707704 and support for Program number HST-GO-12948.04-A was provided by NASA through a grant from the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract NAS5-26555. X.D. acknowledges the financial support from the NASA ADAP programs NNX11AD09G, NNX15AF04G, and NSF grant AST-1413056. We thank the anonymous referee for a careful review of this work. We thank Jeremy D. Schnittman for comments. B.C. thanks Peter Beerli and Paul van der Mark for discussions on parallel computing using Python and Shiuyan Tang for help in generating one plot. The performance analysis was conducted on the HPC cluster at the Research Computing Center at the Florida State University.

REFERENCES

Agol, E. 1997, PhD thesis, Univ. California Santa Barbara
Agol, E., & Krolik, J. H. 2000, ApJ, 528, 161

\textsuperscript{6} Refer to http://rec.fsu.edu/software/MATLAB for the details of submitting parallel MATLAB jobs remotely.