IRSTI 29.05.41 https://doi.org/10.26577/ijmph.2024v15i1a6

N.A. Beissen[1], D.S. Utepova[1,2*], V.N. Kossov[2], S. Toktarbay[1,3], M.K. Khassanov[1], T. Yernazarov[1], A.K. Imanbayeva[1]

[1]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[2]Abai Kazakh National Pedagogical University, Almaty, Kazakhstan
[3]Kazakh National Women's Teacher Training University, Almaty, Kazakhstan
*e-mail: utepova_daniya@mail.ru

Comparing the efficiency of GPU and CPU in gravitational lensing simulation

Abstract. In this study, we investigate the computational advancements in simulating gravitational lensing, particularly focusing on the Schwarzschild black hole model. The traditional approach of back ray tracing, where photons are traced back from the observer to the source, is computationally intensive, especially when aiming to achieve high-resolution images of lensing effects around black holes. By employing a numerical method that integrates the Schwarzschild metric with initial conditions derived from the observer's plane, we map the deflection of light around a black hole to generate simulated images of gravitational lensing.
The core of our study is the comparison between traditional CPU-based (Central Processing Unit-based) computations and GPU-accelerated (Graphics Processing Unit-accelerated) processes using the Numba library. Our findings reveal that GPU acceleration, with its parallel processing capabilities, significantly reduces computation time, particularly as the complexity of the simulation increases with larger grid sizes. This computational efficiency is crucial for simulations of gravitational lensing, where the number of independent calculations grows exponentially with the resolution and accuracy of the desired image.
Our study underscores the importance of leveraging GPU technology for astrophysical simulations on personal computers, offering a substantial improvement in performance over CPU-based methods.
Key words: Gravitational Lensing, GPU Parallelization, Schwarzschild Black Hole, Ray Tracing Methods, Numba Library.

Introduction

One of the most astonishing predictions of Einstein's equations corresponds to the final state of the gravitational collapse of a massive star: a Black Hole. These enigmatic objects, characterized by their intense gravitational fields from which not even light can escape, are described by solutions to Einstein's field equations, known as metrics. The Schwarzschild metric, for example, describes a non-rotating black hole [1], while the Kerr metric accounts for one that rotates [2]. These metrics play a crucial role in understanding the spacetime curvature around black holes and the resultant phenomena such as gravitational lensing.

A monumental achievement in the observation of black holes was made by the Event Horizon Telescope (EHT) collaboration, which captured the first-ever image of a supermassive black hole located at the center of the galaxy M87 [3]. This historic image, showing the shadow of the black hole surrounded by a ring of light distorted by its massive gravitational field, provides unprecedented direct visual evidence of a black hole's existence, and offers a profound confirmation of general relativity in the strong gravity regime. The EHT's success not only marks a significant milestone in observational astronomy but also highlights the critical role of simulations in interpreting the bending of light and the structure of the space around black holes.

Furthermore, the EHT has also studied the supermassive black hole at the center of our own Milky Way galaxy, known as Sagittarius A* (Sgr A*). Recent observations have similarly captured the shadow of Sgr A*, providing additional invaluable data that reinforces our understanding of black hole environments and the behavior of light in intense gravitational fields [4]. These studies of both M87

and Sgr A* enhance our comparative understanding of black hole dynamics and continue to validate the theoretical frameworks underpinning general relativity.

There are various methods to simulate the lensing effects caused by massive objects like black holes [5]. One such method is the back ray tracing method [6], which effectively reverses the path of light from the observer to the source through the curved spacetime around a massive object. This method allows for the accurate simulation of the bending of light as it passes near a massive object, providing insights into the observable phenomena associated with gravitational lensing.

In this work, we will focus on accelerating the computation of the lensing effect, which utilizes the back ray tracing method, by leveraging the parallel processing capabilities of GPUs. The use of GPUs for such simulations represents a significant advancement over traditional CPU-based computations, allowing for a substantial increase in computational efficiency and speed. By employing simple tools available in Python, we aim to demonstrate the feasibility and benefits of using GPU acceleration for the simulation of gravitational lensing effects, offering a more accessible and efficient approach for researchers and enthusiasts in the field of astrophysics.

Models

Schwarzschild Black Hole

A Schwarzschild black hole represents the simplest type of black hole, which is non-rotating and spherically symmetric [1]. We choose to focus on the Schwarzschild metric for this study because it offers the simplest model to illustrate our goals, allowing for a clear understanding of the fundamental principles without the complexities introduced by rotation or charge. It is described by the Schwarzschild metric, a solution to Einstein's field equations in general relativity. The Schwarzschild metric is given by the following equation in spherical coordinates (t, r, θ, φ)

$$ds^2 = -\left(1 - \frac{2GM}{c^2 r}\right) c^2 dt^2 + \frac{1}{1 - \frac{2GM}{c^2 r}} dr^2 + \\ + r^2 d\theta^2 + r^2 \sin^2\theta \, d\varphi^2 \quad (1)$$

where ds^2 is the spacetime interval, G is the gravitational constant, M is the mass of the black hole, c is the speed of light, and r, θ, ϕ are the radial, polar, and azimuthal coordinates, respectively. The term $2GM/c^2$ represents the Schwarzschild radius, beyond which spacetime is significantly curved by the mass of the black hole. Further, we will assume $G = c = 1$, as is commonly accepted in the GR.

The equation of motion for light, or photons, moving in the vicinity of a Schwarzschild black hole can be derived from the Schwarzschild metric. In context of the energy E and the angular momentum L the equation of motion can be written as [7]:

$$\frac{dr}{d\lambda} = v_r \quad (2.1)$$

$$\dot{r} = -[\frac{M}{r^2 - 2M} E^2 + \frac{M}{2Mr - r^2} \dot{r}^2 + \\ + (2M - r)\dot{\theta}^2 + \frac{2M - r}{r^4 \sin^2\theta} L^2] \quad (2.2)$$

$$\frac{d\theta}{d\lambda} = v_\theta \quad (2.3)$$

$$\ddot{\theta} = -\frac{2}{r}\dot{r}\dot{\theta} + \frac{\cos\theta}{r^4 \sin^3\theta} L^2 \quad (2.4)$$

$$\frac{d\varphi}{d\lambda} = \frac{L}{r^2 \sin^2\theta} \quad (2.5)$$

These five equations form the basis for implementing the back ray tracing method, where v_r and v_θ represent the velocity components r and θ directions, respectively. E denotes energy, and L denotes angular momentum, both of which are given in the form [8].

$$E = \sqrt{-g_{tt}} \quad (3)$$

$$L = p_\varphi \quad (4)$$

Back ray tracing

In contrast to natural lensing image generation, there is a noticeable difference when the lensing image is generated computationally. One of the most effective methods of generating this process using computational means is the back ray tracing method, the essence of which is the propagation of photons back in time from the observer to the source of gravity.

To obtain the lensing image of a black hole using the back ray tracing method, we position our black hole model at the origin (see Figure 1a) within Minkowski spacetime and encircle it with a virtual

background sphere, colored in four colors: red, green, yellow, and blue. The observer is located inside this background sphere and looks towards the exact point where all four colors intersect. In the absence of a black hole, the observer would see a picture as depicted in Figure 1b.

Consider two local coordinate systems: (x, y, z) and (r, θ, φ). The first system is associated with the observer, and the second with the black hole. In this setup, the $z-$axis is directed towards the black hole. At the point $z = 0$, there is an observer whose coordinates in the black hole's reference frame are $(r_O, \theta_O, \varphi_O)$.

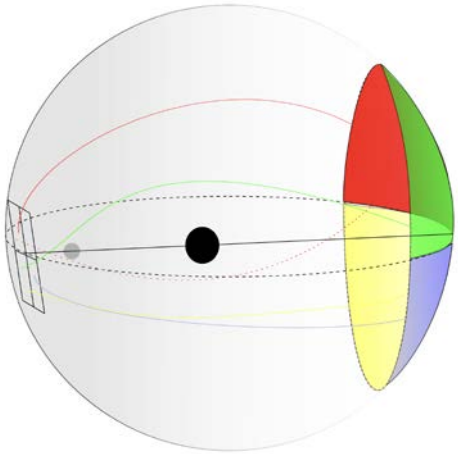

Figure 1a – Schematics of the mapping of Black Hole and observer

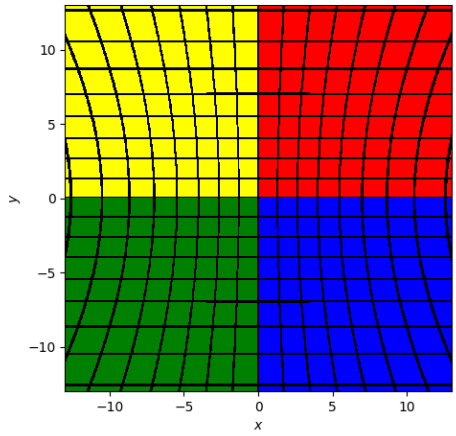

Figure 1b – Grid representing the background sphere in flat spacetime

Because the geodesic equations for light rays are represented in Schwarzschild metric coordinates (r, θ, φ), it is necessary to convert the initial conditions of the light ray on the image plane. This transformation can be done using expressions (5.1) – (5.6). More details about these expressions can be found in the work [9-12]:

$$r(0) = \sqrt{r_O^2 + x^2 + y^2} \tag{5.1}$$

$$\theta(0) = \cos^{-1}\left[\frac{r_O \cos\theta_O + y \sin\theta_O}{r(0)}\right] \tag{5.2}$$

$$\varphi(0) = \\ = \tan^{-1}\left[\frac{(y\cos\theta_O - r_O \sin\theta_O)sin\varphi_O - xcos\varphi_O}{(y\cos\theta_O - r_O \sin\theta_O)cos\varphi_O - xsin\varphi_O}\right] \tag{5.3}$$

and the initial tangent vector $(\dot{x}, \dot{y}, \dot{z}) \approx (0,0,1)$ is translated into

$$\dot{r}(0) = -\frac{r_O}{r(0)} \tag{5.4}$$

$$\dot{\theta}(0) = \frac{(x^2+y^2)\cos\theta_O - y r_O \sin\theta_O}{r^2(0)\sqrt{x^2+(y\cos\theta_O - r_O \sin\theta_O)^2}} \tag{5.5}$$

$$\dot{\varphi}(0) = \frac{x \sin\theta_O}{x^2+(y\cos\theta_O - r_O \sin^2\theta_O)} \tag{5.6}$$

Thus, by discretizing the observer plane and numerically solving equations (2.1)-(2.5) for each cell (x_i, y_i) considering the initial conditions (5.1)-(5.6), we obtain the image of the black hole as shown in Figure 2. The massive computations are performed by Runge-Kutta-4 method.

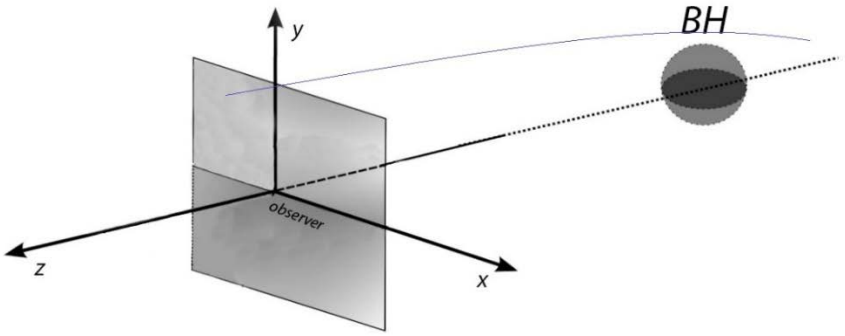

Figure 2 – Creating an image of a black hole's lensing effect from the viewpoint of a distant observer requires tracing the paths of light rays backward to each pixel on the observer's image plane

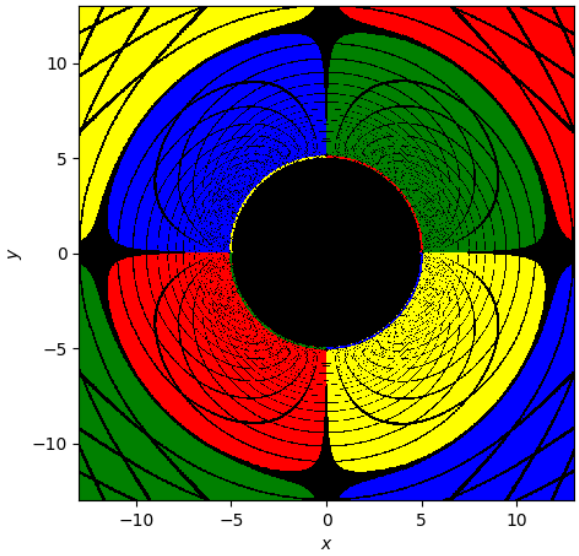

Figure 3 – Lensing of Schwarzschild Black Hole with mass $M = 1$. The observer is at initial position $r_O = 15$, in the equatorial plane ($\theta_O = \pi/2$). The background sphere is at $r = 30$. The image contains 1000×1000 cells. ($G = c = 1$, geometric units)

As anticipated, the shadow depicted in figure 3 forms a precise circle, reflecting the spherical symmetry characteristic of the Schwarzschild Black Hole. It's crucial to highlight several aspects of the image [13].

- Photons associated with large absolute values of x and y are considered direct photons. This means they do not circumnavigate the Black Hole on their way to the background. Additionally, the further we move away from the Black Hole, the more the spacetime resembles Minkowski spacetime.
- In addition to the shadow, examining Figure 3 reveals two distinct areas (an inner and an outer zone). The inner zone is associated with photons that have circled the Black Hole once.
- Should there be a white spot in the background image, as depicted in Figure 1a, the act of lensing would result in the creation of an Einstein ring. This ring would precisely align at the boundary separating the inner and outer zones mentioned previously

Results and Discussion

Acceleration by python tools

Here, we will examine the code's flowchart for obtaining the image of a lensing object and discuss modifying this code to run on a graphics processing unit (GPU) using the Numba package.

First, let's look at the flowchart of the code designed to run on a CPU (see Figure 4). As illustrated in Figure 4, the program is structured as follows: it begins with the initialization of input parameters, then calculates the components of the metric tensor (in our case, the Schwarzschild metric). Following this, a ray is launched from each cell on the observer's plane in the reverse direction with unique initial conditions, and the ray's position is calculated at each step until it either enters the event horizon area or leaves the outer sphere. The history of each ray is calculated sequentially, one after another. After determining the history of all rays, the image is formed.

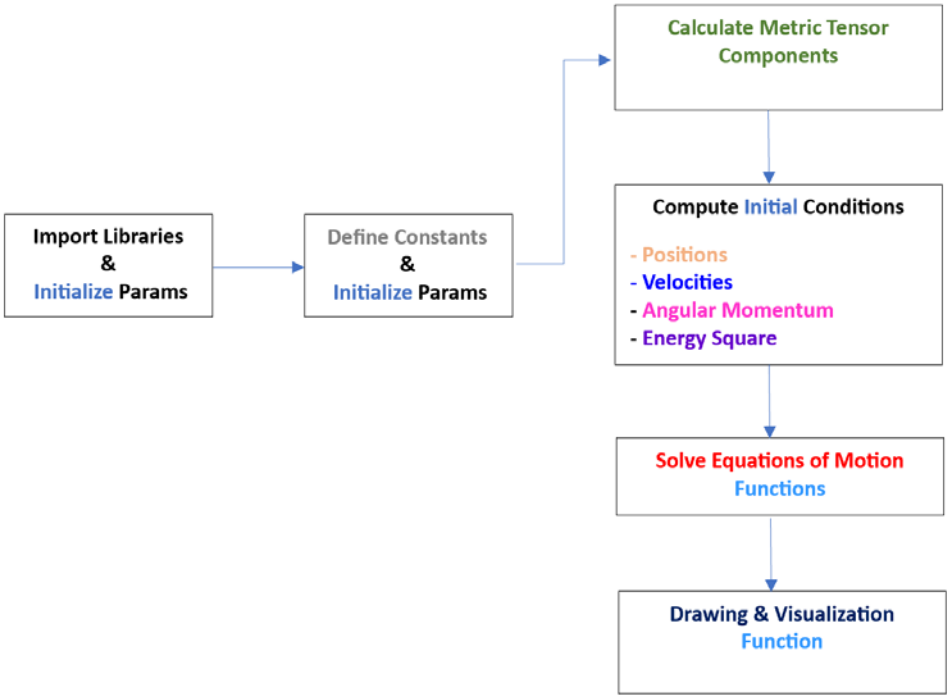

Figure 4 – A simplified block diagram representing the flow and functionalities of the code running on the CPU

From the flowchart presented, it becomes immediately clear that this task is highly suited for parallel computation since it is evident that tracking the history of each ray in parallel, rather than waiting for the completion of the previous one, is the most efficient approach. The easiest way to implement this is to use the Numba library [14]. Numba is a just-in-time compiler that accelerates Python code, especially for numerical computations, by converting it to machine-level code. It supports both CPU and GPU execution, making it ideal for parallel computing tasks. With Numba, developers can achieve significant performance improvements without major changes to their existing Python code, leveraging the power of GPUs for faster data processing and analysis. To parallelize the code and run it on a graphics processor using the Numba library, it suffices to add the @jit(target_backend='cuda') decorator to all functions where calculations are performed relative to a given ray. As a result, we obtain code designed for execution on a graphics processor, the flowchart of which is presented in Figure 5.

Next, we will present a graph comparing the computational time of the code executed on a CPU versus the code executed on a GPU using the Numba package.

The comparison plot (Fig. 6) shows that the calculation time of the original (non-parallel) method shows a linear increase on a logarithmic scale, indicating a power law relationship between grid size and calculation time. In contrast, the modified method using CUDA shows a nonlinear increase. This nonlinearity arises from the overhead associated with parallel processing and memory management on the GPU. As grid size increases, the benefits of parallelization become more pronounced, but the initial overhead and complexity of managing large data sets across multiple cores results in a nonlinear trend in computational efficiency.

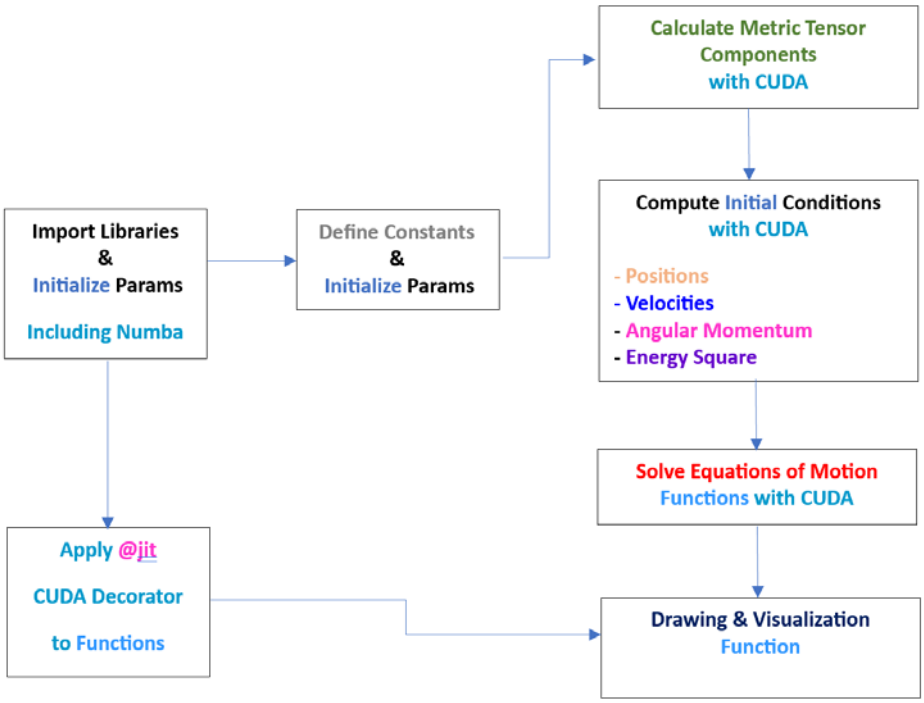

Figure 5 – A modified block diagram representing the flow and functionalities of the code running on the GPU

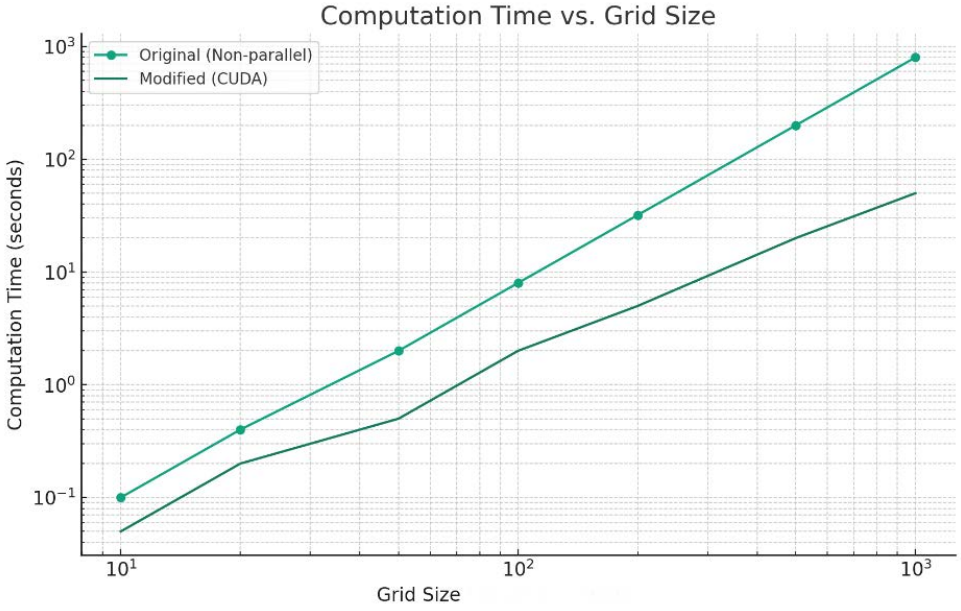

Figure 6 – The graph illustrates the comparison of computation time between the original (non-parallel) and modified (CUDA-parallel) methods when increasing the grid size from 10x10 to 1000x1000. Here both axes are presented on a logarithmic scale

Conclusion

Our study highlights the considerable computational benefits of applying GPU acceleration with the Numba library for gravitational lensing simulations. By transitioning from CPU to GPU execution, we noted a significant reduction in computation time, especially for larger grid sizes, underscoring the efficiency of parallel processing. The use of Numba stands out as the simplest method for parallelizing Python code to run on a GPU, offering a straightforward path to enhance simulation speed. This technique not only renders complex simulations more accessible but also paves the way for broader and more detailed astrophysical studies. In the future, we plan to consider various quadrupolar space-times, both static [15-21] and stationary [22] and in this case, using the GPU for calculations will be even more time efficient, since in this scenario the geodesy equations become more complex and require more computational time in the context of a single light beam.

Acknowledgments

This research has been funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP14972943).

References

1. Schwarzschild K. Über das Gravitationsfeld eines Massenpunktes nach der Einsteinschen Theorie. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften*. 1 (1916) 189-196.
2. Kerr R.P. Gravitational Field of a Spinning Mass as an Example of Algebraically Special Metrics. *Physical Review Letters*. 11(5) (1963) 237-238.
3. Event Horizon Telescope Collaboration, Kazunori Akiyama, et al. First M87 Event Horizon Telescope Results. IV. Imaging the Central Supermassive Black Hole. *Astrophys. J.* 875(1) (2019): L4. DOI: 10.3847/2041-8213/ab0e85.
4. The Event Horizon Telescope Collaboration (May 1, 2022). "First Sagittarius A* Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole in the Center of the Milky Way". *The Astrophysical Journal Letters*. 930 (2): L12.
5. Hilbert S., Hartlap J., White S.D.M., & Schneider P. Gravitational Lensing with Three-Dimensional Ray Tracing: A Comparative Study of Ray Tracing Methods. (2011) *arXiv:*1110.4894.
6. Nightingale J.W., Hayes R.G., Kelly A., Amvrosiadi, A., Etherington A., He Q., Li N., Cao X., Frawley J., Cole S., Enia A., Frenk C.S., Harvey D.R., Li R., Massey R.J., Negrello M., Robertson A. PyAutoLens: Open-Source Strong Gravitational Lensing. *Journal of Open Source Software*. 6(58) (2021), 2525. DOI: 10.21105/joss.02825
7. Weinberg, S. Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity. *John Wiley & Sons*. (1972) ISBN 0471925675. P.185.
8. Velásquez-Cadavid J.M., Arrieta-Villamizar J.A., Lora-Clavijo F.D. et al. OSIRIS: a new code for ray tracing around compact objects. *Eur. Phys. J.* 82 (2022) 103 DOI: 10.1140/epjc/s10052-022-10054-0.
9. Younsi Z. General relativistic radiative transfer in black hole systems. Ph. D. Thesis. April (2014).
10. Younsi Z., Zhidenko A., Rezzolla L., Konoplya R., and Mizuno Y. New method for shadow calculations: Application to parametrized axisymmetric black holes. *Phys. Rev. D*. 94(8) (2014) 084025. DOI: 10.1103/PhysRevD.94.084025
11. Hung-Yi Pu, Kiyun Yun, Ziri Younsi, and Suk-Jin Yoon. Odyssey: A Public GPU-Based Code for General-Relativistic Radiative Transfer in Kerr Spacetime. *Astrophys. J.* 820(2) (2016) 105. DOI: 10.3847/0004-637X/820/2/105.
12. Lin F.L., Patel A. & P, H.Y. Black hole shadow with soft hairs. *J. High Energ. Phys.* 117 (2022). DOI: 10.1007/JHEP09(2022)117.
13. Pedro V.P. Cunha, Nelson A.Eiró, Carlos A.R. Herdeiro, José P.S. Lemos. Lensing and shadow of a black hole surrounded by a heavy accretion disk., *Journal of Cosmology and Astroparticle Physics*. (2020), March. DOI: 10.1088/1475-7516/2020/03/035.
14. Numba, High-Performance Python with CUDA Acceleration: (2013). https://developer.nvidia.com/blog/numba-python-cuda-acceleration/.
15. Boshkayev K., Gasperın E., Gutierrez- Pineres A.C., Quevedo H., Toktarbay S. Motion of test particles in the field of a naked singularity, *Phys. Rev. D*. 93 (2016) 024024. DOI: 10.1103/PhysRevD.93.024024.
16. Abishev M., Beissen N., Belissarova F., Boshkayev K., Mansurova A., Muratkhan A, Quevedo H., Toktarbay S. Approximate perfect fluid solutions with quadrupole moment. *International Journal of Modern Physics D*. 30(13) (2021) 2150096. DOI: 10.1142/S0218271821500966.
17. Toktarbay S., Quevedo H., Abishev M., Muratkhan A. Gravitational field of slightly deformed naked singularities, *European Physical Journal C*. 82(4) (2022). DOI: 10.1140/epjc/s10052-022-10230-2.
18. Muratkhan A., Orazymbet A., Zhakipova M., Assylbek M., Toktarbay, S. A shadows from the static black hole mimickers, *International Journal of Mathematics and Physics*. 13(2) 2023 44–49. DOI: 10.26577/ijmph.2022.v13.i2.06.
19. Beissen N., Utepova D., Abishev M., Quevedo H., Khassanov M., Toktarbay S. Gravitational refraction of compact objects with quadrupoles, Symmetry. (2023) 15(614). DOI: 10.3390/sym15030614.
20. Beissen N., Utepova D., Muratkhan A., Orazymbet A., Khassanov M., Toktarbay S. Application of GBT theorem for gravitational deflection of light by compact objects, *Recent Contributions to Physics*. 1(84) (2023) 15-21. DOI: 10.26577/RCPh.2023.v84.i1.02.

21. Quevedo H. Mass Quadrupole as a Source of Naked Singularities. *Int. J. Mod. Phys. D*. 20 (2011) 1779-1787. DOI: 10.1142/S0218271811019852.

22. Toktarbay S., Quevedo H. A stationary q-metric // *Gravitation and Cosmology*. Vol. 20, № 4. (2014) 252-254.

Information about authors:

Beissen N.A. – Candidate of Physical and Mathematical Sciences, associate Professor, Dean of the Faculty of Physics and Technology Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: nurzada.beissen@kaznu.edu.kz

Utepova D.S. – PhD student, Al-Farabi Kazakh National University, Abai Kazakh National Pedagogical University, Almaty, Kazakhstan, e-mail: utepova_daniya@mail.ru

Kossov V.N. – Doctor of Physical and Mathematical Sciences, Professor, Abai Kazakh National Pedagogical University, Almaty, Kazakhstan, e-mail: kosov_vlad_nik@list.ru

Toktarbay S. – PhD, Al-Farabi Kazakh National University, Kazakh National Women's Teacher Training University, Almaty Kazakhstan, e-mail: Saken.Toktarbay@kaznu.edu.kz

Khassanov M.K. – PhD, Al-Farabi Kazakh National University, Almaty Kazakhstan, e-mail: Manas.Khassanov@kaznu.edu.kz

Yernazarov T. – PhD student, Al-Farabi Kazakh National University, Almaty Kazakhstan, e-mail: Tursynbek.Yernazarov@kaznu.edu.kz

Imanbayeva A.K. – Candidate of Physical and Mathematical Sciences, Al-Farabi Kazakh National University, Almaty Kazakhstan, e-mail: akmaral@physics.kz

Received 29 March 2024
Accepted 23 May 2024