OCTOPUS

Welcome to OCTOPUS, a relativistic ray-tracing code designed for asymptotically flat, spherically symmetric spacetimes. This code enables simulations of black hole images, redshift factor images, gravitational lensing effects, light curves, and other related phenomena. The code consists of 9 Fortran source files: dynamic.f90, emission.f90, functions.f90, initial condition.f90, mainprogram.f90, method.f90, model.f90, operations.f90, and parameters.f90.

To implement a new black hole model, users need to:

① Provide the metric potential and its first three derivatives in the designated section of model.f90

② Define the corresponding metric parameters in the metric_parameters module within parameters.f90

The parameters in parameters.f90 are detailed in the following table. For complete implementation guidelines and theoretical background, please consult our preprint at arxiv.org:

"OCTOPUS: A Versatile, User-Friendly, and Extensible Public Code for General-Relativistic Ray-Tracing in Spherically Symmetric and Static Spacetimes".

Module	Variable in Code	Variable in Article	Description
metric_parameters	r_s	r_s	Dark matter halo parameters. It can be modified to other variables based on the metric, but please ensure the variable names do not conflict with those in other modules.
	rho_s	ρ_s	
	math_pi	π	This variable must be retained, even if it does not appear in other metric potential.
black_hole_parameters	mass_ratio	Γ	Mass ratio of the black hole to the solar mass.
	distance_obs	D	Observation distance in Mpc. Please note that this observation distance is not used for ray-tracing, but for calculating the angular diameter of the black hole shadow.
gravitational_parameters	zeta	ζ	Used for calculating gravitational radiation.
	iota	ι	

	eta	η	
	light_speed	c	
	gravitational_constant	G	
	sun_mass	$M_\odot$	
observation_parameters	t_obs	t	Initial time of the light ray, set to 0 by default.
	r_obs	r_{obs}	Distance from the observer to the black hole.
	theta_obs	ω	Observation inclination (Viewing angle)
	phi_obs	φ_{obs}	Observation azimuth, set to 0 by default.
	resolution_x		Parameters of the observation screen.

	resolution_y		A higher resolution will result in significantly increased computation time. These quantities, though not explicitly denoted with specific symbols in the paper, are straightforward to understand.
	x_ini, x_end		
	y_ini, y_end		
raytracing_parameters	hit_error	ε	A small value used to determine if a ray crosses the event horizon, typically set to 0.0005 to avoid numerical singularities.
	observer		The radius of the celestial sphere in gravitational lensing image simulations.
	ini_step	h_0	The initial step size fed into the RKF56 for null geodesic integration, used to improve RKF56's convergence speed.
	step_ratio	n	Scaling parameter for h_0.
accretion_disk_parameters	r_inner	r_{in}	Inner boundary of the accretion disk.
	r_outer	r_{out}	Outer boundary of the accretion disk.

	disk_model		Used to control the inner and outer boundaries of the accretion disk. disk_model = 1, 2, 3 correspond to the inner boundary being located at the event horizon, critical photon orbit, and innermost stable circular orbit, respectively. When disk_model = 4, the user-input r_{in} takes effect. This parameter does not affect r_{out}.
	max_hit	$N_{\max}$	The maximum number of intersections between the light ray and the radiation source. It must satisfy $0 < N_{\max} \leq 4$.
lightsource_parameters	source_x source_y source_z	(x', y', z')	Coordinates of the point light source in the black hole coordinate system.
	source_radius	r_{source}	Radius of the point light source.
	j_0	j_0	Scaling factors for the emission and absorption coefficients in the radiative transfer process, used to control the strength of radiation or absorption. These only take effect when calculating the specific intensity of point sources.
	alpha_0	α_0	

particle_parameters	t_ini	t_0	Initial conditions for simulating timelike geodesics.
	r_ini	r_0	
	theta_ini	θ_0	
	phi_ini	φ_0	
	pt_ini	p_{t0}	
	pphi_ini	$p_{\varphi 0}$	
	interval		Timelike geodesics are simulated using the fixed-step RK6 method with a step-size of "interval".
	sample		During the simulation of timelike geodesics, data is recorded every "sample" points.
	counts		Number of sampled points in the timelike geodesic. The total integration time is calculated as t_e = interval × sample × counts

	path_class		Controls the initialization method for timelike geodesics. The values path_class = 1, 2, 3, 4, 5 correspond to **Method** 1, 2, 3, 4, 5 in the paper, respectively.
task_parameters	task_model		task_model controls the type of task performed by the CODE. ● task_model = 1: Simulates the light curve of a hot-spot. Generates data file “time sequence.dat”, visualized using “light curve plot.py”. ● task_model = 2: Simulates a hot-spot animation. Generates data file “hot-spot image.dat”, visualized using “lensing animation plot.py”. ● task_model = 3: Simulates timelike geodesics. Generates data files “particle path.dat” and “GW emission.dat”, visualized using "path plot.py" and “GW plot.py” respectively. ● task_model = 4: Simulates the black hole critical curve (shadow). Generates data file “shadow.dat”, visualized using “shadow profile plot.py”. ● task_model = 5: Simulates the accretion disk structure (not

			an image). Generates data file “disk profile.dat”, visualized using “disk profile plot.py”. ● task_model = 6: Simulates the number of times light rays cross the accretion disk. Generates data file "hit.dat", visualized using “hit plot.py”. ● task_model = 7: Simulates redshift factor images. Generates data files “redshift1.dat”, “redshift2.dat”, “redshift3.dat”, and “redshift4.dat”, corresponding to the 1st, 2nd, 3rd, and 4th-order redshift factors respectively, visualized using "redshift plot.py". ● task_model = 8: Simulates black hole images. Generates data files “image86.dat” and “image230.dat”, corresponding to the 86 GHz and 230 GHz images respectively, visualized using “image plot.py”. ● task_model = 9: Simulates the gravitational lensing image of a point source. Generates data file “lensing.dat”, visualized using “lensing plot.py”. ● task_model = 10: Simulates the specific intensity image of a

			point source. Generates data file "hot-spot image.dat", visualized using “image plot.py”. ● task_model = 11: Calculates the Hamiltonian error of the rays. Generates data file “hamiltonian error.dat”, visualized using “error plot.py”. Please note that when plotting, be sure to check the consistency between the plotting resolution and the simulation resolution.
	cpu		Number of CPU cores/threads used for parallel computation.