Code Analysis and Explanation: Search for Feasible Planetary Orbit under Extreme Relativistic Conditions

This Python code performs a numerical search to determine whether a planet can orbit a maximally spinning Kerr black hole in a way that satisfies two extreme conditions simultaneously:

* A time dilation factor of approximately $\gamma = 3.154 \times 10^{12}$, meaning that 1 second on the planet corresponds to ~100,000 years on Earth.
* An orbital period of $T = 9.8156 \times 10^{19}$ seconds, or roughly 3.1 trillion Earth years.

The code aims to validate the feasibility of such an orbit based on the framework of General Relativity and Kerr spacetime. It does so by optimizing over the black hole mass (M) and scanning a range of orbital radii (r) for each M, calculating relativistic time dilation and orbital period at each point.

Step-by-Step Functional Description

1. Constants and Inputs

The code sets fundamental physical constants:
* Gravitational constant G
* Speed of light c

Target values:
* Time dilation factor γ_target $= 3.154 \times 10^{12}$
* Orbital period T_target $= 9.8156 \times 10^{19}$ seconds

2. Kerr Spin and Orbital Functions

* kerr_spin(M): Calculates the Kerr spin parameter $a = GM/c^2$ assuming maximal rotation.
* orbital_velocity(M, r): Computes orbital velocity under Newtonian approximation for circular orbits.
* gamma_kerr(M, r): Returns approximate gravitational time dilation factor from a circular equatorial orbit around a Kerr black hole.
* orbital_period(M, r): Approximates the relativistic orbital period using a corrected Keplerian formula.
* isco_radius(M): Returns a conservative approximation of the innermost stable circular orbit (ISCO) for a prograde orbit.

3. Error Evaluation

* log_error(M_log10): This core evaluation function:
  - Converts log-scaled mass back to linear scale.
  - Scans a logarithmic range of radii starting just beyond ISCO.
  - For each pair (M, r), calculates γ and T.
  - Compares both with target values and calculates squared log-error.
  - Retains the configuration with the minimum combined error.

4. Optimization Routine

* minimize_scalar(...): Searches over $\log_{10}(M)$ from 10^{30} kg to 10^{60} kg to find the best mass yielding minimal error.
  For each candidate mass, the best orbital radius is computed by scanning 500 points logarithmically spaced from just beyond ISCO up to 10^{30} meters.

5. Output

* If a valid configuration is found, it prints:
  - Optimal mass M
  - Best radius r
  - Computed time dilation factor
  - Computed orbital period
  - Combined error metric
* If no viable configuration satisfies both constraints, it reports failure.

Significance and Findings

This code performs an exhaustive yet stable numerical scan using logarithmic scaling to avoid instability. The results consistently show that:
* No combination of black hole mass and radius satisfies both extreme time dilation and long orbital period.
* Strong dilation occurs at small radii with short orbital periods.
* Long orbital periods occur at large radii with negligible time dilation.

Hence, the proposed parameters are incompatible under known Kerr geometry physics, even with maximal spin.

Dependencies

The code uses:
* numpy for numerical operations.
* scipy.optimize.minimize_scalar for scalar optimization.
* matplotlib.pyplot (optional for plotting).