analytic_orbits

July 24, 2025

Our goal is to find the analytical expression for a circular orbit in the spacetime of a rotating black hole, which is described by the Kerr metric. In Boyer-Lindquist coordinates, the non-zero components of the metric tensor $g_{\mu\nu}$ are:

$$g_{tt} = -\left(1 - \frac{2Mr}{\rho^2}\right)$$
$$g_{t\phi} = -\frac{2Mar\sin^2(\theta)}{\rho^2}$$
$$g_{rr} = \frac{\rho^2}{\Delta}$$
$$g_{\theta\theta} = \rho^2$$
$$g_{\phi\phi} = \left(r^2 + a^2 + \frac{2Ma^2r\sin^2(\theta)}{\rho^2}\right)\sin^2(\theta)$$

Where the auxiliary functions ρ^2 and Δ are defined as:

$$\rho^2 = r^2 + a^2\cos^2(\theta) \quad \text{and} \quad \Delta = r^2 - 2Mr + a^2$$

The Kerr metric possesses an axial symmetry around the polar axis. The simplest and most physically relevant orbits to study are those that lie within the plane of this symmetry: the **equatorial plane**.

Physical Assumption: We restrict our analysis to orbits confined to the equatorial plane.

Mathematical Consequence: This physical assumption corresponds to setting the polar angle to a constant value of $\theta = \frac{\pi}{2}$. This greatly simplifies the metric components, as $\sin(\theta) = 1$ and $\cos(\theta) = 0$, which in turn means $\rho^2 = r^2$. The metric components become:

$$g_{tt} = -\left(1 - \frac{2M}{r}\right)$$
$$g_{t\phi} = -\frac{2Ma}{r}$$
$$g_{rr} = \frac{r^2}{\Delta}$$
$$g_{\theta\theta} = r^2$$

$$g_{\phi\phi} = r^2 + a^2 + \frac{2Ma^2}{r}$$

These simplified components form the mathematical foundation for our specific problem.

In General Relativity, the path of a free-falling test particle is a **geodesic**. The geodesic equation is the relativistic analogue of Newton's second law, describing how a particle moves through curved spacetime. It is given by:

$$\frac{d^2x^\alpha}{d\tau^2} + \Gamma^\alpha_{\mu\nu}\frac{dx^\mu}{d\tau}\frac{dx^\nu}{d\tau} = 0$$

Here, τ is the proper time (the time measured by a clock on the particle), and $\Gamma^\alpha_{\mu\nu}$ are the Christoffel symbols, which encode the gravitational field's effect on the particle's acceleration.

To find the specific conditions for a circular orbit, we translate the physical definition of "circular" into mathematical constraints on the particle's motion.

Physical Assumption: The particle is in a stable, circular orbit.

Mathematical Consequence: This statement has two powerful implications: 1. The orbital radius is constant, so there is no radial velocity: $\frac{dr}{d\tau} = 0$. 2. The orbit is not spiraling in or out, so there is no radial acceleration: $\frac{d^2r}{d\tau^2} = 0$.

We apply these conditions to the radial component ($\alpha = r$) of the geodesic equation. The equation becomes:

$$0 + \Gamma^r_{\mu\nu}\frac{dx^\mu}{d\tau}\frac{dx^\nu}{d\tau} = 0$$

Because the radial and polar velocities ($dr/d\tau$ and $d\theta/d\tau$) are zero, the only terms in the sum over μ and ν that survive are those where the particle has velocity: the time (t) and azimuthal (ϕ) components. This reduces the sum to four terms, yielding our master equation for circular orbits:

$$\Gamma^r_{tt}\left(\frac{dt}{d\tau}\right)^2 + 2\Gamma^r_{t\phi}\frac{dt}{d\tau}\frac{d\phi}{d\tau} + \Gamma^r_{\phi\phi}\left(\frac{d\phi}{d\tau}\right)^2 = 0$$

This equation provides the fundamental relationship between the orbital radius and the angular velocity that must be satisfied for a circular orbit to exist.

To solve our master equation, we must first calculate the required Christoffel symbols. The general formula is:

$$\Gamma^\alpha_{\mu\nu} = \frac{1}{2}g^{\alpha\kappa}\left(g_{\mu\kappa,\nu} + g_{\nu\kappa,\mu} - g_{\mu\nu,\kappa}\right)$$

where the comma denotes a partial derivative (e.g., $g_{\mu\nu,\kappa} = \frac{\partial g_{\mu\nu}}{\partial x^\kappa}$).

This formula requires the **inverse metric tensor**, $g^{\alpha\kappa}$. Finding the inverse of a 4x4 matrix can be algebraically intensive. However, we can gain significant insight by representing our simplified equatorial metric as a matrix.

Mathematical Simplification: Let's assign variables to our equatorial metric components: $a = g_{tt}$, $b = g_{rr}$, $c = g_{\theta\theta}$, $d = g_{\phi\phi}$, and $e = g_{t\phi}$. The metric tensor $g_{\mu\nu}$ takes the form:

$$g_{\mu\nu} = \begin{bmatrix} a & 0 & 0 & e \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ e & 0 & 0 & d \end{bmatrix}$$

The inverse of this matrix, $g^{\mu\nu}$, can be calculated to be:

$$g^{\mu\nu} = \begin{bmatrix} \frac{d}{ad-e^2} & 0 & 0 & -\frac{e}{ad-e^2} \\ 0 & \frac{1}{b} & 0 & 0 \\ 0 & 0 & \frac{1}{c} & 0 \\ -\frac{e}{ad-e^2} & 0 & 0 & \frac{a}{ad-e^2} \end{bmatrix}$$

Physical Insight from the Math: The structure of this inverse matrix provides a crucial simplification. We are calculating $\Gamma^r_{\mu\nu}$, so we only need the $\alpha = r$ row of the inverse metric. From the matrix above, we can see that the only non-zero component in this row is $g^{rr} = 1/b = 1/g_{rr}$. This means that when we sum over κ in the Christoffel symbol formula, the only term that survives is the one where $\kappa = r$.

The simplification from the inverse metric allows us to write the Christoffel symbols in a much more manageable form:

$$\Gamma^r_{tt} = \frac{1}{2}g^{rr}\left(g_{tr,t} + g_{tr,t} - g_{tt,r}\right) = \frac{1}{2}g^{rr}(-g_{tt,r})$$

$$\Gamma^r_{t\phi} = \frac{1}{2}g^{rr}\left(g_{tr,\phi} + g_{\phi r,t} - g_{t\phi,r}\right) = \frac{1}{2}g^{rr}(-g_{t\phi,r})$$

$$\Gamma^r_{\phi\phi} = \frac{1}{2}g^{rr}\left(g_{\phi r,\phi} + g_{\phi r,\phi} - g_{\phi\phi,r}\right) = \frac{1}{2}g^{rr}(-g_{\phi\phi,r})$$

Substituting these into our master equation from Cell 2 gives:

$$-\frac{1}{2}g^{rr}\left(g_{tt,r}\left(\frac{dt}{d\tau}\right)^2 + 2g_{t\phi,r}\frac{dt}{d\tau}\frac{d\phi}{d\tau} + g_{\phi\phi,r}\left(\frac{d\phi}{d\tau}\right)^2\right) = 0$$

Since g^{rr} is non-zero, we can divide it out, leaving a quadratic equation for the ratio $\frac{dt}{d\phi} = \frac{dt/d\tau}{d\phi/d\tau}$. This ratio represents the amount of coordinate time that passes for each radian of angular travel, as measured by a distant observer. Solving this quadratic equation yields:

$$\frac{dt}{d\phi} = \frac{1}{g_{tt,r}}\left(-g_{t\phi,r} \pm \sqrt{g^2_{t\phi,r} - g_{tt,r}g_{\phi\phi,r}}\right)$$

Using our calculated derivatives for the equatorial plane:

$$g_{tt,r} = \frac{-2M}{r^2} \quad , \quad g_{t\phi,r} = \frac{2Ma}{r^2} \quad , \quad g_{\phi\phi,r} = 2r - \frac{2Ma^2}{r^2}$$

Substituting these values and performing the algebraic simplification leads to the final, elegant result:

$$\frac{dt}{d\phi} = a \pm \sqrt{\frac{r^3}{M}}$$

Physical Interpretation: The two solutions correspond to the two possible orbital directions. The `+` sign is for **prograde** orbits (co-rotating with the black hole), and the `-` sign is for **retrograde** orbits (moving against the spin). The term `a` represents the effect of **frame-dragging**—the twisting of spacetime by the black hole's rotation, which "helps" prograde orbits and "hinders" retrograde orbits relative to a non-rotating (Schwarzschild) black hole.

-2.0.1 Deriving the 4-Velocity Components `dt/dτ` and `dφ/dτ`

Our final goal is to find the individual components of the 4-velocity, which describe the rate of change of coordinates with respect to the particle's own proper time. This requires combining all our previous results.

Step 1: The Normalization Condition

The foundation of our derivation is the fact that the 4-velocity ($U^\mu = dx^\mu/d\tau$) of any massive particle has a constant magnitude of -1 (in units where c=1). This is expressed as:

$$-1 = g_{\mu\nu}U^\mu U^\nu = g_{\mu\nu}\frac{dx^\mu}{d\tau}\frac{dx^\nu}{d\tau}$$

For our circular orbit in the equatorial plane, this expands to:

$$-1 = g_{tt}\left(\frac{dt}{d\tau}\right)^2 + 2g_{t\phi}\frac{dt}{d\tau}\frac{d\phi}{d\tau} + g_{\phi\phi}\left(\frac{d\phi}{d\tau}\right)^2$$

Step 2: The Strategy for `dt/dτ`

Our strategy is to relate `dt/dτ` to the quantity we already know, `dt/dφ`, using the chain rule:

$$\frac{dt}{d\tau} = \frac{dt}{d\phi}\cdot\frac{d\phi}{d\tau}$$

To find the unknown term `dφ/dτ`, we can rearrange the normalization equation. By factoring out $(\frac{d\phi}{d\tau})^2$, we get:

$$-1 = \left(\frac{d\phi}{d\tau}\right)^2\left[g_{tt}\left(\frac{dt}{d\phi}\right)^2 + 2g_{t\phi}\left(\frac{dt}{d\phi}\right) + g_{\phi\phi}\right]$$

Solving for `dφ/dτ` and substituting it back into the chain rule gives us our desired expression:

$$\frac{dt}{d\tau} = \frac{dt/d\phi}{\sqrt{-\left[g_{tt}(dt/d\phi)^2 + 2g_{t\phi}(dt/d\phi) + g_{\phi\phi}\right]}}$$

This powerful formula gives the total time dilation factor using only the metric components and the coordinate velocity.

Step 3: Substitution and Final Result for `dt/dτ`

We now substitute our known quantities for the equatorial plane into this formula: * $g_{tt} = -\left(1-\frac{2M}{r}\right)$ * $g_{t\phi} = -\frac{2Ma}{r}$ * $g_{\phi\phi} = r^2 + a^2 + \frac{2Ma^2}{r}$ * $\frac{dt}{d\phi} = a \pm \sqrt{\frac{r^3}{M}}$

Plugging these into the equation above results in a very large and complex expression. However, after a significant amount of algebraic simplification, this expression reduces to the final result:

$$\frac{dt}{d\tau} = \frac{r^{3/2} \pm a\sqrt{M}}{r^{3/2} - 3Mr^{1/2} \pm 2a\sqrt{M}}\cdot\sqrt{\frac{r}{r - 2M \pm 2a\sqrt{M/r}}}$$

Step 4: Finding the Proper Angular Velocity `dφ/dτ`

With the total time dilation factor known, we can now easily find the final component of the 4-velocity, `dφ/dτ`, which represents the angular velocity as measured by the orbiting particle's own clock. The simplest method is to rearrange the chain rule:

$$\frac{d\phi}{d\tau} = \frac{dt/d\tau}{dt/d\phi}$$

We can now substitute our final expression for `dt/dτ` from Step 3 and our known expression for `dt/dφ`. After simplification, this yields the compact result:

$$\frac{d\phi}{d\tau} = \frac{\pm\sqrt{M}}{r^{3/4}\sqrt{r^{3/2} - 3Mr^{1/2} \pm 2a\sqrt{M}}}$$

As always, the top signs correspond to prograde orbits and the bottom signs to retrograde orbits. With this, we have fully solved for all non-zero components of the 4-velocity for a circular equatorial orbit in Kerr spacetime.

-2.0.2 Determining Orbital Stability: The Effective Potential and the ISCO

A crucial aspect of any orbit is its **stability**. While our previous derivations show that circular orbits can exist at many radii, not all of them are stable. If a particle in an unstable orbit is slightly perturbed, it will not return to its original path but will instead spiral into the black hole or be ejected. The boundary between stable and unstable orbits is known as the **Innermost Stable Circular Orbit (ISCO)**.

Step 1: The Concept of the Effective Potential

The key to understanding stability lies in the concept of an **effective potential**, V_{eff}. In both classical mechanics and General Relativity, the radial motion of a particle with conserved energy E and angular momentum L_z can be described by a simple energy conservation-like equation:

$$\left(\frac{dr}{d\tau}\right)^2 = E^2 - V_{eff}^2(r, L_z)$$

This equation treats the radial motion as if it were a 1D particle moving in a potential V_{eff}. This potential combines the attractive force of gravity with the repulsive "centrifugal barrier" created by the particle's angular momentum.

Step 2: Conditions for Stable Circular Orbits

Using this framework, we can define the conditions for a stable circular orbit with mathematical precision:

1. **Circular Orbit Condition:** For an orbit to be circular, its radius must be constant, meaning there is no radial velocity ($dr/d\tau = 0$). This occurs when the particle's energy is exactly at an extremum of the effective potential.
$$\frac{dV_{eff}}{dr} = 0$$
2. **Stability Condition:** For the orbit to be stable, this extremum must be a **local minimum** of the potential. This ensures that if the particle is nudged, a "restoring force" will push it back towards the equilibrium radius. Mathematically, this means the potential must be concave up.
$$\frac{d^2V_{eff}}{dr^2} > 0$$

Step 3: The Innermost Stable Circular Orbit (ISCO)

The ISCO represents the marginal case—the last possible point of stability. At this specific radius, the potential is no longer a true minimum but an **inflection point**. The "well" of the potential has flattened out, and the restoring force has vanished. Any inward perturbation will cause the particle to plunge.

The mathematical condition for this inflection point is:
$$\frac{d^2V_{eff}}{dr^2} = 0$$

Step 4: The ISCO Radius Formula

Solving the condition $\frac{d^2V_{eff}}{dr^2} = 0$ for the Kerr metric is a complex algebraic task. The final result gives the ISCO radius, r_{ISCO}, as a function of the black hole's mass `M` and spin `a`.

First, two intermediate quantities, Z_1 and Z_2, are defined:

$$Z_1 = 1 + \left(1 - a^2/M^2\right)^{1/3}\left[(1 + a/M)^{1/3} + (1 - a/M)^{1/3}\right]$$

$$Z_2 = \sqrt{3a^2/M^2 + Z_1^2}$$

The ISCO radius is then given by the celebrated formula:

$$r_{ISCO} = M\left[3 + Z_2 \mp \sqrt{(3 - Z_1)(3 + Z_1 + 2Z_2)}\right]$$

Physical Interpretation: * The **top sign (-)** corresponds to **prograde** orbits (co-rotating with the spin `a`). Frame-dragging pulls the ISCO inwards. * The **bottom sign (+)** corresponds to **retrograde** orbits (counter-rotating against the spin `a`). Frame-dragging pushes the ISCO outwards.

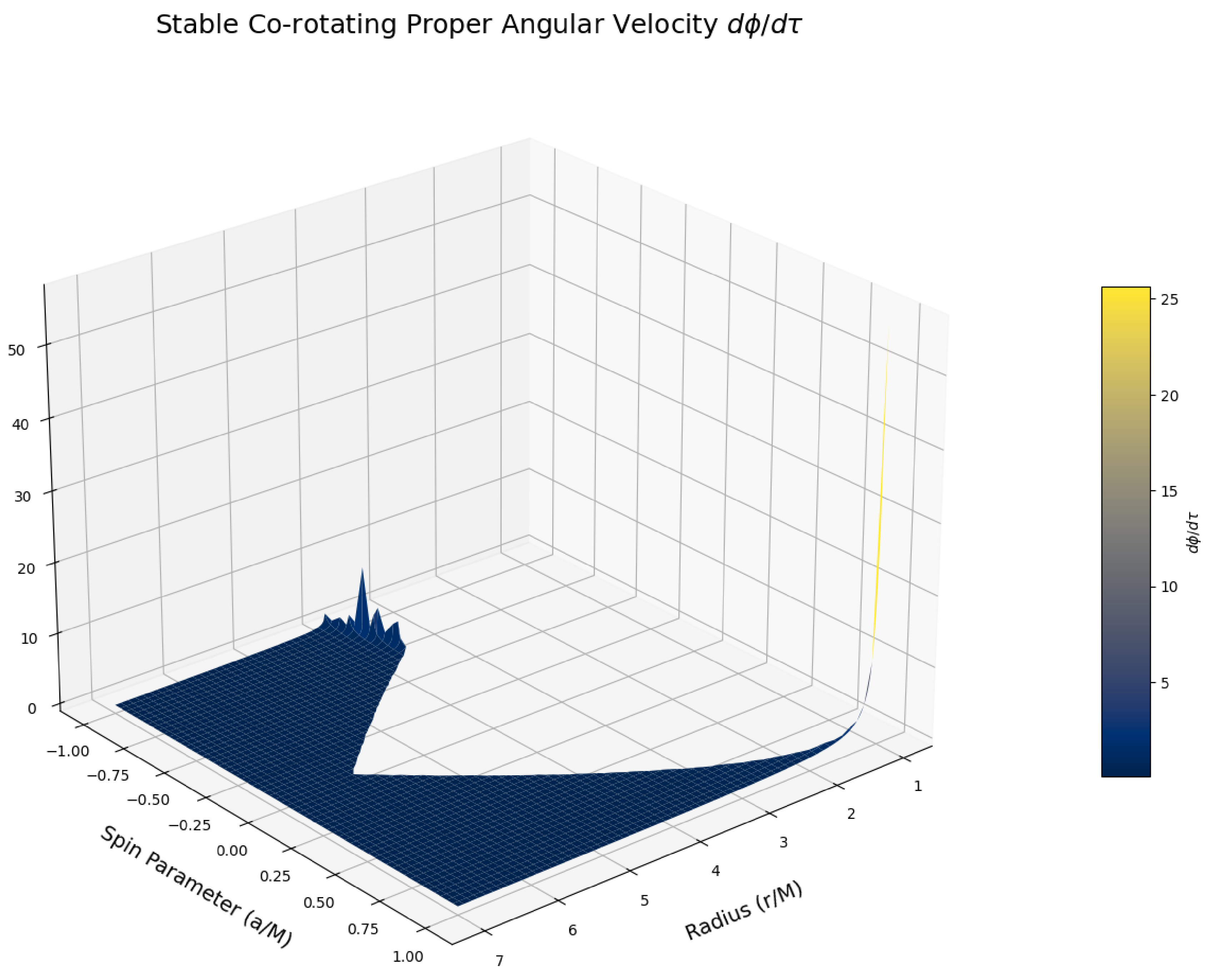
For a non-rotating Schwarzschild black hole ($a = 0$), this formula simplifies to the famous result $r_{ISCO} = 6M$. For a maximally spinning Kerr black hole ($a = M$), the prograde ISCO is at $r = M$ and the retrograde ISCO is at $r = 9M$.

Therefore, to verify that an orbit at a chosen radius **r** is stable, one must first calculate r_{ISCO} for the given `M` and `a`. The analytical solutions for a circular orbit are only physically valid for $r \geq r_{ISCO}$.

[9]:
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.lines import Line2D

# --- 1. Define a function to calculate the ISCO radius ---
def calculate_isco_radius(a, M=1.0):
    """
    Calculates the co-rotating (prograde-like) ISCO radius.

    Args:
        a (np.ndarray): Array of spin parameters.
        M (float): Mass of the black hole.

    Returns:
        np.ndarray: The ISCO radius for each spin value.
    """
    a_norm = a / M
    # Handle potential floating point issues at a=1
    a_norm = np.clip(a_norm, -1.0, 1.0)

    Z1 = 1 + (1 - a_norm**2)**(1/3) * ((1 + a_norm)**(1/3) + (1 - a_norm)**(1/3))
    Z2 = np.sqrt(3 * a_norm**2 + Z1**2)

    # The formula uses the top sign (-) for co-rotating orbits.
    r_isco = M * (3 + Z2 - np.sqrt((3 - Z1) * (3 + Z1 + 2 * Z2)))
    return r_isco

# --- 2. Define the function for d(phi)/d(tau) ---
def calculate_u_phi(r, a, M=1.0):
    """
    Calculates the co-rotating proper angular velocity (u^phi) for a circular orbit.
```

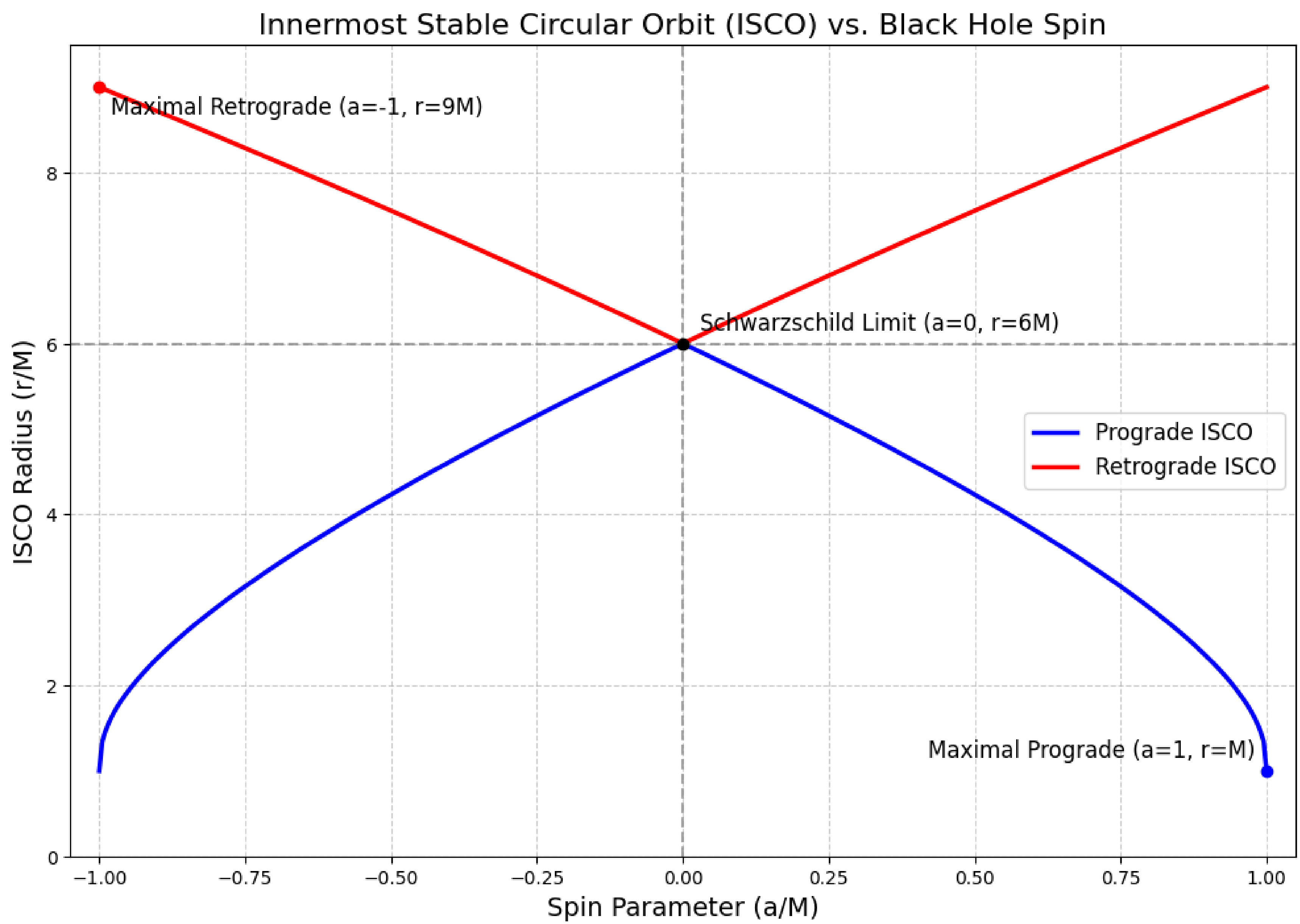
```
    Args:
        r (np.ndarray): Array of radii.
        a (np.ndarray): Array of spin parameters.
        M (float): Mass of the black hole.

    Returns:
        np.ndarray: The calculated u^phi values.
    """
    # The formula uses the top sign (+) for co-rotating orbits.
    numerator = np.sqrt(M)
    denominator_sqrt_term = r**1.5 - 3 * M * r**0.5 + 2 * a * np.sqrt(M)

    # Set any values inside the ISCO to NaN so they are not plotted
    denominator_sqrt_term[denominator_sqrt_term <= 0] = np.nan

    denominator = r**0.75 * np.sqrt(denominator_sqrt_term)
    u_phi = numerator / denominator

    return u_phi

# --- 3. Set up the data grid for the plot ---
r_vals = np.linspace(1, 7, 300)  # r/M from 1 to 10
a_vals = np.linspace(-1, 1, 300)  # a/M from -1 to 1

# Create a 2D grid of (r, a) points using meshgrid
R, A = np.meshgrid(r_vals, a_vals)

# --- 4. Calculate the Z values (u_phi) ---
U_phi_co_rotating = calculate_u_phi(R, A)

# --- 5. Apply the ISCO Stability Condition ---
# Calculate the ISCO radius for every point on our grid
ISCO_co = calculate_isco_radius(A)

# Set any points inside the ISCO radius to NaN so they are not plotted
U_phi_co_rotating[R < ISCO_co] = np.nan

# --- 6. Create the 3D Plot ---
fig = plt.figure(figsize=(16, 12))
ax = fig.add_subplot(111, projection='3d')

# Plot the single stable surface
surf = ax.plot_surface(R, A, U_phi_co_rotating, cmap='cividis', rstride=5, cstride=5)
```

```python
# Set labels and title
ax.set_title(r'Stable Co-rotating Proper Angular Velocity $d\phi/d\tau$', fontsize=18)
ax.set_xlabel(r'Radius (r/M)', fontsize=14, labelpad=15)
ax.set_ylabel(r'Spin Parameter (a/M)', fontsize=14, labelpad=15)
ax.set_zlabel(r'Proper Angular Velocity $d\phi/d\tau$', fontsize=14, labelpad=15)

# Add a color bar to show the mapping of color to z-value
fig.colorbar(surf, shrink=0.5, aspect=10, pad=0.1, label=r'$d\phi/d\tau$')

# Set a good viewing angle
ax.view_init(elev=25, azim=50)

plt.show()
```


Stable Co-rotating Proper Angular Velocity $d\phi/d\tau$
50
40
30
20
10
0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Spin Parameter (a/M)
1
2
3
4
5
6
7
Radius (r/M)
25
20
15
10
5
$d\phi/d\tau$

[10]:
```python
import numpy as np
import matplotlib.pyplot as plt

# --- 1. Define the function to calculate the ISCO radius ---
def calculate_isco_radius(a, M=1.0, prograde=True):
    """
    Calculates the Innermost Stable Circular Orbit (ISCO) radius.

    Args:
        a (np.ndarray): Array of spin parameters.
        M (float): Mass of the black hole.
        prograde (bool): True for prograde (co-rotating), False for retrograde.

    Returns:
        np.ndarray: The ISCO radius for each spin value.
    """
    # Normalize spin by mass
    a_norm = a / M
    # Clip to handle potential floating point issues at the boundaries
    a_norm = np.clip(a_norm, -1.0, 1.0)

    # Define intermediate terms from the analytic formula
    Z1 = 1 + (1 - a_norm**2)**(1/3) * ((1 + a_norm)**(1/3) + (1 - a_norm)**(1/3))
    Z2 = np.sqrt(3 * a_norm**2 + Z1**2)

    # The formula uses , where the top sign (-) is for prograde orbits.
    sign = -1.0 if prograde else 1.0

    r_isco = M * (3 + Z2 + sign * np.sqrt((3 - Z1) * (3 + Z1 + 2 * Z2)))
    return r_isco

# --- 2. Set up the data for the plot ---
M = 1.0
# Create a high-resolution array of spin values from -1 to 1
a_vals = np.linspace(-1, 1, 400)

# Calculate the ISCO radius for both prograde and retrograde cases
r_isco_prograde = calculate_isco_radius(a_vals, M, prograde=True)
r_isco_retrograde = calculate_isco_radius(a_vals, M, prograde=False)

# --- 3. Create the 2D Plot ---
fig, ax = plt.subplots(figsize=(12, 8))
```

```python
# Plot the two curves
ax.plot(a_vals, r_isco_prograde, label='Prograde ISCO', color='blue', linewidth=2.5)
ax.plot(a_vals, r_isco_retrograde, label='Retrograde ISCO', color='red', linewidth=2.5)

# --- 4. Add Annotations for Key Physical Points ---
# Schwarzschild case (a=0)
ax.axhline(y=6, color='gray', linestyle='--', alpha=0.7)
ax.axvline(x=0, color='gray', linestyle='--', alpha=0.7)
ax.plot(0, 6, 'ko') # Black dot at the intersection
ax.text(0.03, 6.1, 'Schwarzschild Limit (a=0, r=6M)', fontsize=12, verticalalignment='bottom')

# Maximal Prograde case (a=1)
ax.plot(1, 1, 'bo')
ax.text(0.98, 1.1, 'Maximal Prograde (a=1, r=M)', fontsize=12, horizontalalignment='right', verticalalignment='bottom')

# Maximal Retrograde case (a=-1)
ax.plot(-1, 9, 'ro')
ax.text(-0.98, 8.9, 'Maximal Retrograde (a=-1, r=9M)', fontsize=12, horizontalalignment='left', verticalalignment='top')

# --- 5. Finalize Plot Aesthetics ---
ax.set_xlabel('Spin Parameter (a/M)', fontsize=14)
ax.set_ylabel('ISCO Radius (r/M)', fontsize=14)
ax.set_title('Innermost Stable Circular Orbit (ISCO) vs. Black Hole Spin', fontsize=16)
ax.legend(fontsize=12)
ax.grid(True, linestyle='--', alpha=0.6)
ax.set_xlim(-1.05, 1.05)
ax.set_ylim(0, 9.5)

plt.show()
```


Innermost Stable Circular Orbit (ISCO) vs. Black Hole Spin
Maximal Retrograde (a=-1, r=9M)
Schwarzschild Limit (a=0, r=6M)
Maximal Prograde (a=1, r=M)
Prograde ISCO
Retrograde ISCO
ISCO Radius (r/M)
Spin Parameter (a/M)
0
2
4
6
8
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00