Probing General Relativity using Cosmic lighthouses

Tom Kimpson [1], Kinwah Wu [1], and Silvia Zane [1]

[1]Mullard Space Science Laboratory, UCL

Intro

General Relativity (GR) is our best theory of space, time and gravity.

We can probe GR in regimes of high spacetime curvature using **pulsars** - cosmic lighthouses - orbiting **massive black holes**

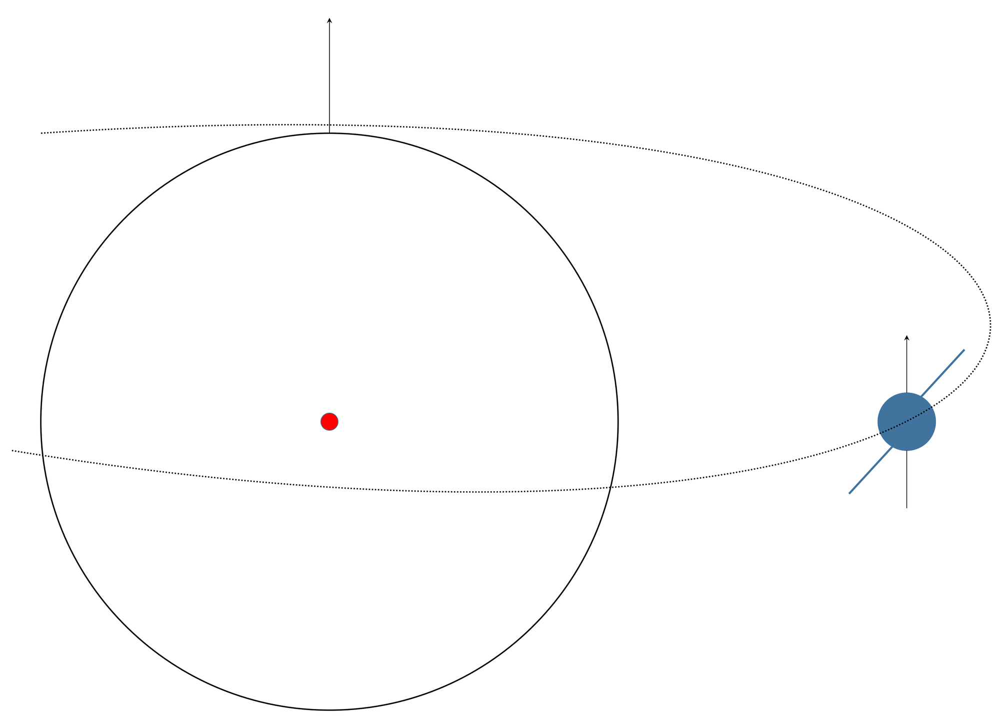

Figure 1. A BH-PSR system

In order to use these systems as probes of strong-field GR in it is necessary to **theoretically model the pulsar timing signal** using computational and numerical methods [1, 2]

Constructing the timing signal

The construction of the timing signal from a pulsar (PSR) orbiting a black hole has 2 primary ingredients:

1. **Covariant Ray Tracing in Curved Spacetime** i.e. *What is the trajectory of light between the observer and the PSR?*
2. **Mathematical Optimization** i.e. *What are the initial conditions of the ray such that the ray connects the the observer and the PSR?*

Astrophysical Background

Black Holes

Black holes are regions of spacetime where the curvature is so extreme that nothing - not even light - can escape.

Originally thought of as purely mathematical artifacts, evidence suggests that they in fact exist as real astrophysical objects. The centre of our own galaxy is thought to host a BH of mass $4 \times 10^6 M_\odot$

The **most extreme gravity in the universe** is found at the surface ('event horizon') of a BH.

Pulsars

Pulsars are exceptionally dense stars composed entirely of neutrons - **a teaspoon of pulsar would weigh 1 billion tonnes**

Pulsars emit beamed radiation. As they spin this light beam sweeps across the universe, analogous to a lighthouse at sea.

The periodic lighthouse 'pulse' is exceptionally stable; pulsars can be thought of as beaconing gyroscopes, providing **the most precise natural clock in the universe**.

Covariant Ray Tracing

Using **geometrical optics**, the trajectory of light can be described as a ray.

The Hamiltonian in covariant form of a general spacetime is,

$$H = \frac{1}{2} g_{\mu\nu} p^\mu p^\nu$$

where $g_{\mu\nu}$ is the *metric* which describes the spacetime and p_μ the *contravariant momenta* of the ray.

The ray path is determined via Hamilton's equations:

$$\dot{x}^\mu = \frac{\partial H}{\partial p_\mu}, \dot{p}_\mu = -\frac{\partial H}{\partial x^\mu}$$

Numerical methods

These equations represent a set of 8 coupled partial differential equations. We can solve these using a **numerical 5th-order Runge Kutta integrator** with adaptive step size [3].

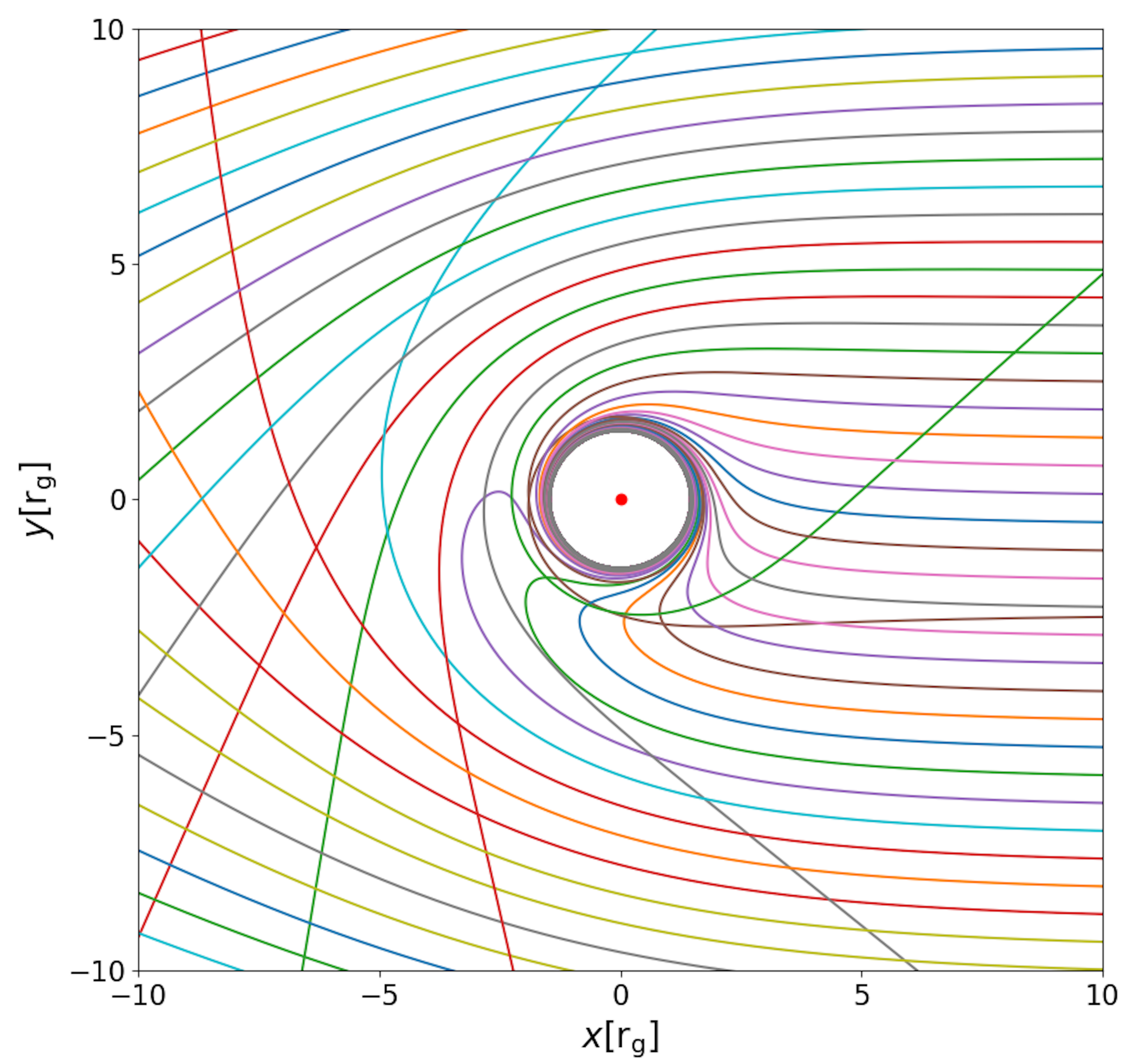

Figure 2. Ray Tracing around a spinning black hole

Finding the intersection

We want to find an intersection of the ray with the pulsar.

We solve this problem **numerically** using a **shooting method**.

This is effectively a **two point boundary value problem**:

Find the initial boundary values (α, β) that minimize $ds = f(\alpha, \beta)$

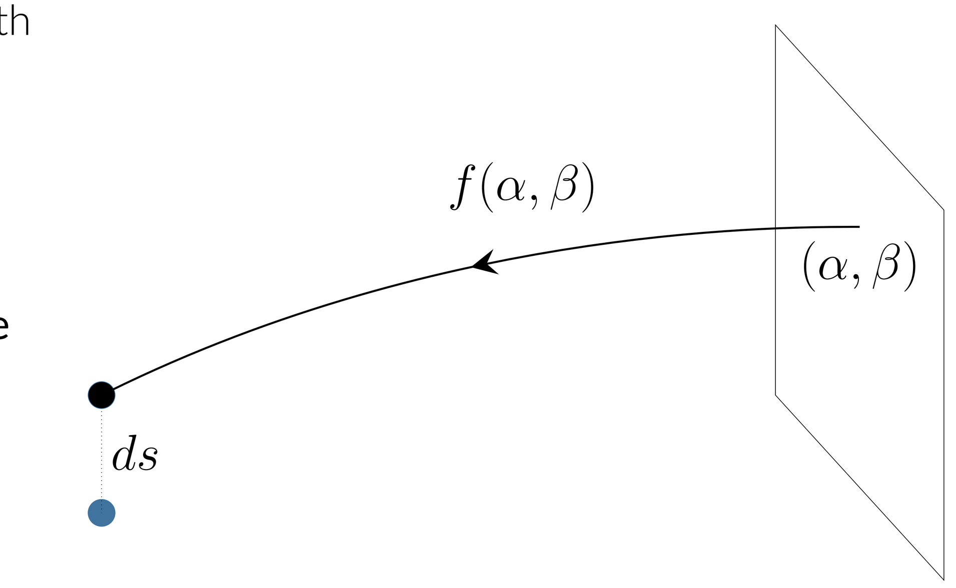

Figure 3. Shooting from the image plane

Blackbox optimization

The ray tracing is effectively a black box function $f(\alpha, \beta)$ which takes some (α, β) and returns the minimum distance ds along the ray from the target value .

We want to minimize $ds = f(\alpha, \beta)$ to within some tolerance ϵ by varying initial conditions (α, β).

The adjustment of (α, β) proceeds via a **nonlinear conjugate gradient descent algorithm** [4].

The conjugate direction vector $\mathbf{h}_j$, is updated at each iteration step j as,

$$\mathbf{h}_{j+1} = \mathbf{g}_{j+1} + \gamma_j \mathbf{h}_j \ , \tag{1}$$

where $\mathbf{g}_j = -\nabla f(\alpha_j, \beta_j)$ and

$$\gamma_j = \frac{\mathbf{g}_{j+1} \cdot \mathbf{g}_{j+1}}{\mathbf{g}_j \cdot \mathbf{g}_j} \ , \tag{2}$$

The variables α, β are then updated at each iteration as,

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \delta_j \mathbf{h}_j \ , \tag{3}$$

for vector $\mathbf{x}_j = (\alpha_j, \beta_j)$ and where δ_j is the variable stepsize, determined via an inexact line search

The analytical form of $f(\alpha, \beta)$ is unknown - we are optimizing a **blackbox function**. The gradients necessary for the optimization are evaluated numerically via the **difference quotient**.

Why not use `vanilla' gradient descent?

- Due to extreme spacetime curvature, the ray path is very sensitive to changes in (α, β).
- Consequently $f(\alpha, \beta)$ often takes the form of an **ill-conditioned narrow valley.**
- Gradient descent becomes **inordinately slow** for such a function surface; the direction of steepest descent is not, in general, in the direction of the minimum and the algorithm instead follows a '**criss-cross' pattern**, oscillating between the sides of the valley.
- This issue is avoided by moving in a direction which is **conjugate** to previous directions.

References

[1] Kimpson, T., Wu, K., Zane, S. , *MNRAS*, 486:360-377, 2019.

[2] Kimpson, T., Wu, K. Zane, S. , *MNRAS*, 484:2411-2419, 2019

[3] Press, W. et al. , Reeves, C., *Numerical Recipes in Fortran*, Cambridge University Press, 1996

[4] Fletcher, R., Reeves, C., *The Computer Journal*, 7:149–154, 1964