Newtonian Black Holes

Ryan Maguire

January 24, 2023

Outline

- Classical Mechanics
- Black Holes
- Euler's Method
- Runge-Kutta Method
- Raytracing
- Cool Pictures

Classical Mechanics

It's been known for some time that the speed of light is not infinite. Newton observed that a finite light speed would explain discrepancies in orbits of Jupiter's moons with the classical theory of gravity. This observation can accurately compute the speed of light as well.

Classical Mechanics

Nothing in classical mechanics says there is a universal speed limit, and there is no indication that the speed of light itself must be constant.

The original idea of *aether* tries to model light after sound, and sound can have variable speed depending on the medium it is travelling through. We'll abuse this a lot to get a very rough estimate of a black hole as far as Newtonian mechanics is concerned. It is not a physically realistic assumption.

Classical Mechanics

If you take a ball and throw it in the air, it comes back. If you had an absolute cannon of an arm, perhaps you could throw the ball at 11,186 meters per second. It would certainly go a lot high, but would it ever come back?

Classical Mechanics

Let's use the universal law of gravitation, which is a classical law (no general relativity here), but is accurate in many scenarios. The potential energy is given by an inverse law:

$$U = -\frac{GMm}{r} \tag{1}$$

Where G is the universal gravitational constant, M is the mass of the earth, m is the mass of the ball, and r is the distance from the ball to the center of the earth. Kinetic energy is given by:

$$K = \frac{1}{2}mv^2 \tag{2}$$

where v is the speed (norm of velocity) of the speed.

Classical Mechanics

Let's try to throw the ball at a speed such that the ball will slow down to 0 m/s *at infinity*. The potential energy at infinity is zero, since $\frac{1}{r} \to 0$ as $r \to \infty$. Invoking conservation of energy:

$$U_0 + K_0 = U_\infty + K_\infty \tag{3}$$

$$U_0 + K_0 = 0 + 0 \tag{4}$$

$$-\frac{GMm}{R} + \frac{1}{2}mv_0^2 = 0 \tag{5}$$

$$v_0 = \sqrt{\frac{2GM}{R}} \tag{6}$$

Where R is the radius of the Earth. This value v_0 is *escape velocity*, the speed at which is needed for an object to *escape* the influence of a bodies gravitational pull.

Black Holes

In the previous slides we were considering the Earth. The radius was constant, as was the mass, and we solved for the escape velocity. Instead, let's suppose the mass M is fixed, and the *escape velocity* v_0 is constant as well. The variable to solve for is the radius R.

$$R = \frac{2GM}{v_0^2} \tag{7}$$

What would happen if we chose $v_0 = c$, the speed of light (ignore the title of this current slide, please)?

Black Holes

The result is a black hole. The gravitational pull is so strong that even light can't escape, so the result is blackness. The formula obtained is the *Schwarzschild radius*:

$$R = \frac{2GM}{c^2} \tag{8}$$

Fun-fact, this is the same formula you get after doing all the rigorous general relativity calculations (Assuming a few conditions, such as the black hole is not rotating).

We're going to try and draw a black hole using the world of classical mechanics.

Euler's Method

Given a vector-valued ordinary differential equation:

$$\frac{\mathrm{d}\mathbf{r}(\mathbf{t})}{\mathrm{d}t} = f\big(\mathbf{r}(\mathbf{t})\big) \tag{9}$$

one of the simplest means of solving this numerically is *Euler's method*. Replacing the differentials with small displacements, we get:

$$\frac{\Delta\mathbf{r}(t)}{\Delta t} \approx f\big(\mathbf{r}(t)\big) \tag{10}$$

Euler's Method

Given initial conditions $\mathbf{r}(t_0) = \mathbf{r}_0$ we can further expand this as follows:

$$\begin{aligned} \Delta\mathbf{r}(t) &\approx f\big(\mathbf{r}(t)\big)\Delta t && (11) \\ \mathbf{r}(t_0 + \Delta t) - \mathbf{r}(t_0) &\approx f\big(\mathbf{r}(t_0)\big)\Delta t && (12) \\ \mathbf{r}(t_1) &\approx \mathbf{r}(t_0) + f\big(\mathbf{r}(t_0)\big)\Delta t && (13) \end{aligned}$$

as Δt gets smaller, the accuracy of the approximation improves. We then obtain a sequence $\mathbf{r}_n$ of approximate values of $\mathbf{r}(t_n)$:

$$\mathbf{r}(t_{n+1}) \approx \mathbf{r}(t_n) + f\big(\mathbf{r}(t_n)\big)\Delta t \qquad (14)$$

Euler's Method

The previous slide is great for first order differential equations, but the equations of motions in classical mechanics deal with second order vector-valued differential equations. This stems from Newton's second law:

$$\mathbf{F} = m\mathbf{a} = m\frac{\mathrm{d}^2\mathbf{r}(t)}{\mathrm{d}t^2} \tag{15}$$

where $\mathbf{a}(t)$ is the acceleration, and $\mathbf{F}$ is the force.

Euler's Method

Let's extend Euler's method to second order differential equations.

$$\frac{\mathrm{d}^2\mathbf{r}(t)}{\mathrm{d}t^2} = f\big(\mathbf{r}(t)\big) \tag{16}$$

with initial conditions $\mathbf{r}(t_0)$ and $\dot{\mathbf{r}}(t_0)$ known to us. We start by defining the velocity:

$$\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{r}(t)}{\mathrm{d}t} \tag{17}$$

The differential equations can be approximated by:

$$\frac{\Delta\mathbf{v}(t)}{\Delta t} \approx f\big(\mathbf{r}(t)\big) \tag{18}$$

so long as the time step Δt is small.

Euler's Method

We mimic the method from before and obtain:

$$\mathbf{v}(t_1) = \mathbf{v}(t_0) + f\big(\mathbf{r}(t_0)\big)\Delta t \tag{19}$$

where $\mathbf{v}(t_0) = \dot{\mathbf{r}}(t_0)$. We can now update the position vector:

$$\mathbf{r}(t_1) = \mathbf{r}(t_0) + \mathbf{v}(t_0)\Delta t \tag{20}$$

And in general, we obtain two sequences:

$$\mathbf{v}(t_{n+1}) = \mathbf{v}(t_n) + f\big(\mathbf{r}(t_n)\big)\Delta t \tag{21}$$

$$\mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + \mathbf{v}(t_n)\Delta t \tag{22}$$

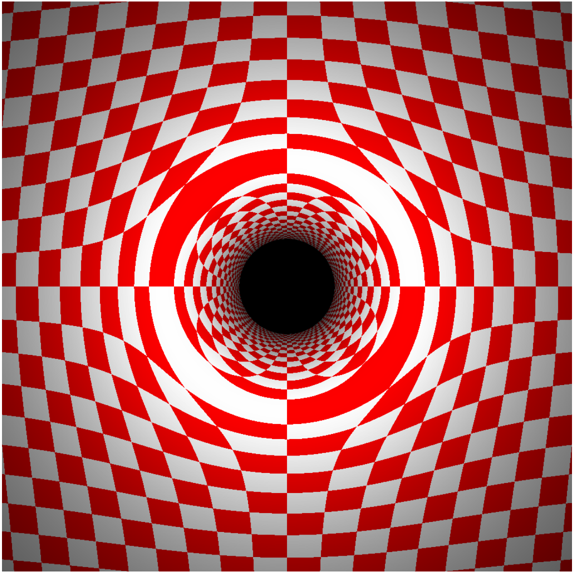
This idea immediately extends to higher order differential equations. The local error in these methods is roughly proportional to the square of the step size Δt. The global error grows linearly.

Runge-Kutta Method

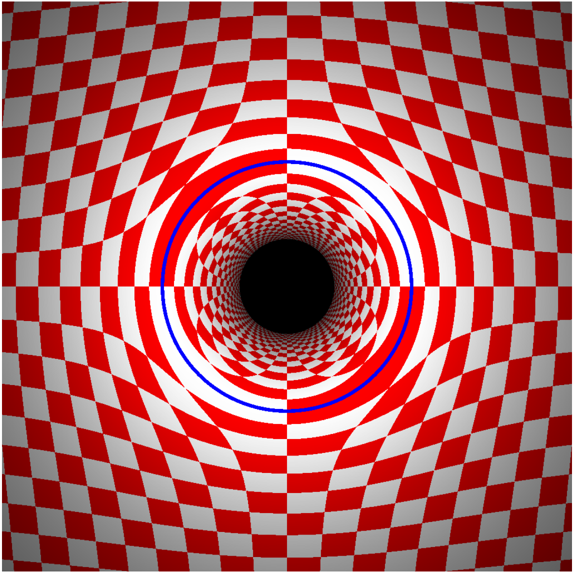
The raytracing we'll do for black holes uses Euler's method with a lot steps (hundreds of thousands) with a small Δt. We would be able to use a larger step size if the error bounds were proportional to higher powers of Δt, and this is where the Runge-Kutta method comes in to play.

For the sake of time, I won't give the derivation (which is much longer than Euler's method), but remark that the error is proportional to Δt^4. The Runge-Kutta programs required far fewer iterations, while achieving similar accuracy.

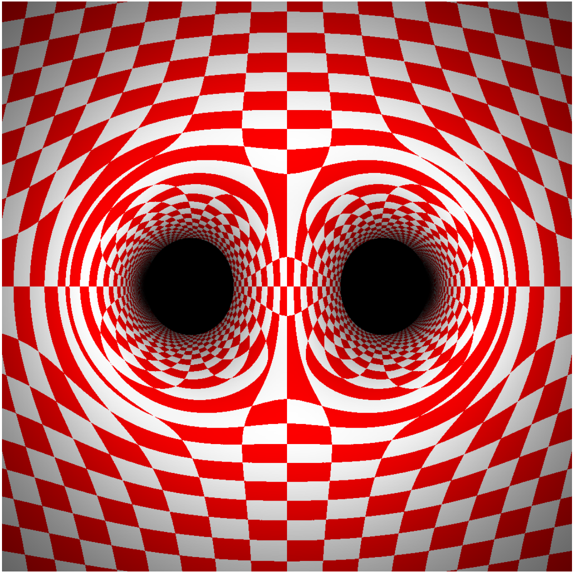
Results



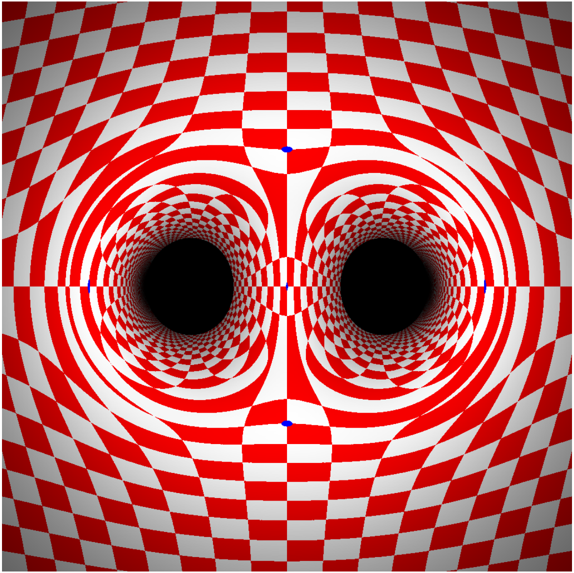
Results



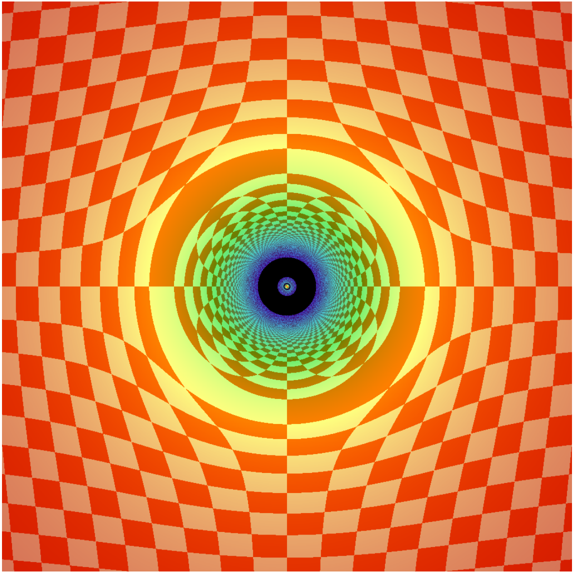
Results



Results



Results



The End

Thanks!