Project 1: Numerical Integrals and ODEs

Carter Matties

September 30, 2025

1 Introduction

In this project, we investigate the use of numerical methods to solve both ordinary differential equations (ODEs) and single-variable integrals. In particular, we examine the ODE governing the (1D) harmonic oscillator of mass m and spring constant k, linearly damped with coefficient γ, and driven by a force

$$F(t) = F_0 \cos(\omega t) \tag{1}$$

for a driving frequency ω and magnitude F_0. To solve this ODE numerically, we write and employ implementations of the explicit Euler's method, as well as a 4th-order Runge-Kutta method to solve a system of first-order ordinary differential equations describing our system.

In addition to numerically solving the aforementioned ODE, we use numerical methods to examine the gravitational field produced at a given point on the z-axis by the portion of a uniform spherical shell spanning $0 \leq \theta \leq \theta_{\max}$, illustrated in Fig. 1.

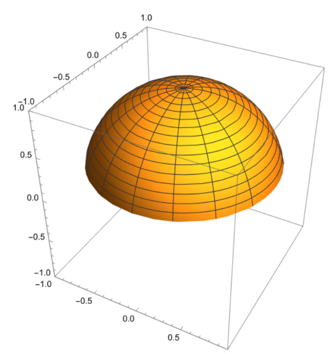


Figure 1: The thin spherical shell of radius 1 with $0 \leq \theta \leq \theta_{\max} = \frac{3\pi}{7}$. In the project, we use numerical integration methods to investigate the gravitational field along the z-axis due to similar shells with various values of $\theta_{\max}$.

In particular, to solve this integral, we employ our own implementations of a left Riemann sum, the trapezoidal rule, and Simpson's rule.

2 Background

2.1 Damped, Driven Spring

Using Newton's second law, we can write out the ODE for our damped, driven harmonic oscillator

$$\ddot{x} + \gamma \dot{x} + \frac{k}{m} x = \frac{F_0}{m} \cos(\omega t). \tag{2}$$

It is then straightforward to transform the 2nd-order ODE (2) into the system of 1st-order equations

$$\begin{cases} \dot{x} & = v, \\ \dot{v} & = \frac{F_0}{m}\cos(\omega t) - \frac{k}{m}x - \gamma v. \end{cases} \tag{3}$$

Since the system of first-order equations (3) is more useful for solving the problem numerically, we shall solve directly solve for $x(t)$ using the single 2nd-order equation (2), and then simply take a time derivative to compute $v(t)$.

We start by solving the homogeneous equation

$$\ddot{x}_0 + \gamma \dot{x}_0 + \frac{k}{m}x_0 = 0. \tag{4}$$

Using the ansatz $x_0(t) = Ae^{i\omega' t}$, we find the real-valued solutions

$$x_0(t) = e^{-\frac{\gamma}{2}t}\left[C_1 \cos\left(\sqrt{\frac{k}{m} - \frac{\gamma^2}{4}}t\right) + C_2 \sin\left(\sqrt{\frac{k}{m} - \frac{\gamma^2}{2m^2}}t\right)\right]. \tag{5}$$

Next, we find the particular solution to Eqn. (2) by solving the ODE

$$\ddot{X} + \gamma \dot{X} + \frac{k}{m}X = \frac{F_0}{m}e^{i\omega t}, \tag{6}$$

and then simply taking the real part. To solve Eqn. (6), we take the ansatz $X = Ae^{i\omega t}$, and solve for the amplitude A. Doing so, we get the solution

$$X(t) = \frac{F_0/m}{\sqrt{\left(\frac{k}{m} - \omega^2\right)^2 + \gamma^2\omega^2}} e^{i\arctan\left(\frac{-\gamma\omega}{k/m-\omega^2}\right)} e^{i\omega t}. \tag{7}$$

Finally, taking the real part of Eqn. (7) and combining with the homogeneous solution Eqn. (5), we get the full solution

$$\begin{aligned} x(t) =& e^{\frac{-\gamma}{2}t}\left[C_1 \cos\left(\sqrt{\frac{k}{m} - \frac{\gamma^2}{4}}t\right) + C_2 \sin\left(\sqrt{\frac{k}{m} - \frac{\gamma^2}{4}}t\right)\right] \\ &+ \frac{F_0}{m\sqrt{\left(\frac{k}{m} - \omega^2\right)^2 + \gamma^2\omega^2}} \cos\left(\omega t - \arctan\left(\frac{\gamma\omega}{k/m - \omega^2}\right)\right). \end{aligned} \tag{8}$$

2.2 Shell Gravitation

With our integral, we want to determine the magnitude of the gravitational field at a point $\vec{r} = z\hat{z}$, due to the portion of a uniform spherical shell of radius R and mass M spanning $\theta \leq \theta_{\max}$, for various values of $\theta_{\max}$. We can see that the integral of interest will be

$$\vec{g} = -\int \frac{G\sigma dA'}{|\vec{r} - \vec{r}'|^3}\left(\vec{r} - \vec{r}'\right), \tag{9}$$

where G is the gravitational constant and σ is the surface mass density of the shell. By rotational symmetry, we can see that the x and y components of the field will cancel out, so we can write the integral

$$g = -\frac{GM}{(1 - \cos\theta_{\max})}\int_0^{\theta_{\max}} \frac{z - R\cos\theta}{\left(R^2 + z^2 - 2Rz\cos\theta\right)^{3/2}} \sin\theta d\theta, \tag{10}$$

where we have glossed over the step of writing $\sigma = M/A$, for the surface area A of the dome.

Evaluating the definite integral in Eqn. (10), we get the exact solution

$$g = -\frac{GM}{1 - \cos\theta_{\max}}\left[\frac{R - z\cos\theta_{\max}}{z^2\sqrt{R^2 + z^2 - 2Rz\cos\theta_{\max}}} - \frac{R - z}{z^2\sqrt{R^2 + z^2 - 2Rz}}\right]. \tag{11}$$

Although it may not be obvious by inspection, Newton's shell theorem tells us that as $\theta_{\max}$ approaches π, we should have

$$g \to -\frac{GM}{z^2},$$

since in this limit the portion of the shell approaches a full spherical shell.

Likewise, we can see that as $\theta_{\max}$ approaches 0, we should have

$$g \to -\frac{GM}{(z-R)^2},$$

since in this limit our "shell" approaches a single point mass M at $(0, 0, R)$.

3 Integrators

3.1 ODE Evolvers

To numerically integrate the system of 1st-order ODEs Eqn. (3), we use the explicit form of Euler's method and a 4th-order Runge-Kutta method.

3.1.1 Euler's Method

To update our position and velocity at each time step with the explicit form of Euler's method, we use the typical rule

$$x(t + \Delta t) = x(t) + \dot{x}(x(t), v(t), t)\Delta t, \tag{12}$$

$$v(t + \Delta t) = v(t) + \dot{v}(x(t), v(t), t)\Delta t. \tag{13}$$

Examining the update rule for x, we can see that, at each timestep, we accumulate a local truncation error on the order of Δt^2. This is because we have the Taylor expansion

$$x(t + \Delta t) = x(t) + \dot{x}(x(t), v(t), t)\Delta t + \frac{\Delta t^2}{2}\ddot{x}(x(t), v(t), t) + \mathcal{O}(\Delta t^3). \tag{14}$$

Hence, to lowest order, we can see that our Euler's method update rule (13) differs from the Taylor expansion (14) by $\mathcal{O}(\Delta t^2)$.

Then, if we want to evolve the system from a time t_0 to a time t_1, then we can see we will have to use $N = \frac{t_1 - t_0}{\Delta t}$ time steps. If the local error accumulated at each of these N time steps is on the order $\mathcal{O}(\Delta t^2)$, then we can see that the global truncation error will be on the order of $N\Delta t^2$. Therefore, the global error in Euler's method is on the order

$$\mathcal{O}(\Delta t)$$

for our function $x(t)$.

3.1.2 Runge-Kutta

To update our system of equations using the 4th-order Runge-Kutta (RK4) method, we first need to compute some coefficients

$$\begin{aligned}
k_{1,x} &= \Delta t\dot{x}(x(t), v(t), t) & k_{1,v} &= \Delta t\dot{v}(x(t), v(t), t) \\
k_{2,x} &= \Delta t\dot{x}(x(t) + k_{1,x}/2, v(t) + k_{1,v}/2, t + \Delta t/2) & k_{2,v} &= \Delta t\dot{v}(x(t) + k_{1,x}/2, v(t) + k_{1,v}/2, t + \Delta t/2) \\
k_{3,x} &= \Delta t\dot{x}(x(t) + k_{2,x}/2, v(t) + k_{2,v}/2, t + \Delta t/2) & k_{3,v} &= \Delta t\dot{v}(x(t) + k_{2,x}/2, v(t) + k_{2,v}/2, t + \Delta t/2) \\
k_{4,x} &= \Delta t\dot{x}(x + k_{3,x}, v + k_{3,v}, t + \Delta t) & k_{4,v} &= \Delta t\dot{x}(x + k_{3,x}, v + k_{3,v}, t + \Delta t).
\end{aligned}$$

Then we can compute the new values

$$x(t + \Delta t) = x(t) + \frac{1}{6}\left(k_{1,x} + 2k_{2,x} + 2k_{3,x} + k_{4,x}\right) \tag{15}$$

$$v(t + \Delta t) = v(t) + \frac{1}{6}\left(k_{1,v} + 2k_{2,v} + 2k_{3,v} + k_{4,v}\right). \tag{16}$$

To examine the local and global errors with RK4, let us once again simplify and examine just the x update rule. We can first observe using the Taylor expansion (14) that

$$k_{1,x} = x(t + \Delta t) - x(t) - \mathcal{O}(\Delta t^2). \tag{17}$$

Following this line, it is possible to see that the 4th-order Runge-Kutta method gives us a local truncation error on the order $\mathcal{O}(\Delta x^3)$, and hence a global truncation error of $\mathcal{O}(\Delta x^4)$.

3.2 Definite Integrals

To evaluate the definite integral in Eqn. (10), we shall use a left Riemann sum, the trapezoidal rule, and Simpson's rule.

3.2.1 Riemann Sum

Let us consider the definite integral of some function $f(x)$ from a to b. To approximate this integral with a left Riemann sum, we choose a number of "slices" N and define $\Delta x = \frac{b-a}{N}$. Then we can write

$$\int_a^b f(x)dx \approx \sum_{n=0}^{N-1} \Delta x f(a + n\Delta x). \tag{18}$$

To determine the truncation error with this method, let us consider the integral of f from a to $a + \Delta x$. Taking the Taylor expansion of f at a, we have that

$$f(a + x) = f(a) + xf'(x) + \frac{x^2}{2} f''(a) + \mathcal{O}(x^3). \tag{19}$$

Thus, approximating, we can see that

$$\begin{aligned}
\int_a^{a+\Delta x} f(x)dx &= \int_a^{a+\Delta x} \left[f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2} f''(a) + \frac{(x-a)^3}{6} f'''(a) + \mathcal{O}((x-a)^4) \right] \\
&= \left[xf(a) + \frac{(x-a)^2}{2} f'(a) + \frac{(x-a)^3}{6} f''(a) + \frac{(x-a)^4}{24} f'''(a) + \mathcal{O}((x-a)^5) \right]_a^{a+\Delta x} \\
&= \Delta x f(a) + \frac{\Delta x^2}{2} f'(a) + \frac{\Delta x^3}{6} f''(a) + \frac{\Delta x^4}{24} f'''(a) + \mathcal{O}(\Delta x^5).
\end{aligned} \tag{20}$$

However, note that the first term here is precisely the Riemann sum approximating the integral $\int_a^{a+\Delta x} f(x)dx$ with $N = 1$. Thus, we can see that for each "slice" Δx in our Riemann sum, we accumulate a local truncation error on the order $\mathcal{O}(\Delta x^2)$, or equivalently, $\mathcal{O}(N^{-2})$.

Then, since we can write our integral

$$\int_a^b f(x)dx = \sum_{n=0}^{N-1} \int_{a+n\Delta x}^{a+(n+1)\Delta x} f(x)dx, \tag{21}$$

with N terms in the sum on the RHS, and our Riemann sum approximates each of these terms to order $\mathcal{O}(N^{-2})$, it follows that our Riemann sum has a global truncation error on the order of $\mathcal{O}(N^{-1})$, or equivalently, $\mathcal{O}(\Delta x)$.

3.2.2 Trapezoidal Rule

To approximate and integral with the trapezoidal rule, we simply average the left and right Riemann sums for a number of "slices" N, i.e.

$$\int_a^b f(x)dx \approx \sum_{n=0}^{N-1} \frac{f(a + n\Delta x) + f(a + (n+1)\Delta x)}{2}. \tag{22}$$

Let us once again consider the integral from a to $a + \Delta x$, with $N = 1$. Then we have the trapezoidal rule approximation

$$\int_a^{a+\Delta x} f(x)dx \approx \frac{\Delta x}{2} \left(f(a) + f(a + \Delta x) \right). \tag{23}$$

However, we can use the Taylor expansion of $f(x)$ at $x = a$ to rewrite

$$f(a + \Delta x) = f(a) + \Delta x f'(a) + \mathcal{O}(\Delta x^2). \tag{24}$$

Then, plugging this back into the single-slice trapezoidal approximation, we have

$$\begin{aligned} \int_a^{a+\Delta x} f(x)dx &\approx \frac{\Delta x}{2} \big(f(a) + f(a) + \Delta x f'(a) + \mathcal{O}(\Delta x^2) \big) \qquad (25) \\ &= \Delta x f(a) + \frac{\Delta x^2}{2} f'(a) + \mathcal{O}(\Delta x^3). \qquad (26) \end{aligned}$$

Now, comparing this to Eqn. (20), we can see that we now have precisely the first two terms in the Taylor expansion of the integral over our slice. Therefore, it follows that the local truncation error with the trapezoidal rule is on the order $\mathcal{O}(\Delta x^3)$, or $\mathcal{O}(N^{-3})$.

We can use the same logic as before, that since each slice has an error of order $\mathcal{O}(N^{-3})$ and we divide the integral from a to b up into N slices, we have a global truncation error of the order $\mathcal{O}(N^{-2})$, or $\mathcal{O}(\Delta x^2)$.

3.2.3 Simpson's Rule

To evaluate integrals using Simpson's rule, we approximate the desired integral by dividing up the interval $[a, b]$ into N slices as usual, approximating f across pairs of slices as a parabola, and then integrating.

In order to approximate f at the points $x = a - \Delta x, a, a + \Delta x$, we shall fit f to the parabola

$$y(x) = A(x - a)^2 + B(x - a) + C.$$

This gives us the system of equations

$$\begin{cases} f(a - \Delta x) = A(-\Delta x)^2 + B(-\Delta x) + C \\ f(a) = C \\ f(a + \Delta x) = A(\Delta x)^2 + B(\Delta x) + C \end{cases}$$

for A, B, C, resulting in the fact that f can be approximated by the parabola

$$y(x) = \frac{1}{\Delta x^2} \left[\frac{f(a - \Delta x) + f(a + \Delta x)}{2} - f(a) \right] (x - a)^2 + \frac{1}{2\Delta x} \left[f(a + \Delta x) - f(a - \Delta x) \right] (x - a) + f(a). \tag{27}$$

This then gives us the approximation

$$\begin{aligned} \int_{a-\Delta x}^{a+\Delta x} f(x)dx \approx & \left\{ \frac{1}{3\Delta x^2} \left[\frac{f(a - \Delta x) + f(a + \Delta x)}{2} - f(a) \right] (x - a)^3 \right. \\ & \left. + \frac{1}{4\Delta x} \left[f(a + \Delta x) - f(a - \Delta x) \right] (x - a)^2 + f(a)(x - a) \right\}_{a-\Delta x}^{a+\Delta x} \\ = & \frac{2\Delta x}{3} \left[\frac{f(a - \Delta x) + f(a + \Delta x)}{2} - f(a) \right] + 2\Delta x f(a). \qquad (28) \end{aligned}$$

Now let us observe that

$$\frac{f(a + \Delta x) + f(a - \Delta x)}{2} = \frac{2f(a) + 2\frac{\Delta x^2}{2} f''(a) + \mathcal{O}(\Delta x^4)}{2}. \tag{29}$$

Thus, it follows that the approximation of Eqn. (28) is equal to

$$\int_{a-\Delta x}^{a+\Delta x} f(x)dx \approx 2\left[\Delta x f(a) + \frac{\Delta x^3}{6} f''(a) + \mathcal{O}(\Delta x^5)\right]. \tag{30}$$

Compare this with the approximation of the integral that we get by Taylor expanding $f(x)$ around $x = a$

$$\int_{a-\Delta x}^{a+\Delta x} f(x)dx = \Delta x[2f(a)] + \frac{\Delta x^2}{2}(f'(a) - f'(a)) + \frac{\Delta x^3}{6}[2f''(a)] + \frac{\Delta x^4}{24}[f'''(a) - f'''(a)] + \mathcal{O}(\Delta x^5), \tag{31}$$

and we can see that our approximation with Simpson's method has a local truncation error of order $\mathcal{O}(\Delta x^5)$, or $\mathcal{O}(N^{-5})$.

Finally, in order to determine the global truncation error with Simpson's method, let us observe that

$$\int_a^b f(x)dx = \sum_{n=0}^{N-2} \int_{a+n\Delta x}^{a+(n+2)\Delta x} f(x)dx. \tag{32}$$

Therefore, to approximate the integral from a to b, we need on the order of N slices, each of which contribute local error $\mathcal{O}(N^{-5})$, so the global truncation error is $\mathcal{O}(N^{-4})$, or $\mathcal{O}(\Delta x^4)$.

4 Results

4.1 ODE

First, let us examine our ODE solvers with the initial conditions $(x_0, v_0) = (0, 1)$ and the parameters $(k, m, F_0, \omega, \gamma) = (1, 2, 0.1, 4, 0.5)$, and $dt = 0.001$. Then, as we can see in Fig. (2), our Euler and Runge-Kutta solvers appear to be in good agreement with both the exact solution and the SciPy implementation of 4th-order Runge-Kutta. Although our numerical and exact solutions appear to be in good agreement for

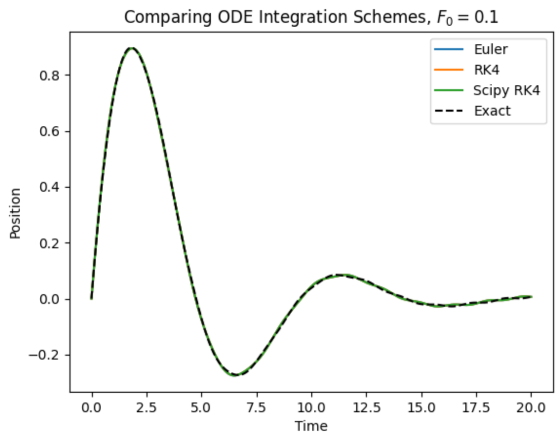


Figure 2: Comparing our implementations of Euler and 4th-order Runge-Kutta ODE evolvers with the exact solution and the SciPy implementation of RK4. We can see that they all appear to be in good agreement.

a small-amplitude driving force $F_0 = 0.1$, we run into some issues when we increase the driving force to $F_0 = 1$, keeping all other parameters the same (Fig. 3). When we increase the driving force's amplitude, we

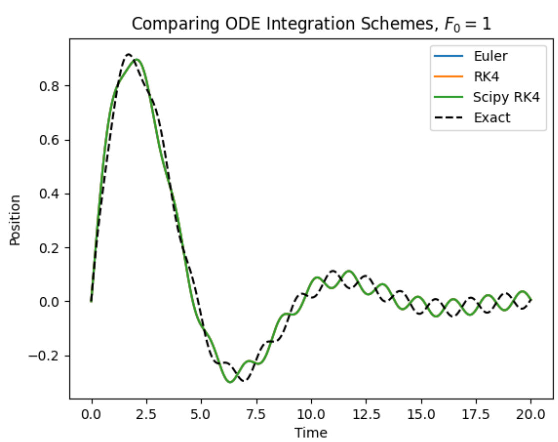


Figure 3: Numerical and exact solutions to the same setup as in Fig. (2), but with $F_0 = 1$. We can see that all of the numerical solutions appear to be in good agreement with one another, but our exact solution eventually gets out of phase with the rest.

can see that our exact solution eventually gets perfectly out of phase with the numerical solutions. Because there is disagreement in the long-term behavior, or the steady state solution, while the transient behavior of our particle is in better agreement between numerical and exact solutions, the issue is likely related to the particular solution to our ODE, which accounts for the driving force $F_0/m \cos(\omega t)$. However, although the exact solution gets out of phase with our numerical solutions, it is remarkable that the frequency of the steady-state solutions appears to agree between our numerical and analytical methods.

Finally, we examine the global truncation error in the Euler and RK4 integration schemes by using the same initial conditions and physical parameters as in Fig. (2), while varying the time step dt. Doing so, we get the plot in Fig. (4). Note that, although there is some oscillation in the truncation error for both

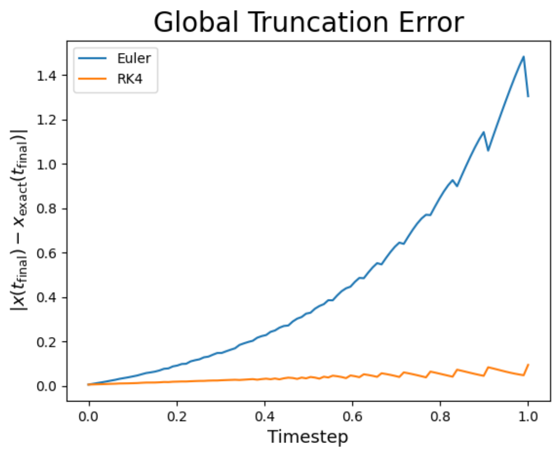


Figure 4: Truncation error scaling for explicit Euler and 4th-order Runge-Kutta numerical integration. Observe that the error from Euler's method increases approximately as dt^2, while the error from RK4 increases much more slowly.

of these integration techniques, the error from Euler's method scales approximately as dt^2, while the error increases much more slowly for RK4, in agreement with the dt^4 scaling in global error we saw previously.

4.2 Integral

Let us first examine our integral of interest on its own, i.e.

$$\int_0^{\theta_{\max}} \frac{(z - R\cos\theta)\sin\theta}{(R^2 + z^2 - 2Rz\cos\theta)^{3/2}} d\theta. \tag{33}$$

Setting $N = 200$, we plot the fractional difference between the exact solution or the scipy implementation, when applicable, and the result from our implementation of the Riemann, trapezoidal, and Simpson's rules in Fig. (5).

Examining Fig. (5), we observe that the disagreement between our implementations of the trapezoidal rule and Simpson's rule is very small, on the order of 10^{-11} and 10^{-13}, respectively. Although this difference is probably larger than we would expect from pure floating point error, it is undoubtedly small enough to confidently say that our implementation of these integrators agree with that of SciPy.

Additionally, note that Simpson's rule is in better agreement with the exact solution than the trapezoidal rule is, which is in turn better than the Riemann sum. This aligns with our derivations of the truncation errors, that Riemann gives an error $\mathcal{O}(N^{-1})$, trapezoidal gives an error $\mathcal{O}(N^{-2})$, and Simpson's rule gives an error $\mathcal{O}(N^{-4})$.

We further investigate the truncation error by fixing $\theta_{\max} = \pi/2$ and comparing the solutions from each integrator for different values of N (Fig. 6).

Examining our plot in Fig. (6), we can see that each curve is linear on the log-log plot. Furthermore, we can see that each slope is approximately equal to the exponent for the corresesonding method's error scaling, i.e. the slope of -2 for the trapezoidal rule agrees with the error scaling $\mathcal{O}(N^{-2})$.

We can also take note of the two spikes on the Simpson's rule curve in Fig. (6). These reveal a problem in our implementation of Simpson's rule, namely that our implementation is significantly less accurate for N odd than for N even. This conclusion comes from the fact that these spikes occur at $N = 75$ and $N = 125$.

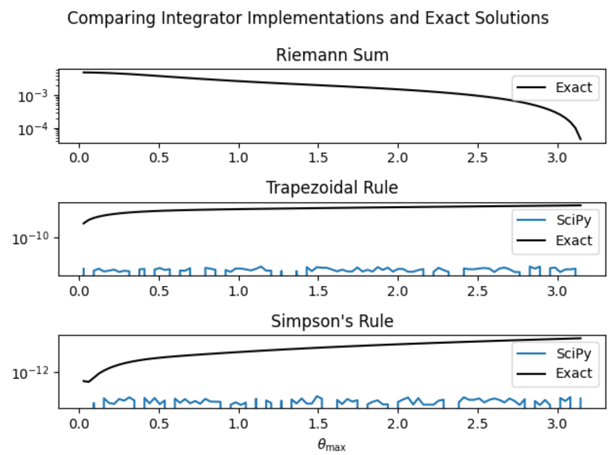


Figure 5: Evaluating our integrator implementations against the exact solutions and implementations from the SciPy package for the full range of $\theta_{\rm max}$ with $N = 200$. For each of the Riemann sum, trapezoidal rule, and Simpson's rule, we plot the fractional difference between our solution and the exact solution (black) or the SciPy implementation (blue), when applicable. We conclude that our trapezoidal and Simpson's rule implementations are in excellent agreement with those from SciPy.

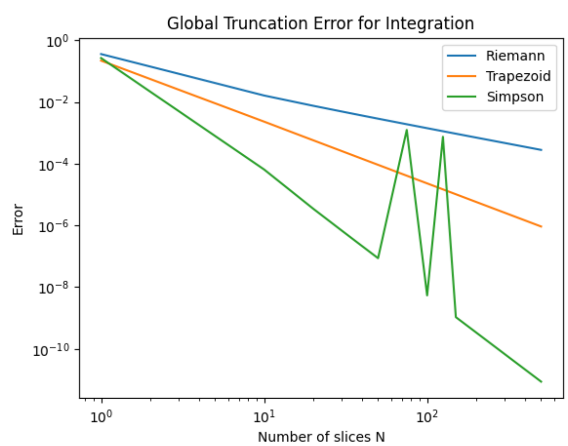


Figure 6: Comparing global truncation errors between different integration methods for various numbers of slices N. Note that each curve (with the exception of two explainable spikes in Simpson's rule) is linear on the log-log plot. Each of these lines is in agreement with our derived truncation error scalings.

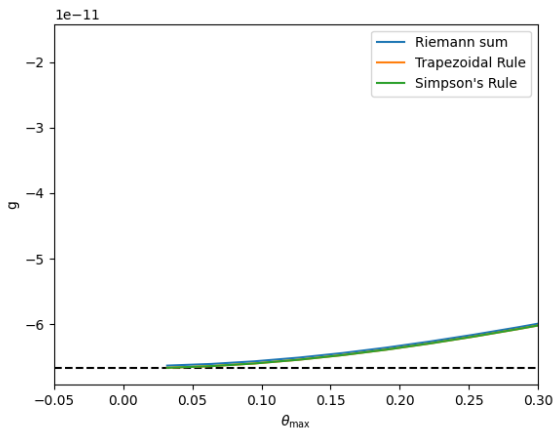
This issue could likely be resolved by being more careful about how our Simpson's rule method sums over the integrals of parabolas that have been fit to the integrand $f(x)$.

Finally, we validate our integral result by returning to the gravitational field Eqn. (10) and examining the limits $\theta_{\rm max} \to 0$ and $\theta_{\rm max} \to \pi$. As previously mentioned, we should have

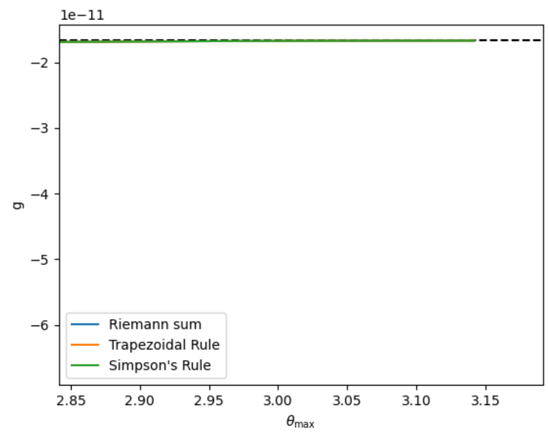
$$\lim_{\theta_{\rm max} \to 0} g = -\frac{GM}{(z - R^2} \tag{34}$$

$$\lim_{\theta_{\rm max} \to \pi} g = -\frac{GM}{z^2}. \tag{35}$$

Indeed, we can plainly see from Fig. (7) that these are both the case.



(a)



(b)

Figure 7: Examining the gravitational field in the limits $\theta_{\max} \to 0$ (a) and $\theta_{\max} \to \pi$ (b). Observe that in both cases, all of our solvers approach the expected value (dashed line).