PHY607 Project 1:
Numerical Methods for Physics

Lauren Sdun

(Dated: September 30, 2025)

This project investigates the accuracy and reliability of numerical integration methods applied to two classical physics problems: the damped harmonic oscillator and the Biot–Savart law for a finite current-carrying wire. For the oscillator, three ordinary differential equation (ODE) solvers were implemented: the explicit Euler method, the fourth-order Runge–Kutta method (RK4), and SciPy's adaptive Runge–Kutta method (RK45). Their performance was validated against the analytic solution in the underdamped regime, focusing on displacement, energy decay, and error analysis. Results confirmed that Euler's method diverges rapidly, while RK4 and RK45 closely reproduce the analytic behavior with the expected error scaling. For the Biot–Savart law, numerical integration was performed using midpoint Riemann sums, the trapezoidal rule, and Simpson's rule, along with SciPy's built-in functions. These methods were validated against the analytic finite-wire solution and tested in the limiting cases of short and infinite wire length. Convergence studies revealed error scaling consistent with theoretical predictions: first order for Riemann, second order for trapezoidal, and fourth order for Simpson's rule. These results highlight the strengths and limitations of different numerical integration schemes.

I. INTRODUCTION

The purpose of this project is to compare several numerical integration methods for two different physics problems.

The first problem solved is the time evolution of a damped harmonic oscillator, which is governed by a second-order ODE. Three ODE solvers are implemented to solve this problem: the explicit Euler method, the classical fourth-order Runge–Kutta method (RK4), and the adaptive Runge–Kutta method of order 5(4) (RK45) from the SciPy library. These solvers were validated against the analytic solution in the underdamped regime, specifically looking at displacement and total mechanical energy.

The second problem solved is the magnetic field of a finite current-carrying wire, which requires the evaluation of a definite integral using the Biot–Savart law. The integration techniques compared here are Riemann sums, the trapezoid rule, Simpson's rule, and their SciPy counterparts against the known analytic solution. The results are further tested in the limiting cases of small and large wire length, where the magnetic field should vanish or approach the infinite-wire expression, respectively.

Together, these two problems provide a comprehensive study of numerical methods, for both solving initial-value ODEs and assessing the accuracy of definite integration methods.

II. NUMERICAL METHODS

The following numerical methods are defined in the `Integration.py` file.

A. ODE Solvers

Ordinary differential equation (ODE) solvers are used to approximate solutions to ODEs by using the initial state to estimate the state at some later time. The standard form of the equation for a damped harmonic oscillator is

$$m\ddot{x}(t) + \gamma\dot{x}(t) + kx(t) = 0,$$

where m is the mass, γ is the damping coefficient, and k is the spring constant.

To implement numerical solvers, this second-order ODE is recast as a system of coupled first-order equations by introducing a new variable for velocity,

$$v = \dot{x}$$

,

so that the system becomes

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = -\frac{k}{m}x - \frac{\gamma}{m}v$$

This formulation allows direct application of iterative numerical schemes such as Euler, RK4, and adaptive RK45.

1. *Euler's Method*

Euler's method is the simplest explicit numerical scheme for solving ODEs. Consider the first-order system

$$\frac{dS(t)}{dt} = F(t, S(t)),$$

where $S(t)$ is the state vector. Given a known state $S(t_j)$ at time t_j, Euler's method advances the solution by a timestep h according to

$$S(t_{j+1}) = S(t_j) + hF(t_j, S(t_j)). \tag{1}$$

Here it is assumed that $t_0 = 0$ and $t_f = Nh$, where N is a positive integer. Each step introduces a small error, meaning that the approximation drifts further from the actual solution as more steps are taken. Euler's method is first-order accurate, so its global error scales as $O(h)$ and a higher accuracy can be obtained by implementing a smaller timestep. Thus, it provides a useful baseline against which more accurate methods can be compared.

For the damped harmonic oscillator, the state vector is $S(t) = (x(t), v(t))$, where x is the displacement and $v = \dot{x}$ is the velocity. The Euler updates are then

$$x_{j+1} = x_j + hv_j, \qquad v_{j+1} = v_j + h\left(-\tfrac{k}{m}x_j - \tfrac{\gamma}{m}v_j\right).$$

2. Fourth-Order Runge-Kutta Method

The fourth-order Runge-Kutta method is a method for integrating ODEs obtained from the Taylor series with higher accuracy. It advances the state vector $S(t)$ from t_j to $t_{j+1} = t_j + h$ by evaluating the derivative function F at several intermediate points within the step:

$$k_1 = F(t_j, S_j), \tag{2}$$
$$k_2 = F\big(t_j + \tfrac{h}{2},\, S_j + \tfrac{h}{2}k_1\big), \tag{3}$$
$$k_3 = F\big(t_j + \tfrac{h}{2},\, S_j + \tfrac{h}{2}k_2\big), \tag{4}$$
$$k_4 = F(t_j + h,\, S_j + hk_3). \tag{5}$$

The state is then updated as

$$S_{j+1} = S_j + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{6}$$

The RK4 method is fourth-order accurate, with global error scaling as $O(h^4)$. It is one of the most commonly used integration methods due to its high accuracy and relative simplicity.

For the damped harmonic oscillator, where $S(t) = (x(t), v(t))$, each k_i represents a slope estimate for the coupled equations

$$\dot{x} = v, \qquad \dot{v} = -\tfrac{k}{m}x - \tfrac{\gamma}{m}v.$$

3. SciPy Integrator RK45

There are functions within the SciPy library in Python designed to solve initial value problems for systems of ODEs. One of these solvers is the default Runge–Kutta method of order 5(4), known as RK45, which is available within the `solve_ivp` function. This algorithm assumes accuracy of the fourth-order method to adaptively refine the timestep, but takes steps utilizing the fifth-order accurate formula. This solver was implemented to be compared to the hand-written Euler and RK4 methods outlined previously.

A key feature of the RK45 solver is its ability to automatically adjust the step size during the integration process to maintain a desired level of accuracy. If the error exceeds a specified tolerance, the step is rejected and retried with a smaller h. If the error is much smaller than the tolerance, a larger h is used.

For the damped harmonic oscillator, the function

$$F(t, S) = \begin{bmatrix} v \\ -\frac{k}{m}x - \frac{\gamma}{m}v \end{bmatrix}, \quad S = \begin{bmatrix} x \\ v \end{bmatrix},$$

is supplied to `solve_ivp`, which then returns the trajectory $S(t)$ over the specified interval.

Because the step size is chosen automatically, SciPy RK45 closely matches the analytic solution while avoiding unnecessary computations. It provides a useful reference to which the other implemented integration methods can be compared.

B. Definite Integrals

Many problems in physics require the evaluation of definite integrals to compute physical quantities. Numerical quadrature methods approximate these integrals by replacing the continuous function with a weighted sum over discrete sample points. The accuracy of such methods depends on the number of subdivisions and method used to approximate the function between points.

In this project, numerical integration techniques are applied to the Biot–Savart law for a finite current-carrying wire. The magnetic field at an observation point a perpendicular distance (R) from the wire is given by

$$B_y = \frac{\mu_0 I}{4\pi}\int_{-L/2}^{L/2} \frac{R}{(R^2 + z^2)^{3/2}}, dz,$$

where μ_0 is the permeability of free space, I is the current, L is the length of the wire, and z is the coordinate along the wire.

Because this integral cannot always be expressed easily in closed form, it provides a useful test case for comparing numerical integration methods. By applying Riemann sums, the trapezoid rule, Simpson's rule, and SciPy's built-in routines, we can assess how the choice of numerical scheme affects the accuracy relative to the analytic solution and limiting cases.

1. Riemann Sums

Riemann sums provide a simple way to approximate definite integrals by summing the areas of a series of rectangles under a curve. A definite integral

$$\int_a^b f(x)\,dx$$

can be approximated by the summation

$$\sum_{i=1}^{N} f(x_i^*)\,(x_i - x_{i-1}),$$

where each rectangle has height $f(x_i^*)$ and width $(x_i - x_{i-1})$. The sample point x_i^* is chosen within the subinterval $[x_{i-1}, x_i]$.

In this project, I implemented the *midpoint Riemann sum*, in which

$$x_i^* = \frac{x_{i-1} + x_i}{2},$$

the midpoint of the subinterval. The midpoint rule generally produces a more accurate approximation than left or right Riemann sums, since its error decreases on the order of $1/N^2$, whereas the error of left or right Riemann sums decreases only on the order of $1/N$.

2. Trapezoid Rule

The trapezoid rule provides a more accurate approximation to a definite integral than a basic Riemann sum. Instead of using rectangles to approximate the area under the curve, the trapezoid rule connects successive points of the function with straight line segments and sums the areas of the resulting trapezoids. For a function $f(x)$ defined on the interval $[a, b]$, divided into N equal subintervals of width $\Delta x = (b - a)/N$, the trapezoid rule approximation is

$$\int_a^b f(x)\,dx \;\approx\; \frac{\Delta x}{2}\Big(f(x_0) + 2\sum_{i=1}^{N-1} f(x_i) + f(x_N)\Big),$$

where $x_0 = a$, $x_N = b$, and the points $x_1, \ldots, x_{N-1}$ are the interior grid points. Because each interior point is shared by two trapezoids, its contribution is weighted by a factor of 2. The trapezoid rule has a global error that decreases as $O(1/N^2)$.

3. Simpson's Rule

Simpson's rule provides an even more accurate approximation to a definite integral by fitting quadratic polynomials through the function values, rather than piecewise linear segments. The interval $[a, b]$ is divided into N equal subintervals, where N must be even, with spacing $\Delta x = (b-a)/N$. For each pair of subintervals, Simpson's rule uses the values at the endpoints and the midpoint to construct a quadratic interpolant, which exactly integrates any polynomial up to degree three.

The composite Simpson's rule formula is

$$\int_a^b f(x)\,dx,$$

$$\approx \frac{\Delta x}{3}\Big(f(x_0) + 4\sum_{i=1,\,\text{odd}}^{N-1} f(x_i) + 2\sum_{i=2,\,\text{even}}^{N-2} f(x_i) + f(x_N)\Big)$$

where $x_0 = a$, $x_N = b$, and the intermediate points are $x_i = a + i\Delta x$.

The alternating weights of 4 and 2 arise because midpoints are used more heavily in the quadratic interpolation. Simpson's rule has a global error that decreases as $O(1/N^4)$, making it much more accurate than the trapezoid rule or midpoint Riemann sums, provided the function is sufficiently smooth.

4. SciPy Integration Functions

The `scipy.integrate` subpackage provides several functions for numerical integration, including trapezoidal integration and Simpson's rule. The `scipy.integrate.trapezoid` function implements the trapezoidal rule, which approximates the integrand by connecting sample points with straight line segments. This function provides reliable accuracy for simple numeric integration such as piecewise linar functions. The `scipy.integrate.simpson` function implements Simpson's rule, which approximates a function with quadratic polynomials over pairs of subintervals. For smooth functions, the error decreases as $O(1/N^4)$, making this method more efficient for achieving high accuracy than trapezoidal integration.

III. VALIDATION

The purpose of this section is to validate the implemented numerical methods against the known analytic solution of the damped harmonic oscillator and the magnetic field of the infinite current-carrying wire.

A. Damped Harmonic Oscillator

1. Analytic Solution

Figure 1 shows the displacement of the damped harmonic oscillator obtained using the analytic solution and three numerical solvers: the explicit Euler method, the classical Runge–Kutta method (RK4), and SciPy's adaptive Runge–Kutta method (RK45).

The analytic solution serves as the exact reference, while the numerical methods approximate the system's behavior over time. As expected, the Euler method quickly deviates from the analytic curve, overestimating the amplitude of oscillations due to its first-order accuracy. The RK4 solution, being fourth-order accurate, closely follows the analytic solution throughout the simulation, with only small phase differences appearing at later times. Finally, the adaptive RK45 solver maintains excellent agreement with the analytic solution, demonstrating both its accuracy and stability for this problem.

This comparison highlights the limitations of low-order methods such as Euler for oscillatory systems, and the benefits of higher-order schemes like RK4 and adaptive solvers, especially over longer periods of time. In this case, the accuracy of Euler's method could be improved by using a smaller step size, though this does not eliminate long term-instability.

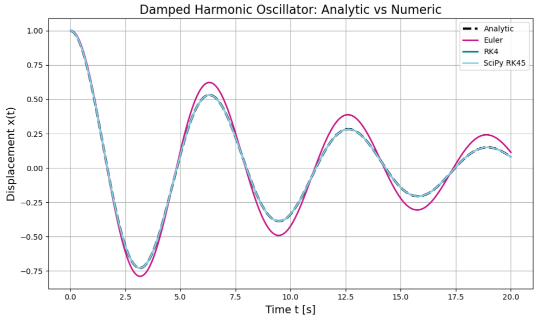

FIG. 1. Displacement of the damped harmonic oscillator obtained from the analytic solution (black dashed) compared with numerical solutions using Euler's method (magenta), RK4 (teal), and SciPy's adaptive RK45 (light blue).

2. *Damped Harmonic Oscillator for Different Regimes*

The equation of motion for the damped harmonic oscillator is given by

$$\ddot{x} + \frac{\mu}{m}\dot{x} + \frac{k}{m}x = 0,$$

which is a homogeneous second-order differential equation with constant coefficients.

To solve such equations, we employ an exponential ansatz of the form

$$x(t) = Ce^{\lambda t},$$

which, when substituted into the equation of motion, yields the characteristic equation

$$\lambda^2 + \frac{\mu}{m}\lambda + \frac{k}{m} = 0.$$

The solutions to this quadratic are

$$\lambda_{1,2} = \frac{-\mu \pm \sqrt{\mu^2 - 4mk}}{2m}.$$

Thus, the general homogeneous solution is

$$x(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}.$$

The qualitative behavior of the damped harmonic oscillator depends on the discriminant

$$\Delta = \gamma^2 - 4mk.$$

When $\Delta < 0$, the system is underdamped and exhibits oscillatory motion with an exponentially decaying envelope. When $\Delta = 0$, the system is critically damped and returns to equilibrium in the shortest possible time without oscillations. When $\Delta > 0$, the system is overdamped, in which case the oscillator returns to equilibrium exponentially without oscillating.

3. *Underdamped Regime*

In the underdamped case, $\Delta < 0$, so the square root is imaginary. Defining

$$\lambda_{1,2} = -\frac{\mu}{2m} \pm i\omega_d,$$

with

$$\omega_d = \sqrt{\frac{k}{m} - \left(\frac{\mu}{2m}\right)^2},$$

the solution takes the form

$$x(t) = e^{-\mu t/(2m)} \left[A\cos(\omega_d t) + B\sin(\omega_d t)\right],$$

which describes oscillations with exponentially decaying amplitude.

Critically Damped Regime

When $\Delta = 0$, the two roots coincide,

$$\lambda = -\frac{\mu}{2m}.$$

In this case, the solution is

$$x(t) = (A + Bt)e^{-\mu t/(2m)},$$

which describes a non-oscillatory return to equilibrium in the shortest possible time.

Overdamped Regime

When $\Delta > 0$, both roots are real and negative,

$$\lambda_{1,2} = \frac{-\mu \pm \sqrt{\mu^2 - 4mk}}{2m}.$$

The general solution is

$$x(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t},$$

with $\lambda_1 \neq \lambda_2$. This describes an exponential return to equilibrium without oscillations, slower than the critically damped case.

Results

The damping coefficient μ determines the qualitative behavior of the oscillator. In the underdamped regime ($\mu^2 < 4mk$), the system oscillates with a gradually decaying amplitude. In the critically damped case ($\mu^2 = 4mk$), the system returns to equilibrium in the shortest possible time without oscillations. Finally, in the overdamped regime ($\mu^2 > 4mk$), the system also returns to equilibrium without oscillations, but more slowly than in the critically damped case. These behaviors are illustrated in Figure 2, which shows the displacement as a function of time for the three regimes.

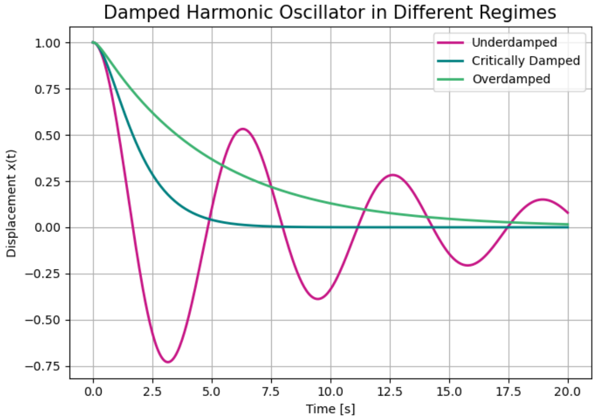

FIG. 2. Displacement $x(t)$ for the damped harmonic oscillator in the underdamped, critically damped, and overdamped regimes.

4. Total Mechanical Energy Decay

A key characteristic of the damped harmonic oscillator is the presence of a damping force that opposes the motion. This force is non-conservative, and therefore the total mechanical energy of the system decreases with time. Physically, the damping extracts energy from the oscillator, leading to a gradual loss of amplitude.

The total mechanical energy is defined as

$$E(t) = \tfrac{1}{2}m\dot{x}^2(t) + \tfrac{1}{2}kx^2(t).$$

Using the analytic solution derived in the previous section, the position $x(t)$ is obtained directly, while the velocity $\dot{x}(t)$ is calculated as its time derivative. For the numerical integration methods (Euler, RK4, and SciPy's RK45), the velocity values are obtained as part of the state evolution.

Figure 3 shows the total energy $E(t)$ of the underdamped oscillator computed with each method. As expected, the analytic solution predicts a smooth monotonic decrease in energy. The numerical methods capture the same qualitative trend, although Euler and RK4 exhibit small oscillations in the computed energy due to numerical error. In this case, Euler's method deviates noticeably from the analytic solution. The SciPy RK45 solution matches the analytic curve most closely. In all cases, the total energy tends to zero as $t \to \infty$, consistent with the physical effect of damping.

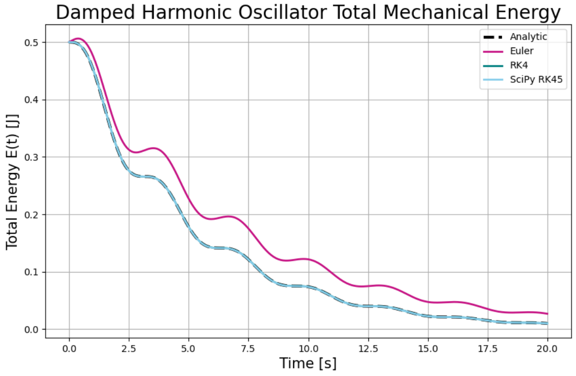

FIG. 3. Total mechanical energy $E(t)$ of the underdamped harmonic oscillator computed using the analytic solution, Euler, RK4, and SciPy RK45 methods.

B. Validation of Biot–Savart Implementation

The numerical integration routines were first validated against known solutions of the Biot–Savart law. Figure 4 shows the magnetic field as a function of wire length L at a fixed observation distance $R = 0.1\,\mathrm{m}$. The results from the hand-written Riemann, trapezoidal, and Simpson's rules, as well as SciPy's built-in routines, are plotted alongside the analytic solution.

Figure 4 compares the magnetic field computed numerically (using Riemann, trapezoidal, and Simpson's rule, along with SciPy's built-in functions) to the analytic solution as a function of wire length L. For small and moderate L, all methods closely reproduce the analytic result. However, as the wire length increases, the solutions begin to diverge slightly, most noticeably for the handwritten Simpson method. Although the general trends remain consistent, the calculations of long wires are much more sensitive to errors caused by the chosen integration method. The integration performed with Simpson's rule, in particular, deviates noticeably from the analytic solution. This could be caused by round-off error.

1. Limiting cases of L

Next, the limiting cases $L \to 0$ and $L \to \infty$ were examined. In the short-wire limit, the magnetic field tends to zero, consistent with the vanishing contribution of a negligibly small current segment. In the long-wire limit,

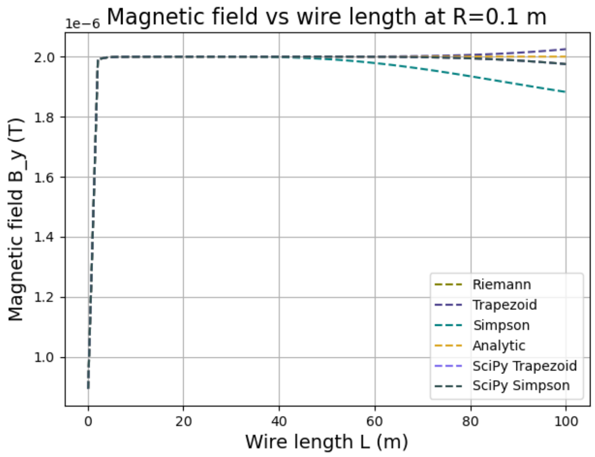

FIG. 4. Magnetic field as a function of wire length L at fixed radius $R = 0.1\,\text{m}$. Numerical integration methods agree with the analytic Biot–Savart solution.

the numerical results approach

$$B = \frac{\mu_0 I}{2\pi R},$$

which is the well-known expression for the magnetic field of an infinitely long straight wire. These behaviors are shown in Figure 5, further validating that the numerical implementation respects the physical constraints of the Biot–Savart law.

IV. ERROR ANALYSIS

In this section, the accuracy and limitations of the numerical methods are explored. This helps to establish the conditions under which a method can be trusted for predictive calculations.

A. ODE Solvers

To quantify the accuracy of the numerical integration methods, we evaluated both the local and global errors of the explicit Euler and Runge–Kutta 4 (RK4) schemes for the damped harmonic oscillator.

The local error is defined as

$$e_n = |x_n - x(t_n)|,$$

where x_n is the numerical solution at time t_n and $x(t_n)$ is the exact analytic solution. Figure 6 shows the local error as a function of time for a step size of $h = 0.05$. As expected, Euler's method exhibits larger errors that grow steadily with time, while RK4 maintains significantly smaller local errors across the entire interval.

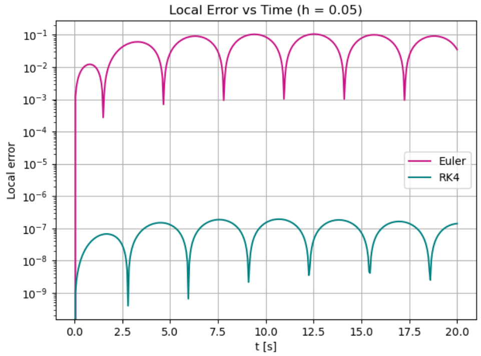

FIG. 6. Local error versus time for the damped harmonic oscillator ($h = 0.05$). Euler's method accumulates error more rapidly than RK4.

The oscillatory structure of the local error in Fig. 6 reflects the periodic nature of the damped harmonic oscillator. At each zero-crossing of the displacement, the numerical and analytic solutions nearly coincide, producing sharp dips in the error. Between crossings, small discrepancies in phase and amplitude accumulate, causing the error to grow until the next crossing. This repeating pattern leads to the characteristic "bumps" in the local error curves. The effect is much more pronounced for Euler's method, which suffers from large phase and amplitude errors, while RK4 maintains errors several orders of magnitude smaller but still exhibits the same periodic behavior caused by the oscillation of the system.

The global error measures the maximum deviation from the analytic solution over the full simulation window,

$$e_{\text{global}} = \max_n |x_n - x(t_n)|.$$

By computing the global error for a range of step sizes h, we can examine the convergence rate of each method. Figure 7 shows the global error as a function of h on a log-log scale. Since the error is defined as the difference between two quantities with the same units, it is a relative measure and therefore unitless. The Euler method displays first-order convergence, with error scaling as $\mathcal{O}(h)$. The RK4 method achieves fourth-order accuracy, scaling as $\mathcal{O}(h^4)$. Both slopes are consistent with theoretical expectations. The Euler result begins to deviate more from the expected slope value with larger step size h, which is expected.

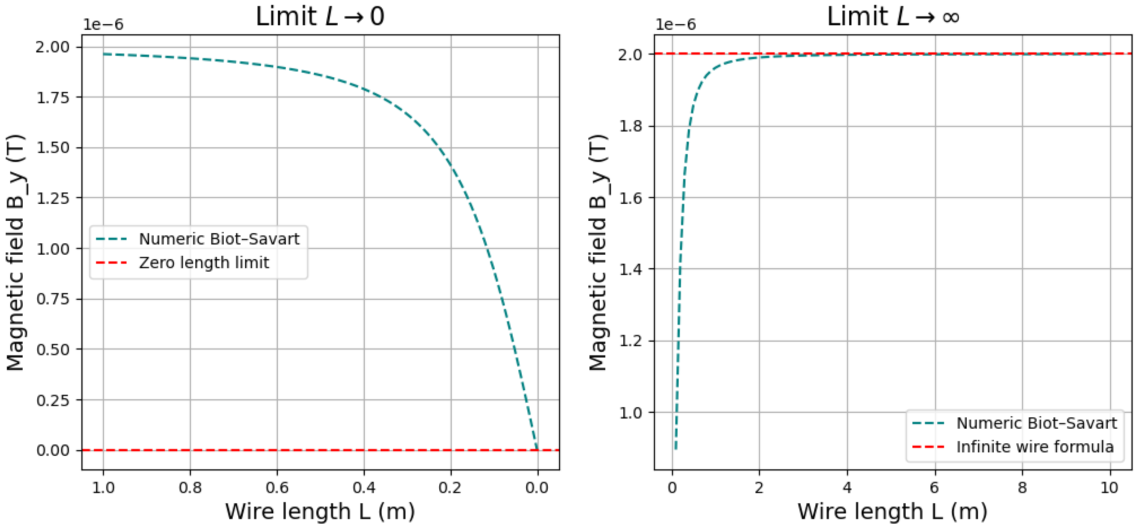

FIG. 5. Verification of limiting behavior of the Biot–Savart law. (Left) In the limit $L \to 0$, the magnetic field vanishes. (Right) In the limit $L \to \infty$, the numerical result approaches the analytic expression for an infinite wire.

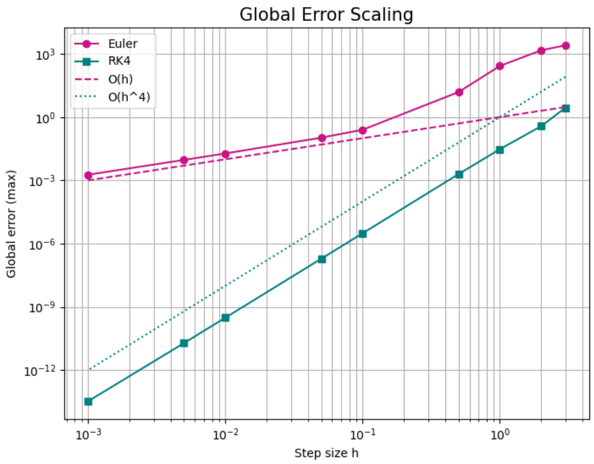

FIG. 7. Global error scaling for Euler and RK4 methods. Reference lines with slopes 1 and 4 are shown for comparison.

These results confirm that the implementations of Euler and RK4 are correct and that their accuracy behaves as predicted. While Euler is useful for simple understandings, its poor accuracy and stability make RK4 the preferred choice for practical applications.

B. Biot–Savart Integration

To assess the accuracy of the numerical integration methods applied to the Biot–Savart law, we compared the Riemann sum, trapezoidal rule, and Simpson's rule against the analytic solution for the magnetic field of a finite straight wire. For a fixed wire length $L = 1.0\,\mathrm{m}$ and observation point at a perpendicular distance $R = 0.1\,\mathrm{m}$, the integral

$$B_y = \frac{\mu_0 I}{4\pi} \int_{-L/2}^{L/2} \frac{R}{(R^2 + z^2)^{3/2}}\, dz$$

was evaluated numerically using increasing numbers of subintervals n. The absolute error was computed as

$$e = \left| B_{\text{numeric}} - B_{\text{analytic}} \right|,$$

where B_{analytic} is obtained from the closed-form solution of the Biot–Savart law.

The convergence of each method was quantified by plotting the error on a log–log scale against n. Linear regression of the slope of these plots provided numerical estimates of the convergence rate. As expected, the Riemann sum exhibited first-order convergence ($\mathcal{O}(h)$), the trapezoidal rule second-order convergence ($\mathcal{O}(h^2)$), and Simpson's rule fourth-order convergence ($\mathcal{O}(h^4)$). To aid visual comparison, reference lines with slopes of -1, -2, and -4 were also plotted on the log–log error graph. The close alignment of the numerical error curves with these reference slopes validates both the implementation and the theoretical convergence behavior.

Figure 8 shows the error convergence plot for the three methods, demonstrating the difference in efficiency. For the same number of integration points, Simpson's rule achieves substantially lower error compared to Riemann and trapezoidal methods.

V. CONCLUSION

In this work, numerical integration methods for two problems in classical physics, the Biot–Savart law for a

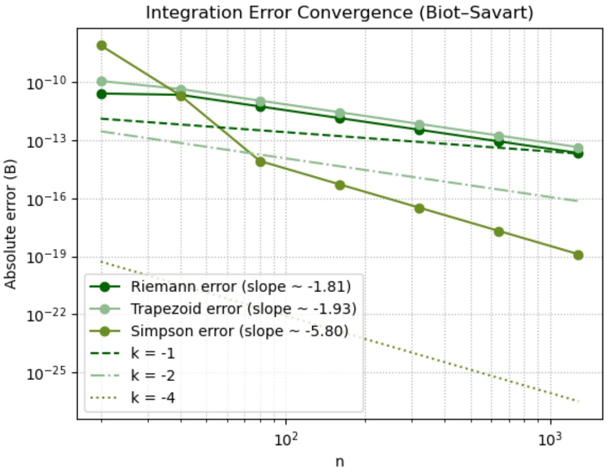

FIG. 8. Error convergence of numerical integration methods applied to the Biot–Savart law for a finite wire. Reference slopes of −1, −2, and −4 are plotted for comparison, corresponding to the expected convergence orders of Riemann, trapezoidal, and Simpson's rule, respectively.

finite current-carrying wire and the damped harmonic oscillator, were validated and analyzed. For the Biot–Savart case, numerical integrators were shown to reproduce the analytic solution across a wide range of wire lengths, and the correct limiting behavior was verified as $L \to 0$ and $L \to \infty$. The convergence study confirmed the expected error scaling: first order for the Riemann sum, second order for the trapezoidal rule, and fourth order for Simpson's rule.

For the harmonic oscillator, explicit Euler, RK4, and SciPy's RK45 integrators were compared against the analytic solution. The results demonstrated the rapid accumulation of error in Euler's method, in contrast to the accuracy and stability of RK4 and RK45. Local and global error analysis further confirmed the theoretical error orders, with Euler scaling as $\mathcal{O}(h)$ and RK4 as $\mathcal{O}(h^4)$. In addition, energy plots displayed physically realistic damping behavior for different damping regimes.

Together, these studies highlight the importance of validation and error analysis in computational physics. Validation ensures that numerical solvers align with known analytic limits and physical laws, while error analysis quantifies the reliability of each method. These methods provide the foundation for the use of numerical methods to study systems where exact solutions are not available.

[1] A. Bayen, Q. Kong, and T. Siauw, *Python Programming and Numerical Methods: A Guide for Engineers and Scientists* (Academic Press, 2020).

[2] Beltoforion, "Harmonic Oscillator," `https://beltoforion.de/en/harmonic_oscillator/`.

[3] Griffiths, David J. (David Jeffery), 1942-. Introduction to Electrodynamics. Boston :Pearson, 2013.

[4] "Matplotlib 3.10.6 Documentation." Matplotlib Documentation, matplotlib.org/stable/.

[5] ”Patrick Walls, Mathematical Python, (2022), Github repository, `https://patrickwalls.github.io/mathematicalpython/`”

[6] "SciPy v1.16.2 Manual." Integration (Scipy.Integrate), `docs.scipy.org/doc/scipy/tutorial/integrate.html`.