Numerical Integration of Damped Harmonic Oscillator and Gaussian Functions

Luis Rufino

September 28, 2024

Abstract

This report presents the implementation of numerical integration methods applied to two classical physics problems: the damped harmonic oscillator and the Gaussian integral. Euler's method, the 4th order Runge-Kutta method (RK4), and external algorithms (e.g., from SciPy) are used to numerically solve the equations of motion and evaluate integrals. Results are compared to analytic solutions, and the accuracy of each method is discussed in terms of truncation and global errors.

1 Introduction

In this project, I explore two classical problems:

- The Damped Harmonic Oscillator, described by a second-order differential equation.
- The Gaussian integral, a common integral in physics and probability theory.

The goal is to numerically solve these problems using different integration algorithms and analyze their performance, comparing results with exact solutions where possible.

2 Gaussian Integral and Numerical Integration Methods

In addition to solving the damped harmonic oscillator, I also tested the performance of our integration methods on a classic mathematical problem: the Gaussian integral. The Gaussian function,

$$f(x) = e^{-x^2},$$

is ubiquitous in physics and mathematics, and its integral over the entire real line has a well-known analytic solution:

$$I = \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

In this project, I approximate this integral numerically using finite integration limits and compare the results from different numerical methods: Riemann Sum, Trapezoidal Rule, Simpson's Rule, and the SciPy `quad` function.

2.1 Numerical Approximation of the Gaussian Integral

To compute the integral numerically, I choose finite integration limits $a = -10$ and $b = 10$, as this range captures most of the area under the Gaussian curve. The numerical methods implemented for this task are:

- **Riemann Sum**: A basic numerical integration method that approximates the integral as the sum of rectangular areas. Mathematically, it is expressed as:

$$I_{\text{Riemann}} = \sum_{i=1}^{n} f(x_i)\Delta x$$

where $f(x_i)$ is the function value at point x_i and Δx is the width of each interval.

- **Trapezoidal Rule**: Improves upon the Riemann sum by approximating the area under the curve using trapezoids. The integral is given by:

$$I_{\text{Trapezoidal}} = \frac{\Delta x}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]$$

where x_0 and x_n are the endpoints, and $f(x_i)$ is the function evaluated at intermediate points.

- **Simpson's Rule**: A higher-order method that fits parabolas to sections of the function. The integral is approximated as:

$$I_{\text{Simpson}} = \frac{\Delta x}{3}\left[f(x_0) + 4\sum_{\text{odd } i} f(x_i) + 2\sum_{\text{even } i} f(x_i) + f(x_n)\right]$$

where x_0 and x_n are the endpoints, and the sums are over odd and even-indexed points, respectively.

2.2 Results and Discussion

The exact value of the Gaussian integral in the limits $[-10, 10]$ can be computed with high precision using the SciPy `quad` function. I then compare the results from the different numerical methods, as shown in Table 1.

Method	Approximate Value
Riemann Sum	1.7706813970546111
Trapezoidal Rule	1.7724538509055168
Simpson's Rule	1.7724538509055163
SciPy Quad	1.772453850905516
Exact Value	$\sqrt{\pi} \sim 1.77245385090551602$

Table 1: Comparison of numerical methods for the Gaussian integral in the range $[-10, 10]$. Using 1000 bins to approximate the integral

All numerical methods provide reasonable approximations to the Gaussian integral within the given range. Simpson's Rule, the Trapezoidal Rule, and SciPy's `quad` function perform similarly, achieving high accuracy and closely matching the exact value. These methods benefit from their ability to account for the smooth curvature of the Gaussian function, leading to minimal error.

However, the Riemann Sum method exhibits slightly reduced accuracy compared to the other methods. This discrepancy arises from its less sophisticated treatment of the function's shape.

Figure 1 illustrates the convergence behavior of the numerical methods as a function of the number of integration points.

For this particular integral, the Riemann Sum, Trapezoidal Rule, and Simpson's Rule converge at a similar rate. None of these methods provide reliable results until at least 20 bins are used, highlighting the need for a sufficient number of bins to achieve accuracy.

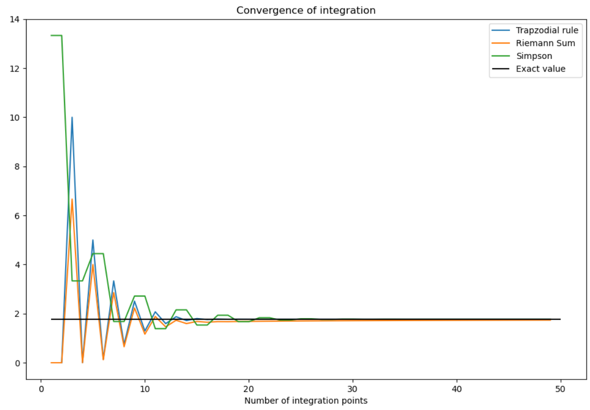

Figure 1: Convergence of the numerical methods for the Gaussian integral as the number of points increases.

2.3 Conclusion

The Gaussian integral provides a valuable test case for evaluating the performance of different numerical integration methods. While the Riemann Sum and Trapezoidal Rule provide simple and quick approximations, they are outperformed by Simpson's Rule, which delivers significantly better accuracy. The SciPy `quad` function serves as a benchmark for high-precision integration.

3 Damped Harmonic Oscillator and Numerical Methods

The damped harmonic oscillator is governed by the following equation of motion:

$$m\frac{d^2q}{dt^2} + b\frac{dq}{dt} + kq = 0, \tag{1}$$

where $q(t)$ is the position, m is the mass, b is the damping coefficient, and k is the spring constant.

For small damping ($b \ll 1$), the system exhibits underdamped oscillations with decreasing amplitude. As b increases, the system moves into critical or overdamped regimes. I aim to solve the underdamped case numerically using the methods described below.

3.1 Numerical Algorithms

In this project, I aim to solve the equations of motion for a damped harmonic oscillator numerically. The Lagrangian formulations can describe the system. However, since the damped harmonic oscillator involves a non-conservative damping force, I modify the Lagrangian to account for this force and use it to derive the equations of motion. I then apply various numerical methods to approximate the solution.

3.2 Lagrangian Formulation with Non-Conservative Forces

In classical mechanics, the Lagrangian $L(q, \dot{q})$ is typically given by:

$$L(q, \dot{q}) = T - V = \frac{1}{2}m\dot{q}^2 - \frac{1}{2}kq^2, \tag{2}$$

where T is the kinetic energy and V is the potential energy. However, for the damped harmonic oscillator, the damping force, $-b\dot{q}$, is non-conservative and cannot be derived from a potential. As a result, I must modify the Lagrangian formulation to account for this non-conservative force.

One way to handle this is by incorporating Rayleigh's dissipation function, $F(\dot{q})$, which models energy loss due to damping. The dissipation function for the damping term is given by:

$$F(\dot{q}) = \frac{1}{2} b \dot{q}^2, \tag{3}$$

where b is the damping coefficient. The equation of motion is then modified by including this force term into the Euler-Lagrange equation as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial F}{\partial \dot{q}} = 0. \tag{4}$$

Substituting the expressions for L and F, I recover the equation of motion for the damped harmonic oscillator:

$$m\ddot{q} + b\dot{q} + kq = 0. \tag{5}$$

3.3 Exact Solution to the Damped Harmonic Oscillator

The equation of motion for a damped harmonic oscillator is given by:

$$m\ddot{q} + b\dot{q} + kq = 0, \tag{6}$$

where m is the mass, b is the damping coefficient, and k is the spring constant. This is a second-order linear differential equation, and its exact solution depends on the value of the damping coefficient b.

The characteristic equation corresponding to the differential equation is:

$$m\lambda^2 + b\lambda + k = 0, \tag{7}$$

which has the solution:

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4mk}}{2m}. \tag{8}$$

There are three possible cases based on the discriminant $\Delta = b^2 - 4mk$:

1. **Underdamped case** ($\Delta < 0$): In this case, the solution is oscillatory with an exponentially decaying amplitude:

$$q(t) = e^{-\frac{b}{2m}t}\left(A\cos(\omega_d t) + B\sin(\omega_d t)\right), \tag{9}$$

where $\omega_d = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2}$ is the damped natural frequency, and A and B are constants determined by the initial conditions.

2. **Critically damped case** ($\Delta = 0$): The system returns to equilibrium as quickly as possible without oscillating:

$$q(t) = (A + Bt)e^{-\frac{b}{2m}t}. \tag{10}$$

3. **Overdamped case** ($\Delta > 0$): The system returns to equilibrium slowly without oscillating:

$$q(t) = Ae^{\lambda_1 t} + Be^{\lambda_2 t}, \tag{11}$$

where λ_1 and λ_2 are the two distinct roots of the characteristic equation.

In our case, the underdamped scenario is of primary interest, where the system oscillates with a gradually decreasing amplitude. This exact solution will serve as a benchmark for validating our numerical methods. Specifically, I can compare the numerically computed phase space trajectories with those predicted by the analytical solution.

3.4 Euler's Method

Euler's method is a simple numerical integration technique based on a first-order approximation of the system's evolution. For a differential equation of the form:

$$\frac{dq}{dt} = f(q, t), \tag{12}$$

Euler's method approximates the value of q at the next time step as:

$$q_{n+1} = q_n + \Delta t \cdot f(q_n, t_n), \tag{13}$$

where Δt is the time step size. For the damped harmonic oscillator, using the Hamiltonian formulation, the position q and momentum p are updated as follows:

$$q_{n+1} = q_n + \Delta t \cdot \frac{p_n}{m}, \tag{14}$$

$$p_{n+1} = p_n - \Delta t \cdot (kq_n + bp_n). \tag{15}$$

Though straightforward, Euler's method is limited by its relatively high truncation error, making it less accurate for large-time steps.

3.5 Runge-Kutta 4th Order Method (RK4)

The Runge-Kutta 4th order method (RK4) improves upon Euler's method by utilizing intermediate steps to more accurately estimate the position and momentum at each time step. For the damped harmonic oscillator, the system of equations is given by:

$$\frac{dq}{dt} = \frac{p}{m}, \quad \frac{dp}{dt} = -kq - bp$$

In the RK4 method, four increments (k_1, k_2, k_3, and k_4) are calculated at each time step for both position q and momentum p:

$$\begin{aligned}
k_1 &= f(q_n, p_n), && (16)\\
k_2 &= f\left(q_n + \frac{\Delta t}{2} k_{1,q}, p_n + \frac{\Delta t}{2} k_{1,p}\right), && (17)\\
k_3 &= f\left(q_n + \frac{\Delta t}{2} k_{2,q}, p_n + \frac{\Delta t}{2} k_{2,p}\right), && (18)\\
k_4 &= f(q_n + \Delta t \cdot k_{3,q}, p_n + \Delta t \cdot k_{3,p}), && (19)
\end{aligned}$$

where $k_{i,q}$ and $k_{i,p}$ represent the changes in position and momentum, respectively. The system's position and momentum are then updated as a weighted average of these increments:

$$q_{n+1} = q_n + \frac{1}{6}(k_{1,q} + 2k_{2,q} + 2k_{3,q} + k_{4,q}),$$
$$p_{n+1} = p_n + \frac{1}{6}(k_{1,p} + 2k_{2,p} + 2k_{3,p} + k_{4,p}).$$

This method achieves higher accuracy by taking into account both the current and future states of the system, making it more suitable for solving stiff or complex systems such as the damped harmonic oscillator.

$$q_{n+1} = q_n + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \tag{20}$$

This method provides significantly better accuracy than Euler's method, particularly for larger time steps.

3.6 SciPy's ODE Solver (`solve_ivp`)

For more complex systems or when high accuracy is required, I use the adaptive step-size ODE solver provided by SciPy's `solve_ivp` function. This solver automatically adjusts the time step size to control the error and efficiently handles stiff problems. The equations of motion for the damped harmonic oscillator are integrated as:

$$\frac{dq}{dt} = \frac{p}{m}, \tag{21}$$

$$\frac{dp}{dt} = -kq - bp. \tag{22}$$

`solve_ivp` combines multiple methods, including the Runge-Kutta-Fehlberg method, to dynamically balance accuracy and computational cost.

4 Validation and Testing

In this section, I validate the numerical integration methods by comparing their results against the exact solutions of the damped harmonic oscillator for different damping regimes: underdamped, critically damped, and overdamped. I test the Euler method, Runge-Kutta 4th-order (RK4) method, and SciPy's `solve_ivp` function to evaluate their performance and accuracy.

The exact solution for the damped harmonic oscillator is given by:

$$q(t) = Ae^{-\frac{b}{2m}t} \cos(\omega_d t + \phi),$$

where $\omega_d = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2}$ is the damped angular frequency in the underdamped case. For critically damped and overdamped regimes, the solution has the following forms:

- **Critically damped** ($b^2 = 4mk$):

$$q(t) = (C_1 + C_2 t)e^{-\frac{b}{2m}t},$$

where C_1 and C_2 are constants determined by the initial conditions.

- **Overdamped** ($b^2 > 4mk$):

$$q(t) = C_1 e^{r_1 t} + C_2 e^{r_2 t},$$

where r_1 and r_2 are the roots of the characteristic equation, given by

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

I compare the exact solutions to the results from the Euler, RK4, and SciPy ODE solvers, focusing on each damping regime.

4.1 Underdamped Case

In the underdamped regime, where $b^2 < 4mk$, the solution shows oscillatory behavior with a decaying amplitude. This is characterized by the damped frequency ω_d. Figure 3 shows the oscillations (position vs time) for this case.

In this case, the Euler method does not match the exact solution, though it does capture the key feature of a decaying amplitude. The implemented RK4 method and the SciPy ODE solver match the exact solution almost exactly. Since the default method used by SciPy is also the RK4 method, it is no surprise that the results from these two methods are closely aligned, appearing nearly identical in the simulation.

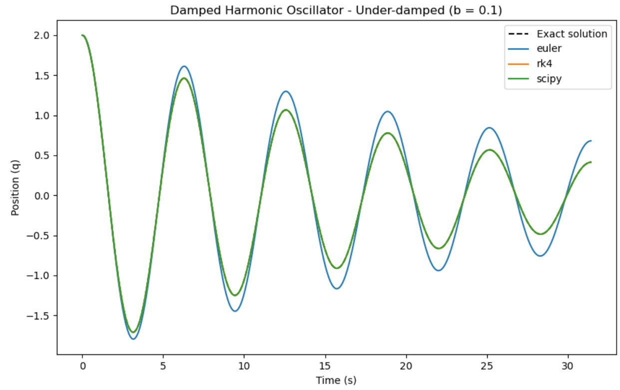

Figure 2: Oscillations of the underdamped harmonic oscillator ($b = 0.1$) comparing Euler, RK4, and SciPy ODE solver results with the exact solution.

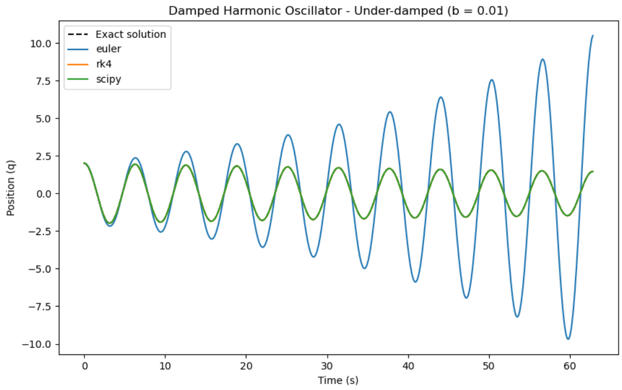

Figure 3: Oscillations of the underdamped harmonic oscillator ($b = 0.01$) comparing Euler, RK4, and SciPy ODE solver results with the exact solution.

Most of the numerical methods align closely with the exact solution for the damped harmonic oscillator. However, as time progresses, the Euler method begins to exhibit an increasing amplitude, deviating significantly from the exact solution. This divergence illustrates the limitations of the Euler method, particularly in long-term simulations of oscillatory systems, where more accurate methods, such as the Runge-Kutta 4th order and the SciPy ODE solver, demonstrate superior performance in maintaining agreement with the exact analytical results.

4.2 Critically Damped Case

For the critically damped case ($b^2 = 4mk$), the system returns to equilibrium without oscillating.

In this case, all methods closely capture the behavior of the exact solution, with minimal deviations, particularly in the later stages of the system's return to equilibrium.

4.3 Overdamped Case

In the overdamped regime, where $b^2 > 4mk$, the system returns to equilibrium slowly without oscillating.

In the overdamped case, all methods perform very well, with each approaching the exact solution instantly. The Euler method, RK4 method, and SciPy ODE solver all show good agreement with the exact solution, particularly in the system's slow approach to equilibrium.

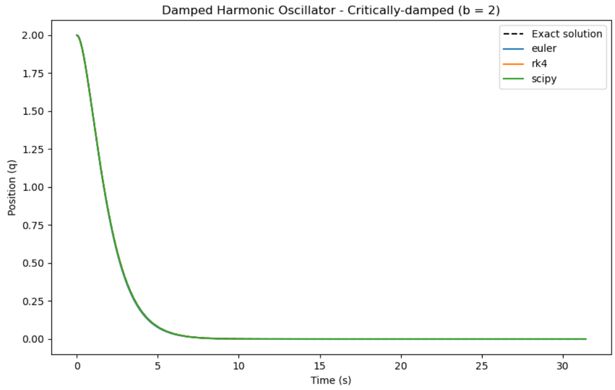

Figure 4: Position vs time of the critically damped harmonic oscillator ($b = 2$) comparing Euler, RK4, and SciPy ODE solver results with the exact solution.

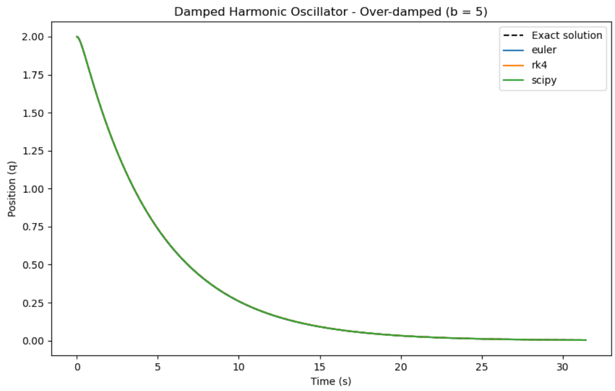

Figure 5: Position vs time of the overdamped harmonic oscillator ($b = 5$) comparing Euler, RK4, and SciPy ODE solver results with the exact solution.

4.4 Comparison of Exact Solutions and Numerical Methods

To further analyze the performance of the numerical integration methods, I calculated the absolute differences between the exact solutions and the numerical results obtained from the Euler method, the Runge-Kutta 4th order method (RK4), and SciPy's ODE solver for the damped harmonic oscillator under different damping scenarios: underdamped, critically damped, and overdamped.

For each damping case, I defined the exact solution based on the damping coefficient b. The exact solutions were computed as before

For each damping case, I varied the time step size and computed the numerical solutions over a specified time interval. The absolute difference between the numerical solutions and the corresponding exact solutions was calculated. The results were plotted to visually compare the discrepancies for each numerical method across the different damping scenarios.

The 6 illustrated how the Euler method, RK4, and SciPy's ODE solver performed relative to the exact solutions. This analysis allows you to see the accuracy of each numerical method.

4.5 Conclusion

I explored the dynamics of the damped harmonic oscillator through analytical solutions and various numerical methods, including Euler's method, the Runge-Kutta 4th order method, and

SciPy's ODE solver. Our analysis confirmed that while all methods exhibit the general behavior of the system, discrepancies arise, particularly with Euler's method in the underdamped regime. The Euler method demonstrated an increasing amplitude over time.

Both the RK4 and SciPy methods maintained close agreement with the exact analytical solution across all damping regimes, showcasing their superior performance and reliability. The rigorous comparison of these numerical methods highlighted the importance of selecting an appropriate integration technique.

Overall, the damped harmonic oscillator not only validated the effectiveness of advanced numerical techniques.

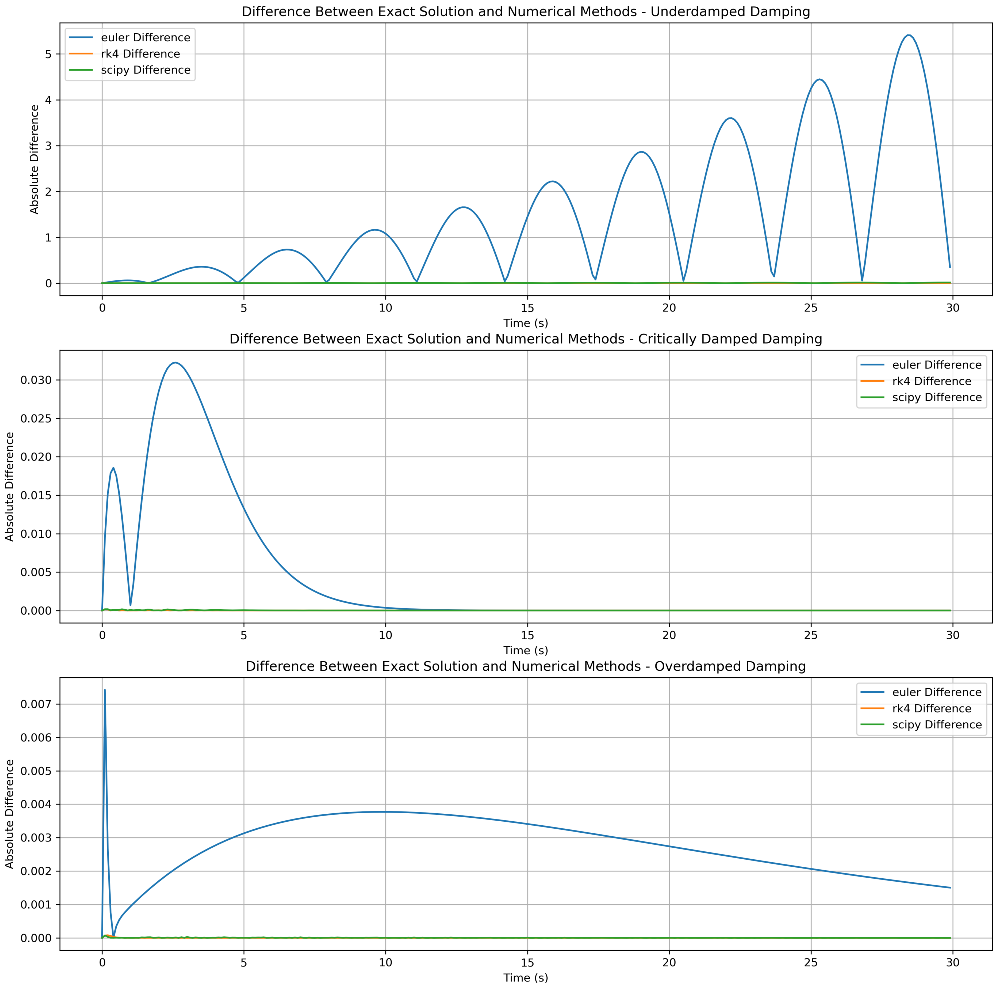

Figure 6: Absolute difference between the exact solution and numerical methods (Euler, RK4, and SciPy) for the damped harmonic oscillator across different damping scenarios: underdamped, critically damped, and overdamped.