Task: You are tasked to write an academic article on a ground breaking methodology your research lab has discovered called ”Rung-Kutta” iterations for numerical differentiation. Although this method actually exists, this assignment should be done as if you are the one who has discovered it and are hereby trying to publish it.

Abstract

Given the ubiquity of differential equations in mathematical modeling, the design of suitable numerical algorithms for accurately approximating solutions is essential. In this article, we present the Runge-Kutta iterative methodology for numerical differentiation, offering a substantial improvement in accuracy and convergence over classical techniques. Namely, we described the fundamental flaw with Euler's Method; it only has first-order convergence, lending itself to rapid error accumulation over large enough step sizes. Through the synthesis of Euler's method, midpoint approximations, and weighted averaging, the Runge-Kutta 4 scheme accomplishes a robust fourth-order global truncation error ($\mathcal{O}(h^4)$), minimizing the aggregate error for small and large step sizes. Alongside its fast convergence, RK-4 can achieve similar results compared to Euler's method with only roughly $\frac{1}{4}$ of the necessary computations, enabling larger step sizes and lowering the overall computational cost. Additionally, our study exemplifies the strength of RK-4 in two applications such as modeling decay and epidemics, where the latter does not have a realized solution. Nonetheless, the RK-4 exhibited extraordinary precision, widening the difference in error against RK-1 by up to 10^8 times for the smallest step size and 10 times for the largest step size. Hence, our results agree with the theoretical analysis posed in this article, suggesting that Runge-Kutta can be well extrapolated to a broad range of models with reliability. Ultimately, our findings establish Runge-Kutta as a transformative algorithm in numerical analysis with boundless applications in STEM related disciplines.

1 Introduction

1.1 Motivation

Differential equations have been widely used and play a crucial role in mathematical and scientific modeling. Although analytical solutions can be found for some differential equations, others require the development of iterative solutions. With existing numerical differentiation techniques, their mileage varies in capturing important relationships and reliably fitting into models.

In response to these challenges, our research lab has proposed a new iterative approach, named the “Runge-Kutta (RK)” method. Our experimental work has proven that Runge-Kutta promises a significant improvement in accuracy, convergence, stability, and adaptability of numerical differentiation that other methods may fail to satisfy. With these ideas in mind, we aim to show that Runge-Kutta is a step forward in numerical differentiation that STEM professionals and researchers can benefit from.

1.2 Overview of the Article

In this report, we present the theoretical foundation and analysis of the Runge-Kutta methodology and highlight its strengths in numerically solving differential equations with accuracy and stability. In Section 2, we provide a detailed explanation and derivation of the fourth-order Runge-Kutta algorithm. In Section 3, we demonstrate the Runge-Kutta 1, 2, and 4 methods to two relevant differential equations, providing side-by-side comparisons and analysis. Lastly, conclusions and opinions are offered in Section 4.

2 Methodology

2.1 Euler’s Method

The primary challenges confronting a numerical analyst of ordinary differential equations appear in a simple form; we wish to construct a reasonable approximation to the unique solution of the initial value problem (IVP):

$$\frac{dy}{dt} = f(y,t), \quad y(t_0) = y_0. \tag{1}$$

The very simplest of these numerical solutions is Euler’s method, which introduces the notion of discretizing a continuous space into partitioned intervals. More precisely, starting at t_0, we define successive mesh points $t_0 < t_1 < t_2 < \cdots < t_n$. Without loss of generality, we impose a uniform step size h such that

$$h = t_{k-1} - t_k > 0 \tag{2}$$

is independent of k. Figure 2.1 shows how one partitions a domain, adding h to t_k to obtain the subsequent mesh point t_{k+1}. The choice of h is not often done

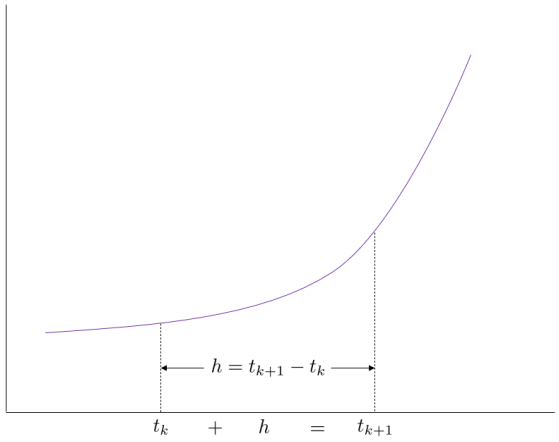


Figure 2.1: Step sizes and Mesh points

naively as seen here; it is often assumed to be relatively small.

The easiest approximation for $y(t)$ is through linearization, using the line tangent to $y(t)$ as an estimation for $y(t_{k+1})$. Recall that we can obtain a tangent line through a first-order Taylor Series. Centering about t_{k+1}, we have

$$y(t_{k+1}) \approx y_k + (t_{k+1} - t_k)\frac{dy}{dt} \tag{3}$$

$$y(t_{k+1}) \approx y_k + hf(y_k, t_k) \tag{4}$$

Here we used equations (1) and (2) to convert (3) into (4), thereby deriving the iterative scheme for Euler's Method.

As sketched in Figure 2.2, Euler's Method approximates the solution pretty well at the start, but gradually accrues error and, in turn, lowers the overall accuracy. To understand how the error is accumulated, we define the h^2 term of the second-order Taylor Series for $y(t_{k+1})$ to

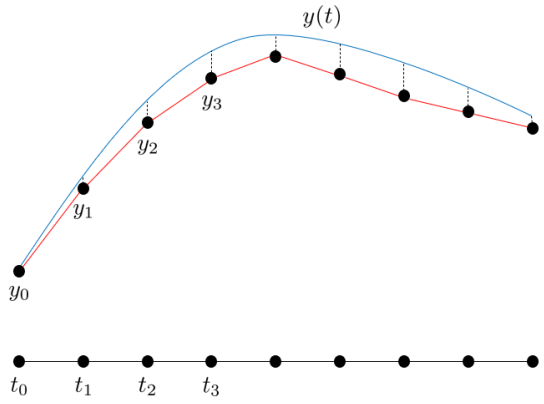


Figure 2.2: Euler's Method

be the local truncation error

$$y(t_{k+1}) \approx y_k + hf(y_k, t_k) + \frac{h^2}{2}f'(\xi(y_k, t_k)), \tag{5}$$

where ξ is the error with respect to y_k, t_k. Therefore, the local error scales quadratically. If we set h to be 10 times smaller, then the error shrinks by 100 times, making it important to set h small enough to make the error insignificant. However, set h to be too large, and Euler's Method diverges, over or undershooting the realized solution. By letting $N = \frac{T}{h}$ as the total number of mesh points, then we obtain the global truncation error

$$N\frac{h^2}{2}f'(\xi(y_k, t_k)) = \frac{hT}{2}f'(\xi(y_k, t_k)), \tag{6}$$

suggesting that Euler's method follows an $\mathcal{O}(h)$ order of convergence, implying that the error depends linearly on the step size h.

2.2 Runge-Kutta Method

What we will find is that Euler's model is an elementary incarnation of numerical schemes in ordinary differential equations. The Runge-Kutta iterative method, to be derived in this section, synthesizes Euler's and Midpoint Approximations to generate a robust model for numerical differentiation. In this section, we will introduce the RK-1, RK-2, and RK-4 methods, each corresponding to their order of convergence and number of "slope samples" required.

We proceed with some notation. We denote c_k, $k = 1, 2, 3, 4$, as the four necessary slope approximations. For instance, the RK-1 method, takes on the form

$$\begin{aligned} c_1 &= hf(y_k, t_k) \\ y_{k+1} &= y_k + c_1. \end{aligned} \tag{7}$$

Comparing to equation (4), RK-1 is actually Euler's Method rewritten with our notation. Here, we say that RK-1 uses one slope sample, c_1, to estimate y_{k+1}. By adding more slope samples and taking their weighted average, we can "leverage," or influence, the consistency of our approximation. Instead of taking one full time step, we can go halfway and also take the slope at $t_k + \frac{h}{2}$, serving as the midpoint between t_k and t_{k+1}. We use the previous slope, c_1, to estimate y at the midpoint, given by $y_k + \frac{1}{2}c_1$.

The new slope, c_2, is obtained by evaluating f at this new t and y. So, each successive slope sample is merely an updated version of the previous one. If we stop here, then we have the set of updates

$$\begin{aligned} c_1 &= hf(y_k, t_k) \\ c_2 &= hf\left(t_k + \frac{h}{2}, y_k + \frac{c_1}{2}\right) \\ y_{k+1} &= y_k + c_2, \end{aligned} \tag{8}$$

which is the Runge-Kutta 2 method as pictured by Figure 2.3.

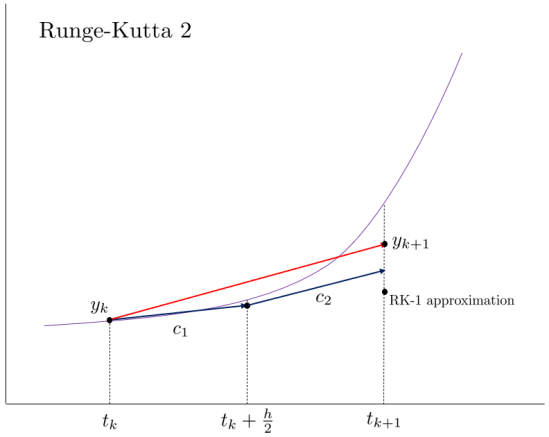


Figure 2.3: Runge-Kutta 2

The blue arrows indicate the two slope samples we constructed and the red arrow extends the slope c_2, from y_k, through an entire time step to guess y_{k+1}, obtaining a global $\mathcal{O}(h^2)$ order of convergence. Had we used RK-1, we would simply extend the slope from c_1 from y_k to y_{k+1}, producing a larger error.

Although there is already a noticeable improvement moving from RK-1 to RK-2, we can further strengthen the convergence by adding more slope samples. Moving forward from RK-2,

apply the following steps to reach RK-4:

1. Take c_2 and apply it to a half step, starting at (t_k, y_k).
2. Then, apply the same idea that we used to calculate c_2: Evaluate f at $t_k + \frac{h}{2}$ and $y_k + \frac{1}{2}c_2$ to acquire c_3.
3. Take c_3 and apply it to a full step, starting at (t_k, y_k).
4. c_4 is computed by evaluating f at t_{k+1}, y_{k+1}.
5. The finalized slope is the weighted average of our four slope samples. Recall that we used full steps for c_1 and c_4, but only half steps for c_2, c_3. To make them full slopes, multiply by 2 (i.e. $2c_2, 2c_3$).
6. This means that c_2 and c_3 make two contributions to the overall slope, but c_1, c_4 contribute just once. The weighted average is reached by summing the slope contributions and dividing by the total number of slope contributions. This completes one iteration of RK-4.

Mathematically, the steps taken to achieve RK-4 are summarized by the series of updates.

$$\begin{aligned} c_1 &= hf(t_k, y_k) \\ c_2 &= hf\left(t_k + \frac{h}{2}, y_k + \frac{c_1}{2}\right) \\ c_3 &= hf\left(t_k + \frac{h}{2}, y_k + \frac{c_2}{2}\right) \\ c_4 &= hf(t_k + h, y_k + c_3) \\ y_{k+1} &= y_k + \frac{1}{6}(c_1 + 2c_2 + 2c_3 + c_4). \end{aligned} \tag{9}$$

Figure 2.4 illustrates how successive

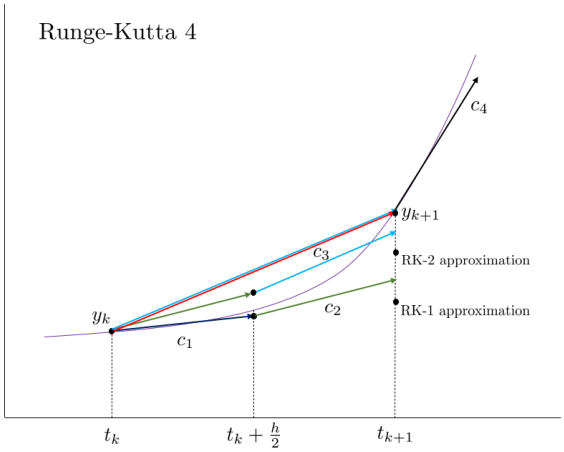


Figure 2.4: Runge-Kutta 4

slope samples are created and ultimately refined to achieve a precise and accurate estimation of y_{k+1}, which is where RK-1 and RK-2 fell short. This is due to the higher order of convergence, namely fourth-order $\mathcal{O}(h^4)$. Formulating a

fifth-order Taylor Series about t_{k+1} yields

$$y\left(t_{k+1}\right) = y_k + hf(y_k, t_k) + \frac{h^2}{2}f'(y_k, t_k) + \frac{h^3}{6}f''(y_k, t_k) + \frac{h^4}{24}f'''(y_k, t_k) + \frac{h^5}{120}f^{(4)}\left(\xi(y_k, t_k)\right). \tag{10}$$

Once again, $\frac{h^5}{120}f^{(4)}\left(\xi(y_k, t_k)\right)$ is the fifth-order local truncation error. Computing each derivative is an exhaustive application of the Chain Rule, requiring a Taylor Series expansion of each term. In the end, the powers of h will end up agreeing with this Taylor Series expansion, obtaining a global fourth-order of convergence with respect to h once multiplying over total number of mesh points $N = \frac{T}{h}$. Additionally, we have that for each mesh point, 5 calculations$-$4 slope samples and 1 update$-$ are required to successfully complete one iteration of RK-4. Therefore, it is imperative that an appropriate choice for h is mindfully considered to strike a balance with accuracy and computational cost. The trade-off can be examined by taking the limit

$$\lim_{h\to 0} N = \lim_{h\to 0} \frac{T}{h}, \tag{11}$$

implying that our aggregate error gets infinitesimally small at the expense of computing infinitely many iterations.

So, each iteration of the RK-4 method maintains the same computational effort as four RK-1 iterations or two RK-2 iterations. Additionally, a more revealing comparison is given by h: If we approximate a solution using RK-4, RK-1 would require $\frac{h}{4}$ time steps to reach comparable accuracy. However, this in turn, requires four times as many computations. Therefore, the computational expense is practically the same, but RK-4 will achieve the best results.

3 Results and Analysis

3.1 Carbon-15 Half-Life

We proceed by applying the Runge-Kutta 1, 2, 4 iterative methods to model the half-life of a Carbon-15 isotope, whose differential equation is given by

$$\frac{dy}{dt} = -\frac{\ln(2)}{\tau}y \tag{12}$$

where $\tau = 2.45$ is the specified half-life in seconds. The analytic solution to the differential equation (12) is

$$y(t) = \left(\frac{1}{2}\right)^{\frac{t}{\tau}}. \tag{13}$$

On the following page, Figure 3.1 illustrates the exact solution y, alongside the RK-1, RK-2, and RK-4 approximations for $t \in C[0, 15]$ seconds. We selected four different step sizes$-h = 0.01, 0.1, 1, 5$, corresponding to 1500, 150, 15, and 3 mesh points$-$to test the accuracy and overall convergence of each iterative method.

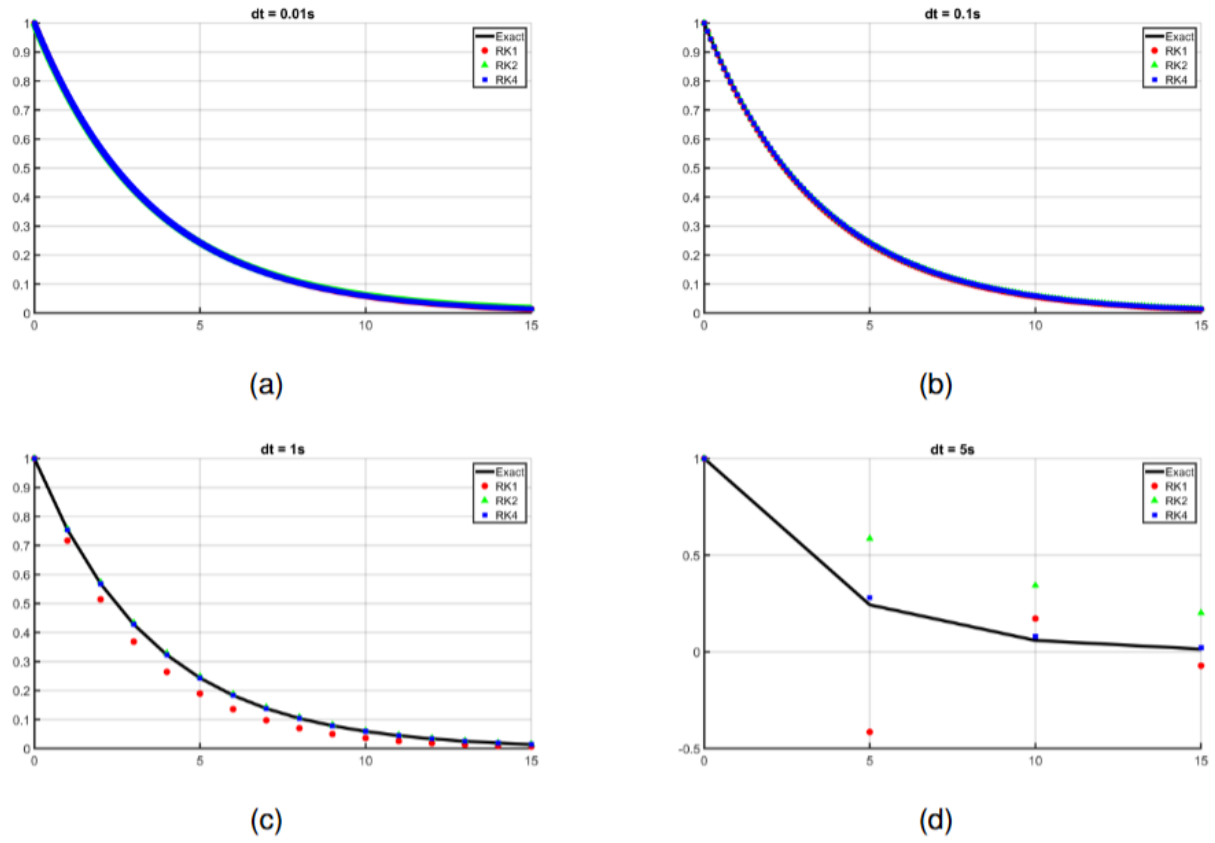


Figure 3.1: Runge-Kutta models for $h =$ (a) 0.01, (b) = 0.1, (c) = 1, and (d) = 5 seconds

We observe no issues at $h = 0.01, 0.1$, as anticipated for a small choice of h. However, starting at $h = 1$, RK-1 undervalues the true decay of the isotope (though it still captures the overarching pattern), whereas RK-2 and RK-4 still converge to the exact solution. At $h = 5$, RK-2 follows the correct trend, but has a modest residual compared to $y(t)$. RK-1 appears to have completely diverged from y: When connecting each point, the resulting curve implies that the isotope decays, then grows and decays again, misaligned with a true decaying isotope. RK-4, on the other hand, promises an almost perfect solution with a substantially lower error than the other two models.

h	RK-1	RK-2	RK-4
0.01	3×10^{-4}	3×10^{-7}	10^{-13}
0.1	3×10^{-3}	3×10^{-5}	10^{-9}
1.00	3×10^{-2}	3×10^{-3}	10^{-5}
5.00	2×10^{-1}	2×10^{-1}	2×10^{-2}

Table 3.2: Mean Errors for the Carbon-15 Half-Life model

The poor performance of RK-1 and RK-2 is further accentuated in Table 3.2, which lists the mean squared error (MSE) for each choice of h. It is clear that RK-4 has phenomenally good accuracy, outperforming its rivals by having the smallest error for all h.

Moreover, it is worth noting that the change in error over each time step agrees with the local truncation error described in Section 2. Recall that we proved the respective orders of convergence for RK-1, RK-2, and RK-4: $\mathcal{O}(h), \mathcal{O}(h^2)$, and $\mathcal{O}(h^4)$. As seen by the table, each decrease in h by a factor of $\frac{1}{10}$ improves the accuracy in the completed solution by 4 additional decimal digits, in accordance with its status as a fourth-order method. Likewise, with the same dilation in h, RK-2 and RK-1 see a 100 times and 10 times increase in accuracy, respectively.

Lastly, take $h = 5$ seconds. In order for RK-1 to reach the same standard of accuracy as RK-4, h needs to be partitioned into 4 additional time steps, or use $h = 1.25$. This matches the intuition of computational effort in Section 2. While we did not use $h = 1.25$, the accuracy is roughly the same at $h = 1$ for RK-1 (3×10^{-2}) as it is when $h = 5$ for RK-4 (2×10^{-2}).

3.2 SIR Model

The differential equation model for decay, as shown in the Carbon-15 isotope, yields an analytic, or explicit solution. However, we should also examine how our method fares against a model where an analytic solution cannot be ascertained through existing techniques.

The SIR (Susceptible-Infected-Recovered) model for pandemic prediction is one such model. Plainly, the SIR model depicts the dynamics of a pandemic (without vaccines) with the change of susceptible individuals $S(t)$, infectious individuals $I(t)$, and removed individuals (including recovered, immune, and deceased) $R(t)$. The interaction to be observed uses the set of differential equations

$$\frac{dS}{dt} = -\beta \frac{IS}{N_{pop}} \tag{14}$$

$$\frac{dI}{dt} = \beta \frac{IS}{N_{pop}} - \gamma I \tag{15}$$

$$\frac{dR}{dt} = \gamma I, \tag{16}$$

where parameter β denotes the average number of contacts per person per time multiplied by the probability of disease transmission in a contact between a susceptible and infectious subject (quantifies the “transmissibility” of the disease) and parameter γ is the recovery rate (including death). N_{pop} is the total population. Generally, SIR models are applied in predicting outbreaks and quickly identifying intervention methods to “flatten the curve.”

Our goal is to use each Runge-Kutta method to solve the differential equation with initial conditions $S(0) = 997$, $I(0) = 3, R(0) = 0$, setting $\beta = 0.2$, $\gamma = 0.04$ and $N_{pop} = 1000$ as the remaining parameters. We choose $h = 1$ and $h = 10$ days as our two step sizes, corresponding to 100 and 10 total mesh points.

On the following page, Figure 3.3

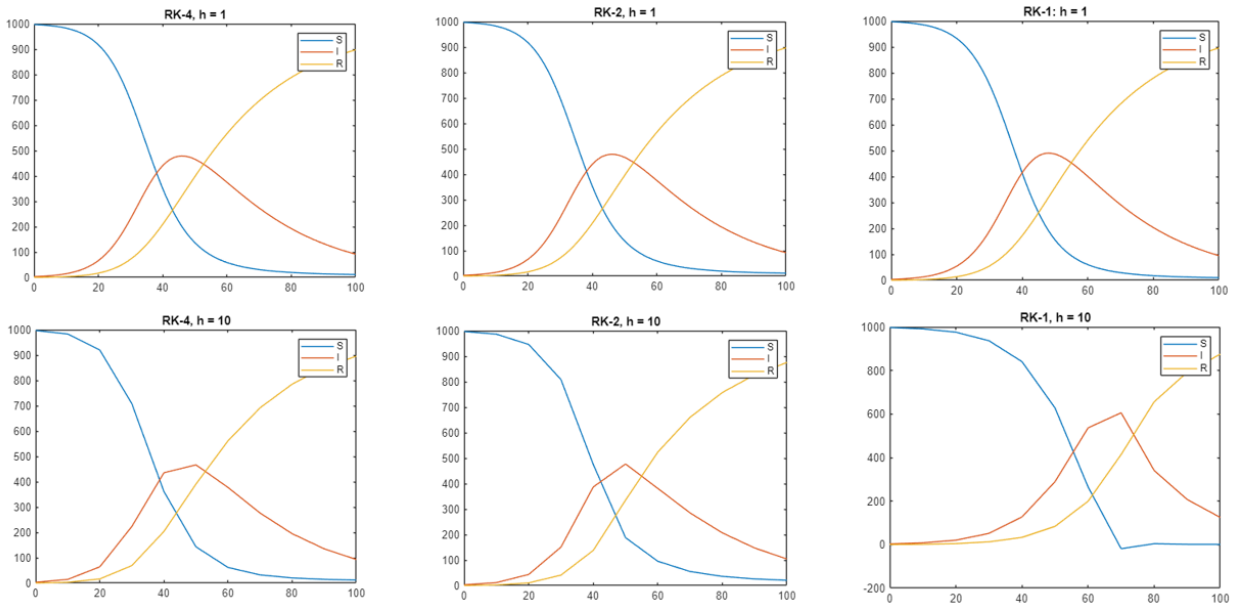


Figure 3.3: Runge-Kutta Approximations for SIR Model using $h = 1, 10$ days

displays the approximated solutions to the unique SIR model with said time steps. We hope to observe a consistent decrease in susceptible individuals and increase in recovered individuals with the choices for β and γ. In addition, there should be a peak for infected individuals and eventually a decline.

Because there is no analytic solution for the SIR model, we cannot compute accuracy empirically. Instead, we rely on how much our solutions change as we perturb h. For $h = 1$, the three RK estimates are in agreement with the expected trend, and we will use this as a reference while controlling for an increased step size. When we increase h tenfold, the models retain the desired trend. However, the behavior of RK-1 is somewhat altered. The steep descent in susceptible individuals occurs much later (say, 20 days). In turn, this delays the steep ascent of infected individuals. Had an international health organization followed the RK-1 model, they would be inclined to believe they have more time to readily prepare for an outbreak, although the $h = 1$ approximations suggest that time is rather limited. Additionally, the peak of infected individuals is skewed to the right and the solution for susceptible individuals stops receiving updates, halted at 0 after 80 days. In contrast, the RK-2 and RK-4 models mostly match the peak for $I(t)$ and $S(t)$ steadily converges to 0.

With these considerations, RK-4 and even RK-2 have a phenomenal grasp on the underlying pattern that we were looking for and can thus be applied to non-autonomous differential equations with large step sizes.

4 Conclusion

The Runge-Kutta iterative scheme establishes a novel approach for numerical differentiation that has proved to offer superior accuracy and versatility compared to its predecessors. Previous techniques are limited to singular updates to slopes which inhibits convergence if not applied carefully. By refining Euler's method with midpoint methods, we founded a rigorous derivation of the Runge Kutta scheme with the theoretical analysis to validate its strong fourth-order convergence.

Our primary objectives were to develop a robust differentiation technique that can deliver promising accuracy while maintaining a reasonable computational cost. The results confirm that our method achieves these goals. For models whose differential equation has an analytical solution such as the half-life of a Carbon isotope, Runge Kutta 4 consistently provided remarkable precision by having the smallest mean error for any choice of step size, seeing as much as 10^8 times less error when h is small. Since we have no hope of solving the majority of differential equations explicitly, we also demonstrated that Runge Kutta, through its convergence to a unique solution in SIR modeling, is an adaptive numerical algorithm in these vast situations.

By satisfying these objectives, we believe that Runge Kutta is broadly applicable to a myriad of scientific and engineering frameworks. Besides the one discussed in this article, we envision the Runge Kutta schemes will quickly integrate itself into branches of physics, such as aerospace engineering, fluid dynamics, mechanical systems, and simulations.

While the Runge-Kutta methodology is already capable of numerically solving complicated models, there remain opportunities for further refinement. In the future, we could explore adaptations of Runge-Kutta in advanced mathematical fields such as partial differential equations, stochastic differential equations, and unstable differential equations.

Ultimately, our findings illustrate an immense breakthrough in numerical methods and analysis, and we anticipate that the Runge Kutta algorithm will naturally blend into new research methods and practical implementations across scientific and engineering disciplines.

References

1. Arnold, Mark. *Order of Convergence.*

2. Dawkins, Paul. "Differential Equations - Euler's Method." Tutorial.math.lamar.edu, 3 Dec. 2018, tutorial.math.lamar.edu/Classes/DE/EulersMethod.aspx.

3. Trench, William. "Euler's Method." LibreTexts, math.libretexts.org/Courses/MonroeCommunityCollege

/MTH225DifferentialEquations/033A NumericalMethods/3.013AEuler's Method.

4. Zeltkevic, Michael. "Forward and Backward Euler Methods." Web.mit.edu, 15 Apr. 1998, web.mit.edu/10.001/Web/Course Notes/DifferentialEquationsNotes/ node3.html.

Appendices

Figure 3.1: Runge-Kutta plots for Carbon-15 Half-Life

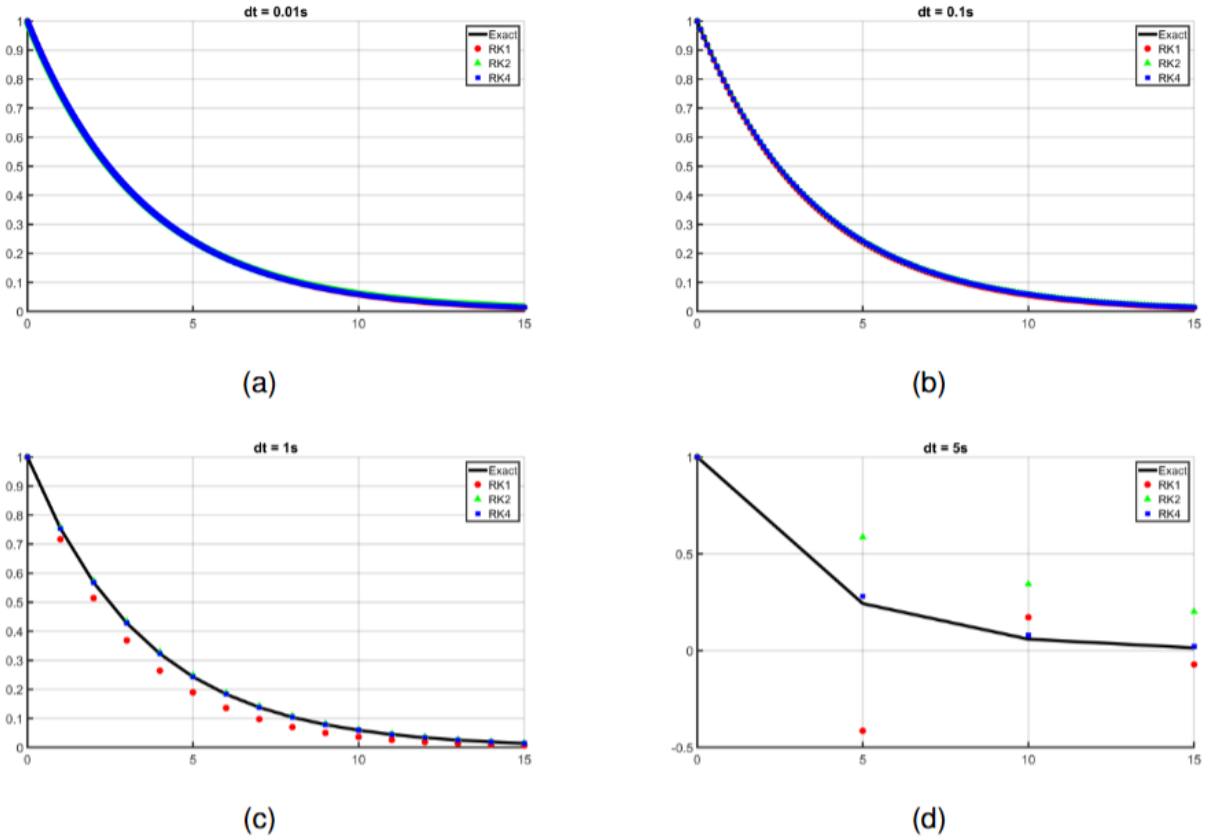
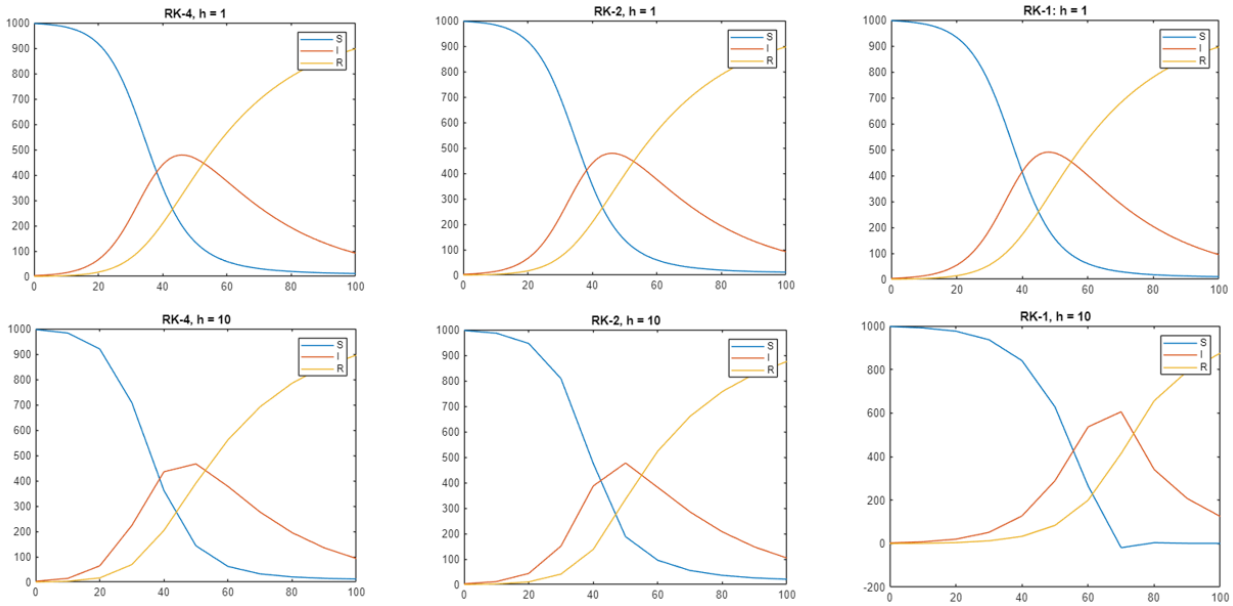


Table 3.2: Precise Mean Error for the Carbon-15 Half-Life model with respect to the analytical solution

h	RK-1	RK-2	RK-4
0.01	3.08×10^{-4}	2.91×10^{-7}	1.16×10^{-13}
0.1	3.09×10^{-3}	2.95×10^{-5}	1.18×10^{-9}
1.00	3.11×10^{-2}	3.42×10^{-3}	1.38×10^{-5}
5.00	2.14×10^{-1}	2.03×10^{-1}	1.64×10^{-2}

Figure 3.3: Runge-Kutta plots for SIR Model



Extra Page

1. The target audience is the editor and publisher of this article.
2. Introduce and publish a new methodology for numerical differentiation, namely Runge-Kutta, and get the theory accepted into the mathematical/STEM field.
3. The main research I did in preparation was reviewing the Explicit Euler method, forward/backward differencing, and developing a better intuition of order of convergence. The sources I used are published from academic institutions and appear to be reliable.
4. Using the differential equation representing the Half-Life of a Carbon-15 isotope, the Runge-Kutta 4 method numerically solves the differential equation with the lowest residual sum of squared error. It is also present within the graphs that the iterations of the Runge-Kutta 4 curve, when connected, best captures/fits the decay of the isotope (a lot of overlapping or substantially low deviance). Additionally, when applied to the SIR model, the RK methods fit the pattern for $h = 1$, but RK-1 falls apart for $h = 10$.
5. The results, as mentioned in the previous part, applies the differential equation of a Carbon-15 isotope's half-life. There are 4 plots, each with increasing time steps (dt = 0.01, 0.1, 1, 5 seconds), generating the numerical solution to the half-life model by the Runge-Kutta 1, 2, and 4 iterative methods. A table summarizes, from the same models shown in the graph, the **mean** error between the predicted and observed output. For the SIR model, I used $h = 1$ as a benchmark for comparison when analyzing the $h = 10$ approximations.
6. After introducing and deriving the Runge-Kutta method for numerical differentiation, we conclude that this iterative approach (namely RK-4) significantly lowers the aggregate error compared to conventional finite-difference methods (i.e. Explicit Euler, forward/backward Euler). As expected with a high order of convergence $(\mathcal{O}(h^4))$, we showed that Runge-Kutta 4 exhibited great stability and accuracy when applied to differential equations.