Assignment Report

MA3105

Numerical Analysis

Analysis of Runge Kutta Methods

Author:
Nimish Sharma - 21MS184
Abhay Saxena - 21MS086

Monday 20th November, 2023

1 The Math

1.1 Runge Kutta 2nd Order

1.1.1 Introduction

The Runge-Kutta Second Order (RK2) method is a fundamental numerical technique for solving ordinary differential equations (ODEs). As a member of the Runge-Kutta family, RK2 strikes a balance between simplicity and accuracy, making it a versatile tool in various scientific and engineering applications.
The RK2 method, also known as the midpoint method, is a second-order method that improves upon the accuracy of simpler methods like Euler's method. The central idea is to use a weighted average of function evaluations at two points within a given time step to estimate the next value of the solution. The RK2 update formula involves two function evaluations and a weighted sum, providing a more accurate approximation compared to first-order methods.

1.1.2 Derivation

The Second Order Runge-Kutta algorithm described above was developed in a purely ad-hoc way. It seemed reasonable that using an estimate for the derivative at the midpoint of the interval between x_0 and x_0+h (i.e., at $x_0+\frac{1}{2}$h) would result in a better approximation for the function at x_0+h, than would using the derivative at x_0[1]

To start, we are expressing our differential equation as

$$\frac{dy(x)}{dx} = y'(x) = f(y(x), x) \tag{1}$$

Next, We define our approximations to the derivative as:

$$\begin{aligned} k_1 &= f(y^*(x_0), x_0) \\ k_2 &= f(y^*(x_0) + \beta h k_1, x_0 + \alpha h) \end{aligned} \tag{2}$$

In all cases α and β will represent fractional values between 0 and 1. These equation state that k_1 is the approximation to the derivative based on the estimated value of y(x) at x=x_0 (i.e., $y^*(x)$) and the position at x_0. The value of k_2 is based upon the estimated value, $y^*(x_0)$, plus some fraction of the step size, βh, times the slope k_1, and the position at $x_0 + \alpha h$

To update our solution with the next estimate of y(x) at $x_0 + h$, we use the equation

$$y^*(x_0 + h) = y^*(x_0) + h(ak_1 + bk_2) \qquad \text{update equation} \tag{3}$$

This equation states that we get the value of $y^*(x_0 + h)$ from the value of $y^*(x_0)$ plus the time step, h, multiplied by a slope that is a weighted sum of $k_1$and k_2). Our goal now is to determine, a, b, α and β that result in low error. Starting with the update equation (above) we can substitue the previously given expressions for k_1 and k_2) which yields

$$y^*(x_0 + h) = y^*(x_0) + h\left(af\left(y^*(x_0), x_0\right) + bf\left(y^*(x_0) + \beta h k_1,\, x_0 + \alpha h\right)\right) \tag{4}$$

We can use a two-dimensional Taylor Series (where the increment in the first dimension is $\beta h k_1$, and the increment in the second dimension is αh) to expand the rightmost term.

$$\begin{aligned} f\left(y^*(x_0) + \beta h k_1,\, x_0 + \alpha h\right) &= f + f_y \beta h k_1 + f_x \alpha h + \cdots \\ &= f + f_y f \beta h + f_x \alpha h + \cdots \end{aligned} \tag{5}$$

In the last line we used the fact the $k_1 = f$. Substituting this into the previous expression for $y^*(x_0 + h)$ and rearranging we get:

$$\begin{aligned} y^*(x_0 + h) &= y^*(x_0) + h\left(af + b\left(f + f_y f \beta h + f_x \alpha h + \cdots\right)\right) \\ &= y^*(x_0) + haf + hbf + h^2 b\beta f_y f + h^2 f_x b\alpha + \cdots \\ &= y^*(x_0) + h(a+b)f + h^2 f_x b\alpha + h^2 b\beta f_y f + \cdots \end{aligned} \tag{6}$$

To finish we compare this approximation with the expression for a Taylor Expansion of the exact solution (going from the first line to the second we used the chain rule for partical derivatives). In this expression the ellipsis represents terms that are third order or higher.

$$\begin{aligned} y^*(x_0+h) &= y^*(x_0) + h\left(af + b\left(f + f_y f\beta h + f_x \alpha h + \cdots\right)\right) \\ &= y^*(x_0) + haf + hbf + h^2 b\beta f_y f + h^2 f_x b\alpha + \cdots \\ &= y^*(x_0) + h(a+b)f + h^2 f_x b\alpha + h^2 b\beta f_y f \end{aligned} \tag{7}$$

Comparing this expression with our final expression for the approximation,

$$y^*(x_0+h) = y^*(x_0) + hf(a+b) + h^2 f_x b\alpha + h^2 b\beta f_y f + \cdots \tag{8}$$

We see that they agree up to the error terms (third order and higher) represented by the ellipsis if we define the constants, a, b, α and β such that

$$\begin{aligned} a+b=1, a &= \frac{1}{2} = b \\ b\alpha &= \frac{1}{2}, \alpha = 1 \\ b\beta &= \frac{1}{2}, \beta = 1 \end{aligned} \tag{9}$$

1.1.3 Error Analysis

1. Local Error
   We define the local Truncation error at $n+1^{th}$ term as:

   $$e_{n+1} = |y(x_{n+1}) - \tilde{y}_{n+1}| \tag{10}$$

   where $y(x_{n+1})$ is the actual $n+1^{th}$ term derived from taylor expansion and $\tilde{y}_{n+1}$ is the approximate we get from rk2 data assuming that y_n is exact.

   $$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2} f''(x) \pm +O(h^3) \quad \text{Taylor expansion} \tag{11}$$

   We get,

   $$y(x_0+h) = y(x_0) + hf + \frac{h^2}{2!}(f_t + f_y f) + \frac{h^3}{3!} y^{'''}(\xi) \tag{12}$$

   where, ξ is some number between x_n and x_{n+1}
   Using Equation (8) as our $\tilde{y}_{n+1}$ and using the Taylor expansion as our $y(x_{n+1})$, we get,

   $$e_{n+1} = |\frac{h^3}{6} y^{'''}(\xi)| = \frac{Mh^3}{6} \tag{13}$$

   Hence, we conclude that the Local Truncation Error at $n+1^{th}$ term will be of order 3.

2. Global Error
   Let $y(t)$ be the exact solution, and w_n be the numerical approximation at the nth step using the RK2 method. The global error at each step is given by:

   $$E_n = y(t_n) - w_n$$

   Subtracting the RK2 update formula from the exact solution expansion, we get:

   $$E_{n+1} = y(t_{n+1}) - w_{n+1} = \frac{h^2}{2} \cdot y''(t_n) + O(h^3)$$

The global error E at any time t is the sum of the local errors:

$$E = \sum_{n=0}^{N} E_n$$

Substituting the expression for E_{n+1} into the sum, we get:

$$E = \sum_{n=0}^{N} \frac{h^2}{2} \cdot y''(t_n) + O(h^3)$$

Simplifying, we obtain the formula for the global error of the RK2 method:

$$E = \frac{h^2}{2} \cdot \sum_{n=0}^{N} y''(t_n) + O(h^3)$$

This formula indicates that the global error is proportional to h^2 and depends on the second derivative of the exact solution evaluated at each time step.

1.2 Runge Kutta 4th Order

1.2.1 Introduction

The Runge-Kutta (RK4) method is a widely used numerical technique for solving ordinary differential equations (ODEs). It belongs to the family of Runge-Kutta methods, which are iterative procedures that provide approximate solutions to ODEs. RK4 is particularly favored for its simplicity and high accuracy, making it a popular choice in various scientific and engineering applications.
The RK4 method is a fourth-order method, meaning it achieves a high level of accuracy compared to simpler methods like Euler or RK2. The basic idea is to use weighted averages of function evaluations at different points within a given time step to estimate the next value of the solution. The RK4 update formula involves four function evaluations and a weighted sum, providing a good compromise between accuracy and computational efficiency.

1.2.2 Derivation

Consider the Taylor expansion of the exact solution:

$$y(t_{n+1}) = y(t_n) + h \cdot y'(t_n) + \frac{h^2}{2} \cdot y''(t_n) + \frac{h^3}{6} \cdot y'''(t_n) + \frac{h^4}{24} \cdot y''''(t_n) + \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \tag{14}$$

Now, compare this expansion with the RK4 update rule, in a similar way we did for RK2 method:

$$\begin{aligned}
k_1 &= h \cdot f(t_n, y_n) \\
k_2 &= h \cdot f(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\
k_3 &= h \cdot f(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\
k_4 &= h \cdot f(t_n + h, y_n + k_3) \\
y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{15}$$

Now, write down the coefficients by comparing the terms in the Taylor expansion:

$$\alpha = 0, \quad \beta = \frac{1}{2}, \gamma = \frac{1}{2}, \quad \delta = 1 \tag{16}$$

For the weights b_i, consider the coefficients of h^2 in the expansion:

$$a = \frac{1}{6}, \quad b = \frac{1}{3}, \quad c = \frac{1}{3}, \quad d = \frac{1}{6} \tag{17}$$

These coefficients correspond to the weights in the RK4 method.
Thus, by comparing the terms in the Taylor expansion and the RK4 update rule, we can derive the coefficients a_i and weights b_i for the RK4 method.

1.2.3 Error Analysis

1. Local Error
   The local error E_{n+1} for the RK4 method is defined as the difference between the exact solution $y(t_{n+1})$ and the numerical approximation w_{n+1} at the same point in time:

   $$E_{n+1} = y(t_{n+1}) - w_{n+1} \tag{18}$$

   Now, consider the Taylor expansion given in Equation (14) of the exact solution around t_n up to the fifth order.

   Subtracting the RK4 update formula from the exact solution expansion, we get the local error for RK4:

   $$\begin{aligned}
   E_{n+1} &= y(t_{n+1}) - w_{n+1} \\
   &= \left(y(t_n) + h \cdot y'(t_n) + \frac{h^2}{2} \cdot y''(t_n) + \frac{h^3}{6} \cdot y'''(t_n) + \frac{h^4}{24} \cdot y''''(t_n) + \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \right) \\
   &- \left(w_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \right)
   \end{aligned} \tag{19}$$

Now, simplify the expression:

$$
\begin{aligned}
E_{n+1} = {} & y(t_n) + h \cdot y'(t_n) + \frac{h^2}{2} \cdot y''(t_n) + \frac{h^3}{6} \cdot y'''(t_n) + \frac{h^4}{24} \cdot y''''(t_n) + \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \\
& - w_n - \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{20}
$$

Now, let's substitute the expressions for k_1, k_2, k_3, k_4 from the RK4 method:

$$
\begin{aligned}
E_{n+1} = {} & y(t_n) + h \cdot y'(t_n) + \frac{h^2}{2} \cdot y''(t_n) + \frac{h^3}{6} \cdot y'''(t_n) + \frac{h^4}{24} \cdot y''''(t_n) + \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \\
& - w_n - \frac{1}{6}\left(h \cdot f(t_n, w_n) + 2h \cdot f\left(t_n + \frac{h}{2}, w_n + \frac{k_1}{2}\right) + 2h \cdot f\left(t_n + \frac{h}{2}, w_n + \frac{k_2}{2}\right) + h \cdot f(t_n + h, w_n + k_3)\right)
\end{aligned} \tag{21}
$$

Simplify further:

$$
\begin{aligned}
E_{n+1} = {} & \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \\
& - \frac{h}{6}\left(f(t_n, w_n) + 2f\left(t_n + \frac{h}{2}, w_n + \frac{k_1}{2}\right) + 2f\left(t_n + \frac{h}{2}, w_n + \frac{k_2}{2}\right) + f(t_n + h, w_n + k_3)\right)
\end{aligned} \tag{22}
$$

Finally, this expression represents the simplified local error for the RK4 method. Now, let's examine the order of this error term.

The leading term is $\frac{h^5}{120}$, and the error term is proportional to h^5, indicating that the RK4 method has local error of order h^5.

2. Global Error

The global error E is the sum of the local errors over all steps:

$$
E = \sum_{n=0}^{N} E_n \tag{23}
$$

Substituting the expression for the local error E_{n+1} into the sum, we get:

$$
E = \sum_{n=0}^{N} \frac{h^5}{120} \cdot y'''''(t_n) + O(h^6) \tag{24}
$$

Now, since we are summing over N steps, the dominant term in the sum will be $\frac{h^5}{120} \cdot N \cdot y'''''(t_n)$. However, considering N steps, the error per step decreases, and the total error becomes proportional to h^4:

$$
E \propto h^4 \tag{25}
$$

Therefore, the global error for the RK4 method is of order h^4.

2 Numerical Results

2.1 Problem Statement

Consider the following differential equations and their respective initial condition:

- Problem A
  1. $\frac{dy}{dx} = y - x$
  2. y(0) $= \frac{2}{3}$
- Problem B
  1. $\frac{dy}{dx} = y - x^2$
  2. y(0) $= \frac{2}{3}$

2.2 Data Table

The following data table compares the actual solution with RK2 and RK4 solutions for different step sizes:

2.2.1 Problem A

n	x	y_{actual}	rk2	$\text{error}_{\text{rk2}}$	rk4	$\text{error}_{\text{rk4}}$
1	0.000000	0.666667	0.666667	0.000000	0.666667	0.000000
2	0.500000	0.950426	0.958333	0.007907	0.950521	0.000095
3	1.000000	1.093906	1.119792	0.025886	1.094218	0.000312
4	1.500000	1.006104	1.069661	0.063558	1.006875	0.000771
5	2.000000	0.536981	0.675700	0.138719	0.538677	0.001695
6	2.500000	-0.560831	-0.276988	0.283844	-0.557338	0.003493
7	3.000000	-2.695179	-2.137605	0.557574	-2.688268	0.006911
8	3.500000	-6.538484	-5.473608	1.064876	-6.525192	0.013292
9	4.000000	-13.199383	-11.207113	1.992270	-13.174339	0.025044
10	4.500000	-24.505710	-20.836559	3.669151	-24.459262	0.046448

Table 1: Comparison of RK2 and RK4 with actual solution

2.2.2 Problem B

n	x	y_{actual}	rk2	$\text{error}_{\text{rk2}}$	rk4	$\text{error}_{\text{rk4}}$
1	0.000000	0.666667	0.666667	0.000000	0.666667	0.000000
2	0.500000	1.051705	1.020833	0.030872	1.051432	0.000273
3	1.000000	1.375624	1.315104	0.060520	1.375148	0.000477
4	1.500000	1.274415	1.199544	0.074870	1.274006	0.000408
5	2.000000	0.147925	0.105509	0.042416	0.148297	0.000371
6	2.500000	-2.993325	-2.891047	0.102278	-2.990568	0.002757
7	3.000000	-9.780716	-9.291702	0.489014	-9.772213	0.008503
8	3.500000	-22.903936	-21.536515	1.367421	-22.882971	0.020965
9	4.000000	-46.797533	-43.590587	3.206946	-46.751095	0.046438
10	4.500000	-88.772842	-81.897204	6.875638	-88.676284	0.096558

Table 2: Comparison of RK2 and RK4 with actual solution

3 Plots

3.1 Direction Fields

Graphical representation of the solutions to our first-order differential equation as Slope fields.It visually depicts the behavior of solutions by showing the slope of the solution curve i.e, the $f(x, y) = \frac{dy}{dx}$ at various points in the plane. The direction field is particularly useful for understanding the qualitative behavior of solutions without explicitly solving the differential equation.

3.1.1 Problem A

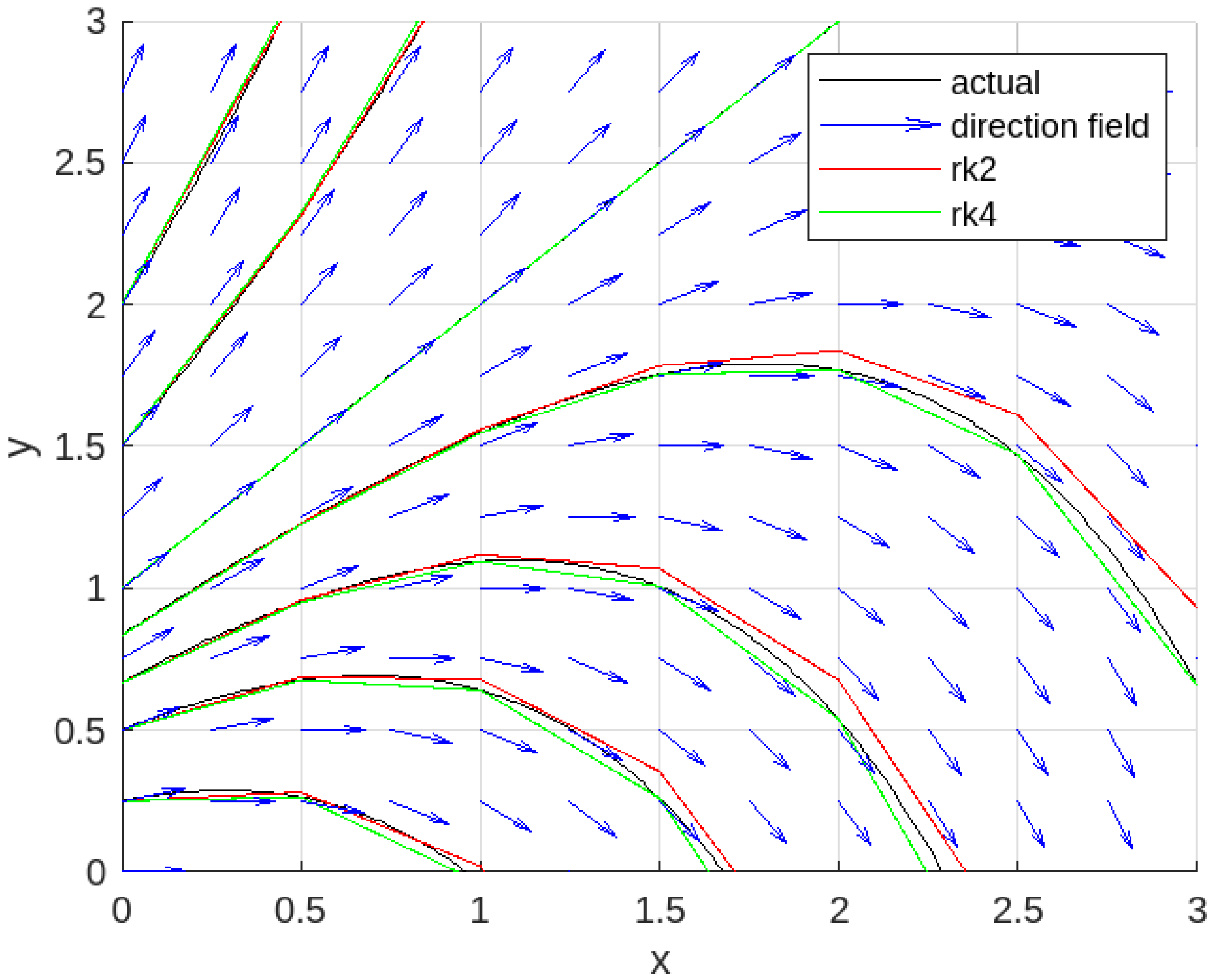

Figure 1: Different initial conditions

3.1.2 Problem B

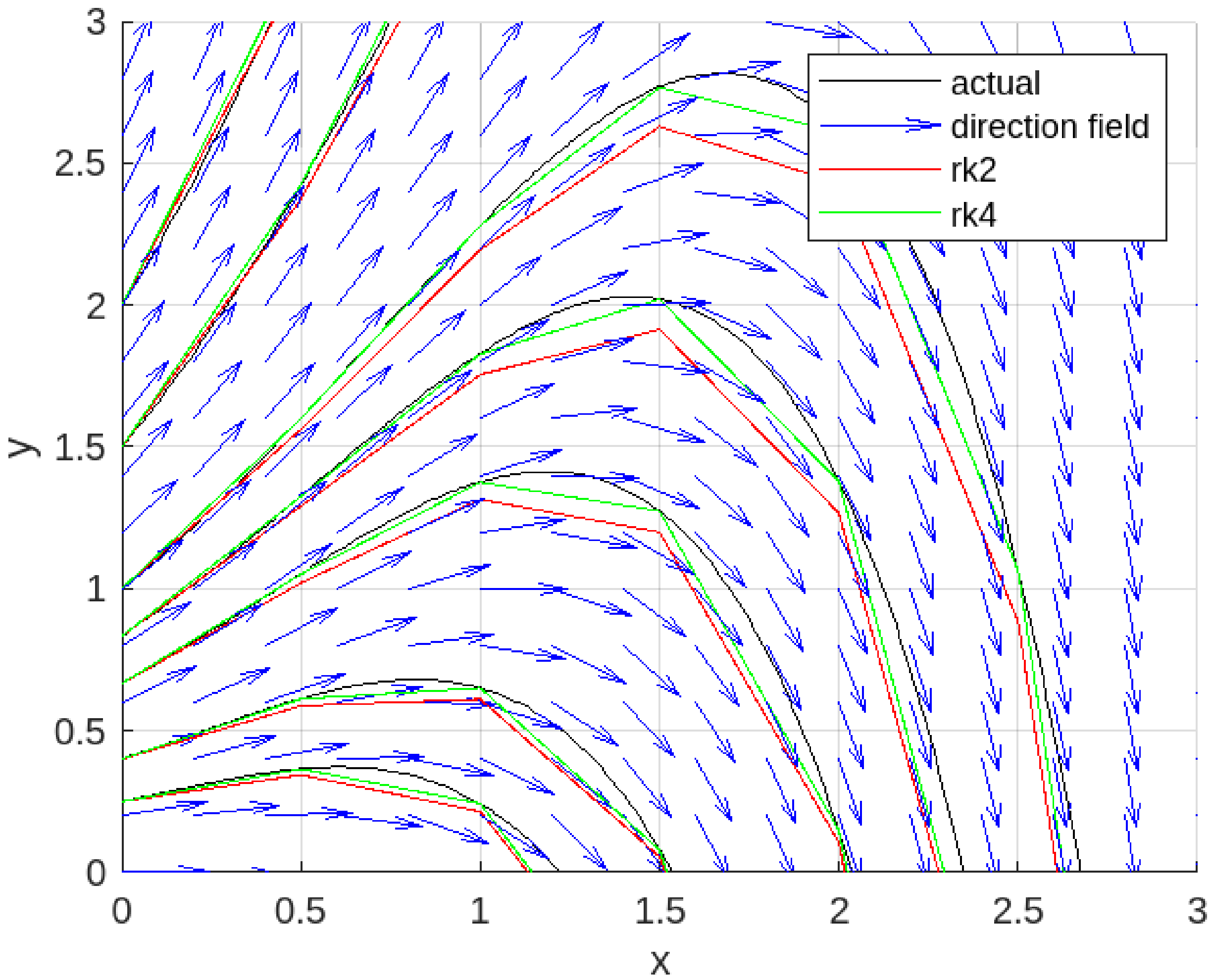

Figure 2: Different initial conditions

3.2 RK4 vs RK2

RK4 is more accurate than RK2. RK4 is a fourth-order method, meaning it achieves an error on the order of h^5 (where h is the step size), while RK2 is a second-order method with an error on the order of h^3. This implies that, for a given step size, RK4 generally provides more accurate results.

3.2.1 problem A

We see how for h = 1, RK4 approximates function better that rk2 We directly compare both errors of RK4 and RK2 to conclude the same, as RK4 approximates better, significantly. from power law relationship

$$\varepsilon = kh^p \tag{26}$$

where ε is global error and h is the step size.
hence, we say: $log(\varepsilon) = log(k) + plog(h)$
i.e a straight line plot of slop same as order of error.

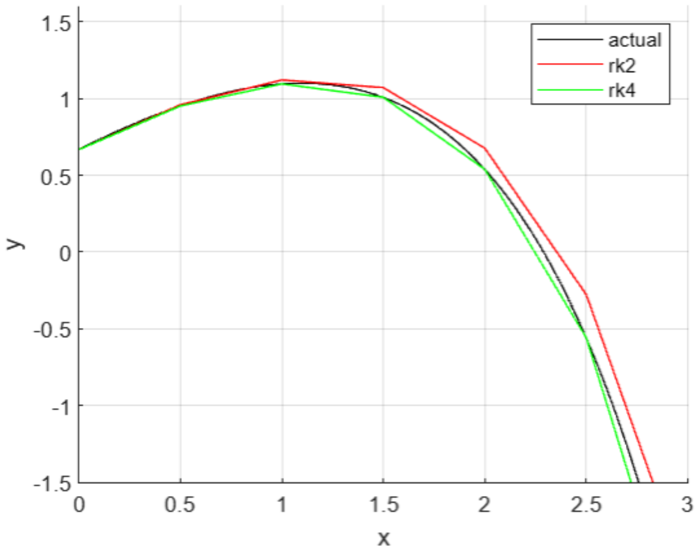

Figure 3: direct plot comparison

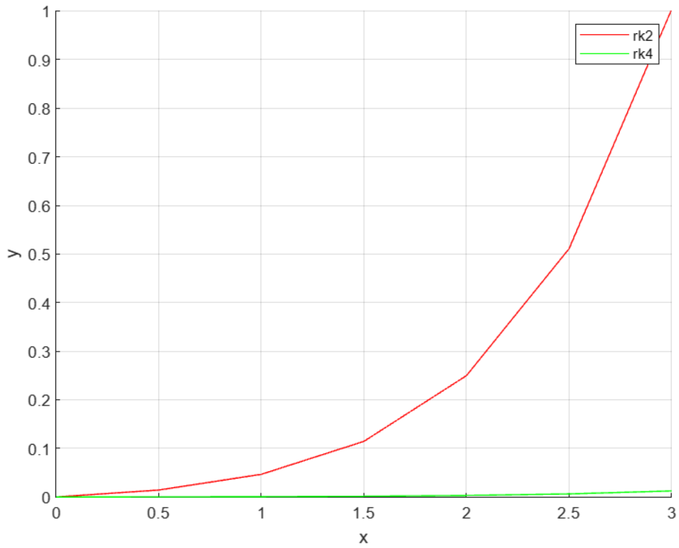

Figure 4: error comparison at different x for h = 0.5

3.2.2 problem B

again from from power law relationship, we find log(error) vs log(h) is a straight line.

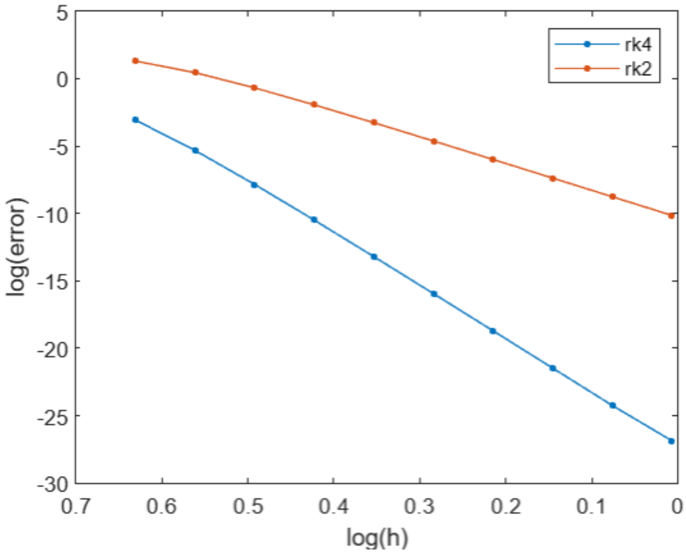

Figure 5: log of error vs log h at x = 5 (slope 4 for rk4, 2 for rk2)

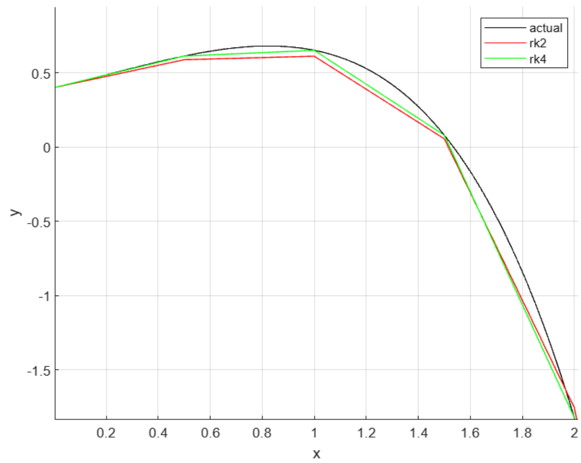

Figure 6: direct plot comparison

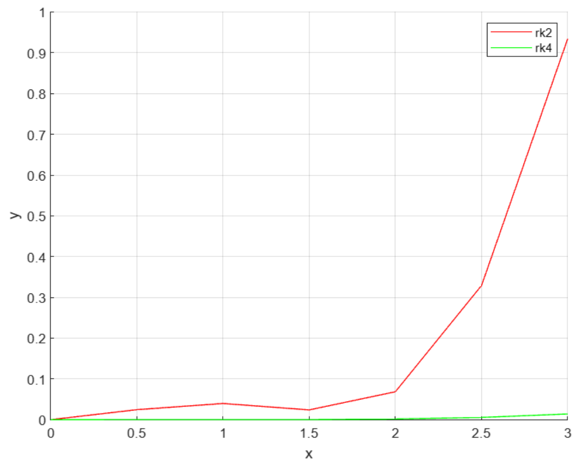

Figure 7: error comparison at different x for h = 0.5

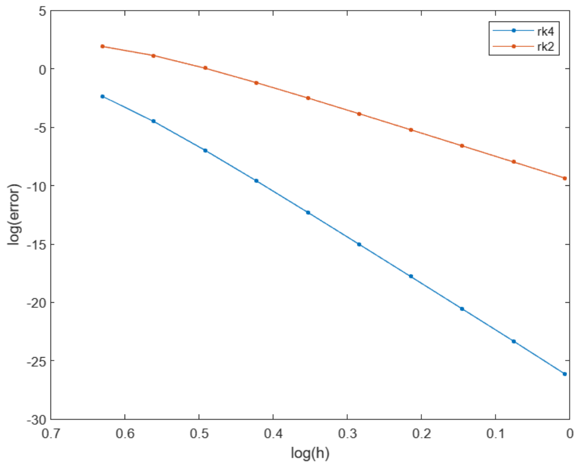

Figure 8: log of error vs log h at x = 5 (slope 4 for rk4, 2 for rk2)

3.3 RK2 vs h

Like many numerical methods, RK2 has a region of stability and convergence for certain step sizes. If the step size is too large, the method might become numerically unstable, and the solution may not accurately reflect the behavior of the differential equation.

RK2 is a second-order method, which means that the local truncation error is on the order of h^3. This implies that, in principle, decreasing the step size by a factor of 10 should reduce the error by a factor of 10^3, assuming the problem is well-behaved and the method is converging at its expected order.

3.3.1 Problem A

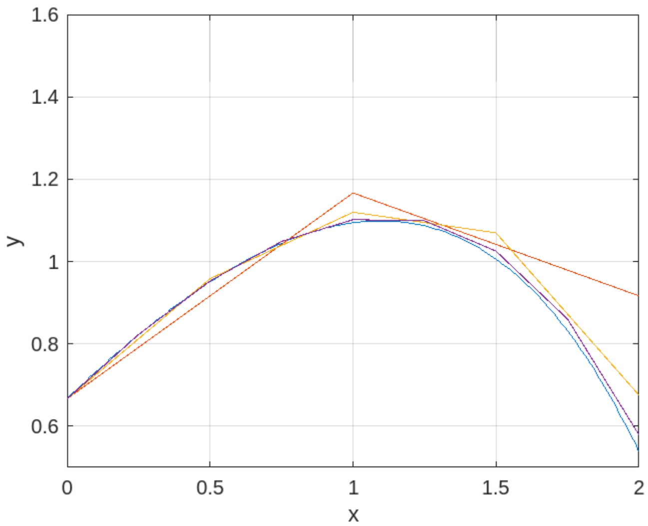

Figure 9: RK2 for different h (real solution in blue)

We see as the h is decreased, approximation gets closer to the actual solution.

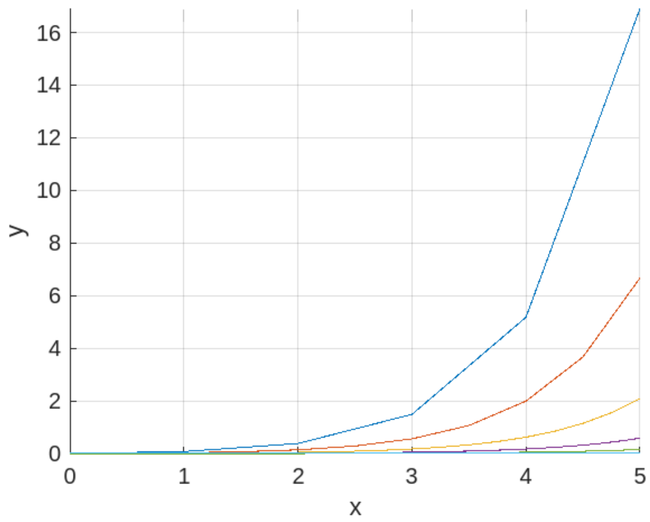

Figure 10: y error for for different h (decreasing the h converges it towards x-axis)

We see as the h is decreased, error converges towards x axis.

3.3.2 Problem B

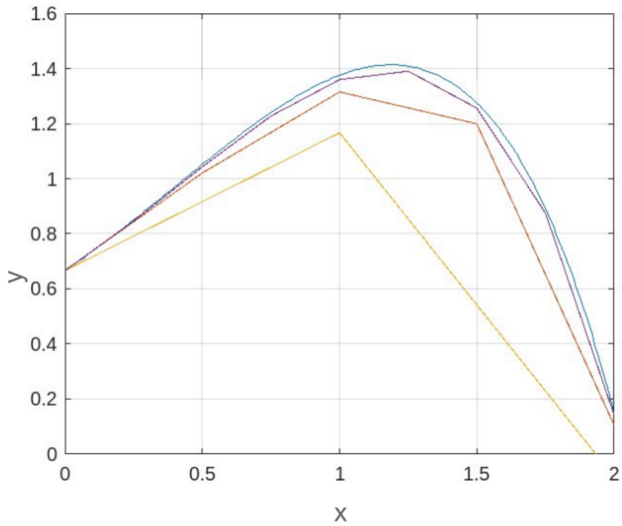

Figure 11: RK2 for different h (real solution in blue)

We see as the h is decreased, approximation gets closer to the actual solution.

We see as the h is decreased, error converges towards x axis.

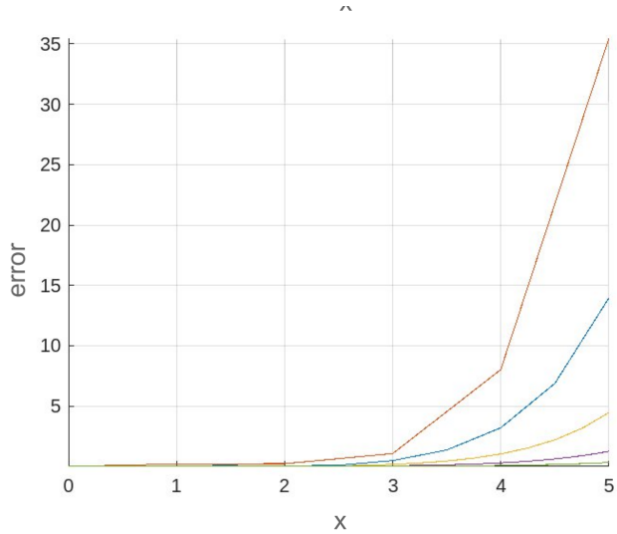

Figure 12: y error for for different h (decreasing the h converges it towards x-axis)

3.4 RK4 vs h

RK4 is a fourth-order method, which means that the local truncation error is on the order of h^5. This implies that, in principle, decreasing the step size by a factor of 10 should reduce the error by a factor of 10^5, assuming the problem is well-behaved and the method is converging at its expected order.

3.4.1 Problem A

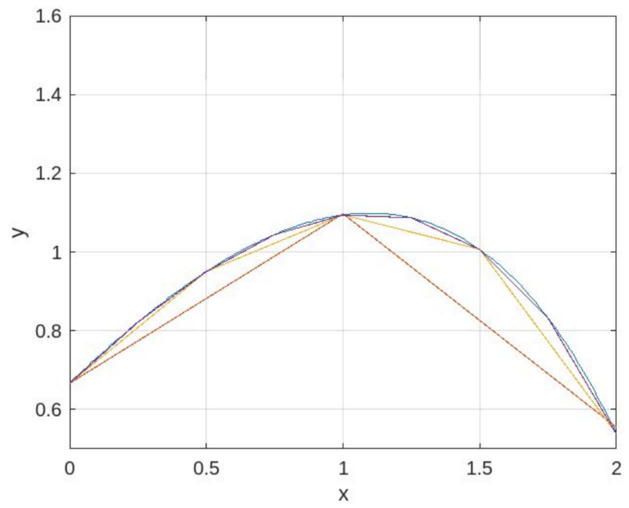

Figure 13: RK4 for different h (real solution in blue)

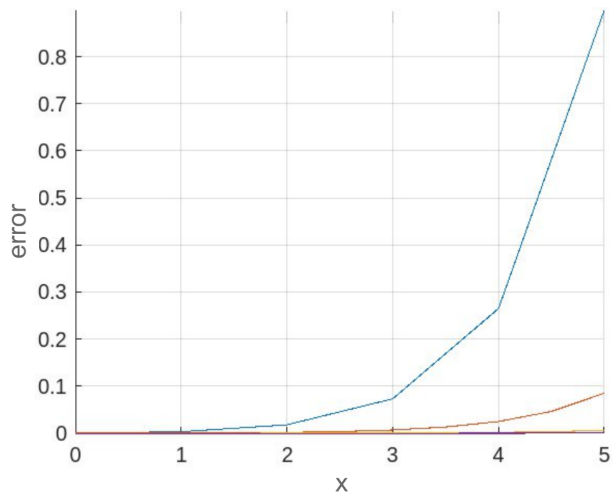

Figure 14: y error for for different h (decreasing the h converges it towards x-axis)

We see as the h is decreased, error converges towards x axis.

3.4.2 Problem B

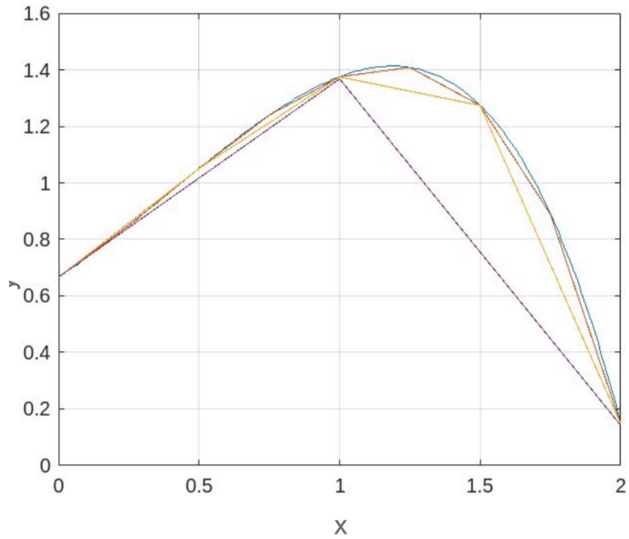

Figure 15: RK4 for different h (real solution in blue)

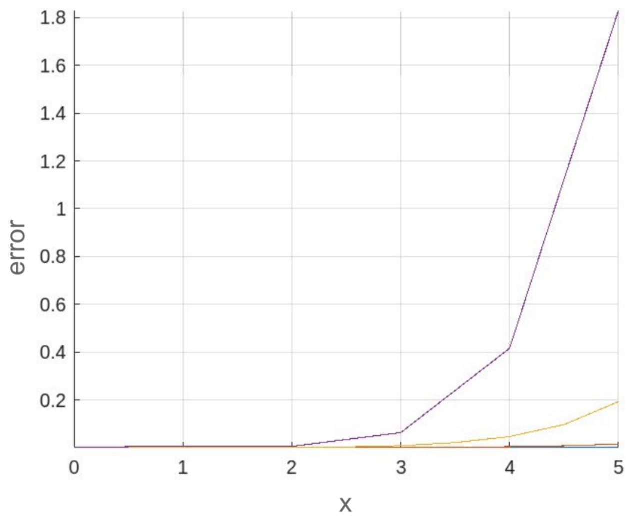

Figure 16: y error for for different h (decreasing the h converges it towards x-axis)

We see as the h is decreased, error converges towards x axis.

4 Conclusion

1. In this study, we conducted a detailed analysis of the Runge-Kutta methods, specifically focusing on the second-order (RK2) and fourth-order (RK4) variants. The mathematical derivations of these methods were presented, shedding light on their underlying principles and the rationale behind their formulations.

2. The RK2 method, also known as the midpoint method, provides a balance between simplicity and accuracy. It was derived through an ad-hoc approach, resulting in a set of coefficients that yield a second-order accurate numerical solution. The error analysis revealed that the local truncation error of RK2 is of order 3, and the global error is proportional to h^2.

3. On the other hand, the RK4 method, a fourth-order technique, was derived by comparing its update formula with the Taylor expansion of the exact solution. The coefficients and weights were determined, leading to a method with a local truncation error of order 5. The global error analysis indicated that the overall accuracy of RK4 is proportional to h^4.

4. To validate the theoretical findings, numerical experiments were conducted on two differential equations, labeled as Problem A and Problem B. The comparison of RK2 and RK4 solutions with the actual solutions demonstrated the superior accuracy of RK4, especially for smaller step sizes. The direction field plots provided visual insights into the behavior of the solutions.

5. Further, the comparison of RK4 and RK2 errors as a function of step size reaffirmed the theoretical expectations. RK4 consistently outperformed RK2 in terms of accuracy, demonstrating its effectiveness in approximating solutions to ordinary differential equations.

6. In conclusion, the Runge-Kutta methods, particularly RK2 and RK4, are valuable tools in numerical analysis for solving ordinary differential equations. The choice between these methods depends on the desired trade-off between computational simplicity and accuracy. RK4, with its higher order of accuracy, is a preferable choice when precision is crucial, while RK2 remains a viable option for simpler problems where computational efficiency is prioritized.