ENHANCED KNOWLEDGE IN SCIENCES AND TECHNOLOGY
e-ISSN: 2773-6385

EKST

Vol. 5 No. 2 (2025) 63-71
https://publisher.uthm.edu.my/periodicals/index.php/ekst

Comparison Between the Runge Kutta Method and Euler Method for Solving Dengue Fever Disease Model

Muhammad Fikreey Mohamad Razapi[1], Mahathir Mohamad[1]*

[1] *Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, UTHM Kampus Cawangan Pagoh, Hab Pendidikan Tinggi Pagoh, KM 1, Jalan Panchor, 84600 Pagoh, Muar, Johor, MALAYSIA.*

*Corresponding Author: mahathir@uthm.edu.my
DOI: https://doi.org/10.30880/ekst.2025.05.02.009

Article Info

Received: 30 December 2024
Accepted: 16 January 2025
Available online: 19 December 2025

Keywords

Dengue Fever, SEIR Model, Euler Method, Runge-Kutta Method, Numerical Simulation

Abstract

Dengue fever remains a global health challenge due to its rapid transmission and severe complications. This study evaluates two numerical methods, Euler and fourth-order Runge-Kutta (RK4), for solving a dengue fever disease model. Utilizing the SEIR framework, the research highlights the RK4 method's superior accuracy and stability compared to Euler's simplicity and computational efficiency. These findings underscore the importance of selecting appropriate numerical techniques for reliable epidemiological modeling, aiding in better-informed public health interventions.

1. Introduction

Dengue fever, caused by the dengue virus (DENV) and transmitted by Aedes aegypti, posed significant global health risks, particularly in tropical and subtropical regions [1]. The disease affected millions annually and presented severe complications such as dengue hemorrhagic fever and dengue shock syndrome [4]. Factors such as urbanization, climate change, and increased human mobility exacerbated its spread, presenting challenges for healthcare systems [3]. Accurate prediction and control of dengue transmission dynamics were critical for designing effective public health interventions, including vaccination campaigns, vector control, and community education programs [2].

The lack of specific antiviral treatment for dengue fever makes prevention and control measures the primary strategies to combat the disease. Vaccination has been developed, but its effectiveness varies across different serotypes of the virus, and its availability remains limited in certain regions. Consequently, vector control methods, such as insecticide spraying, larval source reduction, and the use of genetically modified mosquitoes, have been widely implemented to minimize the spread of the disease. However, these methods require continuous monitoring and adaptation due to the evolving resistance of mosquitoes to insecticides and environmental changes that influence their breeding patterns.

Mathematical modelling emerged as an essential tool for understanding and predicting the spread of infectious diseases like dengue fever. Among these models, the SEIR (Susceptible-Exposed-Infectious-Recovered) framework was widely utilized to simulate disease dynamics and assess the impact of various intervention strategies [3]. The SEIR model extends the traditional SIR (Susceptible-Infectious-Recovered) model by incorporating an exposed compartment, which accounts for individuals who have been infected but are not yet symptomatic or infectious. This additional compartment provides a more realistic representation of the disease progression and enhances the accuracy of epidemic forecasting.

These models relied on solving complex ordinary differential equations (ODEs), which often did not have closed-form analytical solutions. Thus, numerical methods such as Euler and Runge-Kutta were employed to approximate solutions to these equations [3]. The choice of numerical method significantly impacts the accuracy and efficiency of the simulations. The Euler method, despite its simplicity, tends to accumulate errors over time, making it less reliable for long-term predictions. In contrast, the RK4 method provides greater accuracy and

This is an open access article under the CC BY-NC-SA 4.0 license.

stability, making it more suitable for complex epidemiological models. Understanding the strengths and limitations of these numerical methods is crucial for optimizing disease modelling techniques and improving public health decision-making.

The objectives of this study were to solve the dengue fever disease model using both Euler and RK4 methods, to compare their accuracy and computational performance, and to assess the stability of the RK4 method for long-term predictions. By providing a comparative analysis of these numerical methods, this study aims to contribute to the development of more reliable simulation models for dengue fever dynamics and inform more effective control measures.

Nomenclature is included if necessary	
α	Transmission rate between host and vector
β	Recovery rate of the host population
δ_1	Death rate of the vector population
γ	Birth rate of the vector population
μ_n	Recruitment rate of the host population

2. Methodology

2.1 Dengue Fever Model

The SEIR model is defined by the following ODEs:

$$\frac{dx}{dt} = \mu_n\big(1 - x(t)\big) - \alpha x(t)z(t) \quad (1)$$

$$\frac{dy}{dt} = \alpha x(t)z(t) - \beta y(t) \quad (2)$$

$$\frac{dz}{dt} = \gamma\big(1 - z(t)\big)y(t) - \delta_1 z(t) \quad (3)$$

$$r(t) = 1 - x(t) - y(t) \quad (4)$$

Where:

- t: time variable
- $x(t)$: proportion of the susceptible host population
- $y(t)$: proportion of the infectious host population
- $z(t)$: the proportion of the infectious vector
- $r(t)$: the proportion of the recovered population

2.2 Euler method

The Euler method approximates the next value of variables using the following equations:

$$\frac{dx}{dt} = f(t, x, y, z) \quad (5)$$

$$\frac{dy}{dt} = g(t, x, y, z) \quad (6)$$

$$\frac{dz}{dt} = j(t, x, y, z) \quad (7)$$

With value $x(0) = x_0, y(0) = y_0, z(0) = z_0$ and choose the fixed h value that is $h = t_{n=1} - t_n$ is

$$x_{n+1} = x_n + hf(t_n, x_n, y_n, z_n) \quad (8)$$

$$y_{n+1} = y_n + hf(t_n, x_n, y_n, z_n) \quad (9)$$

$$z_{n+1} = z_n + hf(t_n, x_n, y_n, z_n) \tag{10}$$

For $n = 0,1,2,3, \ldots.$
Our SIR model's nonlinear equations are a system of differential equations (1)-(3)), which results in a system of simultaneous nonlinear equations.

$$\frac{dx}{dt} = f(t, x, y, z) = \mu_n(1 - x(t)) - \alpha x(t)z(t) \tag{11}$$

$$\frac{dy}{dt} = g(t, x, y, z) = \alpha x(t)z(t) - \beta y(t) \tag{12}$$

$$\frac{dz}{dt} = j(t, x, y, z) = \gamma(1 - z(t))y(t) - \delta_1 z(t) \tag{13}$$

Using the Euler method we can solve equation (10)-(12) as

$$x_{n+1} = x_n + h[\mu_n(1 - x_n) - \alpha x_n z_n] \tag{14}$$

$$y_{n+1} = y_n + h[\alpha x_n z_n - \beta y_n] \tag{15}$$

$$z_{n+1} = z_n + h[\gamma(1 - z_n)y_n - \delta_1 z_n] \tag{16}$$

2.3 Runge-Kutta method

The 4th-order Runge-Kutta method (RK4) is a highly accurate and widely used numerical method for solving ordinary differential equations (ODEs).

The RK4 method approximates the solution as:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{17}$$

Where:

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= hf(t_n + h, y_n + hk_3) \end{aligned} \tag{18}$$

Where:

- y_n is the current value of the solution at time t_n.
- h is the time step
- $f(t, y)$ is the differential equation to solve

Parameters and Initial Conditions:
Assume parameter values and initial conditions as:

- $\mu_n = 0.1, \alpha = 0.2, \beta = 0.3, \gamma = 0.4, \delta_1 = 0.1$
- $x_0 = 0.99, y_0 = 0.01, z_0 = 0.05$
- Time step $h = 0.1$

RK4 Formula:

For a general differential equation $\frac{dy}{dt} = f(t, y)$:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (19)$$

Where:

$$k_1 = hf(t_n, y_n)$$
$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \quad (20)$$
$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$
$$k_4 = hf(t_n + h, y_n + hk_3)$$

Step-by-Step Calculation:
Compute k_1 for each equation:

- $k_1^x = h[\mu_n(1 - x_0) - \alpha x_0 z_0]$
- $k_1^y = h[\alpha x_0 z_0 - \beta y_0]$ (21)
- $k_1^z = h[\gamma(1 - z_0)y_0 - \delta_1 z_0]$

Compute k_2 for each equation:

- $k_2^x = h\left[\mu_n\left(1 - (x_0 + \frac{k_1^x}{2}\right) - \alpha(x_0 + \frac{k_1^x}{2})(z_0 + \frac{k_1^z}{2})\right]$
- $k_2^y = h\left[\alpha(x_0 + \frac{k_1^x}{2})(z_0 + \frac{k_1^z}{2}) - \beta(y_0 + \frac{k_1^y}{2})\right]$ (22)
- $k_2^z = h[\gamma\left(1 - (z_0 + \frac{k_1^z}{2}\right)(y_0 + \frac{k_1^y}{2}) - \delta_1(z_0 + \frac{k_1^z}{2})]$

Compute k_3 for each equation:

- $k_3^x = h\left[\mu_n\left(1 - (x_0 + \frac{k_2^x}{2}\right) - \alpha(x_0 + \frac{k_2^x}{2})(z_0 + \frac{k_2^z}{2})\right]$
- $k_3^y = h\left[\alpha(x_0 + \frac{k_2^x}{2})(z_0 + \frac{k_2^z}{2}) - \beta(y_0 + \frac{k_2^y}{2})\right]$ (23)
- $k_3^z = h[\gamma\left(1 - (z_0 + \frac{k_2^z}{2}\right)(y_0 + \frac{k_2^y}{2}) - \delta_1(z_0 + \frac{k_2^z}{2})]$

Compute k_4 for each equation:

- $k_4^x = h[\mu_n(1 - (x_0 + k_3^x) - \alpha(x_0 + k_3^x)(z_0 + k_3^z)]$
- $k_4^y = h[\alpha(x_0 + k_3^x)(z_0 + k_3^z) - \beta(y_0 + k_3^y)]$ (24)
- $k_4^z = h[\gamma(1 - (z_0 + k_3^z)(y_0 + k_3^y) - \delta_1(z_0 + k_3^z)]$

Using the RK4 formula:

$$x_{n+1} = x_n + \frac{h}{6}(k_1^x + 2k_2^x + 2k_3^x + k_4^x) \quad (25)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1^y + 2k_2^y + 2k_3^y + k_4^y) \quad (26)$$

$$z_{n+1} = z_n + \frac{h}{6}(k_1^z + 2k_2^z + 2k_3^z + k_4^z) \quad (27)$$

3. Results and Discussion

3.1 Numerical Results

Tables below summarize the solutions for $x(t), y(t) and\ z(t)$:

Table 1 Solution Table of Proportion of Susceptible Hosts of $\boldsymbol{x(t)}$

Time	Euler_x	RK4_x	ODE45_x
0.0	0.9900000000	0.9900000000	0.9900000000
0.1	0.9891100000	0.9891148771	0.9891169064
0.2	0.9882340432	0.9882436435	0.9882471646
0.3	0.9873714223	0.9873856034	0.9873900996
0.4	0.9865214529	0.9865400830	0.9865450572
0.5	0.9856834722	0.9857064302	0.9857114040
0.6	0.9848568387	0.9848840134	0.9848885260
0.7	0.9840409310	0.9840722207	0.9840758281
0.8	0.9832351474	0.9832704596	0.9832727335
0.9	0.9824389048	0.9824781558	0.9824786832
1.0	0.9816516385	0.9816947530	0.9816931347

Table hosts $x(t)$ at different time points table 1 displays the results of the Euler, Runge-Kutta, and benchmark ODE45 methods for determining the proportion of susceptible hosts.

Table 2 Solution Table of Proportion of Infectious Hosts of $\boldsymbol{y(t)}$

Time	Euler_y	RK4_y	ODE45_y
0.0	0.0100000000	0.0100000000	0.0100000000
0.1	0.0108900000	0.0108855648	0.0108830936
0.2	0.0117659568	0.0117572271	0.0117528354
0.3	0.0126285777	0.0126156835	0.0126099004
0.4	0.0134785471	0.0134616084	0.0134549428
0.5	0.0143165278	0.0142956548	0.0142885960
0.6	0.0151431613	0.0151184549	0.0151114740
0.7	0.0159590690	0.0159306211	0.0159241719
0.8	0.0167648526	0.0167327467	0.0167272665
0.9	0.0175610952	0.0175254063	0.0175213168
1.0	0.0183483615	0.0183091570	0.0183068653

Table hosts $y(t)$ at different time points table 2 displays the results of the Euler, Runge-Kutta, and benchmark ODE45 methods for determining the proportion of infectious hosts.

Table 3 Solution Table of Proportion of Infectious Vector Hosts of $\boldsymbol{z(t)}$

Time	Euler_z	RK4_z	ODE45_z
0.0	0.0500000000	0.0500000000	0.0500000000
0.1	0.0497850000	0.0497861035	0.0497986786
0.2	0.0495975852	0.0495995159	0.0496242780

0.3	0.0494370812	0.0494395724	0.0494761612
0.4	0.0493028381	0.0493056320	0.0493537147
0.5	0.0491942302	0.0491970776	0.0492563470
0.6	0.0491106550	0.0491133149	0.0491834890
0.7	0.0490515326	0.0490537716	0.0491345923
0.8	0.0490163049	0.0490178971	0.0491091293
0.9	0.0490044349	0.0490051614	0.0491065920
1.0	0.0490154062	0.0490150547	0.0491264920

Table hosts $z(t)$ at different time points table 3 displays the results of the Euler, Runge-Kutta, and benchmark ODE45 methods for determining the proportion of infectious hosts.

Table 4 Solution Table of Recover of r(t) In Ode45, Runge Kutta And Euler

Time	r_Euler	r_RK4	r_ODE45
0.0	0.0000000000000000087	0.0000000000000	0.0000000000000000087
0.1	-0.000000000000000045	-0.0000004419000	-0.000000000000000050
0.2	-0.000000000000000052	-0.0000008706000	0.0000000000000000503
0.3	0.0000000000000000295	-0.0000012869000	-0.000000000000000015
0.4	0.0000000000000000468	-0.0000016914000	0.0000000000000000364
0.5	0.0000000000000000000	-0.0000020850000	-0.000000000000000013
0.6	0.0000000000000000416	-0.0000024683000	0.0000000000000000139
0.7	-0.000000000000000024	-0.0000028418000	-0.000000000000000048
0.8	-0.000000000000000003	-0.0000032063000	0.0000000000000000035
0.9	0.0000000000000000208	-0.0000035621000	0.0000000000000000208
1.0	-0.000000000000000024	-0.0000039100000	0.0000000000000000312

Tables 5 below summarize the error analysis for $x(t), y(t) and\ z(t)$:

Table 5 Error Analysis for $\boldsymbol{x(t), y(t) and\ z(t)}$:

Time	Error_Euler_x	Error_RK4_x	Error_Euler_y	Error_RK4_y	Error_Euler_z	Error_RK4_z
0.0	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000
0.1	0.0000069064	0.0000020293	0.0000069064	0.0000024712	0.0000136786	0.0000125752
0.2	0.0000131215	0.0000035211	0.0000131215	0.0000043917	0.0000266928	0.0000247621
0.3	0.0000186773	0.0000044962	0.0000186773	0.0000057831	0.0000390800	0.0000365889
0.4	0.0000236044	0.0000049742	0.0000236044	0.0000066657	0.0000508765	0.0000480827
0.5	0.0000279318	0.0000049738	0.0000279318	0.0000070588	0.0000621168	0.0000592695
0.6	0.0000316873	0.0000045126	0.0000316873	0.0000069809	0.0000728340	0.0000701742
0.7	0.0000348971	0.0000036074	0.0000348971	0.0000064492	0.0000830597	0.0000808207
0.8	0.0000375862	0.0000022740	0.0000375862	0.0000054802	0.0000928244	0.0000912322
0.9	0.0000397784	0.0000005273	0.0000397784	0.0000040895	0.0001021571	0.0001014306
1.0	0.0000414962	0.0000016184	0.0000414962	0.0000022917	0.0001110858	0.0001114373

Table 6 Error Analysis table of $r(t)$

Time	Error_Euler_r	Error_RK4_r
0.0	0.00000000000000000000	0.00000000000000000000
0.1	0.00000000000000000520	0.00000044190000000000
0.2	0.00000000000000010235	0.00000087060000000000
0.3	0.00000000000000004510	0.00000128690000000000
0.4	0.00000000000000001041	0.00000169140000000000
0.5	0.00000000000000001388	0.00000208500000000000
0.6	0.00000000000000002776	0.00000246830000000000
0.7	0.00000000000000002429	0.00000284180000000000
0.8	0.00000000000000000694	0.00000320630000000000
0.9	0.00000000000000000000	0.00000356210000000000
1.0	0.00000000000000005551	0.00000391000000000000

This table provides the **Error Analysis** for the numerical solutions of $y(t)$using **Euler's method** and **RK4** compared to the highly accurate reference solution obtained from ODE45.

The results show that, while the Euler method is computationally simpler and requires fewer resources, its low accuracy and error accumulation make it unsuitable for precise simulations of dynamics of dengue fever. In contrast, the RK4 method achieves high accuracy with few errors, making it an excellent choice for complex epidemiological models. However, the computational cost of RK4 must be considered, especially when performing large-scale or real-time simulations.

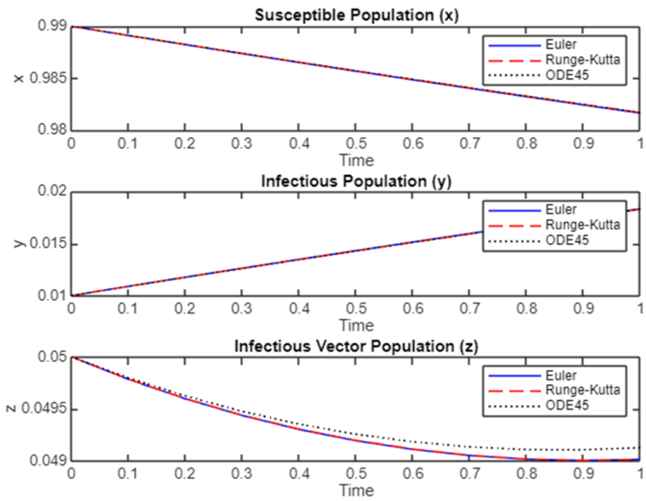

Fig. 1 Visualizes The Convergence of $x(t)$ Solutions Over Time

The RK4 method closely tracks the ODE45 benchmark curve, while the Euler method diverges slightly as t increases.

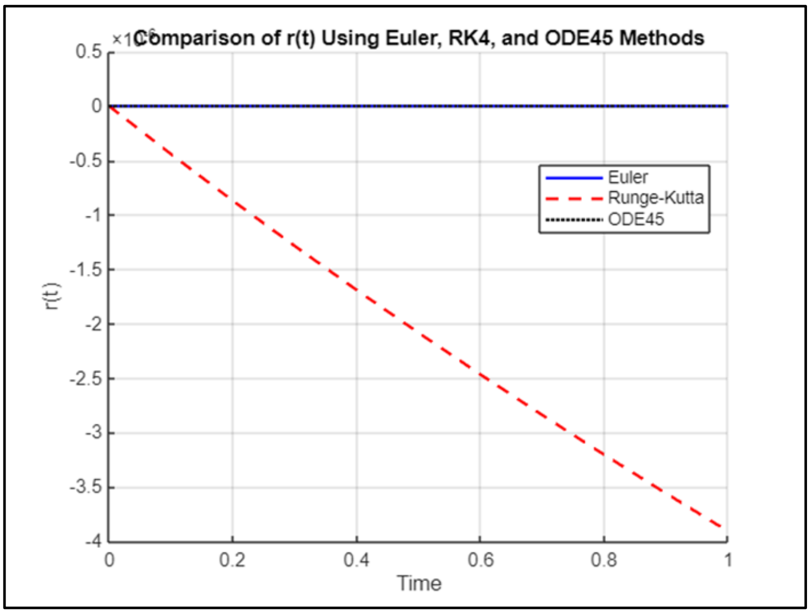

Fig. 2 Solution of Recover $\boldsymbol{r(t)}$ for Euler, Runge Kutta and Ode45

Fig.2 shows the solution of recover $r(t)$for Euler, Runge Kutta and Ode45.

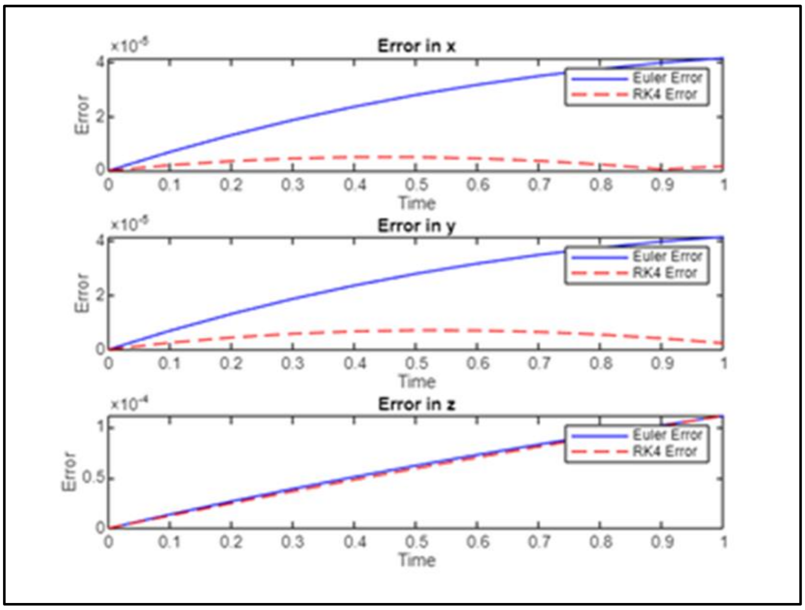

Fig. 3 Error trends for x(t)

Fig. 3 provides a comparative view of the error trends for $x(t)$, demonstrating that RK4 maintains significantly lower errors across all time points compared to the Euler method.

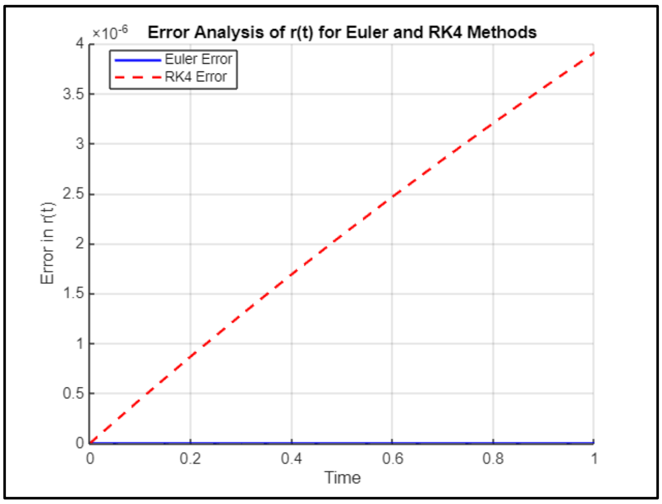

Fig. 4 error analysis of $\boldsymbol{r(t)}$

4. Conclusion

The RK4 method demonstrated superior accuracy and stability for solving the dengue fever model, making it a preferred choice for detailed simulations. However, its higher computational cost may limit its use in resource-constrained settings. The Euler method, despite its simplicity, is less suitable for complex, long-term epidemiological predictions. Future research should explore hybrid numerical methods to balance accuracy and efficiency.

Acknowledgement

The authors would like to thank the Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, for its support.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Muhammad Fikreey Mohamad Razapi;* ***solve the equations:*** *Muhammad Fikreey Mohamad Razapi;* ***analysis and interpretation of results:*** *Muhammad Fikreey Mohamad Razapi, Mahathir Mohamad;* ***draft manuscript preparation:*** *Muhammad Fikreey Mohamad Razapi, Mahathir Mohamad. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Mukhsar, Wibawa, G. N. A., Tenriawaru, A., Usman, I., Firihu, M. Z., Variani, V. I., Mansur, A. B. F., & Basori, A. H. (2023). Stochastic Bayesian Runge-Kutta method for dengue dynamic mapping. *MethodsX*, 10. https://doi.org/10.1016/j.mex.2022.101979

[2] Lede, Y. K., & Mungkasi, S. (2019). Performance of the Runge-Kutta methods in solving a mathematical model for the spread of dengue fever disease. *AIP Conference Proceedings*, 2202. https://doi.org/10.1063/1.5141657

[3] Sanusi, W., Pratama, M. I., Rifandi, M., Sidjara, S., Irwan, & Side, S. (2021). Numerical Solution of SIRS model for Dengue Fever Transmission in Makassar Citywith Runge Kutta Method. *Journal of Physics: Conference Series,* 1752(1). https://doi.org/10.1088/1742-6596/1752/1/012004

[4] Khan, M. B., Yang, Z. S., Lin, C. Y., Hsu, M. C., Urbina, A. N., Assavalapsakul, W., Wang, W. H., Chen, Y. H., & Wang, S. F. (2023). Dengue overview: An updated systemic review. *In Journal of Infection and Public Health* (Vol. 16, Issue 10, pp. 1625–1642). Elsevier Ltd. https://doi.org/10.1016/j.jiph.2023.08.001