

An Efficient GUI-Based Multistep Block Method for SEIR Pandemic Spread Simulation

Nur Tasnem Jaaffar

Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

*Corresponding author : tasnemjaaffar@tmsk.uitm.edu.my

Received: 20 November 2024

Revised: 14 May 2025

Accepted: 17 June 2025

ABSTRACT

This study presents a novel Graphical User Interface (GUI)-based implementation to simulate the SEIR (Susceptible-Exposed-Infectious-Recovered) pandemic model using a multistep block method. The proposed approach enhances computational efficiency by simultaneously approximating two solutions and reducing function evaluations, while maintaining accuracy comparable to the traditional fourth-order Runge-Kutta (RK4) method. A predictor-corrector scheme further improves accuracy. Numerical comparisons demonstrate that our method reduces computation time by approximately 6% and cuts function calls by nearly 50% compared to RK4. The GUI, developed using C with GTK, enables real-time simulations accessible to non-programmers. This makes it a valuable tool for researchers and policymakers engaged in disease forecasting and outbreak control.

Keywords: SEIR model, multistep block method, pandemic modelling, computational efficiency, pandemic preparedness, Graphical User Interface (GUI).

1 INTRODUCTION

Recently, the world has been alarmed by a new, fast-spreading variant of the coronavirus originating in Wuhan, China, known as SARS-CoV-2, which affects the respiratory system and other organs in the body [1]. This disease, named COVID-19, has impacted not only thousands of individuals who have fallen ill or succumbed to the virus but has also disrupted daily life, businesses, global trade, and movement.

These disruptions have significantly affected the global economy [2]. During this crisis, government decisions and interventions are essential to mitigate the virus's spread and estimate the number of susceptible, exposed, infected, and recovered individuals. These estimates support policymakers in making critical decisions on vaccination strategies, movement restrictions, social distancing, and

other preventive actions. It is crucial that these estimates are generated rapidly to facilitate timely responses during this health crisis.

The SEIR model is a mathematical framework that simulates pandemic spread by categorizing individuals into four groups: those at risk of infection (susceptible), those who are infected but not yet contagious (exposed), those who can transmit the disease (infected), and those who have recovered. This model differs from the traditional SIR model by incorporating an incubation, or “exposed” period, representing the 2-3 days before an individual becomes infectious to others after contracting COVID-19.

Previously, researchers have explored various methods for solving the SEIR model. Rustan and Handayani [3] applied the Euler method to a modified SEIR model for COVID-19 in Indonesia, while Aakash et al. [4] used both the Euler and fourth-order Runge-Kutta methods to address the model's complexity. Recently, Palma et al. [5] demonstrated that simulations using fourth- and fifth-order Runge-Kutta methods produced patterns consistent with MATLAB's ODE45 algorithm. Mungkasi [6] introduced a multistage method, utilizing successive approximation and variational iteration methods to derive semi-analytical solutions for the SEIR model. The Mickens method, a non-standard finite difference approach, has also been applied, preserving the SEIR model's dynamic properties while simplifying implementation.

Predictor-corrector techniques have also been employed in this context. For example, Farooqi et al. [7] applied predictor-corrector methods to a non-standard finite difference approach, and Gao et al. [8] utilized a modified predictor-corrector method to solve a fractional-order SIR model. More recently, Salaudeen et al. [9] used the Adams-Bashforth-Moulton method with a predictor-corrector scheme to solve a SIRS model, while Arif et al. [10] combined a third-order explicit predictor-corrector approach with neural networks to simulate the SEIR model. However, to our knowledge, no studies have yet applied a multistep block method with a predictor-corrector approach specifically for the SEIR model.

The SEIR pandemic model is a type of Ordinary Differential Equation (ODE), and selecting efficient numerical algorithms for solving these equations is crucial for reducing computational times. Recent research has focused on developing cost-effective computational methods, underscoring the importance of computational efficiency for solving ODEs [11]. For instance, the Modified Euler method has shown considerable efficiency gains in simulating moderately stiff chemical kinetics in reactive-flow models on GPUs, outperforming traditional methods [12]. This highlights the impact of method selection on computational performance, particularly when handling large systems of independent ODEs.

In this paper, we present a multistep block approach that uses a block strategy to estimate two approximate solutions simultaneously and a multistep technique to reduce the number of functions calls in the algorithm. Together, these two features decrease the algorithm's computational time, making it particularly effective for models with numerous variables and parameters involving large populations. The accuracy of this approach is further enhanced through a predictor-corrector strategy, where the algorithm first predicts approximate solutions using a lower-order method, then corrects them using a higher-order method to improve precision.

Building on this new approach for solving the SEIR model, our next objective is to develop a Graphical User Interface (GUI) based on the proposed multistep block method to simulate pandemic spread. Such a GUI will be especially useful for policymakers, researchers, and health professionals who may lack expertise in mathematical modelling or programming [13]. This user-friendly interface will assist in forecasting disease transmission dynamics, providing essential data-driven recommendations to inform government policies and preventive measures across economic, social, and educational domains.

2 METHODOLOGY

This study is divided into two major objectives. The first one was to develop an efficient multistep block method to solve SEIR model. The next task is to create a GUI that will provide user-friendly interface to the user so that the simulations can be performed by those semi-skilled professionals.

2.1 SEIR Model Description

This study considers the compartmental SEIR model established by Harir et al. [14] that are more complex and involved many parameters so that we can test our method's computational performance. This nonlinear SEIR model of the COVID-19 can be presented as below:

$$\begin{aligned} S'(t) &= -\beta \frac{S(t)I(t)}{N} - \frac{Z}{N}S(t) + (\rho_I + \rho_E) + \left(\frac{e_I}{N} + \frac{e_E}{N}\right)S(t) + vN(t) - \mu S(t), \\ E'(t) &= -\beta \frac{S(t)I(t)}{N} + \frac{Z}{N}S(t) - \alpha E(t) - \left(\frac{e_I}{N} + \frac{e_E}{N}\right)E(t) - \mu E(t) - \sigma E(t), \\ I'(t) &= \alpha E(t) - \gamma I(t) - \left(\frac{e_I}{N} + \frac{e_E}{N}\right)I(t) - \mu I(t), \\ R'(t) &= \gamma I(t) - \mu R(t) + \sigma E(t). \end{aligned} \tag{1}$$

The initial conditions are given as $S(0) = S_0, E(0) = E_0, I(0) = I_0$ and $R(0) = R_0$ and $S(t), E(t), I(t)$ and $R(t)$ are those individuals that are susceptible, exposed, infected, and recovered, respectively with t as the number of days. The total population is given as $N(t) = S(t) + E(t) + I(t) + R(t)$ meanwhile the parameters involved are as follows:

- β = the transmission rate for infected people,
- α = the infection rate for exposed people,
- γ = the average number of people currently infected,
- μ = the natural death rate,
- v = the natural birth rate,

σ = the recovery rate,

Z = the infection rate in standard situation,

ρ_I = the average daily number of international passengers that arrived by air,

ρ_E = the average daily number of domestic passengers that arrived by air,

e_I = the average daily number of international passengers that departed by air,

e_E = the average daily number of domestic passengers that departed by air.

2.2 Multistep Block Method

The SEIR system in equation (1) can be written as the first order ODE as follows:

$$\begin{aligned} S'(t) &= u(t, S(t), E(t), I(t), R(t)), \\ E'(t) &= u(t, S(t), E(t), I(t), R(t)), \\ I'(t) &= u(t, S(t), E(t), I(t), R(t)), \\ R'(t) &= u(t, S(t), E(t), I(t), R(t)), \\ t &\in [a, b]. \end{aligned} \tag{2}$$

In our approach, we need to convert the continuous interval $[a, b]$ into discrete points where the step size involved is $h = \frac{b-a}{M}$ and M is the total number of iterations chosen. Both two approximate solutions at points t_{i+1} and t_{i+2} of each compartment ($S_{i+1}, S_{i+2}, E_{i+1}, E_{i+2}, I_{i+1}, I_{i+2}$ and R_{i+1}, R_{i+2}) will be calculated in block approach to save the computational time.

2.3 Predictor-Corrector Scheme

Then, the multistep block approach is applied by considering fifth order backward points for each two-point solutions. Afterwards, the right-hand side of equation (2) will be integrated twice for each approximate solution points at t_{i+1} and t_{i+2} where the functions $u(t, S(t), E(t), I(t), R(t))$ is approximated by using Lagrange interpolation polynomial of degree 4. Finally, the corrector formula for our multistep method is obtained as follows:

$$\begin{aligned}
 S_{i+1} &= S_i + \frac{h}{720} (251u_{i+1} + 646u_i - 264u_{i-1} + 106u_{i-2} - 19u_{i-3}) \\
 S_{i+2} &= S_i + \frac{h}{90} (29u_{i+2} + 124u_{i+1} + 24u_i + 4u_{i-1} - u_{i-2}) \\
 E_{i+1} &= E_i + \frac{h}{720} (251u_{i+1} + 646u_i - 264u_{i-1} + 106u_{i-2} - 19u_{i-3}) \\
 E_{i+2} &= E_i + \frac{h}{90} (29u_{i+2} + 124u_{i+1} + 24u_i + 4u_{i-1} - u_{i-2}) \\
 I_{i+1} &= I_i + \frac{h}{720} (251u_{i+1} + 646u_i - 264u_{i-1} + 106u_{i-2} - 19u_{i-3}) \\
 I_{i+2} &= I_i + \frac{h}{90} (29u_{i+2} + 124u_{i+1} + 24u_i + 4u_{i-1} - u_{i-2}) \\
 R_{i+1} &= R_i + \frac{h}{720} (251u_{i+1} + 646u_i - 264u_{i-1} + 106u_{i-2} - 19u_{i-3}) \\
 R_{i+2} &= R_i + \frac{h}{90} (29u_{i+2} + 124u_{i+1} + 24u_i + 4u_{i-1} - u_{i-2})
 \end{aligned} \tag{3}$$

Since our approach implements predictor and corrector scheme therefore, we need another multistep block formula for predictor. The procedure to obtain predictor formula is the same as obtaining the corrector formula previously, but this time we will use the fourth order backward points which is one order less than the backwards points for corrector formula. The final predictor formula is obtained as follows:

$$\begin{aligned}
 S_{i+1} &= S_i + \frac{h}{24} (55u_i - 59u_{i-1} + 37u_{i-2} - 9u_{i-3}) \\
 S_{i+2} &= S_i + \frac{h}{3} (27u_i - 44u_{i-1} + 31u_{i-2} - 8u_{i-3}) \\
 E_{i+1} &= E_i + \frac{h}{24} (55u_i - 59u_{i-1} + 37u_{i-2} - 9u_{i-3}) \\
 E_{i+2} &= E_i + \frac{h}{3} (27u_i - 44u_{i-1} + 31u_{i-2} - 8u_{i-3}) \\
 I_{i+1} &= I_i + \frac{h}{24} (55u_i - 59u_{i-1} + 37u_{i-2} - 9u_{i-3}) \\
 I_{i+2} &= I_i + \frac{h}{3} (27u_i - 44u_{i-1} + 31u_{i-2} - 8u_{i-3}) \\
 R_{i+1} &= R_i + \frac{h}{24} (55u_i - 59u_{i-1} + 37u_{i-2} - 9u_{i-3}) \\
 R_{i+2} &= R_i + \frac{h}{3} (27u_i - 44u_{i-1} + 31u_{i-2} - 8u_{i-3})
 \end{aligned} \tag{4}$$

2.4 Initial Conditions

During the first four iterations of this algorithm, which is when $i = 0,1,2,3$ the predictor formula requires four initial approximate solutions, thus we find them by using Euler method order 1 and Modified Euler method order 2 (predictor-corrector scheme as well). Then, starting from the iterations of $i = 4$ until $i = M - 1$, we can proceed to obtain the two approximate solutions simultaneously by using the proposed multistep block method of equation (3) and equation (4).

2.5 Algorithm: Multistep Block Method for SEIR Model

Below is the algorithm for the proposed multistep block method incorporating the predictor-corrector scheme to solve and illustrate the SEIR model.

Input:

Initial conditions: S_0, E_0, I_0, R_0
Parameters: $\beta, \alpha, \gamma, \mu$, etc.
Total population: N
Step size: h
Total number of days: T

Output:

Arrays $S(t), E(t), I(t), R(t)$ over the time interval $[0, T]$

Step 1: Initialize

Set $t_0 = 0$

Set initial values: $S[0] = S_0, E[0] = E_0, I[0] = I_0, R[0] = R_0$

Initialize solution arrays: $S[], E[], I[], R[]$ and derivative functions

Step 2: Initial Solutions using Euler and Modified Euler Method (First 3 steps)

For $i = 1$ to 2

- Compute $N_t[i] = S[i] + E[i] + I[i] + R[i]$
- Evaluate function calls $f_S[i], f_E[i], f_I[i], f_R[i]$
- Predictor step (Euler): Compute predicted S, E, I, R
- Update $N_t[i]$ and evaluate the function calls $f_S[i], f_E[i], f_I[i], f_R[i]$ again.
- Corrector step (Modified Euler): Compute corrected S, E, I, R
- Update $N_t[i+1]$ and evaluate function calls again.

Step 3: Apply Multistep Block Method (i from 3 onwards)

For $i = 3$ to N

PREDICTOR (Adams-Bashforth Order 4):

- Predict $S[i+1], S[i+2], \dots, R[i+1], R[i+2]$ using previous f_* values
- Update $N_t[i+1], N_t[i+2]$
- Evaluate predictor function values f_* at $i+1, i+2$

CORRECTOR (Adams-Moulton Order 5):

- Correct $S[i+1], S[i+2], \dots, R[i+1], R[i+2]$ using predictor values
- Update N_t and function evaluations for $i+1$ and $i+2$

Step 4: Output

- Display numerical values for S, E, I, R

- Print total number of function calls and iteration steps
- Output CPU time taken for execution

End of Algorithm

3 GUI PROGRAM

Next, we will implement our method discussed previously in the GUI program created by using C programming with GTK widget toolkit under macOS operating system. A GUI is a type of computer human interface on a computer, where people and computers communicate with each other [15]. A GUI allows users to interact with software by handling events like mouse clicks, selections, and text input. It responds to these inputs by updating the visible elements (widgets) on the screen [16]. This user-friendly interface benefits users from various backgrounds who may not be familiar with mathematics or programming. By simply inputting parameter values, users can view visual and numerical data of the spread simulation, helping them analyse and recommend suitable intervention solutions.

Figure 1 shows the flowchart of our GUI program. This flowchart depicts the process of developing and running a simulation for the SEIR (Susceptible-Exposed-Infectious-Recovered) model using a C programming interface with GTK widgets. The process begins with designing a user-friendly interface, where users can input various parameters, total population, and step sizes, essential for configuring the simulation. These inputs are then processed through a multistep block method algorithm to solve the SEIR model equations, which simulate the spread of a disease over time. The simulation generates both graphical projections of pandemic trends and numerical values for each population category (S, E, I, R). The simulation continues iteratively, allowing users to analyse changes until they choose to end it. This process provides insights into how different parameters affect pandemic dynamics.

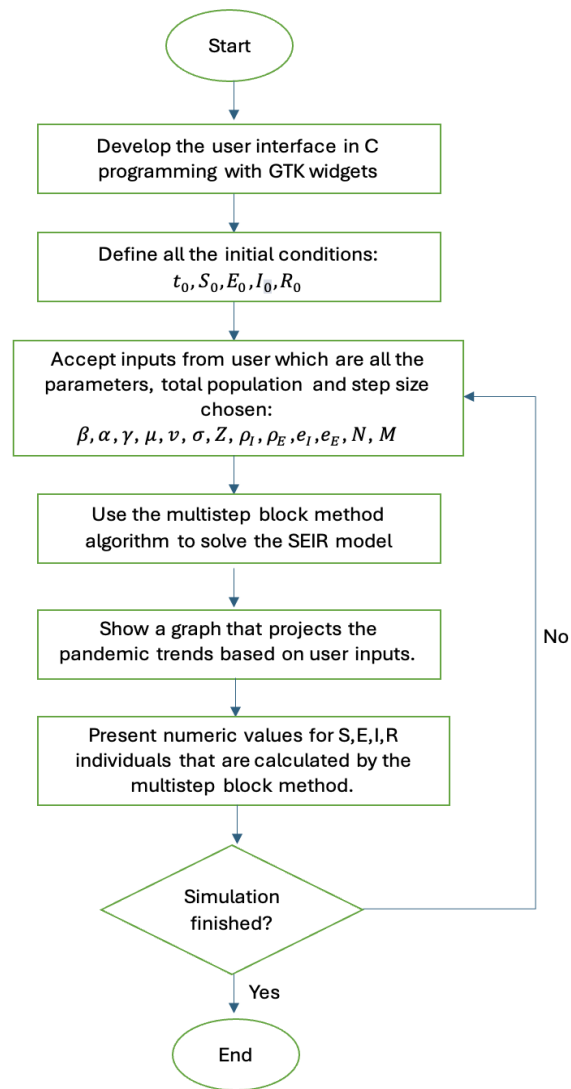
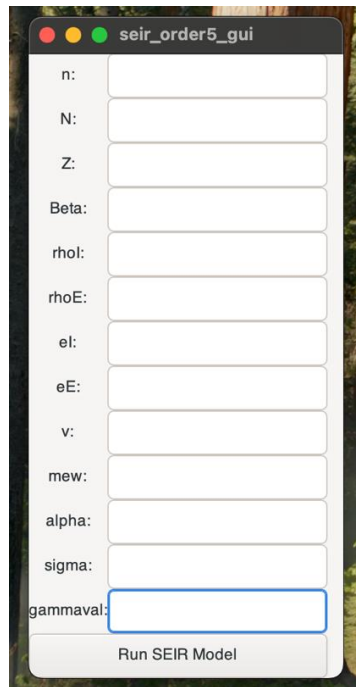


Figure 1: Flowchart of GUI program for pandemic spread.

Upon GUI program execution, the user is presented with the interface illustrated in Figure 2 below:



The image shows a graphical user interface window titled 'seir_order5_gui'. It contains a vertical list of input fields, each preceded by a label: 'n:', 'N:', 'Z:', 'Beta:', 'rhoI:', 'rhoE:', 'eI:', 'eE:', 'v:', 'mew:', 'alpha:', 'sigma:', and 'gammaval:'. The 'gammaval:' field is currently selected with a blue border. At the bottom of the window is a button labeled 'Run SEIR Model'.

Figure 2: GUI interface

4 RESULTS AND DISCUSSION

After entering the input parameters in Figure 2, users can monitor the model's behaviour. By pressing the 'Run SEIR Model' button, a projection graph (as shown in Figure 3) and numerical values (as shown in Table 1) for the S, E, I, and R groups will be displayed.

Figure 3 shows the dynamics of the SEIR (Susceptible-Exposed-Infected-Recovered) model over a period of 20 days. Initially, the Susceptible population (purple line) starts high and steadily decreases as individuals move into the Exposed (blue line) and subsequently the Infected (green line) categories. The Exposed group rises gradually as more susceptible individuals are exposed to the infection, eventually levelling off as these individuals transition to the Infected category. The Infected population grows as exposed individuals become actively infectious, peaking and eventually declining as they recover. The Recovered population (orange line) increases as infected individuals recover, reflecting the cumulative number of recoveries over time.

Similarly, Table 1 illustrates the dynamics of disease spread in the SEIR model in terms of numerical values. The susceptible population declines as individuals transition through the exposed and infected phases, with a corresponding rise in the recovered category. This pattern is typical of

infectious disease progression, where susceptible individuals are gradually replaced by exposed, infected, and eventually recovered individuals over time.

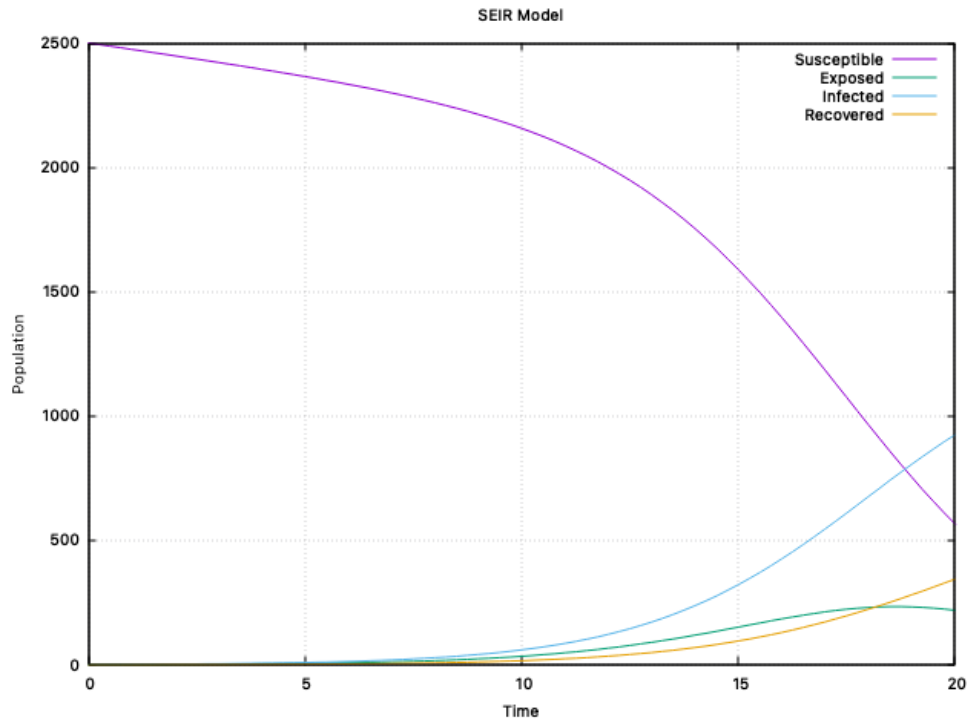


Figure 3: The projection graph of SEIR individuals.

Table 1: The numerical values of SEIR individuals.

t	S approximate	E approximate	I approximate	R approximate
0	2500	1	1	0
2	2448.29599	1.658886	2.624919	0.415349
4	2394.13718	3.61357	5.900195	1.306114
6	2333.79245	7.819432	13.004974	3.254551
8	2259.55187	16.559746	28.155796	7.451931
10	2156.62101	33.985648	59.585725	16.290092
12	1999.53412	66.311176	122.028125	34.269446
14	1753.76283	118.717339	237.33836	68.864603
16	1397.00463	184.13965	425.261704	129.680506
18	965.570782	230.138906	675.578949	223.048155
20	566.538128	219.047677	924.053643	343.778783

Given that a new method is being introduced, it is important to compare its performance with previously established methods to assess and analyse its effectiveness. For this purpose, comparisons will be conducted with the Runge-Kutta order 4 (RK4) method and the Taylor Collocation Method (TCM) [17]. To facilitate comparison between these methods, the input parameters and initial conditions are provided below:

For $t \in [0, 20]$, the initial conditions are given by:

$$S(0) = 2500, E(0) = 1, I(0) = 1, R(0) = 0,$$

$$N(0) = S(0) + E(0) + I(0) + R(0) = 2502.$$

Meanwhile, the parameters are as follows:

$$\beta = 0.8, \alpha = 0.75, \sigma = 0.1, \gamma = 0.05, v = \frac{0.009}{N},$$

$$\mu = 0.01, Z = 0.001, \rho_I = \rho_E = 0.15, e_I = 0.01, e_E = 0.03.$$

We implemented the multistep block method with step size $h = \frac{1}{4}$.

Table 3 shows the comparison of our multistep method numerical results with RK4 and TCM. The comparison table highlights the performance of our proposed multistep method against the established RK4 and TCM methods for the SEIR model. Observing values across all compartments (Susceptible, Exposed, Infected, Recovered) over time, the multistep method maintains comparable values to RK4 and TCM throughout the simulation, with only minor deviations at specific points, such as in the susceptible and recovered compartments at later stages. For instance, at $t = 16$ and $t = 20$, slight differences in susceptible and recovered counts between the methods suggest that the multistep approach closely tracks the population dynamics. Therefore, the multistep method consistently aligns with the results produced by RK4 and TCM, demonstrating its accuracy in modelling disease progression. Accuracy in solving the SEIR model is crucial due to its significant implications for public health, policymaking, economic stability, and overall societal well-being. Reliable and accurate predictions enable effective responses to infectious disease outbreaks, helping to save lives, manage healthcare resources efficiently, and minimize the socio-economic impact of epidemics.

The percentage comparison error in Table 4 compares the accuracy of the proposed Multistep method with the established RK4 and TCM methods across all SEIR compartments over time. From days 0 to 14, the Multistep method produces identical results to both reference methods, showing 0% error. Minor deviations begin to appear at day 10 in the infected compartment (1.695% error) and later in the simulation for susceptible, infected, and recovered values, with all errors remaining below 1%. The only noticeable discrepancy is in the exposed compartment at day 20, showing a 0.905% error versus TCM. These small errors confirm that the Multistep method offers highly accurate results comparable to RK4 and TCM while maintaining computational efficiency, making it a reliable and efficient alternative for simulating pandemic spread.

In addition to accuracy, our method offers clear advantages over the one-step RK4 method by reducing the total number of iteration steps and function calls, which directly impacts computational time. As illustrated in Table 2, our approach achieves faster execution by requiring fewer function calls and iteration steps. Unlike RK4, which is a one-step method needing four function evaluations per step, our method, as a two-point block method, approximates two solutions simultaneously per iteration, effectively halving the iteration count. This difference significantly reduces computational time, a critical benefit for scientific and engineering applications that involve large-scale simulations with millions or even billions of variables. Faster algorithms can thus lead to substantial savings in computational and operational costs.

Table 2: The comparison of computational time

	RK4	Multistep
Total function calls	1280	652
Total step of iterations	80	42
Time taken (seconds)	0.000202	0.000190

Table 3: Comparison of S,E,I,R values

t (days)	S (Susceptible)			E (Exposed)			I (Infected)			R (Recovered)		
	RK4	TCM	Multistep	RK4	TCM	Multistep	RK4	TCM	Multistep	RK4	TCM	Multistep
0	2500	2500	2500	1	1	1	1	1	1	0	0	0
2	2448	2448	2448	2	2	2	3	3	3	0	0	0
4	2394	2394	2394	4	4	4	6	6	6	1	1	1
6	2334	2334	2334	8	8	8	13	13	13	3	3	3
8	2260	2260	2260	17	17	17	28	28	28	7	7	7
10	2157	2157	2157	34	34	34	59	59	60	16	16	16
12	2000	2000	2000	66	66	66	122	122	122	34	34	34
14	1754	1754	1754	119	119	119	237	237	237	69	69	69
16	1398	1398	1397	184	184	184	425	425	425	129	129	130
18	967	967	966	230	230	230	675	675	676	223	223	223
20	567	567	567	219	221	219	923	923	924	343	343	344

5 CONCLUSION

The SEIR model effectively captures disease progression dynamics, showing a decline in the susceptible population as individuals transition through exposure, infection, and recovery stages. The proposed multistep method closely aligns with established methods like RK4 and TCM in terms of accuracy, with minor deviations, affirming its reliability. Notably, it also improves computational efficiency by reducing iteration steps and function calls, offering a faster alternative ideal for large-scale simulations, thus supporting cost-effective and timely analysis in fields requiring extensive computational resources. Future enhancements include extending the algorithm to Susceptible-Exposed-Infected-Recovered-Susceptible (SEIRS) and Susceptible-Exposed-Infected-Recovered-Vaccinated (SEIRV) models, integrating real-time data input, and deploying the GUI in real-time public health monitoring systems.

Table 4: Percentage comparison error

t (days)	S (Susceptible)		E (Exposed)		I (Infected)		R (Recovered)	
	% Error VS RK4	% Error VS TCM	% Error VS RK4	% Error VS TCM	% Error VS RK4	% Error VS TCM	% Error VS RK4	% Error VS TCM
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000	1.695	1.695	0.000	0.000
12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	0.072	0.072	0.000	0.000	0.000	0.000	0.775	0.775
18	0.103	0.103	0.000	0.000	0.148	0.148	0.000	0.000
20	0.000	0.000	0.000	0.905	0.108	0.108	0.292	0.292

REFERENCES

- [1] U. Jain, "Effect of COVID-19 on the organs," *Cureus*, vol. 12, no. 8, p. e9540, 2020, doi: 10.7759/cureus.9540.
- [2] A. Haleem, M. Javaid, and R. Vaishya, "Effects of COVID-19 pandemic in daily life," *Current Medicine Research and Practice*, vol. 10, no. 2, pp. 78–79, 2020, doi: 10.1016/j.cmrp.2020.03.011.
- [3] R. Rustan and L. Handayani, "The outbreak's modeling of coronavirus (COVID-19) using the modified SEIR model in Indonesia," *Spektra: Jurnal Fisika dan Aplikasinya*, vol. 5, no. 1, pp. 61–68, 2020, doi: 10.21009/SPEKTRA.051.07.
- [4] M. Aakash, C. Gunasundari, and Q. M. Al-Mdallal, "Mathematical modeling and simulation of SEIR model for COVID-19 outbreak: A case study of Trivandrum," *Frontiers in Applied Mathematics and Statistics*, vol. 9, p. 1124897, 2023, doi: 10.3389/fams.2023.1124897.
- [5] D. I. Palma and S. Mungkasi, "Fourth and fifth-order Runge-Kutta methods for solving a susceptible-exposed-infected-recovered mathematical model of the spread of COVID-19," in *AIP Conference Proceedings*, vol. 3074, no. 1. AIP Publishing, 2024.
- [6] S. Mungkasi, "Successive approximation, variational iteration, and multistage-analytical methods for a SEIR model of infectious disease involving vaccination strategy," *Communication in Biomathematical Sciences*, vol. 3, no. 2, pp. 114–126, 2021, doi: 10.5614/cbms.2020.3.2.3.
- [7] A. Farooqi et al., "An accurate predictor-corrector-type nonstandard finite difference scheme for an SEIR epidemic model," *Journal of Mathematics*, vol. 2020, no. 1, p. 8830829, 2020.
- [8] W. Gao, P. Veeresha, C. Cattani, C. Baishya, and H. M. Baskonus, "Modified predictor-corrector method for the numerical solution of a fractional-order SIR model with 2019-nCoV," *Fractal and Fractional*, vol. 6, no. 2, p. 92, 2022.
- [9] A. Salaudeen, S. Sathasivam, C. J. Nam, and T. Y. Hang, "Modelling the early outbreak of COVID-19 disease in Malaysia using SIRS model with 4-step Adams-Bashforth-Moulton predictor-corrector method," *Applied Mathematics and Computational Intelligence (AMCI)*, vol. 13, no. 2, pp. 121–136, 2024.
- [10] M. S. Arif, K. Abodayeh, and Y. Nawaz, "A third-order two-stage numerical scheme and neural network simulations for SEIR epidemic model: A numerical study," *Emerging Science Journal*, vol. 8, no. 1, pp. 326–340, 2024.
- [11] J. Cockayne, C. Oates, T. Sullivan, and M. Girolami, "Bayesian probabilistic numerical methods," *SIAM Review*, vol. 61, no. 3, pp. 756–789, 2019, doi: 10.1137/17m1139357.
- [12] K. E. Niemeyer and C. J. Sung, "GPU-based parallel integration of large numbers of independent ODE systems," in *Numerical Computations with GPUs*, V. Kindratenko, Ed. Cham: Springer,

2014, pp. 153–170, doi: 10.1007/978-3-319-06548-9_8.

- [13] N. Mohd Jamil, N. Rosli, and N. Muhammad, “Simulation of COVID-19 outbreaks via graphical user interface (GUI),” *Journal of Public Health Research*, vol. 11, no. 1, p. jphr-2021, 2022.
- [14] A. Harir, S. Melliani, H. E. Harfi, and L. S. Chadli, “Variational iteration method and differential transformation method for solving the SEIR epidemic model,” *International Journal of Differential Equations*, vol. 2020, no. 1, 2020.
- [15] B. J. Jansen, “The graphical user interface,” *ACM SIGCHI Bulletin*, vol. 30, no. 2, pp. 22–26, 1998.
- [16] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, “Graphical user interface (GUI) testing: Systematic mapping and repository,” *Information and Software Technology*, vol. 55, no. 10, pp. 1679–1694, 2013.
- [17] H. Laib, A. Bellour, and A. Boulmerka, “Taylor collocation method for a system of nonlinear Volterra delay integro-differential equations with application to COVID-19 epidemic,” *International Journal of Computer Mathematics*, vol. 99, no. 4, pp. 852–876, 2022.