

Simulation Study on Effects of Order and Step Size of Runge-Kutta Methods that Solve Contagious Disease and Tumor Models

Wang Z^{1*}, Wang Q¹ and Klinke DJ II^{2,3}

¹Department of Computer Science, Mathematics and Engineering, Shepherd University, Shepherdstown, WV 25443, USA

²Department of Chemical Engineering, Mary Babb Randolph Cancer Center, West Virginia University, Morgantown, WV 25606, USA

³Department of Microbiology, Immunology and Cell Biology, West Virginia University, Morgantown, WV 25606, USA

Abstract

Biological processes such as contagious disease spread patterns and tumor growth dynamics are modelled using a set of coupled differential equations. Experimental data is usually used to calibrate models so they can be used to make future predictions. In this study, numerical methods were implemented to approximate solutions to mathematical models that were not solvable analytically, such as a SARS model. More complex models such as a tumor growth model involve high-dimensional parameter spaces; efficient numerical simulation techniques were used to search for optimal or close-to-optimal parameter values in the equations. Runge-Kutta methods are a group of explicit and implicit numerical methods that effectively solve the ordinary differential equations in these models. Effects of the order and the step size of Runge-Kutta methods were studied in order to maximize the search accuracy and efficiency in parameter spaces of the models. Numerical simulation results showed that an order of four gave the best balance between truncation errors and the simulation speed for SIR, SARS, and tumor models studied in the project. The optimal step size for differential equation solvers was found to be model-dependent.

Keywords: Runge-Kutta methods; Simulation metrics; Adaptive evolutionary algorithms

Introduction

It has been observed that virus-originated diseases exhibited similar spread and control patterns [1,2]. These patterns have been researched extensively and a few mathematical models were proposed to theoretically analyze and predict the spread and control patterns of these diseases [1-8]. There are many factors involved in the models which are represented as parameters in the differential equations. In most cases it is impossible to obtain strict mathematical solutions to the equation sets. Researchers have tried to apply various approximation approaches to get solutions with minimal errors; among them numerical simulation is an excellent choice due to the rapidly increasing computing power and the introduction of artificial intelligence techniques these days. Different disease control scenarios such as the effect from various control patterns may be studied and simulated using numerical simulations. It was observed that by changing certain parameters to optimal values in the disease control process, the spreading of the virus could be minimized [7,8]. A classic model for contagious diseases is the SIR (Susceptible, Infected, and Removed) model that contains three differential equations [2]. The SIR model can describe many contagious diseases such as H1N1, SARS (Severe Acute Respiratory Syndrome), Ebola and Zika; however, it has the drawback of over simplifying patterns. Therefore, the SIR model is usually used to study the disease patterns as first-order approximations. A more complex model example of contagious diseases was the SARS model. SARS epidemic attacked Asia in 2003 leaving almost a thousand people dead and many sick. The disease had spread quickly before effective control methods were put into place. Once the quarantine and isolation methods, as well as strict health precautions, went into effect, the disease started to die out. During this process, it has been observed that SARS, like certain other virus-originated diseases, exhibited similar spread and control patterns. The patterns of the SARS virus have been studied and a few closely related models were proposed and studied [3-8]. In these models a group of coupled differential equations were used to describe the SARS virus spread and control process. A few characteristics of SARS virus were studied using the models. Many

factors, which are in the form of parameters, were involved in the models so it was impossible to obtain explicit mathematical solutions to the systems of nonlinear differential equations. Runge-Kutta method of order four and a numerical simulation approach were used to solve the equations under various circumstances. Different disease control circumstances such as effects from various patterns of quarantine rates have been studied and simulated [7]. A calibrated SARS model has the potential to give more accurate predictions on virus-originated disease spread and control patterns in the future. A tumor cell growth model usually involves coupled differential equations with a high dimensional parameter space, which is the result of many factors involved in the tumor growth dynamics [9]. Like contagious disease models, it is almost impossible to analytically obtain mathematical solutions to the equation set. Moreover, it is extremely time consuming to fit all parameters to experimental data with an acceptable error rate. Therefore, efficient numerical approximation methods need to be developed to solve the model and calibrate the model against experimental data. Various approximation techniques have been developed and implemented to get solutions with minimal errors for complex mathematical models [10-14]; among them certain AI (Artificial Intelligence) techniques are excellent choices because they are conceptually simple and easy to implement [10]. Among the search and optimization techniques in the evolutionary method category in AI, GAs (Genetic Algorithms) and the PSO (Particle Swarm Optimization) method were proved to be effective and efficient [11,13,14]. GAs were invented based on

*Corresponding author: Zhijun Wang, Department of Computer Science, Mathematics and Engineering, Shepherd University, Shepherdstown, WV 25443, USA, Tel: +13048765070; E-mail: zwang@shepherd.edu

Received August 30, 2016; Accepted September 22, 2016; Published September 30, 2016

Citation: Wang Z, Wang Q, Klinke DJ II (2016) Simulation Study on Effects of Order and Step Size of Runge-Kutta Methods that Solve Contagious Disease and Tumor Models. J Comput Sci Syst Biol 9: 163-172. doi:10.4172/jcsb.1000234

Copyright: © 2016 Wang Z, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the natural evolution process while the PSO method was inspired by the social behaviour of bird flocking or fish schooling, the collective intelligent behaviours of individuals who interact with each other and their environment. Both GAs and the PSO method help to calibrate the model to fit experimental data within tolerable error ranges in a reasonable time frame. They have been successfully applied in many areas including artificial neural network training, fuzzy systems, and optimization problems. Parameter space search in tumor models is a NP-hard optimization problem, which requires a computationally tractable algorithm to solve. GAs and PSO with Runge-Kutta solvers provide a good solution to these models. The difference between GAs and the PSO method is that no crossover or mutation operation is involved in the PSO search process, while GAs simulate the natural evolution and selection process with crossover and mutation operations. The rest of this paper is organized as follows. Next section describes mathematical models for SARS and tumor growth dynamics. Three following sections give details of the simulation method, demonstrate the simulation setup, and show the simulation results of Runge-Kutta methods with various orders and step sizes. The final section concludes the paper and discusses possible future work.

Related Work

The simulation for SARS spread and control patterns was based on a previously proposed model [4]. The model subdivides a given SARS-affected region into six mutually exclusive subpopulations and it consists of six differential equations 1-6.

$$\frac{dS}{dt} = \Pi - \frac{s(\beta I + \epsilon_E \beta E + \epsilon_Q \beta Q + \epsilon_J \beta J)}{N} - \mu S \quad (1)$$

$$\frac{dE}{dt} = p + \frac{s(\beta I + \epsilon_E \beta E + \epsilon_Q \beta Q + \epsilon_J \beta J)}{N} - (\gamma_1 + \kappa_1 + \mu) E \quad (2)$$

$$\frac{dQ}{dt} = \gamma_1 E - (\kappa_2 + \mu) Q \quad (3)$$

$$\frac{dI}{dt} = \kappa_1 E - (\gamma_2 + \delta_1 + \sigma_1 + \mu) I \quad (4)$$

$$\frac{dJ}{dt} = \gamma_2 I + \kappa_2 Q - (\sigma_2 + \delta_2 + \mu) J \quad (5)$$

$$\frac{dR}{dt} = \sigma_1 I + \sigma_2 J - \mu R \quad (6)$$

From equations 1-6, the total population at time t , denoted by $N(t)$, is given by 7.

$$N(t) = S(t) + E(t) + Q(t) + I(t) + J(t) + R(t) \quad (7)$$

There are fifteen parameters involved in the model [4]. By dividing the whole population (N) into six subpopulations, which are susceptible (S), asymptotically infected (E), quarantined (Q), symptomatic (I), isolated (J) and recovered (R) individuals, the spread and control patterns of SARS can be modelled with the differential equations using ratios and the probability of one subpopulation moving to others. Figure 1 shows the subpopulations and how they flow to each other in the model.

Solving the model with an appropriate parameter set can lead

to finding when an asymptotically stable disease-free equilibrium is reached. The solution then can determine when the disease can stop spreading or can be put into control. The simulation results can also be used to study the spread and control patterns of other virus-originated diseases.

A more complex biological process simulated in the project is tumor cell population growth, which is described by a three-compartment model and mathematically modelled using a group of differential equations. Among the three compartments, the dynamics of nine state variables that are regulated by the governing biological processes were considered [9]. To deal with the practically insolvable high-dimensional parameter space in the model, besides efficient numerical differential equation solvers, evolutionary methods such as GAs and the PSO techniques were implemented to speed up the search process in the multi-dimensional parameter space to find optimal parameter values that fit experimental data from mice cancer cells [9,14]. The fitness function, which measures the difference between calculated results and experimental data, was minimized in the parameter space search process.

Numerical Solutions

There are a few popular numerical methods that solve ordinary differential equations with various speeds and accuracy. Euler method, backward Euler, Runge-Kutta, Adams-Bashforth, Adams-Moulton, and backward differentiation were implemented as equation solvers to solve the models within GAs or the PSO method. Among them Runge-Kutta methods form a family of numerical methods that offer a good balance between efficiency and accuracy for the mathematical models studied. Since equations are solved for every parameter set in every generation in an evolutionary technique, the equation solver is a speed bottleneck in the simulation process. Both implicit and explicit versions of Runge-Kutta methods [15,16] were considered. The explicit versions of Runge-Kutta methods could be implemented in a fairly efficient manner thus became a desired equation solver for the models.

In SARS and tumor models studied in the project, all observed function values, such as infected subpopulations and the adenovirus in lymph node, change with time. Therefore, the variable for all functions is the time t . Assume an initial value problem as specified in equation 8.

$$\dot{y} = f(t, y), y(t_0) = y_0 \quad (8)$$

The general form of Runge-Kutta methods of s -stage can be defined by

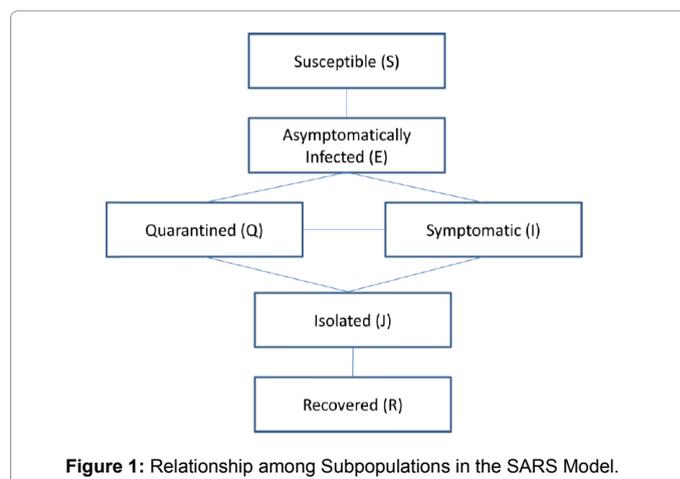


Figure 1: Relationship among Subpopulations in the SARS Model.

$$y_{n+1} = y_n + hg(t_n, y_n, h) \quad (9)$$

where h is the step size and the function g is the approximate gradient that is obtained by

$$g(t_n, y_n, h) = \sum_{i=1}^s c_i k_i \quad (10)$$

With

$$k_i = f\left(t + hw_i, y + h \sum_{j=1}^s b_{ij} k_j\right) \quad (11)$$

And

$$w_i = \sum_{j=1}^s b_{ij} \quad (12)$$

In simulations, the explicit versions of Runge-Kutta methods were used for efficiency purposes. Explicit Runge-Kutta methods satisfy condition 13.

$$j \geq i \rightarrow b_{ij} = 0 \quad (13)$$

The Euler method can be treated as the Runge-Kutta method of order one. An example of an explicit four-stage Runge-Kutta method is specified by equations 14-19, which is one of the four-stage methods implemented in the simulation.

$$t_{j+1} = t_j + h \quad (14)$$

$$y_{j+1} = y_j + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4) \quad (15)$$

$$k_1 = f(t_j, y_j) \quad (16)$$

$$k_2 = f\left(t_j + \frac{h}{3}, y_j + \frac{hk_1}{3}\right) \quad (17)$$

$$k_3 = f\left(t_j + \frac{2h}{3}, y_j + h\left(-\frac{k_1}{3} + k_2\right)\right) \quad (18)$$

$$k_4 = f(t_j + h, y_j + h(k_1 - k_2 + k_3)) \quad (19)$$

Besides simulation speed, simulation error is another important metric. Runge-Kutta methods had been well analyzed in terms of local and global truncation errors. Assume $y(x)$ is the exact solution of the initial value problem, the local truncation error of the Runge-Kutta methods is defined by

$$error = y(t_{n+1}) - y(t_n) - hg(t_n, y_n, h) \quad (20)$$

The order of a Runge-Kutta method is p if p is the greatest number so that

$$error = O(h^{p+1}) \quad (21)$$

The global truncation error is one order lower [16]. Therefore, the step size h is also an important variable for local and total truncation errors. Systematic tests were conducted in simulations to study the relationship between the step size and simulation metrics such as accuracy and speed. Among all Runge-Kutta methods, Runge-Kutta method of order four has been most frequently used to numerically solve coupled differential equations [17]. Assume that $f(t, y)$ is continuous and satisfies a Lipschitz condition in the variable y , and consider an initial value problem 8, the Runge-Kutta method of order four uses the formulas 9 and 10 to approximate the solution to the differential equation using a discrete set of points. Parameters k_1, k_2, k_3 , and k_4 are calculated using formulas similar to the above implemented four-stage method but with different coefficients [14].

Beside Runge-Kutta methods of order four, Euler method,

order-six and order-eight Runge-Kutta methods were also studied in simulations due to a concern of accuracy of the order-four method raised by researchers for the SARS model simulation [7]. All Runge-Kutta methods implemented in simulation programs are defined by formulas 8-12.

Most of evolutionary techniques such as GAs and the PSO method follow a standard procedure with four steps [10]. In this study, adaptive versions were proposed and implemented to maximize the search effectiveness and efficiency. The procedure is as follows.

Step 1. Randomly generate an initial population based on the problem nature and complexity.

Step 2. Calculate and use fitness function values to rank individuals in the first generation.

Step 3. Reproduce the population based on fitness values. Adaptive methods such as dynamic crossovers and mutations are used in GAs. Local bests are used for the PSO method. Calculate and use fitness values to rank individuals in the new generation.

Step 4. If requirements are met or the maximum number of iterations is reached, stop and report results. Otherwise go back to Step 3.

Numerical Simulation

Simulation design

The simulation engine is composed of related classes that implement the search algorithms such as adaptive GAs and the PSO method. Essential modules in the simulation engine include

- i. Numerical equation solver such as a version of Runge-Kutta methods.
- ii. Evolutionary component such as Crossover and Mutation classes for GAs.
- iii. Experimental data processing and comparison component.
- iv. User input, output, and event-handling component.
- v. Exception handling features to handle unexpected run-time errors.
- vi. File output feature that stores results in files for further data mining.

In SIR and SARS simulators, Graphical User Interfaces (GUIs) were designed and implemented for the simulation engines so equation parameters can be changed when the simulation is running and results are shown intuitively as functional curves on the screen, which much facilitate the research process using the models. The simulators with easy-to-use graphical user interfaces may be used by various groups of researchers to experiment with different aspects of virus-originated disease patterns. In the SARS simulator, functions were implemented to cause perturbations to variables and parameters such as quarantine rates for subpopulations. Therefore, the effects of control methods like quarantine rate control can be studied conveniently.

Implementation and validation of simulators

SIR model was implemented using the original version of GAs. The basic architecture is shown in Figure 2. The basic class is the Chromosome class, which is used to instantiate parameter set objects. The Reproduce class contains crossover, mutation, and selection methods, which form the core of GAs. The SIR model, which has three

differential equations, was implemented in the SIR Model class. Pair class defines different ways to pair chromosomes to generate the next generation in GAs.

SIR model and SARS model simulators are shown in figures in the next section with simulation results. GUI components were designed and implemented for both SIR and SARS simulation engines so state variable initial values, parameter ranges, and simulation control variables can be specified directly on the interface.

For the tumor growth dynamic models, adaptive versions of GAs were designed and implemented. Figure 3 shows the simulator interface with GAs and different equation solvers. Besides variable initial values and parameter ranges, step size and GA control parameters can also be specified. Adaptive GAs are able to change the number of crossover points, use random crossover points, use random mutation rates, and dynamically evolve based on the simulation progress, which is evaluated by the best fitness function value achieved at the moment. Adaptive features introduced in GAs, when combined with Runge-Kutta methods of the appropriate order and step size, much facilitated the parameter search process. The simulator using another evolutionary method, the PSO technique, is quite similar to the one using GAs [14].

To conduct an initial check to the simulators, a systematic list of test cases was constructed to test the special cases and extreme cases. After each simulation run, the simulation results from the text-based file outputs are obtained and checked against the expected results. For each test case, the generated results should match the expected results. This systematic procedure validates the simulator to an extent. The more general cases, however, cannot be examined because no analytical solution can be obtained for the models. Numerical error analysis is used for these scenarios.

Simulation results

The simplest model among the three, the SIR model, was examined first. Figure 4 shows a typical SIR simulator interface with numerical

results from the Runge-Kutta method of order four and a step size of one day. Due to its simplicity, SIR model is insensitive to the step size variable, when considering the scale of error measured by the variance between experimental data and calculated results. The solution does not change significantly as a function of the order of Runge-Kutta methods. However, Runge-Kutta method of order eight is desirable if more accurate experimental data becomes available in future. There are only three equations in the model so a solution can be calculated in a reasonable time frame using a higher order of Runge-Kutta methods.

Besides the simple SIR model, a systematic study on the SARS spread and control pattern was conducted using simulation. Disease control scenarios such as the effects from various patterns of quarantine rates were studied. The simulation results show that by moving people between pairs of subpopulations under certain rates, the SARS virus can be put under control more quickly. The results could also provide guidance to the control measure of diseases with similar patterns [7].

The cases without subpopulation exchanges were simulated first. Figure 5, 6 and 7 show that subpopulation functions E, Q, I, J, and R change with time with step sizes of 1 day, 0.5 day, and 0.1 day, respectively. All three figures were calculated using the same set of parameter values in the SARS model. It was observed that curves were quite similar, including intercept patterns between curves, with different step sizes after ten different step sizes were examined. Minor differences showed up when the results were compared numerically. The calculated variable values with a step size of 0.1 day were set to 1.0 and used as units to compare the average relative differences among variables as a function of step size. The results are shown in Figure 8. It was observed that the changes of variables were negligible when step size was less than 0.1 day for the SARS model, when compared with the error scale between experimental data and calculated results.

Typical results for more complex scenarios from the SARS model simulation are shown in Figures 9 and 10. Figure 9 shows that if people are moved by 20% from E to Q and from Q to J, I and J approach

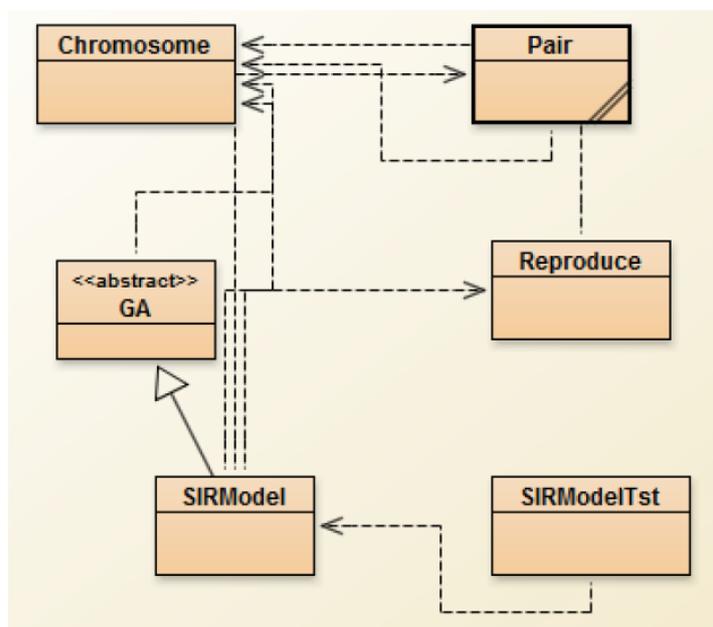


Figure 2: Implementation of SIR Model Solver using Genetic Algorithm.

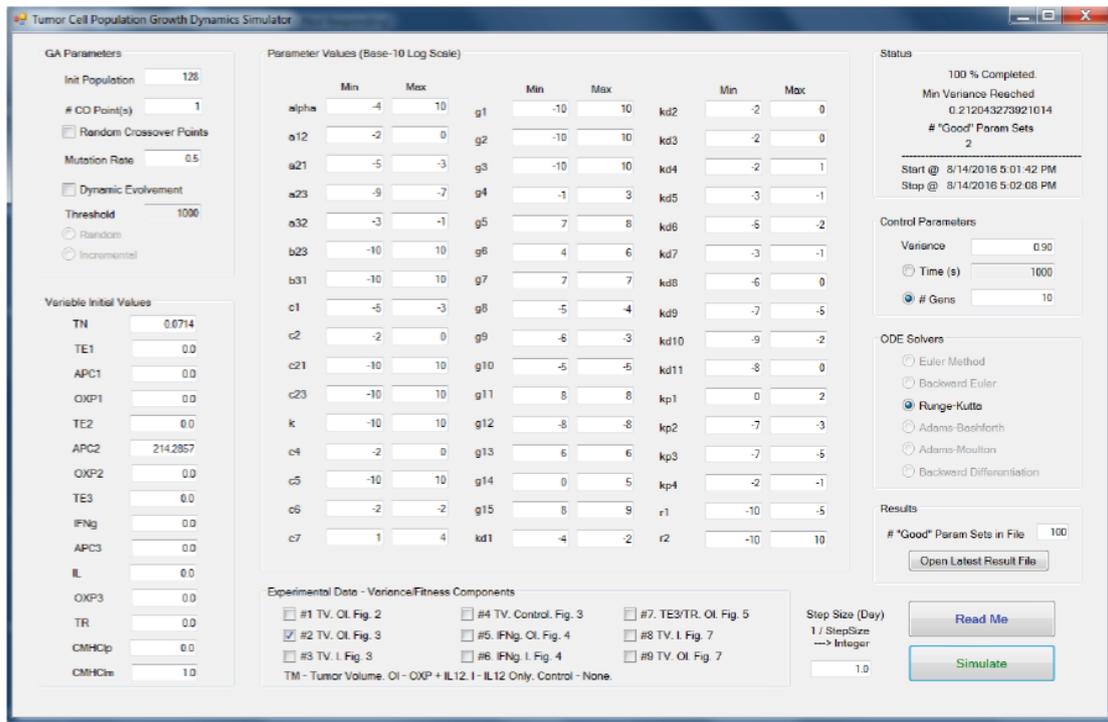


Figure 3: Simulator with Variable Initial Values, Parameter Ranges, and Genetic Algorithm Control Variables.

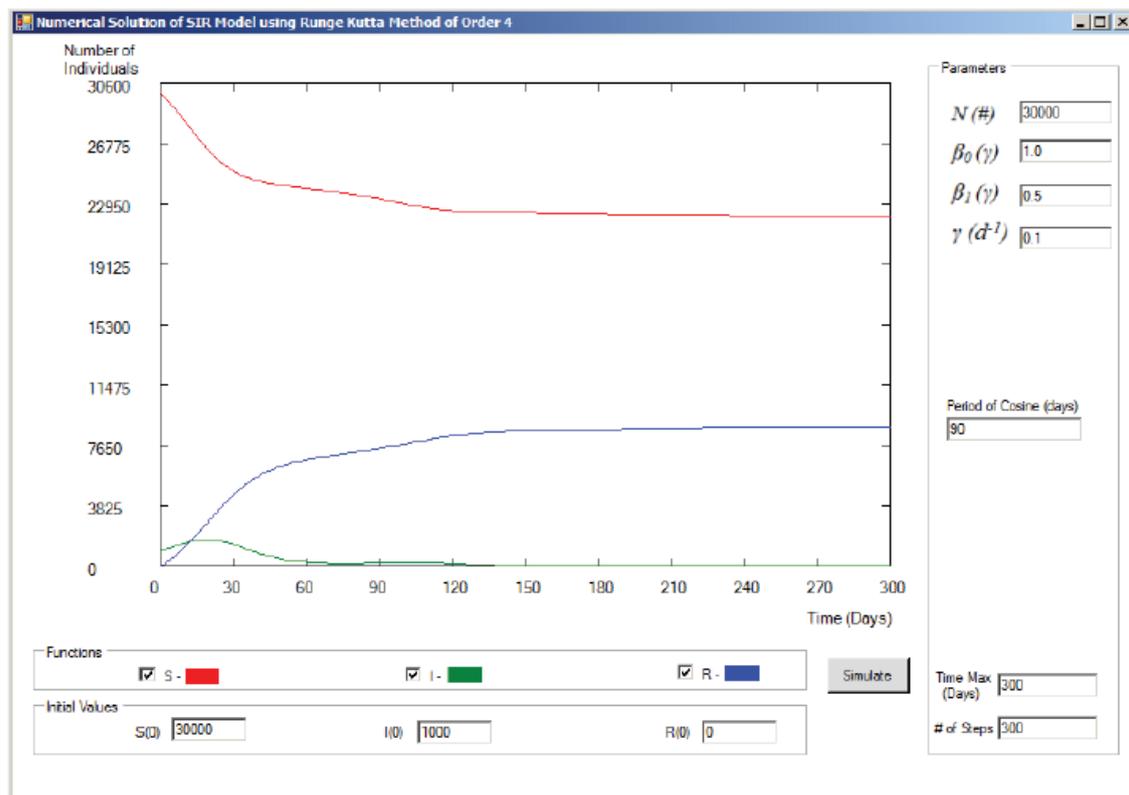


Figure 4: Numerical Results from Runge-Kutta Method of Order Four for the SIR model with a Step Size of 1 Day.

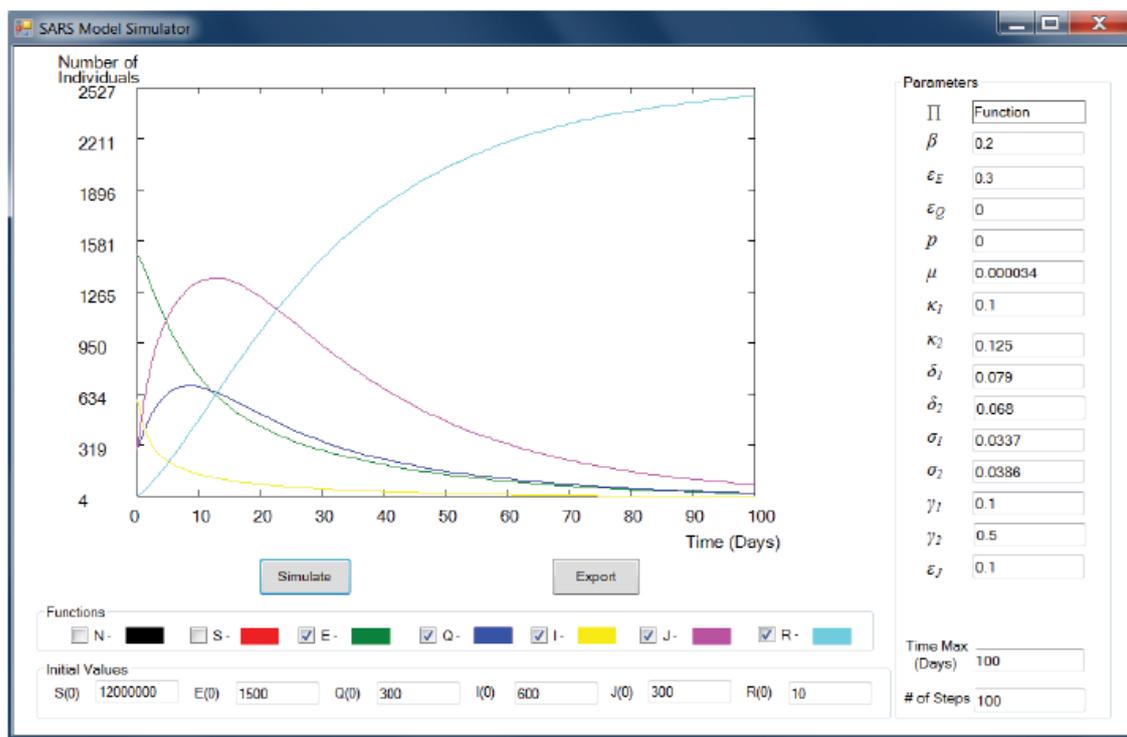


Figure 5: Subpopulation Functions E, Q, I, J and R without Impulsive Control with a Step Size of 1 Day.

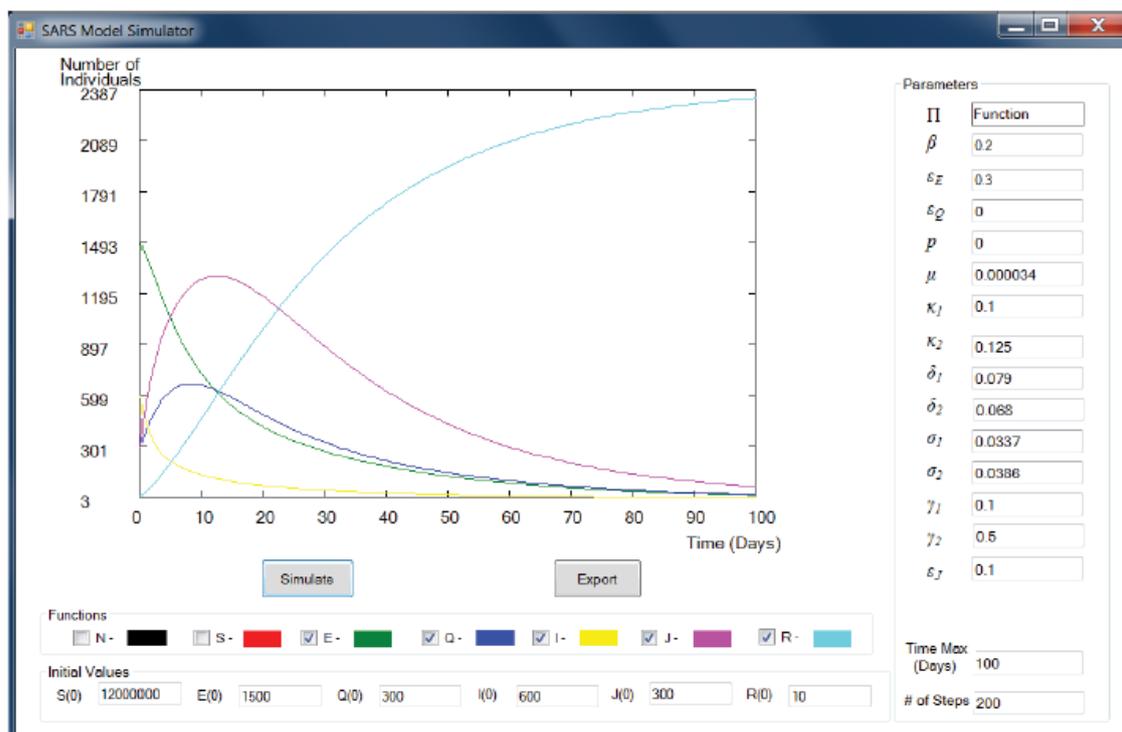


Figure 6: Subpopulation Functions E, Q, I, J, and R without Impulsive Control with a Step Size of 0.5 Day.

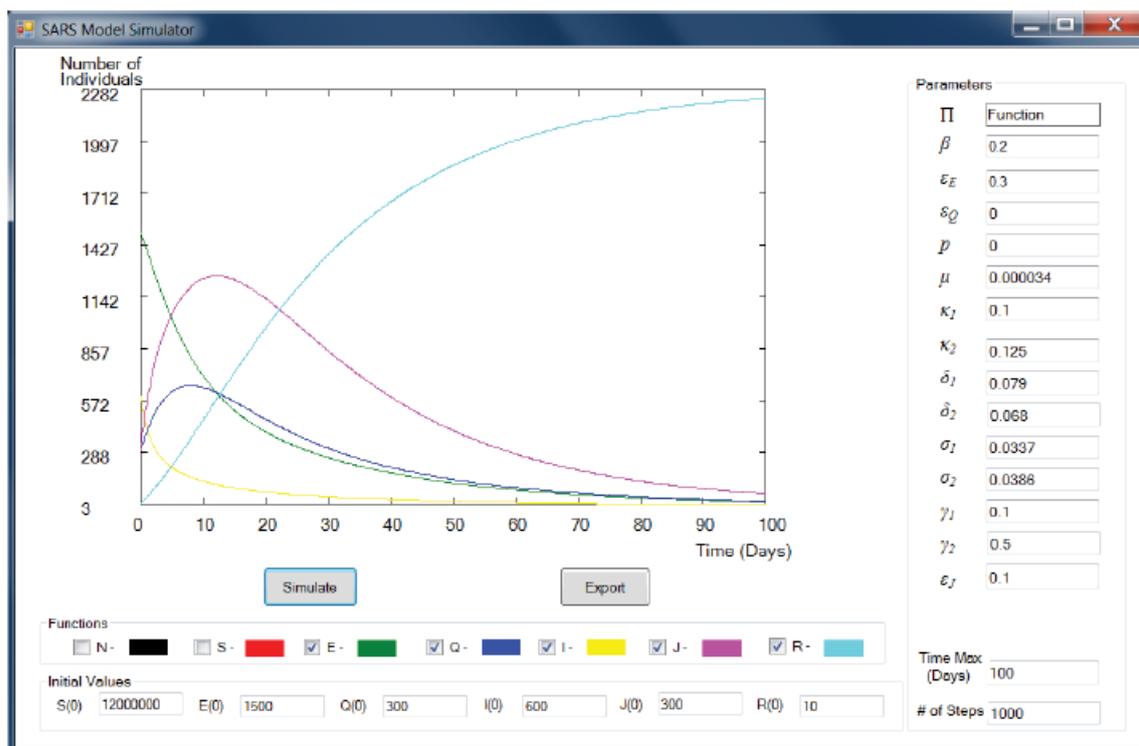


Figure 7: Subpopulation Functions E, Q, I, J and R without Impulsive Control with a Step Size of 0.1 Day.

zero much faster than not having the movements among the four subpopulations. Thus the disease dies out faster under this set of control methods. A step size of 0.2 day was used for Figure 9 and 0.1 day was used for Figure 10. The movement between two subpopulations was implemented using a perturbation to both subpopulations. In real life, this can be done by manually moving people between subpopulations. After the perturbations were added, the effect of different step sizes and orders of Runge-Kutta methods were studied systematically like the impulse-free cases. Numerical results showed that Figure 8 also applied to the cases with impulsive controls. Therefore, the best practical step size for this specific SARS model is 0.1 day.

The difference between simulated order-four and order-eight Runge-Kutta methods can be safely neglected since the average error between experimental data and calculated results was one order greater than the difference. The results on the optimal step size and the optimal order of Runge-Kutta methods can be used to set good control parameter values to optimize the simulation process by saving computing time and increasing accuracy of calculated results for the SARS model and similar models for contagious diseases.

Tumor growth dynamic models were simulated extensively using GAs and the PSO method with various orders of Runge-Kutta methods and step sizes. Among all nine state variables, the total tumor volume is an important measurement for tumours. The tumor volume is a combination of the results from two variables, which are the MHC class I positive tumor cells volume C_{MHC1}^+ and the MHC class I negative tumor cells volume C_{MHC1}^- . Figure 11 shows the comparison between the numerical simulation result and three samples of experimental data for the total tumor volume. Runge-Kutta method of order eight and a step size of 0.2 day were used in Figure 11.

According to Figure 11, the results for the tumor volume are almost the same for the order four [14] and the order-eight Runge-Kutta methods. The difference between different parameter sets for the Runge-Kutta methods with the same order is negligible compared with the truncation error. Among all variables, naïve CD8⁺ T cells, effector CD8⁺ T cells in lymph node, effector CD8⁺ T cells in blood, and interferon gamma have an average of 5% or less difference between order-four and order-eight Runge-Kutta methods, while tumor volume, adenovirus in lymph node, and tumor necrosis factor α have less than 2% of difference on average. Effector CD8⁺ T cells in tumor microenvironment gives the highest difference between order-four and order-eight Runge-Kutta methods, which is around 10% on average.

Table 1 shows the average percentage differences of simulation results for order one, order six, and order eight Runge-Kutta methods, using the calculated variable values from Runge-Kutta method of order four as units, with a step size of 0.2 day and a simulation length of 1,000 generations in GAs. Crossover points were dynamic and mutation rate adjusted itself according to fitness of the population. Initial population was generated randomly according to estimated ranges. A random number was generated for a gene in the range of two neighbour genes, instead of using a random number in the whole permitted parameter range. With the exception of the Euler method, the differences shown in the table are insignificant, when compared with the difference between the calculated results and experimental data. Since Runge-Kutta method of order four is the fastest, it was used as the major equation solver in simulators. For this specific tumor model, a step size of 0.2 day gives the best balance between solution accuracy and simulation speed, which is different from the optimal step size for the SARS model discussed before.

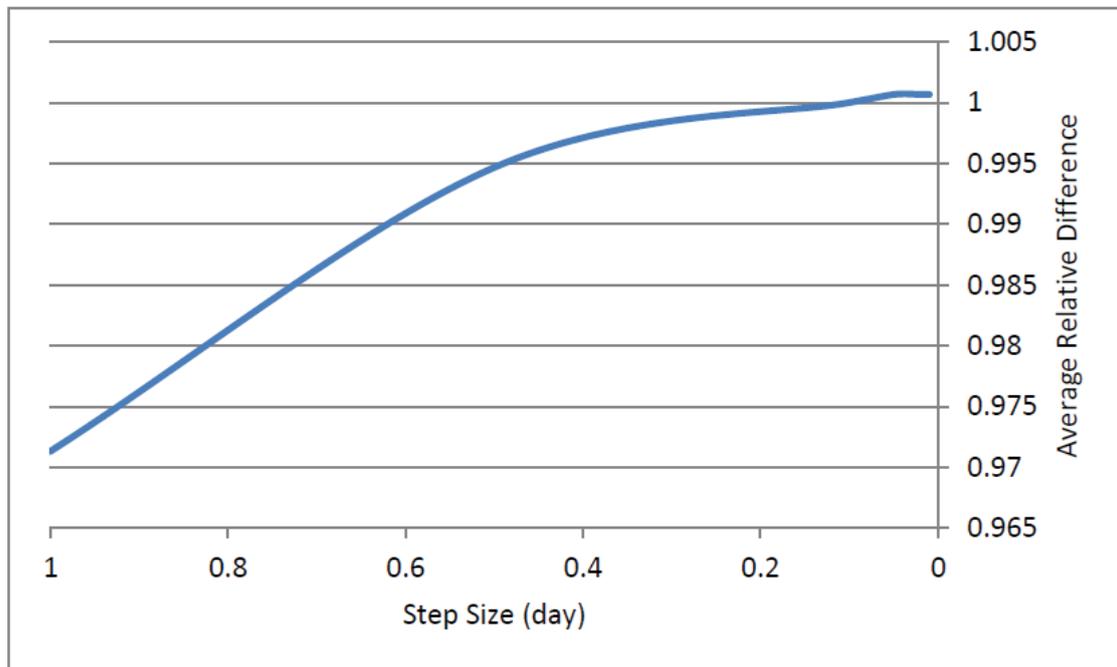


Figure 8: Average Variable Difference as a Function of Step Size.

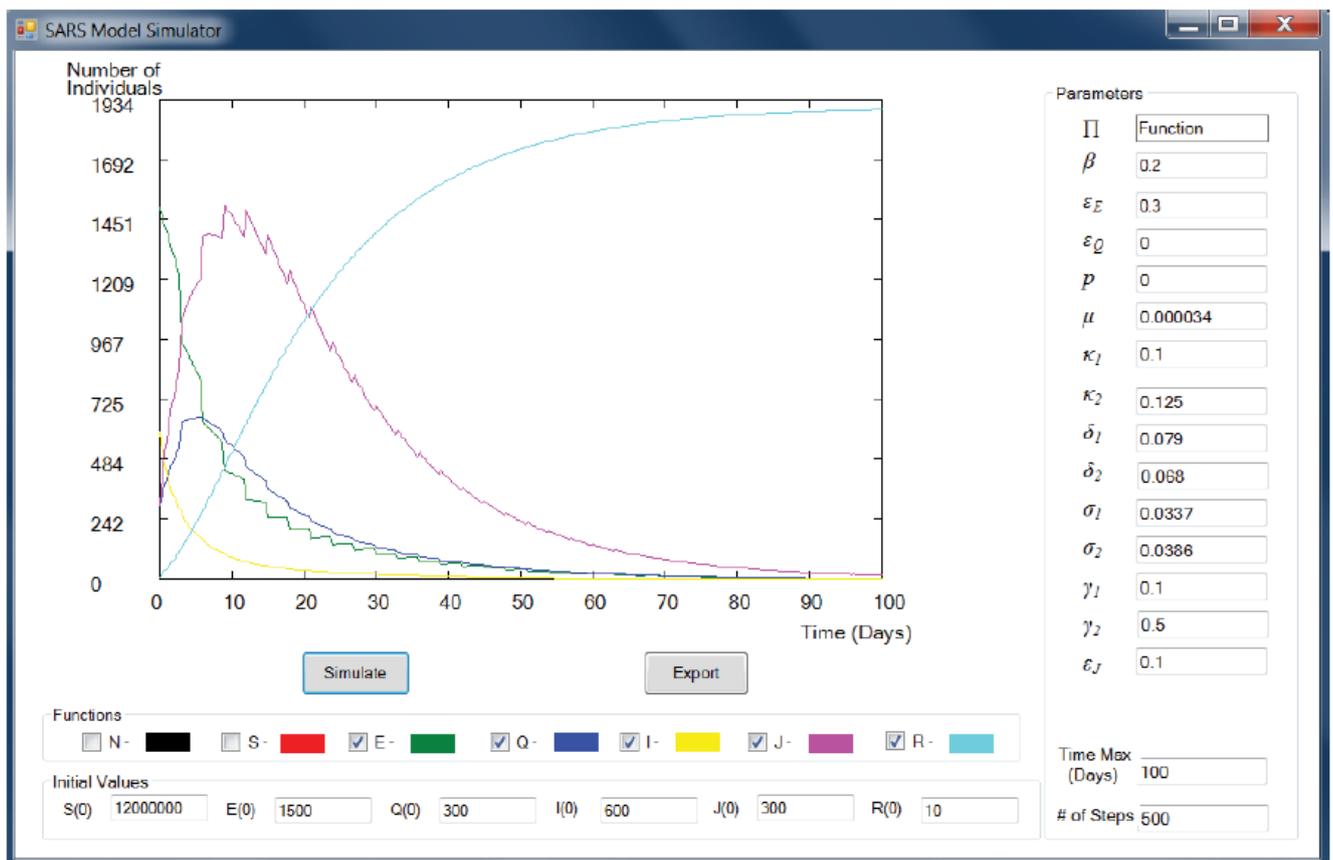


Figure 9: Subpopulation Functions E, Q, I, J, and R with Impulsive Control (20% of E is moved to Q, 20% of Q is moved to J every 5 days). Step size=0.2 day.

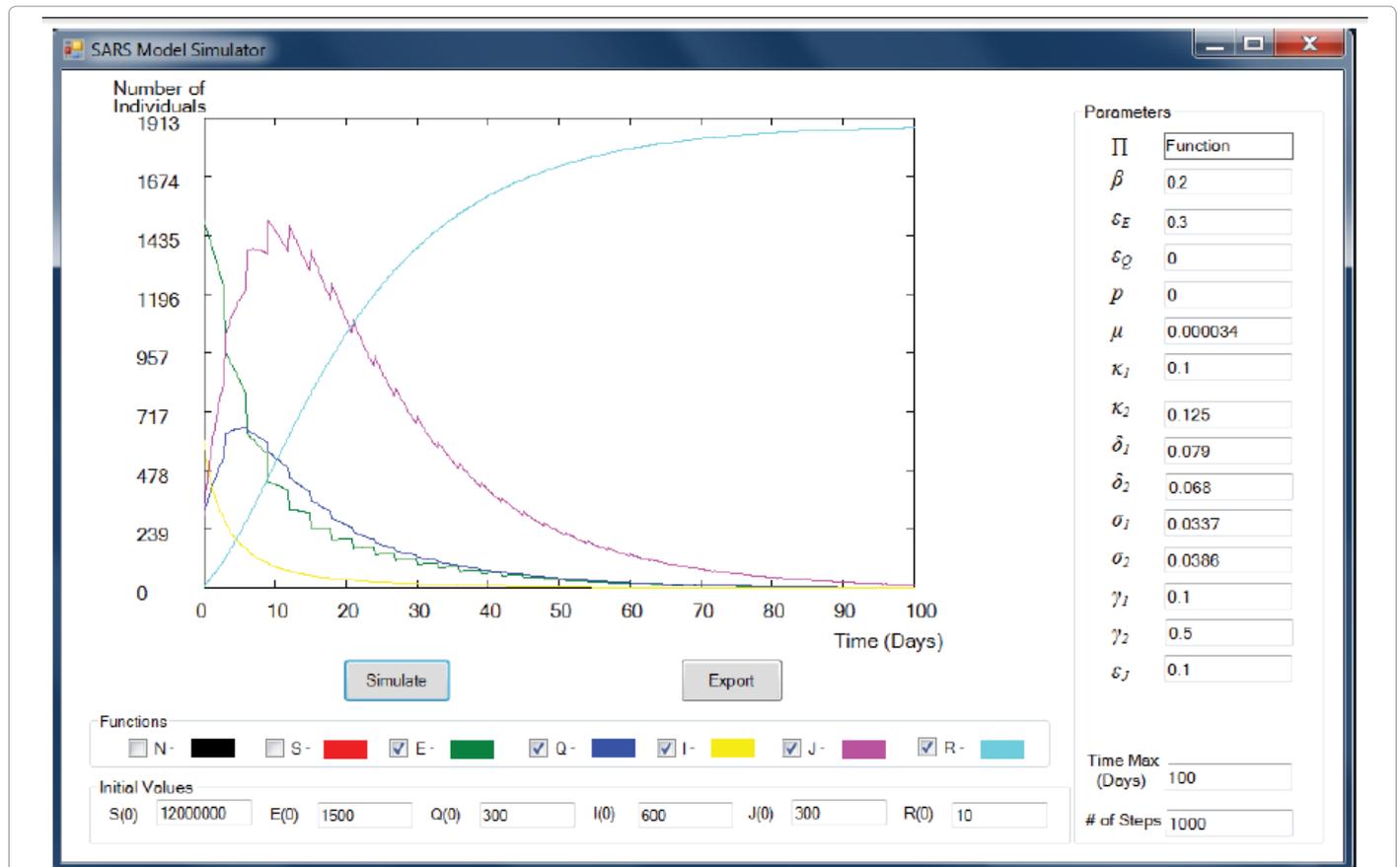


Figure 10: Subpopulation Functions E, Q, I, J, and R with Impulsive Control (20% of E is moved to Q, 20% of Q is moved to J every 5 days). Step size=0.1 day.

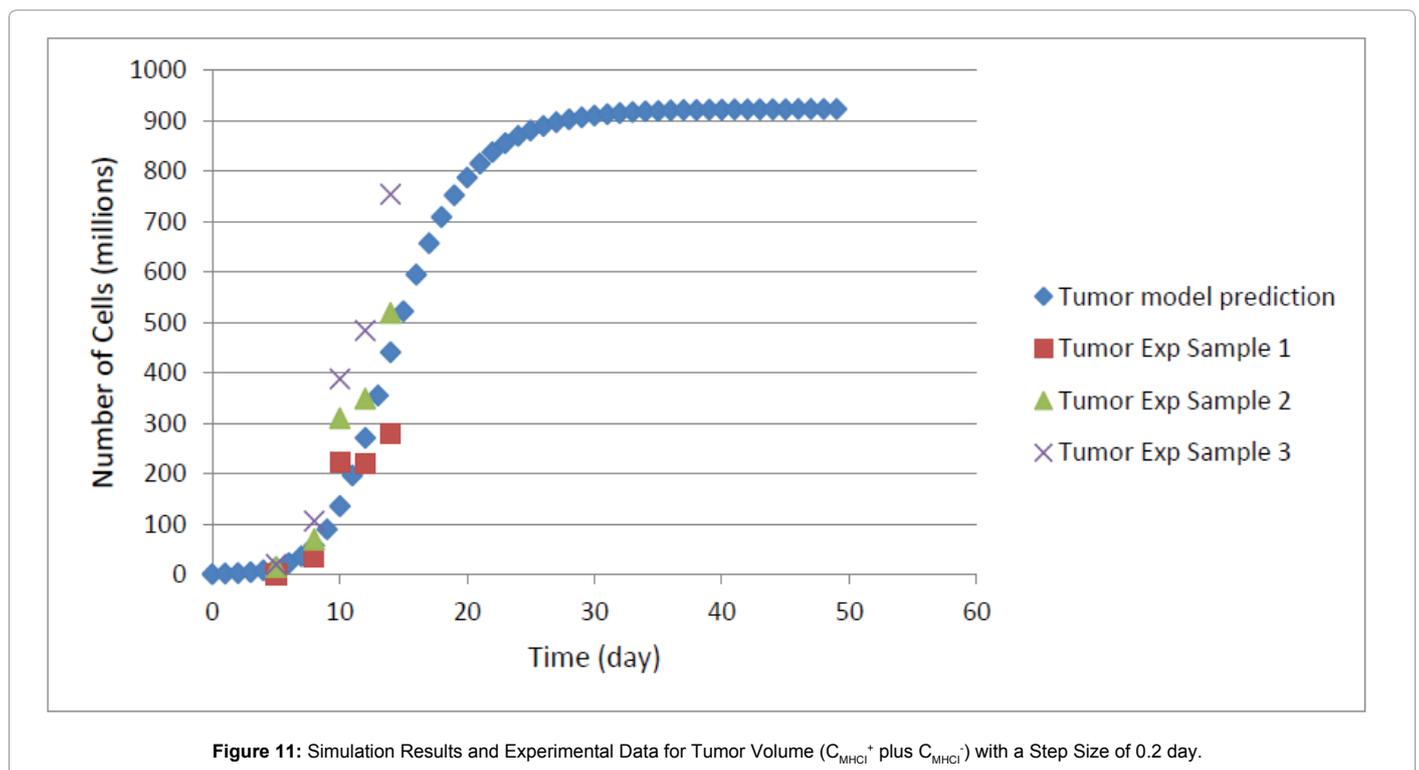


Figure 11: Simulation Results and Experimental Data for Tumor Volume (C_{MHC1^+} plus C_{MHC1^-}) with a Step Size of 0.2 day.

Variable Description	Symbol	Euler (%)	RK6 (%)	RK8 (%)
Naïve CD8 ⁺ T cells	T_N	22.1	4.1	5
Effector CD8 ⁺ T cells in lymph node	T_{E1}	17.5	3.5	4.7
Adenovirus in lymph node	LV	11.7	0.9	1.2
Effector CD8 ⁺ T cells in blood	T_{E2}	13.3	3.9	4.5
MHC class I positive tumor cells	C_{MHC1}^+	10.8	0.7	1.2
MHC class I negative tumor cells	C_{MHC1}^-	15.4	1.3	1.7
Effector CD8 ⁺ T cells in tumor	T_{E3}	22.6	7.7	11.6
Interferon gamma	IFN γ	13.7	2.5	3.8
Tumor Necrosis Factor α	TNF α	8.8	1.6	1.8

Table 1: Average Differences Relative to Results from Runge-Kutta Method of Order Four for Runge-Kutta Methods of Orders One (Euler), Six, and Eight with a Step Size of 0.2 day.

Conclusion and Future Work

A mathematical model for SARS was simulated to study the effect of different quarantine and isolation control methods on the disease subpopulation dynamics. The simulation results could provide guidance to work with spread and control patterns of SARS as well as other similar virus-originated diseases. GAs and the PSO method were identified as efficient optimization techniques for models with high-dimensional parameter spaces. The simulation results from a tumor cell growth dynamics model showed their effectiveness and efficiency. An essential component in simulation programs is the differential equation solver. Among all numerical solvers, Runge-Kutta methods, which could provide a balance between solution accuracy and search speed, were proved to be the most effective and efficient for models studied in the project. It was found that Runge-Kutta method of order four was the most effective method in the Runge-Kutta method family, in terms of models examined. The optimal step size depends on the specific model in the study. A possible future work is to simulate other mathematical models using a systematic group of orders and step sizes for Runge-Kutta methods and possibly develop a uniform theory that could provide guidance for selection of appropriate orders and step sizes for different models.

Acknowledgements

This study was supported by the National Institute of General Medical Sciences of the National Institutes of Health (NIH) grant as part of the West Virginia IDeA Network of Biomedical Research Excellence (P20GM103434).

References

- Arino J, Brauer F (2006) Simple models for containment of a pandemic. *J R Soc Interface* 3: 453-457.
- Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. *Proc Royal Society A: Math Phy Eng Sci* 115: 700.
- Colizza V, Barrat A, Barthelemy M, Vespignani A (2007) Predictability and epidemic pathways in global outbreaks of infectious diseases: the SARS case study. *BMC Medicine* 5: 34-46.
- Gumel AB, Ruan S (2004) Modeling control strategies for controlling SARS outbreaks. *Proc Biol Sci* 271: 2223-2232.
- Hsu S, Roeger L (2007) The final size of a SARS epidemic model without quarantine. *J Math Anal App* 333: 557-566.
- McLeod R, Brewster J, Gumel A, Slonowsky D (2008) Sensitivity and uncertainty analyses for a SARS model with time-varying inputs and outputs. *Math Biosci Eng* 3: 527-544.
- Wang Z, Wang Q (2012) Simulation of SARS model with impulsive control describing spread and control patterns of SARS virus. *Proceed Inter Conf Dynamic Syst Appl* 6: 449-454.
- Yan X, Zou Y (2008) Optimal and sub-optimal quarantine and isolation control in SARS epidemics. *Math Comp Model* 47: 235-245.
- Wang Q, Klinke DJ, Wang Z (2015) CD8⁺ T cell response to adenovirus vaccination and subsequent suppression of tumor growth: modeling, simulation and analysis. *BMC Syst Biol* 9: 27.
- Luger G (2008) Artificial intelligence: structures and strategies for complex problem solving. 6th edn. Addison Wesley, UK.
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. *Proceed IEEE Inter Conf Neural Networks Vol. IV*: 1942-1948.
- Klinke DJ (2009) An empirical Bayesian approach for model-based inference of cellular signaling networks. *BMC Bioinfo*.10: 371.
- Rubner J, Schulten K (1990) Development of feature detectors by self-organization. *Biol Cybernet*. 62: 193-199.
- Wang Z, Wang Q (2016) Numerical simulation of a tumor growth dynamics model using particle swarm optimization. *J Comput Sci Syst Biol* 9: 001-005.
- Butcher JC (1975) A stability property of implicit Runge-Kutta methods. *BIT Num Math* 15: 358-361.
- Butcher JC (2008) Numerical methods for ordinary differential equations. 2nd edn. John Wiley & Sons, USA.
- Tan D, Chen Z (2012) On a general formula of fourth order Runge-Kutta method. *J Math Sci Math Edu* 7: 1-10.