



A Robust and Efficient Numerical Method for RNA-Mediated Viral Dynamics

Vladimir Reinhartz¹, Alexander Churkin², Harel Dahari³ and Danny Barash^{1*}

¹ Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel, ² Department of Software Engineering, Sami Shamoon College of Engineering, Beer-Sheva, Israel, ³ Program for Experimental and Theoretical Modeling, Division of Hepatology, Department of Medicine, Loyola University Medical Center, Maywood, IL, United States

OPEN ACCESS

Edited by:

Henri Orland,
UMR3681 Institut de Physique
Théorique (IPhT), France

Reviewed by:

Marc Delarue,
Institut Pasteur, France
Frédéric Poitevin,
Stanford University, United States
Graziano Vernizzi,
Siena College, United States

*Correspondence:

Danny Barash
dbarash@cs.bgu.ac.il

Specialty section:

This article was submitted to
Mathematics of Biomolecules,
a section of the journal
Frontiers in Applied Mathematics and
Statistics

Received: 29 June 2017

Accepted: 18 October 2017

Published: 31 October 2017

Citation:

Reinhartz V, Churkin A, Dahari H and
Barash D (2017) A Robust and
Efficient Numerical Method for
RNA-Mediated Viral Dynamics.
Front. Appl. Math. Stat. 3:20.
doi: 10.3389/fams.2017.00020

The multiscale model of hepatitis C virus (HCV) dynamics, which includes intracellular viral RNA (vRNA) replication, has been formulated in recent years in order to provide a new conceptual framework for understanding the mechanism of action of a variety of agents for the treatment of HCV. We present a robust and efficient numerical method that belongs to the family of adaptive stepsize methods and is implicit, a Rosenbrock type method that is highly suited to solve this problem. We provide a Graphical User Interface that applies this method and is useful for simulating viral dynamics during treatment with anti-HCV agents that act against HCV on the molecular level.

Keywords: hepatitis C virus, multiscale model, age-structured model, RNA-mediated viral dynamics, partial differential equations, numerical solution, Rosenbrock method

1. INTRODUCTION

Approximately 71 million people worldwide are affected by chronic hepatitis C viral (HCV) infection, which is the primary cause of liver cirrhosis, liver cancer and liver transplant [1]. Approximately 400,000 people die each year from HCV, mostly from cirrhosis and hepatocellular carcinoma [2]. There is no vaccine for HCV and for more than a decade the standard-of-care of pegylated interferon-alpha (IFN) and ribavirin was suboptimal [3]. However the recent advent of direct-acting antivirals (DAAs) allows for interferon-free, all-oral treatment yielding cure rates exceeding 90% with pangenotypic activity and shorter durations of therapy (8–24 weeks) compared to IFN-based therapy (24–48 weeks [4]). While these highly effective DAAs are considered one of the greatest achievements in medicine, significant challenges remain for eliminating HCV infection such as finding an optimal approach to current DAA failures, preventing re-infection, identifying all those infected and the high cost of the new DAAs which represents a major barrier to treating the populations that are most affected by HCV [3]. Thus, there exists a continuing need for more affordable therapies, as well as an effective vaccine [5].

Mathematical models are valuable tools for understanding the *in vivo* serum dynamics of viruses that trigger both persistent infection (e.g., HIV-1 [6–9], hepatitis B virus [10–12], hepatitis D virus [13–15], Theiler murine encephalomyelitis virus [16], herpes simplex virus [17] and HCV [18–20]) and acute infection (e.g., influenza A [21–23] and ebola [24]). Mathematical modeling is also improving our understanding of intracellular viral genome dynamics [25–28] and the quantitative events that underlie the immune response to pathogens [6, 9]. The standard model for HCV kinetics during treatment provided many insights into the effectiveness and mechanism of action of IFN and ribavirin (reviewed in [29, 30]). This model has been able to retrospectively predict the duration of treatment needed for HCV eradication (cure) [31–35] and more recently

was used in real-time (on treatment) to predict the duration of therapy needed to achieve cure with an IFN-free regimen of silibinin +ribarivirin [36]. In the age of DAAs, new models are being developed to meet the challenge associated with these new agents [37–39]. Notably, the first age-based multiscale mathematical model for HCV kinetics has been developed [25, 40, 41] providing a more comprehensive understanding of viral treatment response kinetics observed in patients treated with IFN, HCV protease inhibitors (telaprevir and danoprevir), or the HCV NS5A inhibitor daclatasvir as well as modes of action of these drugs.

The aforementioned model is an extension to the classical Neumann et al. biphasic model [20] that was introduced in 1998 and treated the infected cell as a “black box,” producing virions but without any consideration of the intracellular viral RNA replication and degradation within the infected cell [26, 27, 42]. The biphasic model is a set of three ordinary differential equations (ODEs) with three variables: uninfected target cells (T), productively infected cells (I), and extracellular virus in blood (V). The multiscale model considers the intracellular viral RNA as an additional equation for the variable (R), with the introduction of age-dependency and time-dependency, making it a partial differential equation (PDE) model. When this multiscale model is used to study the dynamics of HCV infection under therapy with DAAs it includes both intracellular viral RNA replication/degradation and extracellular viral RNA (i.e., virus particles) with age-dependency and time-dependency. As such, it is considerably more difficult to solve compared to the standard biphasic model. Previously short-term and long-term analytical approximations were derived [25, 41, 43]. In the short-term approximation, it was assumed that after therapy is initiated the infected cells maintain their steady-state levels of HCV, whereas in the long-term approximation all new infections after the onset of therapy are neglected. While the short-term approximation has been shown to be precise only in the first half-day of treatment, the long-term approximation is in agreement after several days post-treatment initiation with a simple numerical solution that utilized a canned solver (an ODE solver used in higher level languages such as Matlab and Mathematica, or Python) [41].

In this paper, we provide a robust and efficient numerical method for solving the multiscale model. The goal is to considerably improve the numerical solution presented in Rong et al. [41] that used a canned solver (an ODE solver used in higher level languages such as Matlab and Mathematica, or Python), making the numerical solution a flexible and robust entity alongside the analytical approximations. As it turns out, because of the properties of this multiscale model and the fact that the differential equations are stiff, some advanced numerical methods that involve adaptive stepsize are needed. To begin with, the use of a canned solver should be replaced with a full-fledged solver because of the additional integral introduced in the multiscale model for the variable V that needs to be computed at each time step. Unlike the construction of numerical schemes in other applications, for example in the non-linear diffusion of digital images [44–46] where accuracy can be limited, herein it is advisable to construct a stable and efficient scheme that belongs

to the Runge-Kutta family with at least a fourth order of accuracy. However, due to the nature of the differential equations that are stiff and the additional integral that needs to be evaluated at each time step, implicit solvers with adaptive stepsize are considerably more stable and efficient than the standard Runge-Kutta fourth order method. We implement implicit schemes with adaptive stepsize [47] that are highly efficient and stable for use in the multiscale model with age of hepatitis C virus dynamics.

The main contribution of this manuscript is in the use of the Rosenbrock method to solve the pre-existing multiscale model of hepatitis C virus (a system of PDEs), and to provide an open-access graphical user interface (GUI) to solve and simulate the model numerically. To the best of our knowledge, this is the first numerical simulator with a graphical user interface developed for the multiscale model. In addition, a detailed presentation of the Rosenbrock method is provided, as well as updated results. The paper is organized as follows. Section 2 provides the mathematical background for the model. In section 3, we provide the details of the Rosenbrock method including a step-by-step derivation starting from the standard Runge-Kutta fourth order, describing how we applied it to solve our problem. Section 4 presents and discusses the results achieved with our implementation of the Rosenbrock method, culminating with a description of the multiscale model simulator that we developed and is provided for general use. We conclude the paper with a summary of our results and future work.

2. MATHEMATICAL BACKGROUND

2.1. The Standard HCV Model

The standard model that has been used and modified for studying hepatitis C viral dynamics is the Neumann et al. model [20]. The three variables this model keeps track of are the target cells T , in Equation (1a), the infected cells I in Equation (1b) and the extracellular virus V in Equation (1c). The target cells T are produced at constant rate s , and decreased by the number of cells infected by virus in blood V at constant rate β and their death rate d . The infected cells, I , increase with the new infections at rate $\beta V(t)T(t)$ and die at constant rate δ . The virus V is produced at rate p by each infected cell and is cleared at constant rate c . The ϵ term denotes the effectiveness of the anti-viral treatment that decreases viral production from p to $(1 - \epsilon)p$. The previously established ensemble of ODEs for this model is:

$$\left\{ \begin{array}{l} \frac{dT(t)}{dt} = s - \beta V(t)T(t) - dT(t) \end{array} \right. \quad (1a)$$

$$\left\{ \begin{array}{l} \frac{dI(t)}{dt} = \beta V(t)T(t) - \delta I(t) \end{array} \right. \quad (1b)$$

$$\left\{ \begin{array}{l} \frac{dV(t)}{dt} = (1 - \epsilon)pI(t) - cV(t). \end{array} \right. \quad (1c)$$

From the mathematical perspective, the model is simple and can be solved analytically by appropriate assumptions.

2.2. The HCV Age-Based Multiscale Model

The multiscale model of HCV dynamics has been formulated in recent years [25, 41, 43] in order to study HCV dynamics in patients and decide among various treatment options. **Figure 1**

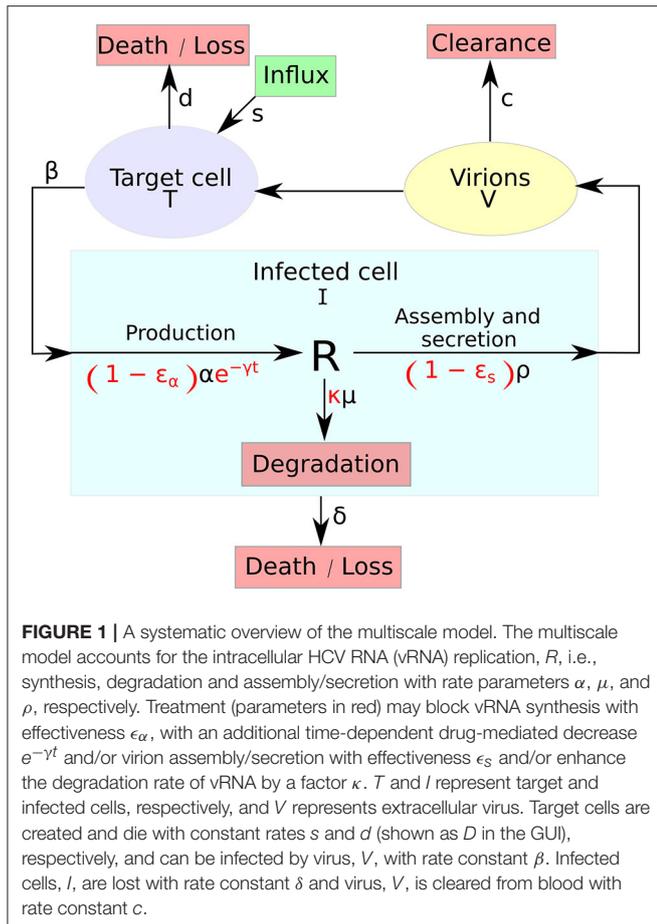


FIGURE 1 | A systematic overview of the multiscale model. The multiscale model accounts for the intracellular HCV RNA (vRNA) replication, R , i.e., synthesis, degradation and assembly/secretion with rate parameters α , μ , and ρ , respectively. Treatment (parameters in red) may block vRNA synthesis with effectiveness ϵ_α , with an additional time-dependent drug-mediated decrease $e^{-\gamma t}$ and/or virion assembly/secretion with effectiveness ϵ_s and/or enhance the degradation rate of vRNA by a factor κ . T and I represent target and infected cells, respectively, and V represents extracellular virus. Target cells are created and die with constant rates s and d (shown as D in the GUI), respectively, and can be infected by virus, V , with rate constant β . Infected cells, I , are lost with rate constant δ and virus, V , is cleared from blood with rate constant c .

depicts a systematic overview of the model, which is described by the following partial differential equations (PDEs):

$$\begin{cases} \frac{dT(t)}{dt} = s - \beta V(t)T(t) - dT(t) & (2a) \\ \frac{\partial I(a,t)}{\partial t} + \frac{\partial I(a,t)}{\partial a} = -\delta I(a,t) & (2b) \\ \frac{dV(t)}{dt} = (1 - \epsilon_s) \int_0^\infty \rho(a)R(a,t)I(a,t) da - cV(t) & (2c) \\ \frac{\partial R(a,t)}{\partial t} + \frac{\partial R(a,t)}{\partial a} = (1 - \epsilon_\alpha)\alpha(a) - [(1 - \epsilon_s)\rho(a) + \kappa\mu(a)] R(a,t), & (2d) \end{cases}$$

subject to the boundary conditions $I(0, t) = \beta V(t)T(t)$, $I(a, 0) = \bar{I}(a)$, $R(0, t) = 1$, and $R(a, 0) = \bar{R}(a)$.

The variable $I(a, t)$ for infected cells, that existed in the standard model as simply $I(t)$, and the newly introduced variable $R(a, t)$ for intracellular viral RNA (vRNA) both depend on the age of infection a and the time duration from therapy initiation t . Hence, a and t are two different times, with the use of partial derivatives in Equations (2b) and (2d). Model parameters of T , Equation (2a), and I , Equation (2b), are similar to the standard model. The quantity of vRNA R , in Equation (2d), depends on its synthesis α and its degradation μ and secretion from the cell as virus particles ρ . The quantity of extracellular virus V shown in

Equation (2c) depends on the number of assembled and released virions and their clearance rate c . A schematic description of the multiscale model is shown in **Figure 1**.

An important consideration in this model is that the treatment starts after the infection has reached its steady state. The steady states of the different functions are $\bar{R}(a, t)$, $\bar{I}(a, t)$, \bar{V} and \bar{T} . Given N , the total number of virions produced by a cell in its life-span, it was shown in Rong et al. [41] that those values are:

$$\begin{aligned} N &= \frac{\rho(\alpha + \delta)}{\delta(\rho + \mu + \delta)} \\ \bar{R}(a, t) &= \frac{\alpha}{\rho + \mu} + \left(1 - \frac{\alpha}{\rho + \mu}\right) e^{-(\rho + \mu)a} \\ \bar{I}(a, t) &= \beta \bar{V} \bar{T} e^{-\delta a} \\ \bar{T} &= c / (\beta N) \\ \bar{V} &= (\beta N s - dc) / (\beta c). \end{aligned} \tag{3}$$

Unlike the standard model, three different antiviral effects of therapy can be simulated in the multiscale model. The decrease in viral RNA synthesis is represented by ϵ_α , the reduction in secretion by ϵ_s and the increase in viral degradation by $\kappa \geq 1$.

Through the method of characteristics, as was derived in Rong et al. [41], an analytical solution was found for the variable $R(a, t)$. The same method was applied to derive a solution for $I(a, t)$. The ensemble of Equations (2) represents the full model. The analytical solutions for $R(a, t)$ and $I(a, t)$ were described as follows:

$$R(a, t) = \begin{cases} \frac{\alpha}{\rho + \mu} + \left(1 - \frac{\alpha}{\rho + \mu}\right) e^{-[\rho + \mu]a} & a < t \\ \frac{\alpha}{(\rho + \mu)} + \left(\bar{R}(a - t) - \frac{\alpha}{(\rho + \mu)}\right) e^{-(\rho + \mu)t} & a \geq t \end{cases} \tag{4}$$

$$I(a, t) = \begin{cases} \beta V(t - a)T(t - a)e^{-\delta a} & a < t \\ \bar{I}(a - t)e^{-\delta t} & a \geq t \end{cases} \tag{5}$$

From the system of Equations (2) it can be noticed that computing $V(t)$ necessitates an integral. If $a < t$, in other words the cell age is younger than the time of treatment, i.e., infection occurs after initiation of treatment, the term $I(a, t)$ of the integral in Equation (2) depends itself on $V(t)$ and $T(t)$ by consideration of Equation (5). As was shown in Guedj et al., Rong et al., Rong and Perelson [25, 41, 43], this makes the analytical solution for $V(t)$ approximative.

3. MATERIALS AND METHODS

The mathematical difficulties in deriving the long-term approximation, itself imprecise, hinders the generalization to more complex models. Numerical solutions are time consuming unless an efficient method with an adaptive stepsize is implemented, herein the Rosenbrock method [47]. We present how it derives from the Runge-Kutta family methods and its implementation for solving the model shown in Equation 2.

Runge-Kutta methods rely on computing the weighted average of a small increment from the starting position [48].

TABLE 1 | Butcher tableau for explicit Runge-Kutta, missing values are null as the c_1 row.

c_2	a_{21}		
\vdots	\vdots	\ddots	
c_S	a_{S1}	\dots	$a_{S,S-1}$
	b_1	\dots	b_S
	b_1^*	\dots	b_S^*

TABLE 2 | Butcher tableau for implicit Runge-Kutta.

c_1	a_{11}	\dots	a_{S1}
\vdots	\vdots	\ddots	\vdots
c_S	a_{S1}	\dots	a_{SS}
	b_1	\dots	b_S
	b_1^*	\dots	b_S^*

Those weights, b_i , are predetermined. An advantage of these methods is that because of the use of Taylor series there exist a different set of weights b_i^* , also known, that allow to compute directly the approximation error. The number of terms needed to compute the next value is called the order S .

3.1. A General Introduction to the Rosenbrock Method

In this section we wish to approximate the function $y(t)$ for solving the differential equation $\frac{dy}{dt} = f(t, y)$, where $f(t, y)$ is known. Finite-difference methods of higher order are used. Given a known value of y at time step n , y_n , compute the value at the next time step y_{n+1} by means of $f(t_n, y_n)$.

3.1.1. Explicit Runge-Kutta

The explicit Runge-Kutta family is a generalization of the Euler method to higher order:

$$y_{n+1} = y_n + h \sum_{i=1}^S k_i b_i \tag{6}$$

$$t_{n+1} = t_n + h, \tag{7}$$

where

$$k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j) \tag{8}$$

and a_{ij}, b_i, c_i are pre-determined constants. They are usually displayed in a Butcher tableau as in **Table 1**. It is important to note the limits of the summation in Equation (8), from 1 to $i - 1$. In this manner, it is straight forward to compute k_i at every step. A drawback of this method is the lack of stability for stiff problems [49]. The implicit Runge-Kutta methods are offering a solution to the instability problem.

TABLE 3 | Butcher tableau for Rosenbrock, missing values are null.

c_1	γ		
\vdots	\vdots	\ddots	
c_S	a_{S1}	\dots	γ
	b_1	\dots	b_S
	b_1^*	\dots	b_S^*

TABLE 4 | The parameters of the model used in all simulations, taken from Rong et al. [41].

Parameter	Value	Parameter	Value
s	13,0000 cells/mL	β	0.00000005 mL day ⁻¹ virion ⁻¹
d	0.01 day ⁻¹	δ	0.14 day ⁻¹
κ	6.36	c	22.5 day ⁻¹
α	40 day ⁻¹	ρ	7.95 day ⁻¹
μ	1 day ⁻¹	γ	0.24 day ⁻¹
ε_S	0.65	ε_α	0.997

3.1.2. Implicit Runge-Kutta

The implicit Runge-Kutta methods are themselves a more stable version of the explicit Runge-Kutta family [48], which is recommended to be used in stiff problems. The equations extend to:

$$y_{n+1} = y_n + h \sum_{i=1}^S k_i b_i \tag{9}$$

$$t_{n+1} = t_n + h \tag{10}$$

$$k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^S a_{ij} k_j) \tag{11}$$

and as before, a_{ij}, b_i, c_i are pre-determined constants. The main difference lies in Equation (11) where the summation is taken from 1 to S . The difference becomes obvious when looking at the updated Butcher tableau (**Table 2**). With the table now full, at each iteration a system of S equations with S unknowns needs to be solved.

3.1.3. The Rosenbrock Method

As shown in Rosenbrock [47] a special case of the implicit Runge-Kutta methods is when all of the values in the Butcher tableau in the upper triangular matrix above the main diagonal are null and all the diagonal elements equal to a single value *gamma* (i.e., $\forall i a_{ii} = \gamma$, **Table 3**). The value of k_i shown in Equation (11) can now be simplified as follows:

$$k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^i a_{ij} k_j). \tag{12}$$

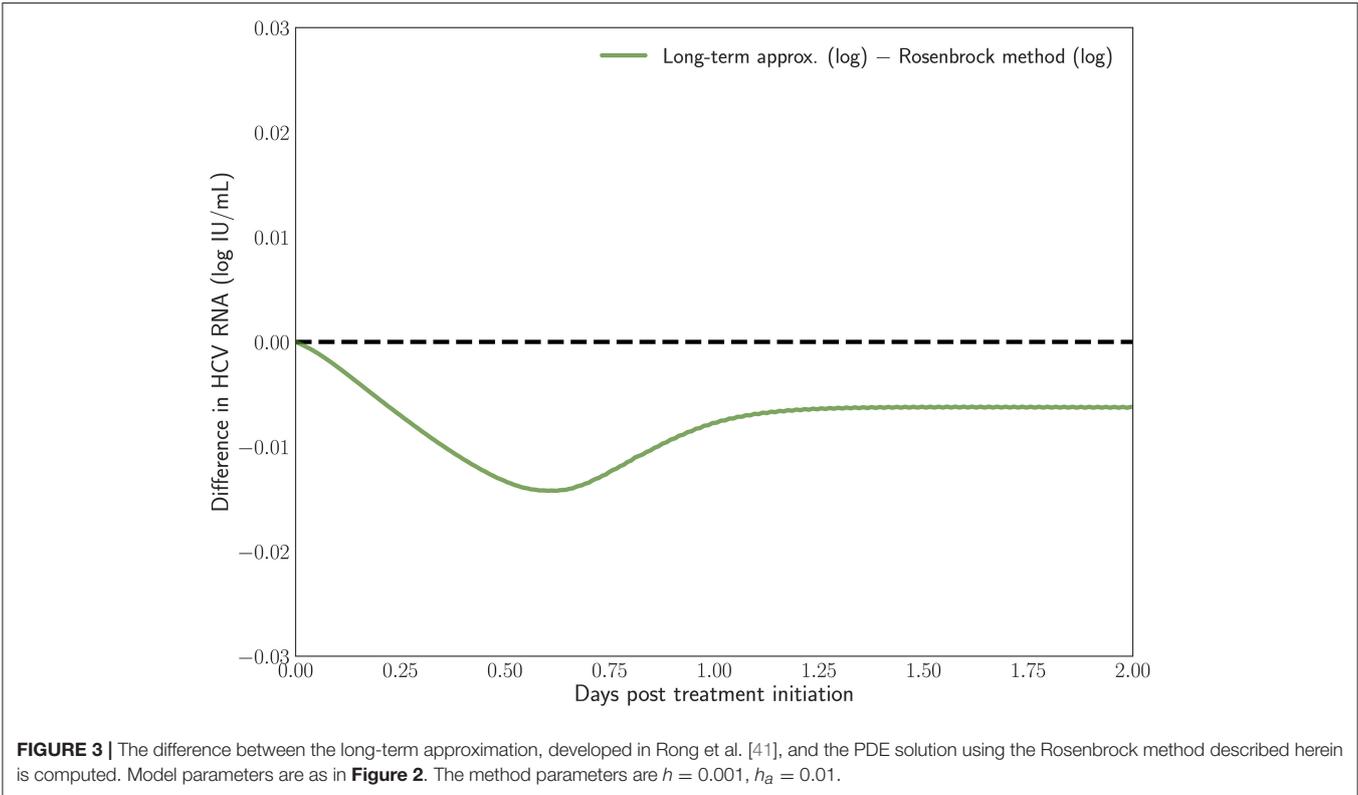
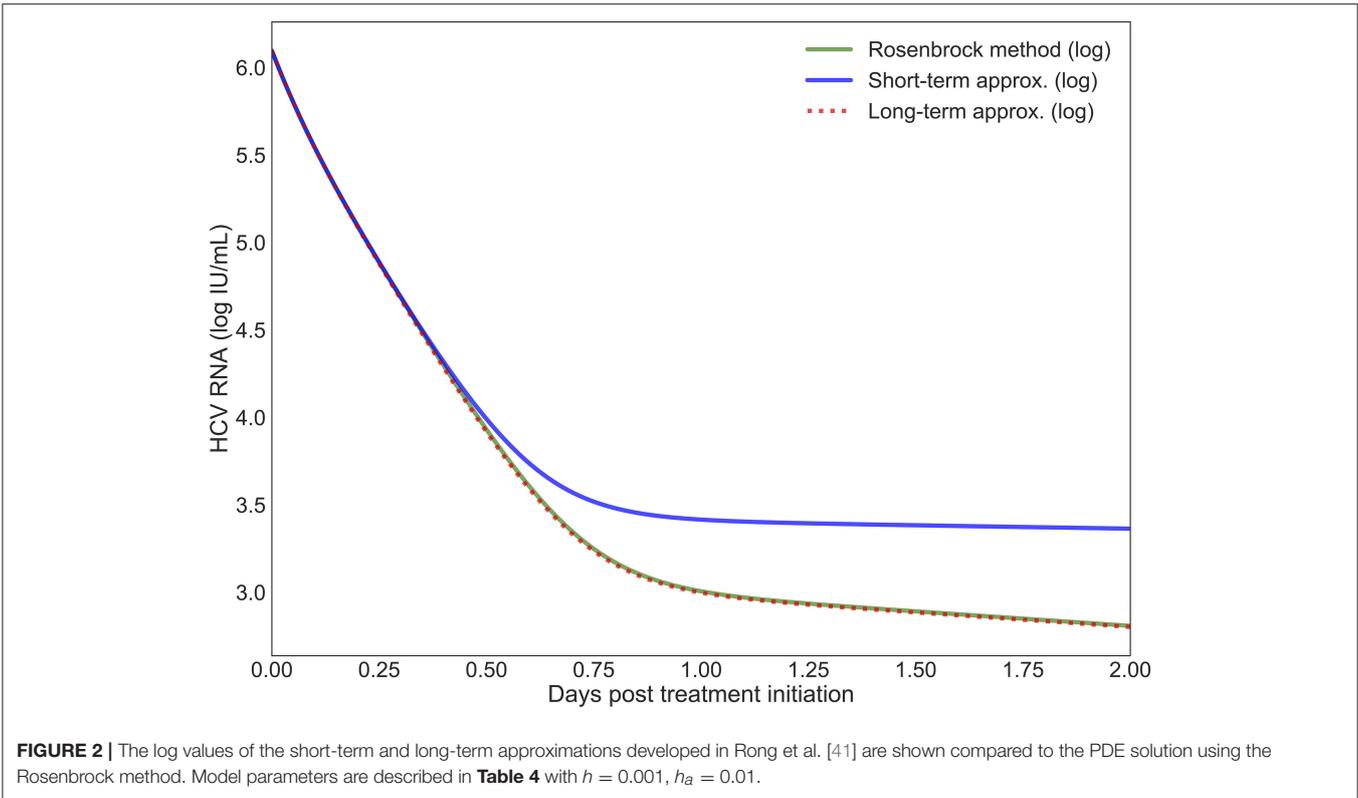


TABLE 5 | Number of iterations taken by the Rosenbrock method given time step values h and h_a compared to the number of iterations taken with the canned method (Default).

h	h_a	Rosenbrock	Default
0.001	0.01	343	2,000
0.01	0.1	112	200
0.1	1	23	20

TABLE 6 | Efficiency comparison of the Rosenbrock method with other methods.

Method	Average time (sec)	Standard deviation (sec)
Rosenbrock	78.68	1.16
Runge–Kutta 4th order	687.51	17.72
Dormand–Prince	46.51	1.26
Default	688.73	18.67

The computing time in seconds were taken by the various methods for 14 days of simulation, averaged over 10 repetitions per each method. The starting time step values were $h = 0.001$ and $h_a = 0.01$ and the runs were performed on a standard PC.

Additional substitutions [50, 51] allow to rewrite the problem as follows:

$$y_{n+1} = y_n + h \sum_{i=1}^S g_i b_i \tag{13}$$

$$(I/h\gamma - J_n)g_i = f \left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} g_j \right) + \frac{1}{h} \sum_{j=1}^{i-1} \gamma_{ij} g_j, \tag{14}$$

where J_n is the Jacobian of f at step n . To evaluate the value y_{n+1} we now need to compute the values of the g_i 's. One can notice how the left term of the equation contains a matrix with the term J_n , which is the same for all g_i 's. This is one of the key strategies of this method since to solve for g_i there is only one matrix to invert per time step. It will be taken advantage of by performing an LU decomposition, as shown in the continuation.

Additionally the error e_{n+1} can be computed as $e_{n+1} := \sum_{i=1}^S g_i b_i^*$ where the b_i^* 's are constants [50]. Given a threshold on the error, the strategy to decide if the step size must be increased or decreased consists in computing the value $\Delta_{n+1} := e_{n+1}/(\text{TOL} \times (y_n + f(t_n, y_n) + 10^{-30}))$. If Δ_{n+1} is smaller than 1 then the step size can be increased, else it must be decreased [52]. A good strategy in the latter case is to recompute the $n + 1$ value but with a smaller step size.

3.2. The Multiscale Model Solution

Since serum HCV RNA is stable in chronic HCV subjects, we assume that the model is at steady state before the start of treatment, as shown in Equation (3) and discussed in Rong et al. [41]. While individuals with recent HCV infection might not be at steady state at initiation of DAA therapy [3], it may be feasible to assume a quasi-steady state with lower pre-treatment HCV RNA levels, as observed in some subjects, which could be adjusted by changing model parameters.

We implemented the multiscale model with the Rosenbrock method of order $S = 4$. We define:

$$f(t_n, y_n) := \begin{pmatrix} \frac{dT}{dt}(t_n, y_n) \\ \frac{dV}{dt}(t_n, y_n) \end{pmatrix} = \begin{pmatrix} s - dT_n - \beta V_n T_n \\ (1 - \epsilon_s) \int_0^\infty \rho(a)R(a, t)I(a, t) da - cV_n \end{pmatrix}, \tag{15}$$

where $y_n = \begin{pmatrix} T_n \\ V_n \end{pmatrix}$ and

$$J_n := \begin{pmatrix} \frac{\partial \frac{dT}{dt}}{\partial T} & \frac{\partial \frac{dT}{dt}}{\partial V} \\ \frac{\partial \frac{dV}{dt}}{\partial T} & \frac{\partial \frac{dV}{dt}}{\partial V} \end{pmatrix} = \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix}. \tag{16}$$

We note that the term $(1 - \epsilon_s) \int_0^\infty \rho(a)R(a, t)I(a, t) da$ is constant at each iteration and is computed between each one once. We can now explicitly write the values of the constants and describe the whole scheme. The Equations (13) and (14) become:

$$\left(2\frac{I}{h} - \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix} \right) g_1 = f(t, y_n) \tag{17}$$

$$\left(2\frac{I}{h} - \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix} \right) g_2 = f(t, y_n + 2g_1) + \frac{1}{h} (-8g_1) \tag{18}$$

$$\left(2\frac{I}{h} - \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix} \right) g_3 = f \left(t, y_n + \frac{48}{25}g_1 + \frac{6}{25}g_2 \right) + \frac{1}{h} \left(\frac{372}{25}g_1 + \frac{12}{5}g_2 \right) \tag{19}$$

$$\left(2\frac{I}{h} - \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix} \right) g_4 = f \left(t, y_n + \frac{48}{25}g_1 + \frac{6}{25}g_2 \right) + \frac{1}{h} \left(\frac{-112}{125}g_1 + \frac{-54}{125}g_2 + \frac{-2}{5}g_3 \right) \tag{20}$$

$$y_{n+1} = y_n + \frac{19}{9}g_1 + \frac{1}{2}g_2 + \frac{25}{108}g_3 + \frac{125}{108}g_4 \tag{21}$$

$$e_{n+1} = \frac{17}{54}g_1 + \frac{7}{36}g_2 + 0g_3 + \frac{125}{108}g_4 \tag{22}$$

Note that the computations of g_3 in Equation (19) and g_4 in Equation (20) use the same value of f , which implies that it needs

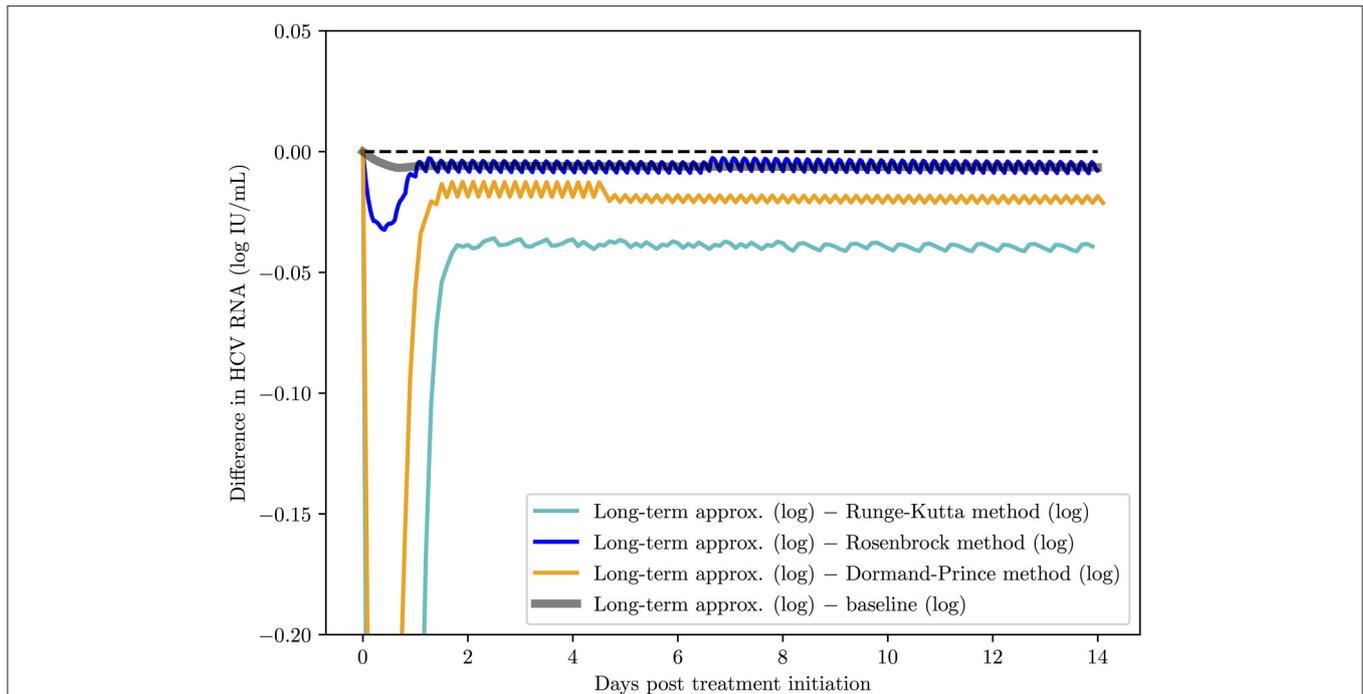


FIGURE 4 | A comparison of the Rosenbrock method with other methods over 14 days of simulation with starting time steps of $h = 0.1$ and $h_a = 0.1$. The baseline (the default implementation in the SciPy library of an ODE solver) was computed with time steps of $h = 0.001$ and $h_a = 0.01$ since with these time step values and smaller, as expected, all methods converged to the solution of the baseline.

TABLE 7 | Sensitivity of the Rosenbrock method.

Parameter	Decreased	Increased	Parameter	Decreased	Increased
s	0.88%	1.12%	β	0.98%	1.01%
d	1.02%	0.98%	δ	1.16%	0.87%
κ	1.08%	0.93%	c	1.13%	0.89%
α	0.88%	1.12%	ρ	0.93%	1.07%
μ	1.08%	0.93%	γ	1.05%	0.96%

Each of the 10 parameters was decreased and increased by 10% of its original value. For each we present the ratio of the result after 2 days over the result with the original value. We omit ϵ_a and ϵ_s since they effectively determine the fraction of α and ρ used in the model.

to be computed only once. The error term also disregards the value of g_3 .

The scheme for one iteration is shown in **Algorithm 1**. An important observation is how the error term e is computed. It derives only from the error induced by the Rosenbrock iteration, not the computation of the integral term $(1 - \epsilon_s) \int_0^\infty \rho(a)R(a, t)I(a, t) da$.

The system $(I/h\gamma - J_n)g_i = f(y_n + \sum_{j=1}^{i-1} \alpha_{ij}g_j) + \frac{1}{h} \sum_{j=1}^{i-1} \gamma_{ij}g_j$ needs to be solved for four different values of i . Since the left hand matrix is constant, it is decomposed into its LU decomposition once. The Jama package [53] is used to perform the LU decomposition for solving the system in a highly efficient manner.

Algorithm 1: General Rosenbrock Scheme. TOL was set to 10^{-4} .

Data: y_n, h

Result: Compute the next value y_{n+1} for a step of size h

integralterm \leftarrow solve $(1 - \epsilon_s) \int_0^\infty \rho(a)R(a, t)I(a, t) da$;

LU \leftarrow LU decomposition of

$$\left(2\frac{I}{h} - \begin{pmatrix} -d - \beta V_n & -\beta T_n \\ 0 & -c \end{pmatrix} \right);$$

for $i \leftarrow 1$ **to** 4 **do**

 solve for g_i : $LUg_i = f(y_n + \sum_{j=1}^{i-1} \alpha_{ij}g_j) + \frac{1}{h} \sum_{j=1}^{i-1} \gamma_{ij}g_j$;

end

$$y_{n+1} = y_n + \frac{19}{9}g_1 + \frac{1}{2}g_2 + \frac{25}{108}g_3 + \frac{125}{108}g_4;$$

$$e_{n+1} = \frac{17}{54}g_1 + \frac{7}{36}g_2 + 0g_3 + \frac{125}{108}g_4;$$

if $e \geq \text{TOL} \times (y_n + f(t_n, y_n) + 10^{-30})$ **then**

 Start again with smaller h ;

else

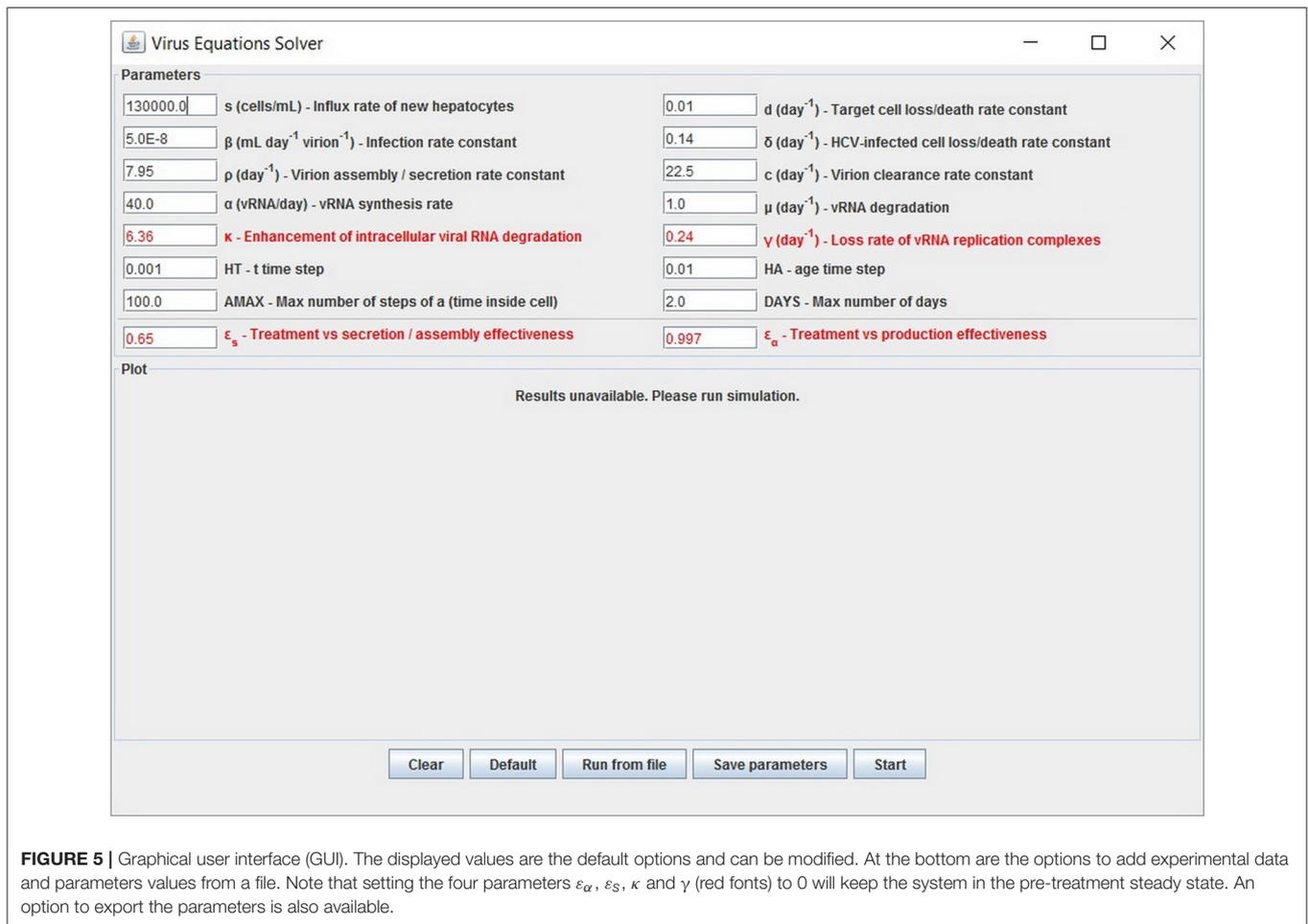
 Increase h ;

return y_{n+1}

end

4. RESULTS AND DISCUSSION

A preliminary version of the Rosenbrock method was first implemented in Python3 using the SciPy library, freely available



at <http://www.cs.bgu.ac.il/~dbarash/HCVnumerics>, and then converted to Java for the purpose of developing a user-friendly simulator with a graphical user interface (GUI) that is freely available at <http://www.cs.bgu.ac.il/~dbarash/HCVsimulator>. Aside of the Rosenbrock method, the default implementation in SciPy of an ordinary differential solver leverages ODEPACK [54] and is referred to as the canned solver *Default*. For this entire section, unless stated otherwise, we used the parameters from Rong et al. [41] and shown in **Table 4**. The parameters that are changed through the results are the number of days, the size h of the steps for the ODEs, and the size h_a of the steps for computing the integral. In the case of the Rosenbrock method that utilizes an adaptive stepsize, we bound the stepsize by h_a as minimum and h_a as maximum.

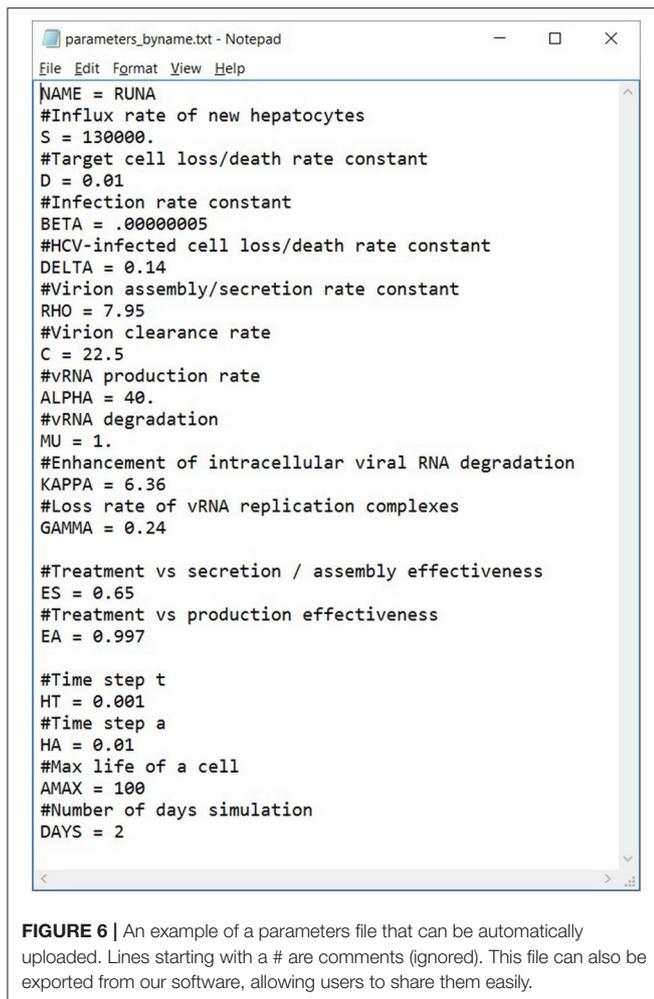
4.1. Short-Term Approximation Holds for Half a Day

The short-term approximation [41] is shown in **Figure 2** in blue. The results are shown for only 2 days since the behavior is smooth and conserved on longer time scales in agreement with previous results (Figure 2A in [41]). It is clear that after 12 h the value converges far from the PDE solution. This is expected since the effect of the treatment on the infection rate is not

taken into account. In practice, most simulations are valuable for more than half a day and are focused on the long-term approximation.

4.2. Long-Term Approximation Underestimates PDE

In Rong and Perelson [43] it is hypothesized that the long-term approximation is an **underestimate** of the PDE model solution since some infection events are being ignored. However, with realistic parameters characteristic of potent therapy, the difference between them is very small. **Figure 3** depicts the result of our scheme, measuring the difference between the long-term approximation and the solution of the multiscale PDE model (Figure 2B in [41]). We show that the long-term approximation is converging and we are indeed obtaining that the long-term approximation is an underestimate of the PDE model solution (as the difference between the long term approximation and the solution of the PDE model is negative). The difference of less than 0.01 that we are obtaining is very small compared to the y -axis units shown in Figure 1 of Rong and Perelson [43]. Therefore, it is possible to view in **Figure 3** this small effect that is an improvement over what was inferred in Rong and Perelson [43] regarding the long-term approximation. This finding is reiterated



```

parameters_byname.txt - Notepad
File Edit Format View Help
NAME = RUNA
#Influx rate of new hepatocytes
S = 130000.
#Target cell loss/death rate constant
D = 0.01
#Infection rate constant
BETA = .00000005
#HCV-infected cell loss/death rate constant
DELTA = 0.14
#Virion assembly/secretion rate constant
RHO = 7.95
#Virion clearance rate
C = 22.5
#vRNA production rate
ALPHA = 40.
#vRNA degradation
MU = 1.
#Enhancement of intracellular viral RNA degradation
KAPPA = 6.36
#Loss rate of vRNA replication complexes
GAMMA = 0.24

#Treatment vs secretion / assembly effectiveness
ES = 0.65
#Treatment vs production effectiveness
EA = 0.997

#Time step t
HT = 0.001
#Time step a
HA = 0.01
#Max life of a cell
AMAX = 100
#Number of days simulation
DAYS = 2

```

FIGURE 6 | An example of a parameters file that can be automatically uploaded. Lines starting with a # are comments (ignored). This file can also be exported from our software, allowing users to share them easily.

here for completeness and is referred to below in the sub-section about the Rosenbrock method.

4.3. The Model Equations Are Stiff

An important distinction of the age-based PDE HCV model equations is their stiffness, or how they can be numerically unstable even with a small stepsize. We have verified that with small time-steps the results are similar for all methods. But as we increase the size of the steps, the instability of the equations can be observed over 20 days. All explicit methods such as the standard version of Runge-Kutta quickly diverge from the solution with small time-steps. On the other hand an implicit Runge-Kutta method tends to the correct solution and the Rosenbrock method does so efficiently as reported below.

4.4. The Rosenbrock Method Is Efficient and Stable

As previously mentioned, the integral term $(1 - \epsilon_s) \int_0^\infty \rho(a)R(a, t)I(a, t) da$ impedes the use of adaptive stepsize when a canned solver is used. Such methods are essential to allow fast computations of complex models like ours. We

present in **Table 5** the number of iterations computed using the Rosenbrock method, shown in the previous section, in comparison with the default canned method for three different time step values of h and h_a . With the smallest time step, the Rosenbrock method is more than five times as efficient as the default canned method, giving results of similar accuracy. This is crucial since the methods with fixed time step can take up to tens of seconds per day of simulation with $h = 0.001$ and $h_a = 0.01$. For most research needs, simulations are expected to be performed numerous times for various instances and parameter values. Under those conditions a five-fold increase in speed, through the decrease in the number of iterations, provides a much needed advantage when testing large number of parameters over long time periods. Interestingly with the highest value of h , the Rosenbrock method takes additional iterations. This is due to the greediness of the time step adjustment, which when increased too quickly and thereby inducing a large error backtracks and starts again with a smaller value.

Simpler methods than Rosenbrock exist with adaptive stepsize, such as Dormand-Prince [48], Fehlberg (RKF) [55] and Cash-Karp (RKCK) [56]. These are explicit methods that can be easily implemented but are vulnerable to the stiffness of the equations, while Rosenbrock is an implicit method that is indeed found to be more stable. In the following comparisons we demonstrate that the Rosenbrock method is more efficient than the Runge-Kutta 4th order method and at the same time it is more stable than both the Runge-Kutta 4th order and the Dormand-Prince methods.

First, a comparison of the computing time between the Rosenbrock method and the other methods mentioned above is reported in **Table 6** with starting time step values of $h = 0.001$ and $h_a = 0.01$ for 14 days of simulation. Each run with a certain method was repeated 10 times, after which the mean and standard deviation was taken. The Rosenbrock method is over 8 times faster than the standard Runge-Kutta 4th order method, which is a non-adaptive time step method. While Dormand-Prince is an adaptive time step method that is even faster than Rosenbrock, as will be shown below, it is an explicit method that does not always converge to the correct solution. The fastest time of execution for the Dormand-Prince method is not surprising since an explicit method is easier to evaluate computationally than an implicit method, in particular because there is no need for matrix inversions.

Second, a comparison of the methods behavior over 14 days of simulation is presented in **Figure 4**. All the methods were started with time steps of $h = 0.1$ and $h_a = 0.1$. The baseline (the default implementation in the SciPy library of an ODE solver) was computed for $h = 0.001$ and $h_a = 0.01$ since those time step values are small enough to ensure that all methods converge to the same solution. The advantage of the Rosenbrock method in terms of stability and convergence to the correct solution is clearly noticed.

Finally, we also report in **Table 7** the sensitivity of the Rosenbrock method. We compute the changed value of the result after 2 days of simulation given a variation of $\pm 10\%$ on 10 parameters. We notice that most of the perturbations are symmetrical. We omit ϵ_α and ϵ_s since they effectively determine

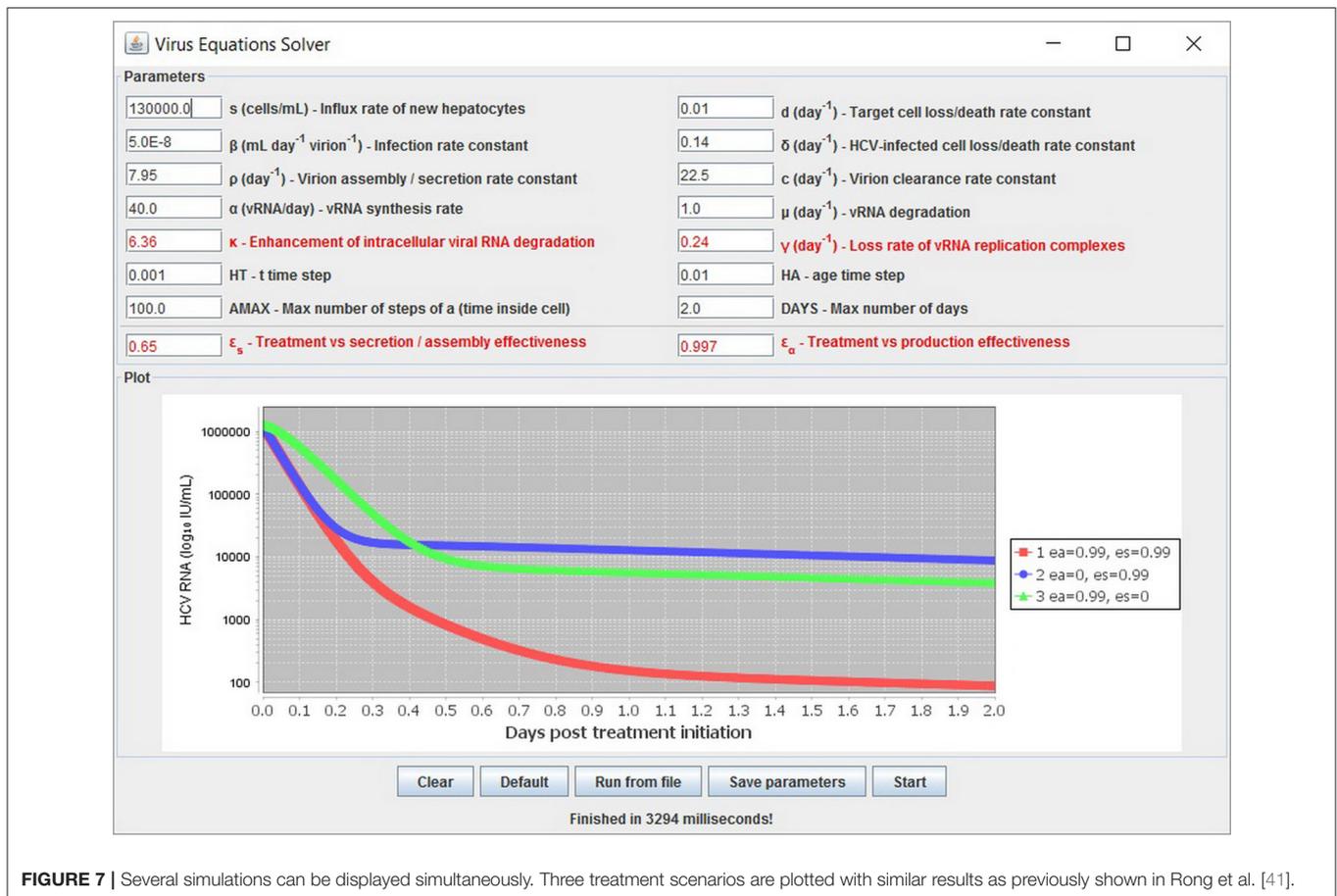


FIGURE 7 | Several simulations can be displayed simultaneously. Three treatment scenarios are plotted with similar results as previously shown in Rong et al. [41].

the fraction of α and ρ used in the model. The results (Table 7) indicate that the Rosenbrock method is sufficiently robust for our needs.

4.5. The Multiscale Model Simulator

We have developed a user-friendly simulator with a GUI for the multiscale model (Figure 5) that is freely available at <http://www.cs.bgu.ac.il/~dbarash/HCVsimulator>. The parameters file input is shown in Figure 6. For illustration, two example simulations are provided, one over 2 days (Figure 7) and one over 28 days (Figure 8). In Figure 7 we show how different parameter ensembles can be displayed simultaneously. In that case we chose ϵ_a and ϵ_s to be 0 or 0.99, resulting in three curves. We can observe that increasing the ϵ_s parameter, which decreases the assembly/secretion of intracellular virions, has a strong short term effect that tempers quickly. In contrast the reduction in the synthesis of vRNA, modeled by an increase of the parameter ϵ_a , shows a slower but more efficient effect that exhibits stronger results after half a day. Combining both factors shows a multiphasic viral decline which was observed under treatment with HCV NS5A inhibitors [25]. In Figure 8 we present the difference between the long-term approximation and the Rosenbrock method for those three sets of parameters over longer (than 2 days) treatment durations (e.g., 28 days). Although the highest error is present when $\epsilon_s = 0$, the error remains below a tenth of the differences of the log values. Thus, in all cases the

trends from the first 2 days continue in the next 4 weeks and the error grows minimally.

5. SUMMARY AND CONCLUSIONS

Modeling intracellular viral RNA dynamics within infected cells is becoming an important mean for considering various curative treatment options. A viral dynamic model that considers intracellular viral RNA replication, namely an age-structured PDE multiscale model, has been recently put forth to study viral hepatitis dynamics during antiviral therapy [25, 41, 43]. This type of model is more complicated to solve than previous ODEs models. The seminal works that introduce the age-based multiscale model predominantly use analytical approximations, with some numerical solutions that are either based on simple first-order methods or canned solvers [25, 41, 43]. Neither of these numerical solutions are satisfactory for realistic simulations of several days of infection and therapy in terms of accuracy, efficiency and stability.

We first showed that the long-term approximation is an underestimate of the PDE model solution, which was anticipated in Rong and Perelson [43] because some infection events are being ignored in this analytical approximation. We then observed that the governing differential equations are stiff and therefore advanced numerical methods are needed. Methods

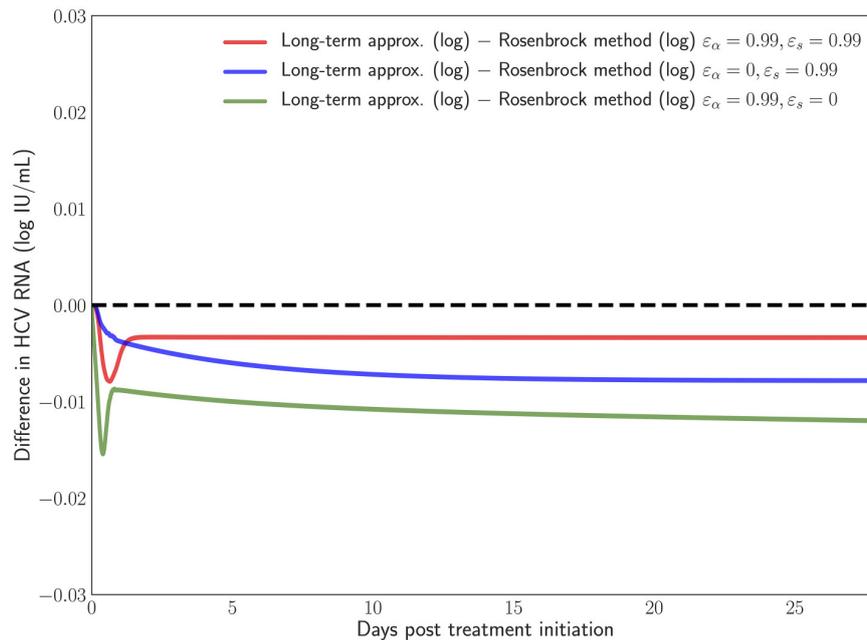


FIGURE 8 | Difference in the log values between the long-term approximation [41] and the Rosenbrock method when choosing ε_α and ε_s from (0, 0.99). One can notice how even at the longer treatment duration the difference remains minimal and mostly constant.

with fixed stepsize or alternatively, the use of canned routines, do not offer a comprehensive solution to the model and are limited in scope. For example, in the multiscale model, there is an integral term that needs to be computed at each time step depending on previous iterations and this is inadequately done by canned solvers. Methods with adaptive stepsize offer a considerable improvement for realistic simulations. Having previously investigated these methods, we found the Rosenbrock method to be the most accurate and efficient among adaptive stepsize methods for this model. We provide a simulator based on the Rosenbrock method with a user-friendly GUI that is freely available at <http://www.cs.bgu.ac.il/~dbarash/HCVsimulator>.

Future work from the numerical standpoint could potentially include a more comprehensive treatment of the associated integral. In particular the adaptive stepsize techniques rely on the evaluation of the error produced at each evaluation of the ODE equations. Therefore, the error produced by the integral itself is never taken into account. Including that term would allow to relax the restrictions on the stepsize increase and potentially decrease further, beyond what is observed in **Table 5** as least number of iterations, the number of iterations that are required to achieve a sufficient accuracy.

The Rosenbrock method implementation provided here can be generalized to potentially assist in understating treatment failure due to drug resistance by the expansion of the age-based model to include viral strains [57]. It could also be used to explore age-based models that include further aspects of intracellular HCV life cycle such as translation positive-strand HCV RNA and the synthesis of positive and negative-strand HCV RNA [58]. Finally, since the current age-based HCV multiscale model is a successful milestone but still fails to predict

cure in some DAA-treated patients [59–61], further model modification will be needed [38, 39] and future developments would necessitate some updates in the numerical method. While such models become quickly overwhelming to solve analytically, an extension of the presented method is straightforward. The simulator provided here based on the Rosenbrock method can be modified accordingly and be developed further to accommodate the modelers needs.

AUTHOR CONTRIBUTIONS

DB conceived research. VR implemented method and developed the corresponding equations with assistance of DB. VR, HD, and DB designed research. VR, AC, HD, and DB performed research and wrote the paper.

FUNDING

The research was supported in part by the U.S. National Institute of Health (NIH) grant R01-AI078881. VR is funded by Azrieli and FQRNT post-doctoral fellowships. The authors would like to thank the Lynne and William Frankel Center for Computer Sciences at Ben-Gurion University for partially supporting the research.

ACKNOWLEDGMENTS

The authors thank Susan Uprichard for careful reading of the manuscript and helpful comments. They would also like to thank Libin Rong and Daniel Coffield for providing the initial code of the multiscale model and the reviewers for their comprehensive comments.

REFERENCES

1. The Polaris Observatory HCV Collaborators. Global prevalence and genotype distribution of hepatitis C virus infection in 2015: a modelling study. *Lancet Gastroenterol Hepatol.* (2017) **2**:161–76. doi: 10.1016/S2468-1253(16)30181-9
2. World Health Organization. *Guidelines for the Screening, Care and Treatment of Persons with Hepatitis C Infection.* Geneva: World Health Organization (2014).
3. Martinello M, Gane E, Hellard M, Sasadeusz J, Shaw D, Petoumenos K, et al. Sofosbuvir and ribavirin for 6 weeks is not effective among people with recent hepatitis C virus infection: the DARE-C II study. *Hepatology* (2016) **64**:1911–21. doi: 10.1002/hep.28844
4. AASLD/IDSA HCV Guidance Panel. Hepatitis C guidance: AASLD-IDSA recommendations for testing, managing, and treating adults infected with hepatitis C virus. *Hepatology* (2015) **62**:932–54. doi: 10.1002/hep.27950
5. Rosen HR. “Hep C, where art thou”: what are the remaining (fundable) questions in hepatitis C virus research? *Hepatology* (2017) **65**:341–9. doi: 10.1002/hep.28848
6. Perelson AS. Modelling viral and immune system dynamics. *Nat Rev Immunol.* (2002) **2**:28–36. doi: 10.1038/nri700
7. Ho DD, Neumann AU, Perelson AS, Chen W, Leonard JM, Markowitz M. Rapid turnover of plasma virions and CD4 lymphocytes in HIV-1 infection. *Nature* (1995) **373**:123. doi: 10.1038/373123a0
8. Perelson AS, Neumann AU, Markowitz M, Leonard JM, Ho DD. HIV-1 dynamics *in vivo*: virion clearance rate, infected cell life-span, and viral generation time. *Science* (1996) **271**:1582. doi: 10.1126/science.271.5255.1582
9. Burg D, Rong L, Neumann AU, Dahari H. Mathematical modeling of viral kinetics under immune control during primary HIV-1 infection. *J. Theor. Biol.* (2009) **259**:751–9. doi: 10.1016/j.jtbi.2009.04.010
10. Ciupe SM, Ribeiro RM, Nelson PW, Dusheiko G, Perelson AS. The role of cells refractory to productive infection in acute hepatitis B viral dynamics. *Proc Natl Acad Sci USA.* (2007) **104**:5050–5. doi: 10.1073/pnas.0603626104
11. Dahari H, Shudo E, Ribeiro RM, Perelson AS. Modeling complex decay profiles of hepatitis B virus during antiviral therapy. *Hepatology* (2009) **49**:32–8. doi: 10.1002/hep.22586
12. Nowak MA, Bonhoeffer S, Hill AM, Boehme R, Thomas HC, McDade H. Viral dynamics in hepatitis B virus infection. *Proc Natl Acad Sci USA.* (1996) **93**:4398–402. doi: 10.1073/pnas.93.9.4398
13. Koh C, Canini L, Dahari H, Zhao X, Uprichard SL, Haynes-Williams V, et al. Oral prenylation inhibition with lonafarnib in chronic hepatitis D infection: a proof-of-concept randomised, double-blind, placebo-controlled phase 2A trial. *Lancet Infect Dis.* (2015) **15**:1167–74. doi: 10.1016/S1473-3099(15)00074-2
14. Guedj J, Rotman Y, Cotler SJ, Koh C, Schmid P, Albrecht J, et al. Understanding early serum hepatitis D virus and hepatitis B surface antigen kinetics during pegylated interferon-alpha therapy via mathematical modeling. *Hepatology* (2014) **60**:1902–10. doi: 10.1002/hep.27357
15. Canini L, Koh C, Cotler SJ, Uprichard SL, Winters MA, Thanda Han MA, et al. Pharmacokinetics and pharmacodynamics modeling of lonafarnib in patients with chronic hepatitis delta virus infection. *Hepatol Commun.* (2017) **1**:288–92. doi: 10.1002/hep4.1043
16. Zhang J, Lipton HL, Perelson AS, Dahari H. Modeling the acute and chronic phases of Theiler murine encephalomyelitis virus infection. *J virol.* (2013) **87**:4052–9. doi: 10.1128/JVI.03395-12
17. Schiffer JT, Abu-Raddad L, Mark KE, Zhu J, Selke S, Margaret A, et al. Frequent release of low amounts of herpes simplex virus from neurons: results of a mathematical model. *Sci Trans Med.* (2009) **1**:ra16. doi: 10.1126/scitranslmed.3000193
18. Dahari H, Layden-Almer JE, Kallwitz E, Ribeiro RM, Cotler SJ, Layden TJ, et al. A mathematical model of hepatitis C virus dynamics in patients with high baseline viral loads or advanced liver disease. *Gastroenterology* (2009) **136**:1402–9. doi: 10.1053/j.gastro.2008.12.060
19. Dahari H, Major M, Zhang X, Mihalik K, Rice CM, Perelson AS, et al. Mathematical modeling of primary hepatitis C infection: noncytolytic clearance and early blockage of virion production. *Gastroenterology* (2005) **128**:1056–66. doi: 10.1053/j.gastro.2005.01.049
20. Neumann AU, Lam NP, Dahari H, Gretch DR, Wiley TE, Layden TJ, et al. Hepatitis C viral dynamics *in vivo* and the antiviral efficacy of interferon- α therapy. *Science* (1998) **282**:103–7. doi: 10.1126/science.282.5386.103
21. Baccam P, Beauchemin C, Macken CA, Hayden FG, Perelson AS. Kinetics of influenza A virus infection in humans. *J Virol.* (2006) **80**:7590–9. doi: 10.1128/JVI.01623-05
22. Pawelek KA, Huynh GT, Quinlivan M, Cullinane A, Rong L, Perelson AS. Modeling within-host dynamics of influenza virus infection including immune responses. *PLoS Comput Biol.* (2012) **8**:e1002588. doi: 10.1371/journal.pcbi.1002588
23. Beauchemin CA, Handel A. A review of mathematical models of influenza A infections within a host or cell culture: lessons learned and challenges ahead. *BMC Public Health.* (2011) **11**:S7. doi: 10.1186/1471-2458-11-S1-S7
24. Madelain V, Oestereich L, Graw F, Nguyen THT, De Lamballerie X, Mentré F, et al. Ebola virus dynamics in mice treated with favipiravir. *Antiv. Res.* (2015) **123**:70–7. doi: 10.1016/j.antiviral.2015.08.015
25. Guedj J, Dahari H, Rong L, Sansone ND, Nettles RE, Cotler SJ, et al. Modeling shows that the NS5A inhibitor daclatasvir has two modes of action and yields a shorter estimate of the hepatitis C virus half-life. *Proc Natl Acad Sci USA.* (2013) **110**:3991–6. doi: 10.1073/pnas.1203110110
26. Dahari H, Sainz B, Perelson AS, Uprichard SL. Modeling subgenomic hepatitis C virus RNA kinetics during treatment with alpha interferon. *J Virol.* (2009) **83**:6383–90. doi: 10.1128/JVI.02612-08
27. Dahari H, Ribeiro RM, Rice CM, Perelson AS. Mathematical modeling of subgenomic hepatitis C virus replication in Huh-7 cells. *J Virol.* (2007) **81**:750–60. doi: 10.1128/JVI.01304-06
28. Neumann AU, Phillips S, Levine I, Ijaz S, Dahari H, Eren R, et al. Novel mechanism of antibodies to hepatitis B virus in blocking viral particle release from cells. *Hepatology* (2010) **52**:875–85. doi: 10.1002/hep.23778
29. Dahari H, Shudo E, Ribeiro RM, Perelson AS. Mathematical modeling of HCV infection and treatment. *Methods Mol Biol.* (2009) **510**:439–53. doi: 10.1007/978-1-59745-394-3_33
30. Dahari H, Guedj J, Perelson AS, Layden TJ. Hepatitis C viral kinetics in the era of direct acting antiviral agents and interleukin-28B. *Curr. Hepatitis Rep.* (2011) **10**:214–27. doi: 10.1007/s11901-011-0101-7
31. Snoeck E, Chanu P, Lavielle M, Jacqmin P, Jonsson EN, Jorga K, et al. A comprehensive hepatitis C viral kinetic model explaining cure. *Clin. Pharmacol. Ther.* (2010) **87**:706–13. doi: 10.1038/clpt.2010.35
32. Dixit NM, Layden-Almer JE, Layden TJ, Perelson AS. Modelling how ribavirin improves interferon response rates in hepatitis C virus infection. *Nature* (2004) **432**:922–4. doi: 10.1038/nature03153
33. Guedj J, Perelson AS. Second-phase hepatitis C virus RNA decline during telaprevir-based therapy increases with drug effectiveness: implications for treatment duration. *Hepatology* (2011) **53**:1801–8. doi: 10.1002/hep.24272
34. Dahari H, Canini L, Graw F, Uprichard SL, Araujo ES, Penaranda G, et al. HCV kinetic and modeling analyses indicate similar time to cure among sofosbuvir combination regimens with daclatasvir, simeprevir or ledipasvir. *J Hepatol.* (2016) **64**:1232–9. doi: 10.1016/j.jhep.2016.02.022
35. Gambato M, Canini L, Lens S, Graw F, Londono MC, Uprichard SL, et al. Modeling early HCV kinetics to individualize direct acting antiviral treatment duration in patients with advanced cirrhosis. *J Hepatol.* (2016) **64**:S795. doi: 10.1016/S0168-8278(16)01550-6
36. Dahari H, Shteingart S, Gafanovich I, Cotler SJ, D’Amato M, Pohl RT, et al. Sustained virological response with intravenous silibinin: individualized IFN-free therapy via real-time modelling of HCV kinetics. *Liver Int.* (2015) **35**:289–94. doi: 10.1111/liv.12692
37. Rong L, Dahari H, Ribeiro RM, Perelson AS. Rapid emergence of protease inhibitor resistance in hepatitis C virus. *Sci Trans Med.* (2010) **2**:30ra32. doi: 10.1126/scitranslmed.3000544
38. Nguyen TH, Guedj J, Canini L, Osinusi A, Pang PS, McHutchison J, et al. “The paradox of highly effective sofosbuvir combo therapy despite slow viral decline,” in *Conference on Retroviruses and Opportunistic Infections CROI 2015.* Seattle, WA (2015). p. 169–70.
39. Goyal A, Lurie Y, Meissner G, Major M, Sansone N, Uprichard S, et al. Modeling HCV cure after an ultra-short duration of therapy with direct acting agents. *Antiv Res.* (2017) **1444**:281–5. doi: 10.1016/j.antiviral.2017.06.019

40. Guedj J, Dahari H, Uprichard SL, Perelson AS. The hepatitis C virus NS5A inhibitor daclatasvir has a dual mode of action and leads to a new virus half-life estimate. *Expert Rev Gastroenterol Hepatol*. (2013) 7:397–9. doi: 10.1586/17474124.2013.811050
41. Rong L, Guedj J, Dahari H, Coffield DJJ, Levi M, Smith P, et al. Analysis of hepatitis C virus decline during treatment with the protease inhibitor danoprevir using a multiscale model. *PLoS Comput Biol*. (2013) 9:e1002959. doi: 10.1371/journal.pcbi.1002959
42. Guedj J, Neumann AU. Understanding hepatitis C viral dynamics with direct-acting antiviral agents due to the interplay between intracellular replication and cellular infection dynamics. *J Theor Biol*. (2010) 267:330–40. doi: 10.1016/j.jtbi.2010.08.036
43. Rong L, Perelson AS. Mathematical analysis of multiscale models for hepatitis C virus dynamics under therapy with direct-acting antiviral agents. *Math Biosci*. (2013) 245:22–30. doi: 10.1016/j.mbs.2013.04.012
44. Weickert J, ter Haar Romeny B, Viergever M. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans Imag Proc*. (1998) 7:398–410. doi: 10.1109/83.661190
45. Barash D, Israeli M, Kimmel R. “An accurate operator splitting scheme for nonlinear diffusion filtering,” in *Proceedings of the 3rd International Conference on ScaleSpace and Morphology. LNCS Series*. Vancouver, BC: Springer-Verlag (2001). p. 281–9.
46. Barash D. Nonlinear diffusion filtering on extended neighborhood. *Appl Num Math*. (2005) 52:1–11. doi: 10.1016/j.apnum.2004.07.002
47. Rosenbrock H. Some general implicit processes for the numerical solution of differential equations. *Comp J*. (1963) 5:329–30. doi: 10.1093/comjnl/5.4.329
48. Dormand J, Prince P. A family of embedded Runge-Kutta formulae. *J Comp Appl Math*. (1980) 6:19–26. doi: 10.1016/0771-050X(80)90013-3
49. Dekker K, Verwer JG. *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*. Vol. 984. North-Holland; Amsterdam: Elsevier Science Ltd (1984).
50. Kaps P, Rentrop P. Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. *Numer Math*. (1979) 33:55–68. doi: 10.1007/BF01396495
51. Hairer E, Wanner G. *Solving Ordinary Differential Equations. Vol. 14 of Springer Series in Computational Mathematics*. 2nd ed. Berlin: Cambridge University Press; Springer-Verlag (1996).
52. Press WH, Teukolsky SH, Vetterling WT, Flannery BP. *Numerical Recipes: The Art of Scientific Computing*. 2nd ed. Cambridge, UK: Cambridge University Press (1997).
53. Hicklin J, Moler C, Webb P, Boisvert RF, Miller B, Pozo R, et al. *Jama: A Java Matrix Package*. Available online at: <http://mathnist.gov/javanumerics/jama> (2000).
54. Hindmarsh AC. ODEPACK, a systematized collection of ODE solvers. *IMACS Trans Sci Comp*. (1983) 1:55–64.
55. Fehlberg E. Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems. *NASA Technical Report 315* (1969).
56. Cash J, Karp A. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Trans Math Softw*. (1990) 16:201–22. doi: 10.1145/79505.79507
57. Canini L, Perelson AS. Viral kinetic modeling: state of the art. *J Pharmacokinetic Pharmacodyn*. (2014) 41:431–43. doi: 10.1007/s10928-014-9363-3
58. Quintela B, Conway J, Hyman J, Reis R, dos Santos R, Lobosco M, et al. “An age-based multiscale mathematical model of the hepatitis C virus life-cycle during infection and therapy: including translation and Replication,” in *VII Latin American Congress on Biomedical Engineering CLAIB 2016*. Bucaramanga: Springer (2017). p. 508–11.
59. Lau G, Benhamou Y, Chen G, Li J, Shao Q, Ji D, et al. Efficacy and safety of 3-week response-guided triple direct-acting antiviral therapy for chronic hepatitis C infection: a phase 2, open-label, proof-of-concept study. *Lancet Gastroenterol Hepatol*. (2016) 1:97–104. doi: 10.1016/S2468-1253(16)30015-2
60. Hasin Y, Shteingart S, Dahari H, Gafanovich I, Floru S, Braun M, et al. Hepatitis C virus cures after direct acting antiviral-related drug-induced liver injury: Case report. *World J Hepatol*. (2016) 8:858–62. doi: 10.4254/wjh.v8.i20.858
61. Shteyer E, Harel D, Gafanovich I, Dery I, Wolf D, Cottler SJ, et al. End of treatment RNA-positive/sustained viral response in an individual with acute hepatitis C virus infection treated with direct-acting antivirals. *Therap Adv Gastroenterol*. (2017) 10:429–30. doi: 10.1177/1756283X17694808

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Reinharz, Churkin, Dahari and Barash. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.