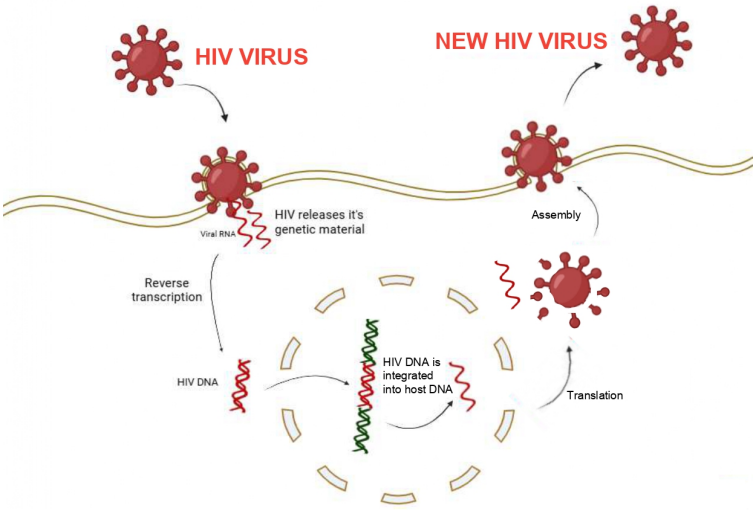
CHALMERS
UNIVERSITY OF TECHNOLOGY



Sensitivity Study of Parameters in a Mathematical Model of HIV Infection Using Clinical Data

Master's thesis in Engineering Mathematics and Computational Science

Messaoud Shaker

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Sensitivity Study of Parameters in a Mathematical Model of HIV Infection Using Clinical Data

Messaoud Shaker

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2025

Sensitivity Study of Parameters in a Mathematical Model of HIV Infection Using Clinical Data

Supervisor: Larisa Beilina, Department of Mathematical Sciences
Examiner: Larisa Beilina, Department of Mathematical Sciences

Master's Thesis 2025
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg

Abstract

Modeling the early stages of HIV infection is essential for understanding virus dynamics and improving treatment strategies. In this project, we solve a system of three ordinary differential equations (ODEs) that describe the interactions between uninfected immune cells, infected cells, and the viral load. Our goal is to reconstruct key parameters and optimize the model using real clinical data from four patients.
The study is divided into two phases. In the first phase, we focus on estimating the viral load $V(t)$ by solving an inverse problem with a time-adaptive optimization method. Two test cases, Test 1 and Test 2, are evaluated, and we find that Test 2 provides better accuracy across all four patients. The results show a good fit between the computed and measured viral load up to day 50, confirming the effectiveness of Test 2 in capturing early infection dynamics.
In the second phase, we extend the model by including the total immune response $\Sigma = T + I$, which represents the sum of uninfected and infected CD4+ T cells. We continue using Test 2, as it provided the best accuracy in the first phase. Additionally, we modify the adjoint equations by incorporating an extra term derived from the adjoint problem to improve the optimization process. The model is applied to clinical data from four patients to analyze how well it reconstructs their immune response and viral load dynamics.
An interesting result emerges from the numerical experiments. In many cases, the optimization improves the fit for both the viral load $V(t)$ and the immune response $\Sigma = T+I$ at the same time. This suggests that the model and the conjugate gradient algorithm can reconstruct both variables in parallel with good accuracy. However, in more challenging situations such as for Patient 1 and 4, it is still difficult to achieve a perfect match for both quantities. These findings highlight the importance of having a good initial guess and confirm the reliability of the optimization method, even though further refinements may still be necessary.
From a theoretical point of view, we implement a time-adaptive reconstruction method that improves numerical stability and allows better control of the parameters. The optimization is based on minimizing a Tikhonov functional, where the regularization term is chosen to balance data fitting and smoothness. Adding extra terms from the adjoint equations improves the reconstruction but also makes it harder to achieve good results for both state variables at the same time.
Overall, our results show that choosing test cases and optimization strategies carefully is important in order to reconstruct both $V(t)$ and $\Sigma = T + I$ effectively. Time adaptivity improves numerical performance, but more improvements are needed to consistently match both outputs. This study contributes to building better mathematical models for HIV infection, which can be useful in clinical prediction and treatment planning.

Acknowledgements

I would like to thank my supervisor, Professor Larisa Beilina, for her guidance and support throughout my master's thesis. Her advice and encouragement have been invaluable.
I also want to express my gratitude to Chalmers Tekniska Högskola and its professors for their mentorship, which has greatly helped me in my studies.
This work is dedicated to my beloved mother, who passed away in the 2023 earthquake in Turkey. She always dreamed of seeing me start and complete my master's, and I carry her dream with me.
A special thanks to my father for his endless love and support, always believing in me.
To my dear wife, who has stood by me through everything—thank you for always being there for me. I would also like to sincerely thank my mother-in-law, who supported me with the same care and kindness as my own mother.
Lastly, I appreciate my friends and family for their constant encouragement.

Messaoud Shaker, Gothenburg, march 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CGA	Conjugate Gradient Algorithm
ODE	Ordinary Differential Equation
PIP	Parameter Identification Problem
LMM	Linear Multistep Method

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

t	Time index
k	Time step index in finite element discretization
n	Iteration index in Newton's method
m	Iteration index in optimization process
i, j	Indices for system variables

Sets

$\mathcal{T}$	Set of time steps in the discretized domain
$\mathcal{X}$	Set of state variables $\{T(t), I(t), V(t)\}$
$\mathcal{L}$	Set of adjoint variables $\{\lambda_T, \lambda_I, \lambda_V\}$

Parameters

β_1	Infection rate of target cells
β_2	Virus production rate from infected cells
μ	Natural death rate of target cells
ρ	Virus production rate by infected cells
c	Clearance rate of free virus particles
$d(t)$	Drug efficacy function
$E(t)$	Time-dependent treatment parameter
τ_k	Time step in finite element discretization

M_n	Maximum iterations in Newton's method
K_n	Maximum iterations for the adjoint system
γ	Regularization parameter in optimization

Variables

$T(t)$	Uninfected target cells
$I(t)$	Infected cells
$V(t)$	Free virus particles
$\lambda_T(t)$	Adjoint variable for target cells
$\lambda_I(t)$	Adjoint variable for infected cells
$\lambda_V(t)$	Adjoint variable for virus particles
$F(T, I, V)$	Right-hand side of the forward system
$\tilde{F}(\lambda_T, \lambda_I, \lambda_V)$	Right-hand side of the adjoint system
$G(E)$	Gradient of the objective function
$J(E)$	Objective function in the optimization

Contents

List of Figures

List of Tables

1
Introduction

HIV remains one of the biggest public health challenges in the world. Since the beginning of the epidemic, it has caused the deaths of around 42.3 million people [1]. Even though treatment options have improved a lot over the years, about 630000 people still died from HIV-related causes in 2023.
By the end of 2023, an estimated 39.9 million people were living with HIV, and 1.3 million new infections were reported globally. The World Health Organization (WHO) African Region continues to be the most affected, accounting for more than 60% of global HIV-related deaths and nearly half of all new infections [1].
There has been a lot of progress. For example, HIV-related deaths have decreased by almost 70% since their peak in 2004 and by 51% since 2010. However, more work is still needed to reach the global goal of ending the AIDS epidemic by 2030.
Even after many years of research and efforts to fight HIV and AIDS, there are still many challenges. A lot of people who are living with HIV, or are at risk of getting it, do not have access to proper prevention, treatment, or medical care. This lack of access creates a big gap in global healthcare. HIV affects not only the health of individuals but also the economy and society, especially in poorer countries where healthcare services are limited. In addition, stigma and discrimination still make it difficult for many people to seek help and receive treatment.
There has been important progress. Over the past ten years, global efforts have helped reduce the number of new HIV infections. Many more people now have access to antiretroviral therapy (ART), which helps them live longer and healthier lives. There have also been major improvements in stopping the spread of HIV from mothers to their babies.
Even with these achievements, problems remain. Many vulnerable people, especially in low-income areas, still struggle to get the healthcare they need because of poverty, weak healthcare systems, and social inequality [2].
In the early stage of HIV infection, called the acute phase, the amount of virus in the blood increases very quickly. This is known as the viral load. During this time, many people get flu-like symptoms such as fever, headache, and swollen lymph nodes. These symptoms usually appear when the virus reaches its highest level in the blood, often exceeding 10^6 RNA copies per millilitre.
After this peak, the viral load starts to decrease. This happens because the immune system begins to fight the virus, and the virus also destroys many of its target cells, like macrophages and CD4+ T cells. Eventually, the amount of virus levels off at a stable point, known as the viral equilibrium point. This marks the beginning of the chronic phase, which can last for many years without symptoms.
It is very important to understand what happens during the acute phase of HIV

infection because this early stage has a big impact on how the disease develops over time. Two important things to look at are how many healthy CD4+ T cells are left at the end of this phase and how high the viral load is. These help predict how the infection might progress. Also, the risk of passing the virus to others is highest during this stage, which is why it is so important to have accurate models and good estimates of the key values in the system.

Mathematical models are very helpful for studying HIV infection. They give us a way to understand how the virus behaves in the body, predict how the disease might develop, and test how different treatments could work. These models use differential equations to describe how the virus interacts with healthy cells and infected cells.

One of the biggest challenges is figuring out the right values for the model parameters. These values need to match real clinical data as closely as possible. When the parameters are accurate, the model gives better predictions and helps us understand the disease more clearly.

This project focuses on using an optimization-based mathematical model to better understand how the immune system responds during the early stage of HIV infection. The model shows how the viral load $V(t)$ interacts with the total number of CD4+ T cells, written as $\Sigma = T + I$, where T is the number of uninfected CD4+ T cells and I is the number of infected CD4+ T cells. This system is modeled using a set of ordinary differential equations (ODEs).

One important feature of this work is the use of a time-adaptive numerical method. This method automatically adjusts the time steps during simulation, based on how accurate the results are. This helps make the calculations both more precise and more efficient.

The project is carried out in two main steps. In the first step, the goal is to estimate the viral load $V(t)$ using two test cases, called test 1 and test 2. These tests are applied to data from four patients. The results show that test 2 works best, so it is used in the second step. In that step, the model is expanded to include both viral load and the total CD4+ T-cell population $\Sigma = T + I$. To improve accuracy, the adjoint equations are updated by adding extra terms from equation (2.14).

This research helps improve how we estimate parameters in HIV models and how we use optimization methods. The results offer useful insights into how well time-adaptive modeling works for understanding HIV infection and supporting treatment evaluation.

2

Mathematical Model and Theoretical Background

2.1 HIV Infection Model

A time-adaptive algorithm for estimating parameters in an HIV infection model was recently developed in [3], based on the model originally proposed in [4]. This model describes the interaction between uninfected target cells (T), infected target cells (I), and viral particles (V).
For this study, we consider the case of HIV-1 infection without treatment, modifying the system from [4] to three equations. The time domain is denoted as $\Omega_t = [0, T_{obs}]$, where T_{obs} is the final observation time.
The system of differential equations considered in this work is:

$$\begin{cases} \dfrac{dT}{dt} = s - \beta_1 T(t)V(t) - \mu T(t), \\ \dfrac{dI}{dt} = \beta_1 T(t)V(t) - d(t)E(t)I(t), \\ \dfrac{dV}{dt} = \rho I(t) - \beta_2 T(t)V(t) - cV(t). \end{cases} \tag{2.1}$$

where:

- $s = 10^4\ ml^{-1} \cdot day^{-1}$ – inflow rate of CD4+ T cells,
- $\beta_1 = 2.4 \times 10^{-8}\ ml \cdot day^{-1}$, $\beta_2 = 2.4 \times 10^{-8}\ ml \cdot day^{-1}$ – virus infectivity rates,
- $\mu = 0.01\ day^{-1}$ – natural death rate of uninfected T cells,
- $d(t) = 0.26\ day^{-1}$ – death rate of infected cells,
- $c = 2.4\ day^{-1}$ – viral clearance rate,
- $\rho = 10^3\ virions \cdot day^{-1}$ – total number of viral particles produced per infected cell.

Note that here we have $d(t)$ as a constant, but it can be a function in time as well.
The initial conditions for this system are given as:

$$T(0) = T^0\ (\text{cells} \cdot ml^{-1}), \quad I(0) = I^0\ (\text{cells} \cdot ml^{-1}), \quad V(0) = V^0\ (\text{copies} \cdot ml^{-1}).$$

We introduce the state variable:

$$X(t) = (T(t), I(t), V(t))$$

and denote

$$f_x = (f_{1x}, f_{2x}, f_{3x})$$

Then the problem 2.1 can be written as:

$$\begin{cases} \dfrac{dT}{dt} = f_{1x} = s - \beta_1 T(t)V(t) - \mu T(t), \\ \dfrac{dI}{dt} = f_{2x} = \beta_1 T(t)V(t) - d(t)E(t)I(t), \\ \dfrac{dV}{dt} = f_{3x} = \rho I(t) - \beta_2 T(t)V(t) - cV(t). \end{cases} \tag{2.2}$$

where

$$\frac{dX}{dt} = f_x, \quad X(0) = X^0, \quad X^0 = (T^0, I^0, V^0).$$

2.2 Parameter Identification Problem (PIP)

The objective of the Parameter Identification Problem (PIP) is to determine the time-dependent immune response function $E(t)$, which is not directly measurable. This function represents the strength of the immune system's response to infection.
Problem Definition: Given clinical time-dependent data for:

- The total CD4+ T cell count:

$$T(t) + I(t) = g_1(t), \quad t \in \Omega_t, \tag{2.3}$$

- The viral load in the bloodstream:

$$V(t) = g_3(t), \quad t \in \Omega_t, \tag{2.4}$$

where $g_1(t)$ and $g_3(t)$ are measured clinical values, the goal is to estimate $E(t)$ in the system (2.2).
To ensure biological plausibility, $E(t)$ is constrained within the following range:

$$M_E = \{E(t) : E(t) \ \in \ [1, 10] \ \text{for all } t \in \Omega_t\}. \tag{2.5}$$

Observations and Time Interval Considerations: The inverse problem is solved over a time interval $[0, \mathbf{T}]$, but clinical data for $g_1(t)$ and $g_3(t)$ may only be available on a shorter observation range $[\mathbf{T}_1, \mathbf{T}_2] \subset [0, \mathbf{T}]$. This limited availability of observations can affect the accuracy of the reconstructed $E(t)$, especially in regions where no data is present.
Numerical results (see Section 4) indicate that the reconstruction of $E(t)$ is less reliable in periods where observations are missing. To improve accuracy, it is crucial to collect viral load measurements as early as possible, ideally from the onset of infection, when the virus first begins replicating in the host.

2.3 Clinical Data and Observations

To estimate key parameters in the mathematical model (2.2), clinical data from four untreated HIV patients were analyzed over a one-year period [6]. The data include measurements of the viral load $V(t)$ and the total CD4+ T cell count $\Sigma = T + I$. These values serve as reference points for validating the model and optimizing the reconstruction process.
The clinical data were collected during both the hyperacute and acute phases of HIV infection, corresponding to Fig. 1-b of [6]. Measurements were taken at eight time points: pre-infection, 0 weeks, 1 week, 2 weeks, 3 weeks, 4 weeks, 6 months, and 1 year. The viral load and CD4+ T cell count for each patient are summarized in Tables 2.1–2.4.

Table 2.1: Clinical data for Patient 1 (from Fig. 1-b of [6]).

Time (weeks)	0	1	2	3	4	6 months	1 year
$\log_{10} V(t)$ (copies/ml^{-1})	0.0	5.5	7.75	6.5	5.5	4.5	5.1
Σ ($\times 10^3$ cells/ml^{-1})	1125	825	675	525	540	550	600

Table 2.2: Clinical data for Patient 2 (from Fig. 1-b of [6]).

Time (weeks)	0	1	2	3	4	6 months	1 year
$\log_{10} V(t)$ (copies/ml^{-1})	0.0	5.5	7.0	5.75	4.8	3.8	4.0
Σ ($\times 10^3$ cells/ml^{-1})	750	630	450	280	750	570	530

Table 2.3: Clinical data for Patient 3 (from Fig. 1-b of [6]).

Time (weeks)	0	1	2	3	4	6 months	1 year
$\log_{10} V(t)$ (copies/ml^{-1})	0.0	5.7	7.4	6.8	4.2	3.8	3.7
Σ ($\times 10^3$ cells/ml^{-1})	700	430	300	480	570	450	510

Table 2.4: Clinical data for Patient 4 (from Fig. 1-b of [6]).

Time (weeks)	0	1	2	3	4	6 months	1 year
$\log_{10} V(t)$ (copies/ml^{-1})	0.0	3.7	5.7	6.0	3.9	3.5	3.0
Σ ($\times 10^3$ cells/ml^{-1})	615	700	450	615	575	520	615

2.4 Mathematical Assumptions

To ensure the stability of the parameter estimation process and numerical simulations, the following assumptions are made:

- The functions defining the system dynamics are assumed to be Lipschitz continuous.

- The immune response function $E(t)$ is restricted to a physiologically meaningful range given in (2.5)

These conditions facilitate the proper numerical treatment of the inverse problem, ensuring that the reconstructed immune response $E(t)$ accurately represents the observed viral load data.

2.5 Optimization Method

To estimate the unknown immune response function $E(t)$, we formulate an optimization problem using a Tikhonov regularization approach, similar to the methods used in [3], [10], and [11]. This method ensures stability in the reconstruction process by incorporating clinical observations while imposing smoothness constraints on $E(t)$.

2.5.1 Tikhonov Functional Formulation

The optimization problem is defined by the Tikhonov functional, which measures the difference or residual between the predicted and observed clinical data:

$$\begin{aligned} J(E) \quad &= \frac{1}{2} \int_{\Omega_t} \left[\log_{10}(T(t) + I(t)) - \log_{10}(g_1(t))\right]^2 z_{1\zeta}(t)\ \mathrm{d}t \\ &+ \frac{1}{2} \int_{\Omega_t} [\log_{10} V(t) - \log_{10} g_2(t)]^2 z_{2\zeta}(t)\ \mathrm{d}t + \frac{1}{2}\gamma \int_{\Omega_t} (E(t) - E^0)^2 dt. \end{aligned} \tag{2.6}$$

In this expression:

- $T(t), I(t), V(t)$ are the model solutions for uninfected cells, infected cells, and viral load, respectively.
- $g_1(t)$ represents the observed total CD4+ T cell population $(T + I)$, and $g_3(t)$ represents the observed viral load $V(t)$.
- E^0 is the initial estimate of $E(t)$, and $\gamma \in (0, 1)$ is a regularization parameter that controls smoothness.
- $z_{1\zeta}(t)$ and $z_{2\zeta}(t)$, with $\zeta \in (0, 1)$, are smoothing functions used to adjust the influence of data points, smoothing functions which can be defined in [11]

2.5.2 Optimization via the Lagrangian Approach

To determine the optimal function $E(t)$, we seek a stationary point of $J(E)$, which satisfies:

$$J'(E)(\bar{E}) = 0, \quad \forall \bar{E} \in H. \tag{2.7}$$

where H is a Hilbert space of admissible solutions.
To determine the minimum of (2.6), we apply a constrained optimization approach using the standard Lagrangian method [8, 12]. The Lagrangian function is introduced as follows:

$$L(v) = J(E) + \int_{\Omega_t} \lambda_x \cdot \left(\frac{dX}{dt} - f_X \right) \, dt, \tag{2.8}$$

where $X(t) = (T(t), I(t), V(t))$ represent the state variable obtained from system (2.2), and $\lambda_x = (\lambda_T, \lambda_I, \lambda_V)$ is the vector of Lagrange multipliers that enforce the system constraints. The optimization variable is now given by:

$$v = (\lambda_x, X, E).$$

Let us define the corresponding functional spaces required for analysis:

$$\begin{aligned} H^1_X(\Omega_t) &= \{X \in H^1(\Omega_t) : X(0) = X_0\}, \\ H^1_{\lambda_x}(\Omega_t) &= \{\lambda \in H^1(\Omega_t) : \lambda(\mathbf{T}) = 0\}, \\ U &= H^1_X(\Omega_t) \times H^1_\lambda(\Omega_t) \times C(\Omega_t), \end{aligned} \tag{2.9}$$

where:

- $X(t) = (T(t), I(t), V(t))$ represents the state variables obtained from system (2.2)
- $\lambda_x(t) = (\lambda_T(t), \lambda_I(t), \lambda_V(t))$ are the Lagrange multipliers enforcing system constraints.
- $X_0 = (T_0, I_0, V_0)$ are the initial conditions.
- $H^1(\Omega_t)$ denotes the Sobolev space of functions with square-integrable derivatives.
- $C(\Omega_t)$ represents the space of continuous functions over the time domain.

where all functions are real-valued.

To derive the Fréchet derivative of the Lagrangian (2.8), we assume that $v = (\lambda_x, T, I, V, E)$ can be independently varied in the sense that:

$$L'(v)(\bar{v}) = 0, \quad \forall \bar{v} = (\bar{\lambda}, \bar{T}, \bar{I}, \bar{V}, \bar{E}) \in U. \tag{2.10}$$

Thus, we compute $L(v + \bar{v}) - L(v)$, extract the linear terms with respect to v, and discard all nonlinear contributions.

The optimality conditions (2.10) lead to the following equations:

$$\begin{aligned} 0 = \frac{\partial L}{\partial \lambda_T}(v)(\bar{\lambda}_T) &= \int_{\Omega_t} \left(\frac{\partial T}{\partial t} - s + \beta_1 TV + \mu T \right) \bar{\lambda}_T \, dt, \quad \forall \bar{\lambda}_T \in H^1_\lambda(\Omega_t), \\ 0 = \frac{\partial L}{\partial \lambda_I}(v)(\bar{\lambda}_I) &= \int_{\Omega_t} \left(\frac{\partial I}{\partial t} - \beta_1 TV + dEI \right) \bar{\lambda}_I \, dt, \quad \forall \bar{\lambda}_I \in H^1_\lambda(\Omega_t), \\ 0 = \frac{\partial L}{\partial \lambda_V}(v)(\bar{\lambda}_V) &= \int_{\Omega_t} \left(\frac{\partial V}{\partial t} - \rho I + cV + \beta_2 TV \right) \bar{\lambda}_V \, dt, \quad \forall \bar{\lambda}_V \in H^1_\lambda(\Omega_t). \end{aligned} \tag{2.11}$$

$$
\begin{aligned}
0 = \frac{\partial L}{\partial T}(v)(\bar{T}) &= \int_{\Omega_t} \frac{1}{(T+I)} \left[\log_{10}(T+I) - \log_{10}(g_1)\right] z_{1\zeta}(t)\, \bar{T}\, dt \\
&+ \int_{\Omega_t} \left(-\frac{\partial \lambda_T}{\partial t} + \lambda_T \beta_1 V + \mu \lambda_T - \lambda_I \beta_1 V + \beta_2 \lambda_V V \right) \bar{T}\, dt, \quad \forall \bar{T} \in H_T^1(\Omega_t), \\
0 = \frac{\partial L}{\partial I}(v)(\bar{I}) &= \int_{\Omega_t} \frac{1}{(T+I)} \left[\log_{10}(T+I) - \log_{10}(g_1)\right] z_{1\zeta}(t)\, \bar{I}\, dt \\
&+ \int_{\Omega_t} \left(-\frac{\partial \lambda_I}{\partial t} + dE\lambda_I - \lambda_V \rho \right) \bar{I}\, dt, \quad \forall \bar{I} \in H_I^1(\Omega_t), \\
0 = \frac{\partial L}{\partial V}(v)(\bar{V}) &= \int_{\Omega_t} \frac{1}{V} \left[\log_{10}(V) - \log_{10}(g_2)\right] z_{2\zeta}(t)\, \bar{V}\, dt \\
&+ \int_{\Omega_t} \left(-\frac{\partial \lambda_V}{\partial t} + \lambda_T \beta_1 T - \lambda_I \beta_1 T + c\lambda_V + \lambda_V \beta_2 T \right) \bar{V}\, dt, \quad \forall \bar{V} \in H_V^1(\Omega_t).
\end{aligned}
\tag{2.12}
$$

$$
0 = \frac{\partial L}{\partial E}(v)(\bar{E}) = \gamma \int_{\Omega_t} (E - E^0)\bar{E}\, dt + \int_{\Omega_t} d\lambda_I I \bar{E}\, dt, \quad \forall \bar{E} \in C(\Omega_t). \tag{2.13}
$$

Using the optimality conditions (2.11)-(2.12), we observe that the equations in (2.11) correspond to the system of model equations (2.1), which defines the forward problem. The equations in (2.12) correspond to the following adjoint problem:

$$
\begin{aligned}
\frac{\partial \lambda_T}{\partial t} &= \lambda_T(t)\beta_1 V(t) + \mu\lambda_T(t) - \lambda_I(t)\beta_1 V(t) + \beta_2 \lambda_V(t) V(t) \\
&+ \frac{1}{(T(t)+I(t))} [\log_{10}(T(t)+I(t)) - \log_{10}(g_1)] z_{1\zeta}, \\
\frac{\partial \lambda_I}{\partial t} &= -\lambda_V(t)\rho + d\lambda_I(t) E(t) \\
&+ \frac{1}{(T(t)+I(t))} [\log_{10}(T(t)+I(t)) - \log_{10}(g_1)] z_{1\zeta}, \\
\frac{\partial \lambda_V}{\partial t} &= \lambda_T(t)\beta_1 T(t) - \lambda_I(t)\beta_1 T(t) + c\lambda_V(t) + \lambda_V(t)\beta_2 T(t) \\
&+ \frac{1}{V(t)} [\log_{10}(V(t)) - \log_{10}(g_2)] z_{2\zeta}, \\
\lambda_X(\mathbf{T}) &= 0, \quad X \in \{T, I, V\}.
\end{aligned}
\tag{2.14}
$$

This system can be rewritten in a compact form as:

$$\begin{cases} \frac{\partial \lambda_x}{\partial t} = \tilde{f}(\lambda_x(t), E(t)), \\ \lambda_X(\mathbf{T}) = 0. \end{cases} \tag{2.15}$$

where:

$$\begin{aligned} \lambda_x(t) &= (\lambda_T(t), \lambda_I(t), \lambda_V(t))^T, \\ 0 &= (\lambda_T(\mathbf{T}), \lambda_I(\mathbf{T}), \lambda_V(\mathbf{T}))^T, \\ \frac{d\lambda_x}{dt} &= \left(\frac{\partial \lambda_T}{\partial t}, \frac{\partial \lambda_I}{\partial t}, \frac{\partial \lambda_V}{\partial t}\right)^T, \\ \tilde{f}(\lambda_x) &= (\tilde{f}_T, \tilde{f}_I, \tilde{f}_V)^T (\lambda_x(t), E(t))^T. \end{aligned} \tag{2.16}$$

Since the adjoint system must be solved backward in time, it requires the previously computed forward solution $(T(t), I(t), V(t))$ from (2.2) and the known clinical measurements $g_1(t)$ and $g_3(t)$.
Using the definition of the Tikhonov functional (2.6), we observe that for all known parameters, we can derive from (2.8):

$$L(v(E)) = J(E), \tag{2.17}$$

which holds when (T, I, V) and λ satisfy the forward system (2.2) and the adjoint system (2.14), respectively. Under these conditions, the Fréchet derivative of the Tikhonov functional takes the following form:

$$J'(E) := J_E(T(E), I(E), V(E), E) = \frac{\partial J}{\partial E}(T(E), I(E), V(E), E) = \frac{\partial L}{\partial E}(v(E)). \tag{2.18}$$

Substituting (2.13) into (2.18), we derive the following expression for the Fréchet derivative of the Tikhonov functional, which will later be utilized in the conjugate gradient update for $E(t)$:

$$J'(E)(t) = \gamma(E - E^0) + d\lambda_I(t) I(t) = 0. \tag{2.19}$$

3

Numerical Techniques for Solving the Optimization-Based Inverse Problem

To simplify calculation, we define the state variables as:

$$x_1(t) = T(t), \quad x_2(t) = I(t), \quad x_3(t) = V(t). \tag{3.1}$$

Then the system 2.1 led to the following system:

$$\begin{cases} \dfrac{dx_1}{dt} = f_1 = s - \beta_1 x_1(t) x_3(t) - \mu x_1(t), \\ \dfrac{dx_2}{dt} = f_2 = \beta_1 x_1(t) x_3(t) - d(t) E(t) x_2(t), \\ \dfrac{dx_3}{dt} = f_3 = \rho x_2(t) - \beta_2 x_1(t) x_3(t) - c x_3(t), \\ x_1(0) = x_1^0 \ ml^{-1}, \\ x_2(0) = x_2^0 \ ml^{-1}, \\ x_3(0) = x_3^0 \ ml^{-1}. \end{cases} \tag{3.2}$$

This system can be written as:

$$\frac{dx}{dt} = f, f = (f_1, f_2, f_3). \tag{3.3}$$

We set $x(0) = x^0$.

3.0.1 Time Discretization

To solve the minimization problem (2.10), we define the computational domain as $\Omega_t = [0, \mathbf{T}]$ and discretize it by dividing the time interval into a set of subintervals $J_\tau = \{J_k\}$, where each subinterval is defined as $J_k = (t_{k-1}, t_k]$, with a corresponding time step size.

$$\tau_k = t_k - t_{k-1}, \quad \forall J_k \in J_\tau. \tag{3.4}$$

To approximate time derivatives, we use the first order approximation of the time derivative.

$$\frac{\partial x_i}{\partial t} \approx \frac{x_i^{k+1} - x_i^k}{\tau_k}, \quad i = 1, 2, 3. \tag{3.5}$$

3.0.2 Newton's Method for Solving the Forward Problem

The forward problem (3.2) is nonlinear, and while more efficient methods exist, Newton's method is used here due to the complexity of the nonlinearity and its simplicity. The variational formulation of the problem (3.2) or (3.3) is:

$$\left(\frac{dx}{dt}, \bar{x}\right) = (F(x), \bar{x}), \quad \forall \bar{x} \in H_x^1(\Omega_t). \tag{3.6}$$

Using the discretization in time (3.5), we obtain the iterative system in time:

$$\left(x^{k+1} - \tau_k F(x^{k+1}) - x^k, \bar{x}\right) = 0, \quad \forall \bar{x} \in H_x^1(\Omega_t). \tag{3.7}$$

To apply Newton's method for (3.11), we introduce $\bar{x} = x^{k+1}$, such that (3.11) transforms to:

$$(\bar{x} - \tau_k F(\bar{x}) - x^k, \bar{x}) = 0 \tag{3.8}$$

Then, denoting

$$V(\bar{x}) = \bar{x} - \tau_k F(\bar{x}) - x^k, \tag{3.9}$$

the Newton's method iterates as:

$$(V(\bar{x}), \bar{x}) = 0, \tag{3.10}$$

$$V(\bar{x}) = 0. \tag{3.11}$$

The Newton's method applied to (3.5) is given by:

$$\tilde{x}^{n+1} = \tilde{x}^n - [V'(\tilde{x}^n)]^{-1} \cdot V(\tilde{x}^n), \quad n = 1, \ldots, M_n, \tag{3.12}$$

where:

$$V'(\tilde{x}^n) = I - \tau_k F'(\tilde{x}^n).$$

Here, n is the number of iterations in Newton's method.
The Jacobian for the system is:

$$F'(\tilde{x}^n) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} (\tilde{x}^n) = \begin{bmatrix} -\beta_1 {x_3}_\tau^n - \mu & 0 & -\beta_1 {x_1}_\tau^n \\ \beta_1 {x_3}_\tau^n & -d\ E & \beta_1 {x_1}_\tau^n \\ -\beta_2 {x_3}_\tau^n & \rho & -\beta_2 {x_1}_\tau^n - c \end{bmatrix}.$$

To provide a more biologically meaningful representation, we substitute the variables x_1, x_2, x_3 with their respective interpretations:

$$x_1 \to T(t), \quad x_2 \to I(t), \quad x_3 \to V(t),$$

Applying this transformation, we obtain the updated Jacobian matrix:

$$F'(\tilde{T}, \tilde{I}, \tilde{V}) = \begin{bmatrix} -\beta_1 V_\tau^n - \mu & 0 & -\beta_1 T_\tau^n \\ \beta_1 V_\tau^n & -d\ E & \beta_1 T_\tau^n \\ -\beta_2 V_\tau^n & \rho & -\beta_2 T_\tau^n - c \end{bmatrix}.$$

3.0.3 Variational Formulation of the Adjoint Problem

The adjoint problem (2.14) is solved backwards in time with the known terminal condition $\lambda(\mathbf{T}) = 0$. The finite element method applies the discretization:

$$(\lambda_X^k - \lambda_X^{k+1} + \tau_k \tilde{F}_X(\lambda_T^k, \lambda_I^k, \lambda_V^k), \bar{\lambda}_X) = 0, \quad \forall \bar{\lambda}_X \in H^1_{\lambda_X}(\Omega_t), \quad X \in \{T, I, V\}. \tag{3.13}$$

Newton's iterations are:

$$\tilde{\lambda}_X^{n+1} = \tilde{\lambda}_X^n - [R'(\tilde{\lambda}_X^n)]^{-1} \cdot R(\tilde{\lambda}_X^n), \quad n = 1, \ldots, K_n. \tag{3.14}$$

Following the same procedure as in $F'(\tilde{x}^n)$, we obtain the Jacobian matrix for the adjoint system:

$$\tilde{f}'(\tilde{\lambda}^n) = \begin{bmatrix} \frac{\partial \tilde{f}_T}{\partial \lambda_T} & \frac{\partial \tilde{f}_T}{\partial \lambda_I} & \frac{\partial \tilde{f}_T}{\partial \lambda_V} \\ \frac{\partial \tilde{f}_I}{\partial \lambda_T} & \frac{\partial \tilde{f}_I}{\partial \lambda_I} & \frac{\partial \tilde{f}_I}{\partial \lambda_V} \\ \frac{\partial \tilde{f}_V}{\partial \lambda_T} & \frac{\partial \tilde{f}_V}{\partial \lambda_I} & \frac{\partial \tilde{f}_V}{\partial \lambda_V} \end{bmatrix} (\tilde{\lambda}^n) = \begin{bmatrix} \beta_1 V_\tau^n + \mu & -\beta_1 V_\tau^n & \beta_2 V_\tau^n \\ 0 & dE & -\rho \\ \beta_1 T_\tau^n & -\beta_1 T_\tau^n & c + \beta_2 T_\tau^n \end{bmatrix}.$$

3.0.4 Optimization Method for Estimating $E(t)$

The gradient descent update is obtained from 2.19 and at iteration method. The method written as follows:

$$G^m(t_i) = \gamma(E_\tau^m(t_i) - E_\tau^0(t_i)) + d(t_i)\ {\lambda_2}_\tau^m(t_i) {x_2}_\tau^m(t_i). \tag{3.15}$$

If all the model parameters are assumed to be nonzero, then the Jacobian matrices in the forward and adjoint systems remain nonsingular. In other words, the determinants satisfy $\det F'(x^n) \neq 0$ and $\det \tilde{F}'(\lambda^n) \neq 0$. As a result, the numerical schemes used in equations (3.12) and (3.14) will converge, provided that the initial values x^0 and λ_x^0 are chosen appropriately.

Theoretical studies have shown that gradient-based and Newton-type optimization methods can converge efficiently under certain conditions; see [7]. One of the most important requirements is that the initial guess for the objective function to be reconstructed (in our case, the initial guess E^0 for E) is sufficiently close to the true solution. This helps ensure that the iterative method remains within the region of convergence. Further details on convergence theory and optimization techniques can be found in [9], [13], [14], and [15].

3.1 Conjugate Gradient Algorithm (CGA)

The Conjugate Gradient Algorithm (CGA) is used to solve the optimization problem by iteratively updating the control variable $E(t)$. The main steps of the algorithm are presented below.

Algorithm 1 Conjugate Gradient Algorithm (CGA)

1: Define a time partition J_τ over the interval $(0, \mathbf{T})$. Initialize the control parameter E_τ^0.

2: Solve the forward problem $x_\tau^m = x_\tau(t, E_\tau^m)$ and the adjoint problem $\lambda_\tau^m = \lambda_\tau(t, E_\tau^m)$ using Newton's method and time discretization.

3: Compute the gradient $G^m(t_i)$ at observation points using equation (3.17).

4: Update the parameter $E(t)$ iteratively:

$$E_\tau^{m+1}(t_i) = E_\tau^m(t_i) + r^m d^m(t_i),$$

where the search direction $d^m(t_i)$ is determined as:

$$d^m(t_i) = -G^m(t_i) + \beta^m d^{m-1}(t_i),$$

with the step size:

$$r^m = -\frac{(G^m, d^m)}{\gamma \|d^m\|^2}. \tag{3.16}$$

The conjugate coefficient β^m is given by:

$$\beta^m = \frac{||G^m(t_i)||^2}{||G^{m-1}(t_i)||^2}.$$

5: Choose a tolerance $0 < \theta < 1$. The iteration continues until one of the following stopping criteria is met:

- $||G^m||_{L_2(\Omega_t)} \leq \theta$, where θ is a predefined tolerance.
- A sudden increase in gradient norm $||G^m||_{L_2(\Omega_t)}$.
- The parameter values $||E_\tau^m||_{L_2(\Omega_t)}$ stabilize over iterations.

Otherwise, update $m := m + 1$ and repeat from Step 2.

3.2 Optimization Strategy for the Parameter Identification Problem (PIP)

To estimate the control function $E(t)$, we use the **Conjugate Gradient Algorithm (CGA)** with a fixed (uniform) time step.
In the CGA method, the time grid is uniform, meaning that each time interval J_k has the same time step τ_k.
To monitor the progress of optimization, the gradient at each iteration m is calcu-

lated at the observation points $\{t_i\}$ using the formula:

$$G^m(t_i) = \gamma \left(E_\tau^m(t_i) - E_\tau^0(t_i) \right) + d(t_i) \cdot \lambda_{I,\tau}^m(t_i) \cdot I_\tau^m(t_i), \tag{3.17}$$

Here, $I_\tau^m(t_i)$ is the number of infected CD4+ T cells and $\lambda_{I,\tau}^m(t_i)$ is the corresponding adjoint variable at time t_i, both obtained by solving the forward and adjoint problems with the updated parameter $E := E_\tau^m$.

3.2.1 How the Newton Method Helps

To solve the problem, we use Newton's method to compute the forward and adjoint solutions, and we use the gradient to update the control function $E(t)$. The forward problem (3.2) is nonlinear, and while more efficient methods exist, we choose Newton's method because it handles the complexity of the nonlinearity well.

4

Results

This chapter presents the results of numerical experiments that evaluate the performance of the conjugate gradient algorithm (CGA) in solving two inverse problems: estimating the immune response parameter $E(t)$ in PIP1. All computations were performed in MATLAB R2024b.

For these experiments, clinical data for the viral load $V(t)$ were compared to observed values $g_3(t)$ for four patients. As shown in Tables 2.1, 2.2, 2.3, and 2.4, the available data points were limited to eight specific time instances per patient. Since numerical simulations require data at daily intervals, the weekly observations were interpolated into daily values using the method described in Subsection 4.0.1. After this preprocessing, all tests were conducted on a refined time grid over the interval $[0, \mathbf{T}] = [0, 363]$.

To ensure good convergence of the CGA, the initial guess for the control parameter $E^0(t)$ was chosen to be as close as possible to its expected value. In addition, parameters such as t_1, t_2, ΔT_1, and ΔT_2 were carefully adjusted to fit the clinical data well. These choices play an important role in improving the quality and reliability of the reconstructed solutions.

All simulations were carried out using a tolerance of $\Theta = 10^{-12}$ in the CGA method. To evaluate the accuracy of the results, the relative error between the computed viral load $V(t)$ and the observed data $g_3(t)$ was calculated using the following formula:

$$e = \frac{\| \log_{10} V(t) - \log_{10} g_3(t) \|_{L_2(\Omega_t)}}{\| \log_{10} V(t) \|_{L_2(\Omega_t)}},$$

where Ω_t is the time interval used in the computations.

Finally, an iterative procedure similar to the one used in [3] was employed to update the regularization parameter γ, starting from an initial value $\gamma_0 = 0.1$, to ensure a good trade-off between stability and accuracy during the optimization.

4.0.1 Interpolation of Data

To convert weekly clinical data into daily values over one year, we use linear spline interpolation on each time interval $[t_k, t_{k+1}]$.

Linear interpolation is a simple method used to estimate unknown values between two known data points. Suppose we have data points (t_k, y_k), where $k = 1, \ldots, m$. The goal is to construct a linear polynomial $p(t) = x_1 + x_2 t$ such that $p(t_k) = y_k$.

For a specific interval $[t_1, t_2]$ with known values y_1 and y_2, we set up the following

system:

$$x_1 + x_2 t_1 = y_1,$$
$$x_1 + x_2 t_2 = y_2.$$

Solving this gives the coefficients:

$$x_1 = \frac{t_2 y_1 - t_1 y_2}{t_2 - t_1},$$
$$x_2 = \frac{y_2 - y_1}{t_2 - t_1}.$$

Using these, the interpolating polynomial becomes:

$$p(t) = \frac{t_2 y_1 - t_1 y_2}{t_2 - t_1} + \frac{y_2 - y_1}{t_2 - t_1}(t - t_1).$$

More generally, for any interval $[t_k, t_{k+1}]$, we use:

$$p(t) = y_k + (t - t_k)\frac{y_{k+1} - y_k}{t_{k+1} - t_k}.$$

In this study, linear interpolation is chosen because the data span one year, which typically consists of 52 weeks or 364 days (assuming 7 days per week). Since we only have 8 real observation points but need 364 values for daily-scale simulation, the data is interpolated in two steps: first from 8 observations to 52 weekly values, then from 52 weeks to 364 daily values.

4.0.1.1 Daily Interpolation of $\log_{10} V(t)$

Figure 4.1 illustrates the interpolation process for the viral load data of Patients 1–4. The **blue lines** represent the original $\log_{10} V(t)$ data sampled at weekly intervals, while the **magenta circles** indicate the interpolated $\log_{10} V(t)$ values at daily intervals.

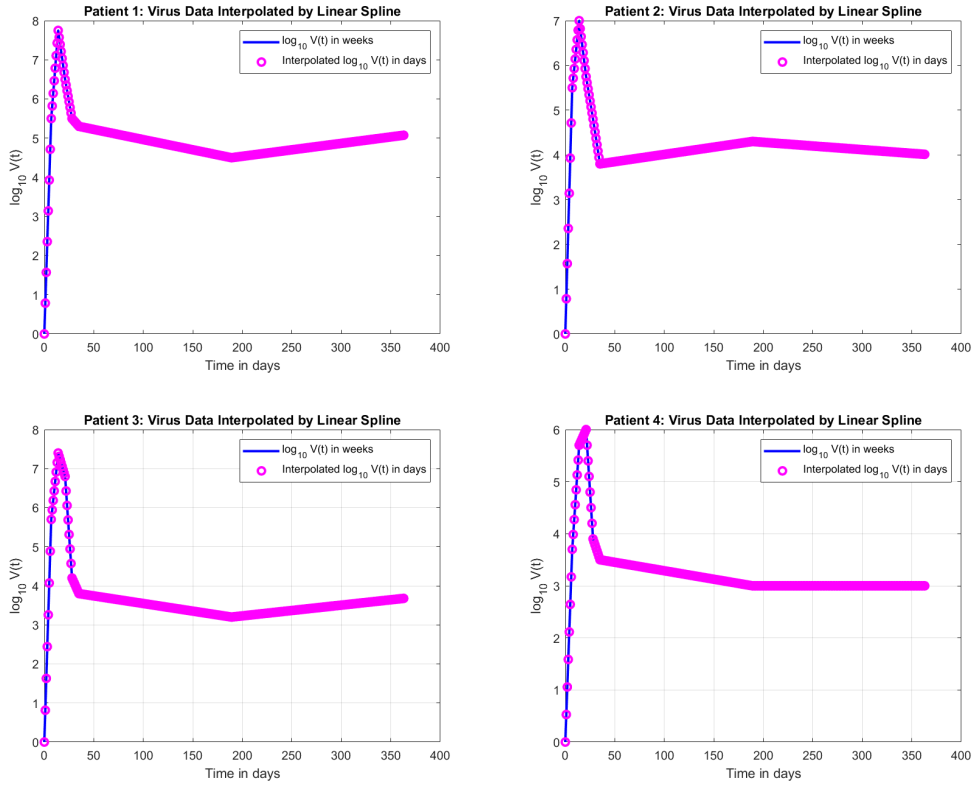


Figure 4.1: Interpolated virus data for Patients 1–4. The blue line represents weekly $\log_{10} V(t)$ data, while the magenta circles represent daily interpolated $\log_{10} V(t)$ values.

4.0.1.2 Daily Interpolation of $\log_{10} \Sigma = T + I$

Figure 4.2 illustrates the linear spline interpolation process for the total CD4+ T-cell population $\Sigma = T + I$ in $\log_{10}$ scale for Patients 1–4. The **blue lines** show the original weekly $\log_{10} \Sigma$ values, while the **magenta circles** represent the interpolated values at daily resolution.

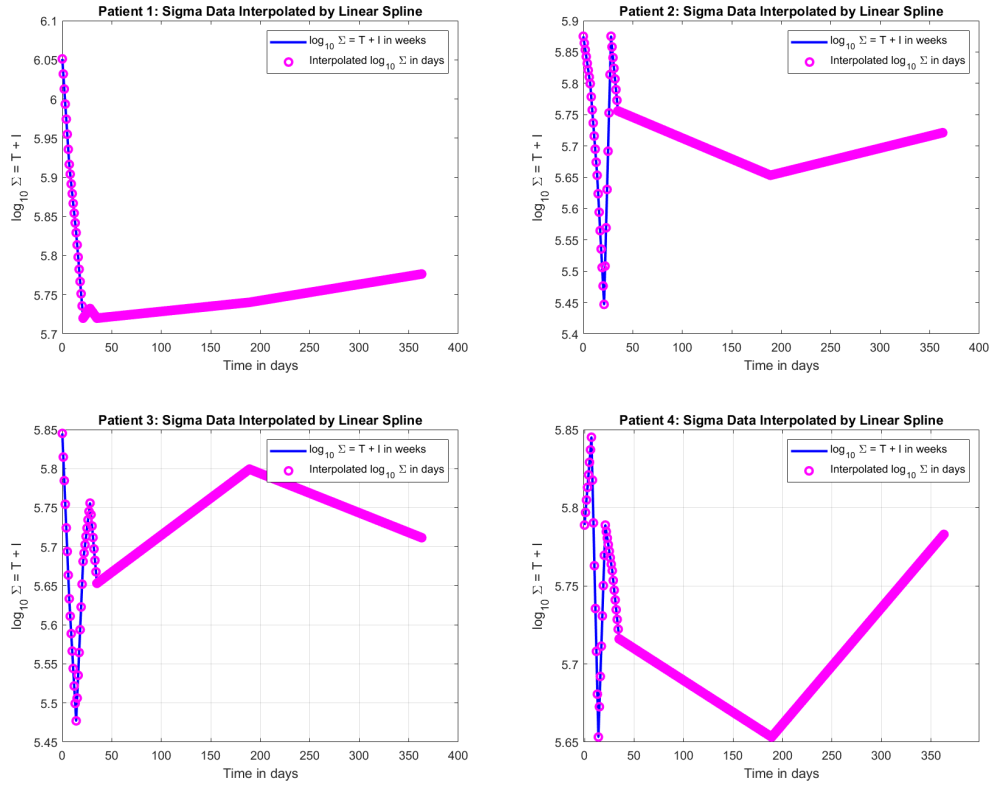


Figure 4.2: Interpolated $\log_{10}\Sigma = T + I$ data for Patients 1–4. The blue lines represent weekly observations, and the magenta circles show the daily interpolated values using linear spline.

4.0.1.3 Daily interpolation of $\log_{10} V(t)$ and $\log_{10}(\Sigma = (T + I))$

Figure 4.3 illustrates the interpolation process for the viral load and immune response data of Patients 1–4. The **red lines** represent the original $\log_{10}(\Sigma = (T + I))$ data sampled at weekly intervals, while the **blue lines** represent the original $\log_{10}(V(t))$ data. The **red circles** indicate the interpolated $\log_{10}(\Sigma = (T + I))$ values at daily intervals, whereas the **blue circles** show the interpolated $\log_{10}(V(t))$ values at daily intervals.

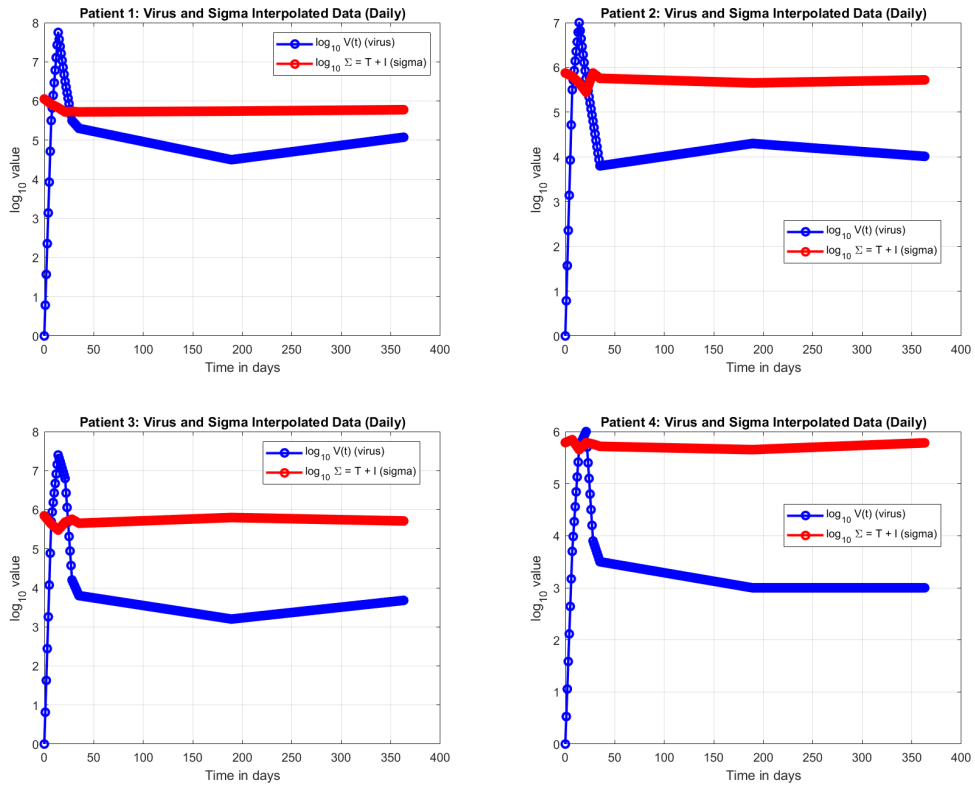


Figure 4.3: Comparison of daily interpolated $\log_{10}(\Sigma)$ vs. $\log_{10}(V(t))$ for multiple patients. The red curves represent $\log_{10}(\Sigma)$, while the blue curves represent $\log_{10}(V(t))$.

4.1 Numerical Solution of the Forward Problem and Reconstruction Results of the First Phase: Modeling the Viral Load $V(t)$

To study the forward problem of the model, we tested different values of the immune response parameter $E(t)$, chosen within the allowable range defined earlier in Equation (2.6). The purpose of this step is to analyze how the solution behaves depending on whether or not the cytotoxic T lymphocyte (CTL) response is included in the dynamics.
The following initial conditions were used:

$$T(0) = 1125000, \quad I(0) = 0, \quad V(0) = 1. \tag{2.2}$$

The fixed model parameters are:

$$s = 10000, \quad \mu = 0.01, \quad \beta_1 = \beta_2 = 2.4 \times 10^{-8}, \quad d = 0.26, \quad c = 2.4, \quad \rho = 1000. \tag{2.3}$$

As described earlier, the project is structured in two phases. The first phase focuses on reconstructing the viral load $V(t)$ using a time-adaptive numerical method. In this phase, two test cases, test 1 and test 2, were selected and applied across all four patients.

For this initial stage, only the second term of the Tikhonov functional (see Equation 2.6) was used in the optimization process. Furthermore, only the terms shown in black in the adjoint system (Equation 2.14) were included in the computations. This means that the red terms, which represent the influence of the total CD4+ T-cell population $\Sigma = T(t) + I(t)$, were not yet considered in the model.

4.1.1 Test 1

In our first experiment, called Test 1, we define the immune response function $E(t)$ using the Cytotoxic T Lymphocyte (CTL) response, based on the model proposed in [5]. This model incorporates the logarithm of the viral load $V(t)$ and is given by the following equations:

$$d \cdot E(t) = d_0 + d_1(t, V(t)), \tag{8.59a}$$

$$d_1(t, V(t)) = \begin{cases} 0, & t < t_1, \\ f(t) \log_{10}(V(t)), & t \geq t_1, \end{cases} \tag{8.59b}$$

$$f(t) = \frac{\beta_{\text{CTL}}}{1 + \kappa e^{-(t-t_1)/\delta T_1}} - \frac{\beta_{\text{CTL}}}{1 + \kappa e^{-(t-t_2)/\delta T_2}}, \tag{8.59c}$$

where $\kappa = 1 + 10^5 \beta_{\text{CTL}}$.

The parameters used in this test are:

- $d_0 = d = 0.26$,
- $\beta_{\text{CTL}} = 0.125$,
- $\delta T_1 = 2.5, \quad \delta T_2 = 5$,
- $t_1 = 1, \quad t_2 = 0$.

Figure 4.4 shows the results of Test 1 for Patient 1. The plots include the immune response function $E(t)$, the weighting function $f(t)$, the resulting $d(t)$, and a comparison of the simulated viral load against the clinical data.

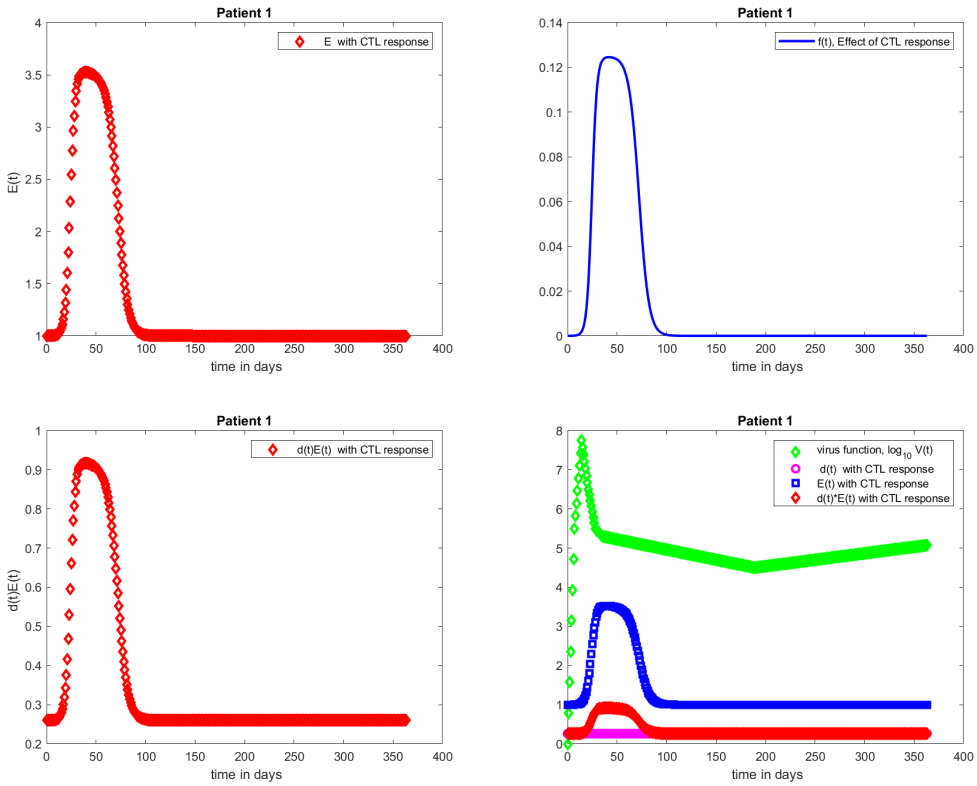


Figure 4.4: Test 1: Immune response function $E(t)$, modulation function $f(t)$, combined profile of $E(t)$ and $d(t)$, and comparison between computed and clinical viral load for Patient 1.

4.1.1.1 Patient 1: Initial Value Analysis of E

The updated results for Test 1 show that there is no significant difference in the viral load $V(t)$ when comparing the cases $E = 1$ and $E = 10$. As shown in Figure 4.5, the left panel corresponds to $E = 1$, and the right panel to $E = 10$. In both cases, the computed $V(t)$ remains nearly identical, indicating that the model is not sensitive to large changes in the initial value of E. Furthermore, the computed $V(t)$ does not align well with the observed data $g_3(t)$, particularly during the early phase, suggesting that simply adjusting E has limited impact on improving the model fit.

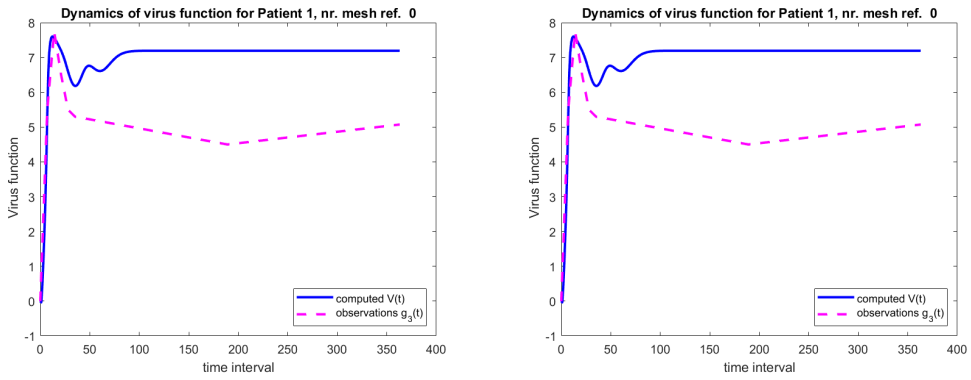


Figure 4.5: Dynamics of the viral load $V(t)$ for Patient 1 in Test 1. The left plot corresponds to $E = 1$, and the right plot to $E = 10$.

The results for Test 1 with $t_1 = 33$ show that the computed $V(t)$ still does not match the observed data $g_3(t)$. Figure 4.6 shows two plots: (a) the viral load $V(t)$ together with the dynamics of $d(t)$, $E(t)$, and $d(t)E(t)$, and (b) the comparison between computed $V(t)$ and the observed data $g_3(t)$.

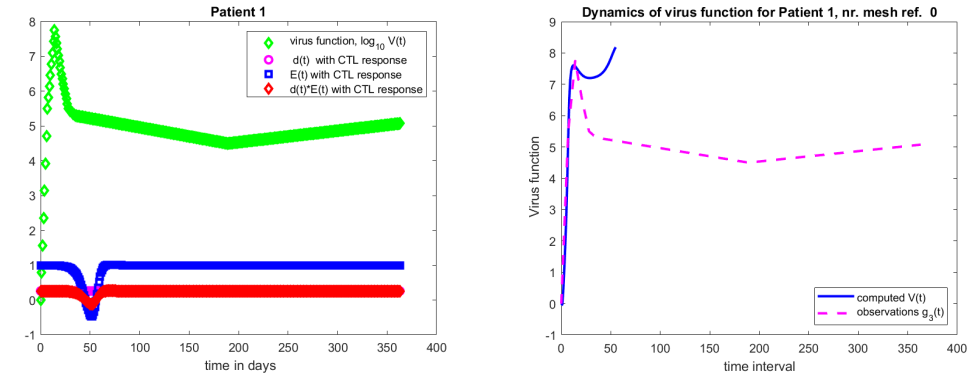


Figure 4.6: Results for Patient 1 with $t_1 = 33$. The left plot shows the viral load $V(t)$ and parameter dynamics, while the right compares computed $V(t)$ to observed data $g_3(t)$.

The results with $t_2 = 33$ show minor differences in $V(t)$, but the computed solution still fails to match $g_3(t)$. As shown in Figure 4.7, increasing t_2 introduces slight changes, but the alignment remains poor. This suggests that both t_1 and t_2 should be chosen small, with t_1 having a stronger effect on accuracy.

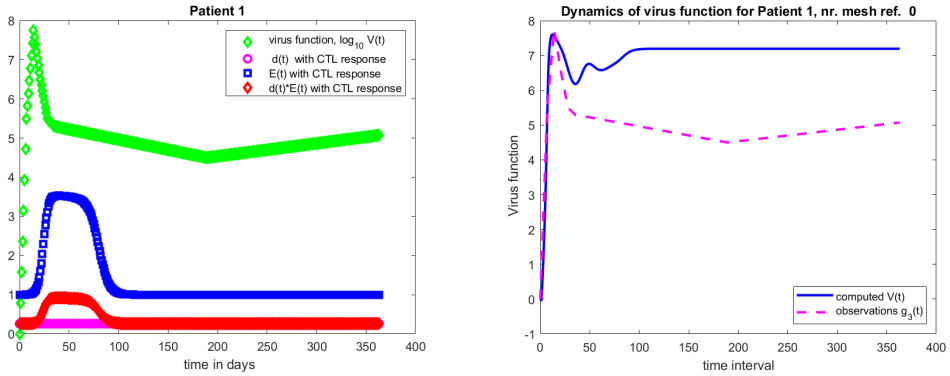


Figure 4.7: Results for Patient 1 with $E = 1$, $t_1 = 1$, and $t_2 = 33$. The left plot shows $V(t)$ and parameter dynamics, while the right compares $V(t)$ to observed data $g_3(t)$.

Choosing $\Delta t_1 = 1.3$ gives the best fit between $V(t)$ and $g_3(t)$. Increasing Δt_1 to 5 significantly worsens the result. As shown in Figure 4.8, the first plot to the left corresponds to $\Delta t_1 = 1.3$, and the second to the right corresponds to $\Delta t_1 = 5$. With the smaller value, $V(t)$ aligns well with $g_3(t)$, especially during the first 50 days. A larger Δt_1 leads to more noticeable deviation between the model and the observed data.

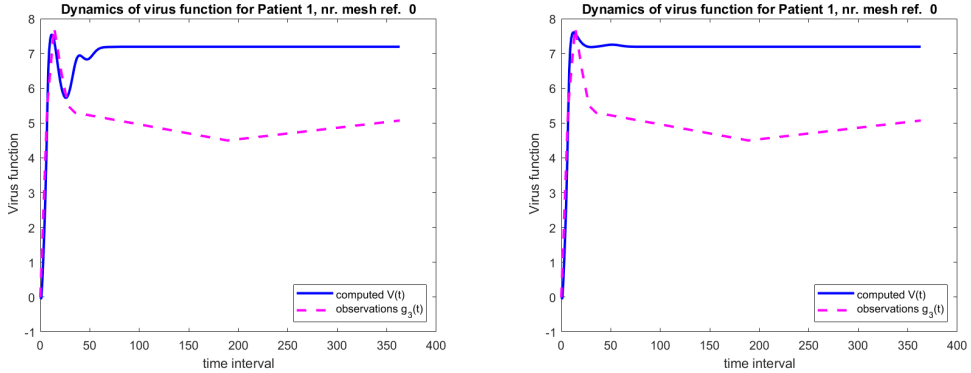


Figure 4.8: Impact of Δt_1 on the fit between $V(t)$ and $g_3(t)$. The left plot corresponds to $\Delta t_1 = 1.3$, and the right plot to $\Delta t_1 = 5$.

The results suggest that while modifying Δt_2 does have some effect on the viral load $V(t)$, its impact is noticeably less significant compared to Δt_1. Figure 4.9 presents a comparison between two cases: in the left panel, $\Delta t_2 = 15$; in the right panel, $\Delta t_2 = 20$. In both cases, Δt_1 is fixed at 1.3.

As Δt_2 increases, the model shows a slightly better match with the observed data $g_3(t)$ during the early phase. However, as time progresses, the viral load curve becomes flatter, which limits the model's ability to reflect realistic viral dynamics. This highlights the importance of choosing Δt_2 carefully and in balance with Δt_1 to maintain both a good early fit and dynamic behavior over time.

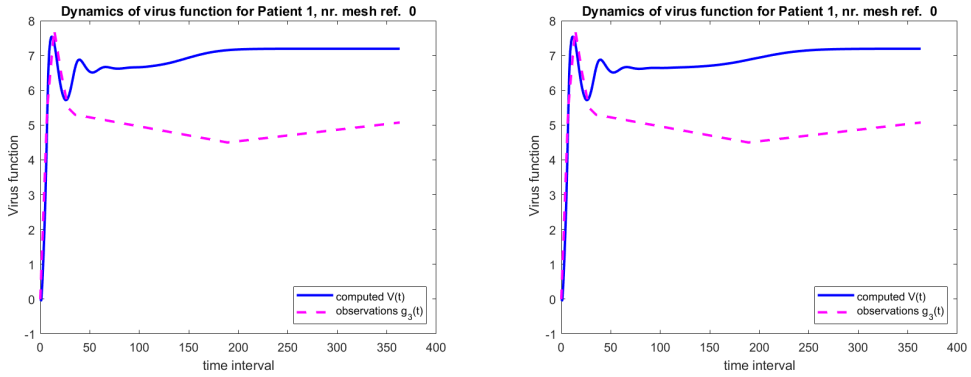


Figure 4.9: Effect of increasing Δt_2 on the viral load $V(t)$. The left plot shows the result for $\Delta t_2 = 15$, while the right plot shows $\Delta t_2 = 20$. In both cases, $\Delta t_1 = 1.3$.

4.1.1.2 Patient 2: Initial Value Analysis of E

For Patient 2, changes in the parameter E have a clear impact on the computed viral load $V(t)$. Figure 4.10 presents the simulation results for three different values of E: 1 (left plot), 2.25 (middle plot), and 3 (right plot).

Out of these, the value $E = 2.25$ provides the best match with the observed viral load $g_3(t)$, particularly during the first 50 days. Increasing E to 3 introduces more pronounced oscillations in $V(t)$ after day 50 and worsens the fit during the early phase. Conversely, choosing $E = 1$ results in an underestimation of the viral load during the same initial period.

These findings highlight that the model is sensitive to the choice of E for Patient 2. Careful tuning of this parameter is essential, with $E = 2.25$ appearing to be the most effective in replicating the clinical data.

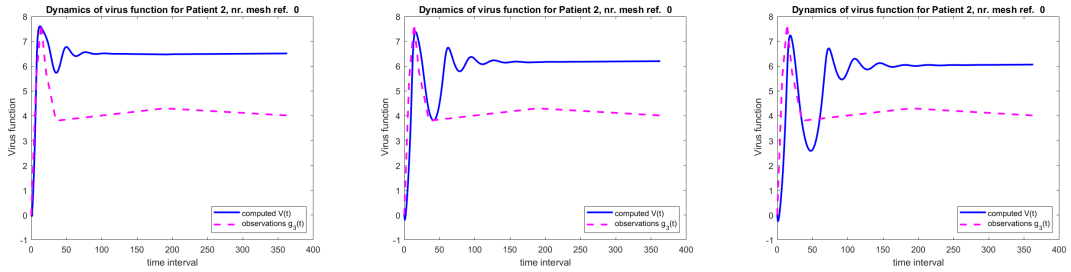


Figure 4.10: Computed viral load $V(t)$ for Patient 2 for three different values of E. The left plot shows the result for $E = 1$, the middle plot for $E = 2.25$, and the right plot for $E = 3$. The best fit with the observed data $g_3(t)$ is achieved with $E = 2.25$, especially during the first 50 days.

4.1.1.3 Patient 3: Initial Value Analysis of E

For Patient 3, the parameter E has a strong influence on the behavior of the computed viral load $V(t)$. Figure 4.11 displays the results for three different values of E: 1 (left plot), 2.35 (middle plot), and 3 (right plot).
Among these, $E = 2.35$ provides the best agreement with the observed data $g_3(t)$, particularly during the first 50 days. When E is increased beyond 2.3, the model begins to deviate from the observed values, and the early-phase fit becomes worse. Similarly, a lower value of $E = 1$ underestimates the viral load compared to the observations.
These results suggest that choosing $E = 2.35$ offers the most accurate representation of the early viral dynamics for Patient 3 under the current modeling approach.

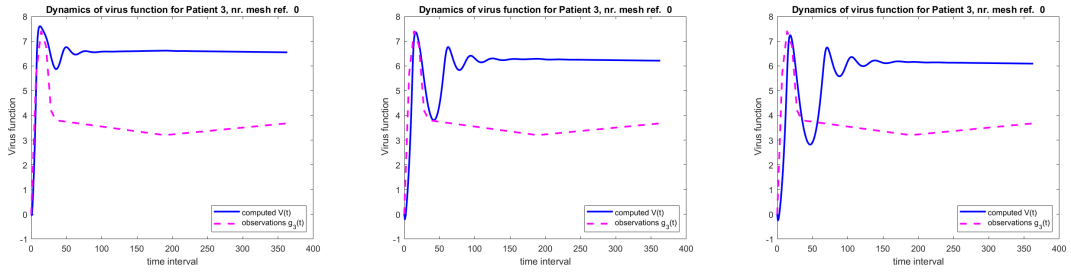


Figure 4.11: Computed viral load $V(t)$ for Patient 3 using different values of E. The left plot shows $E = 1$, the middle plot shows $E = 2.35$, and the right plot shows $E = 3$. The case $E = 2.35$ provides the best fit with the observed data $g_3(t)$ during the first 50 days.

4.1.1.4 Patient 4: Initial Value Analysis of E

For Patient 4, we tested the dynamics of the viral load $V(t)$ under different initial values of the parameter E using Test 1. The parameters used in this test were fixed at $t_1 = 1$, $t_2 = 0$, $\delta T_1 = 2.5$, and $\delta T_2 = 5$. The aim was to identify which initial value of E would produce a viral load that best matches the observed data $g_3(t)$.
Figure 4.12 presents the results for four different initial values: $E = 1.5$ (top left), $E = 2.8$ (top right), $E = 3.0$ (bottom left), and $E = 4.5$ (bottom right). Among

these, the best match with the observed data $g_3(t)$ is achieved when $E = 2.8$, indicating that this value provides the most accurate initialization for modeling the viral dynamics of Patient 4.

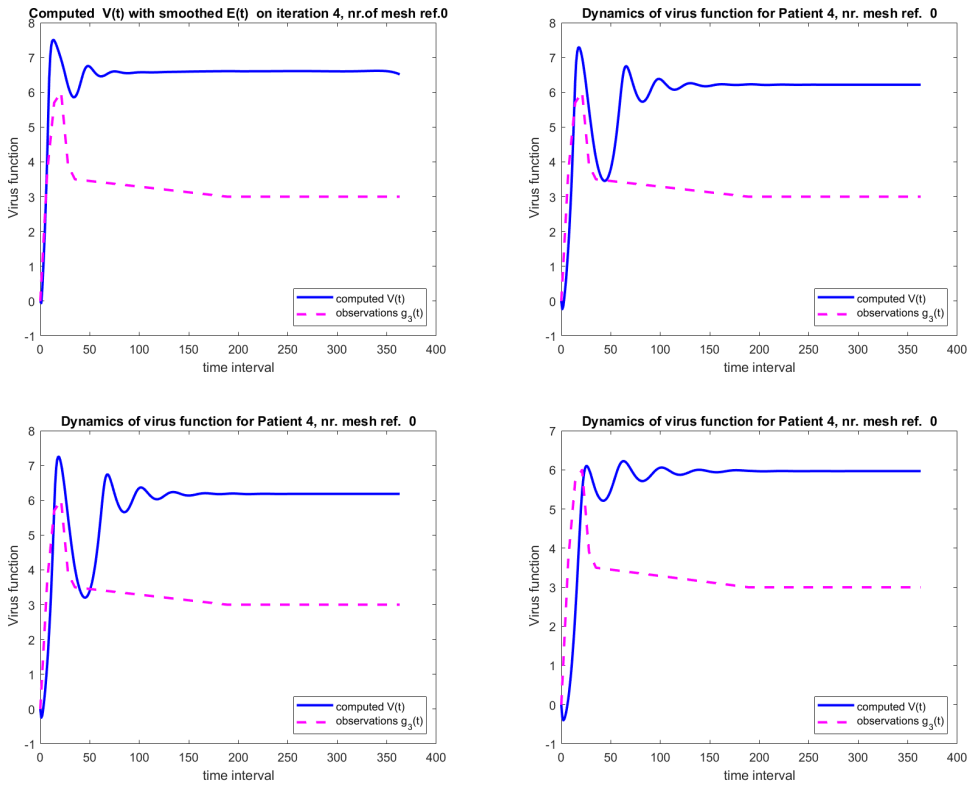


Figure 4.12: Computed viral load $V(t)$ for Patient 4 under different initial values of E. The best agreement with the observed data $g_3(t)$ occurs when $E = 2.8$.

4.1.2 Test 2

In Test 2, we have modeled the functions $d(t)$ and $E(t)$ separately as follows:

$$\begin{aligned} d(t) &= d_0 + d_1(t, V), \\ d_1(t, u_3) &= \begin{cases} 0, & t < t_1, \\ f(t) \log_{10} V, & t \geq t_1, \end{cases} \\ E(t) &= E_0 + d_1(t, V), \\ f(t) &= \frac{\beta_{\text{CTL}}}{1 + \kappa e^{-(t-t_1)/\delta T_1}} - \frac{\beta_{\text{CTL}}}{1 + \kappa e^{-(t-t_2)/\delta T_2}}, \\ \kappa &= 1 + 10^5 \beta_{\text{CTL}}. \end{aligned} \tag{4.1}$$

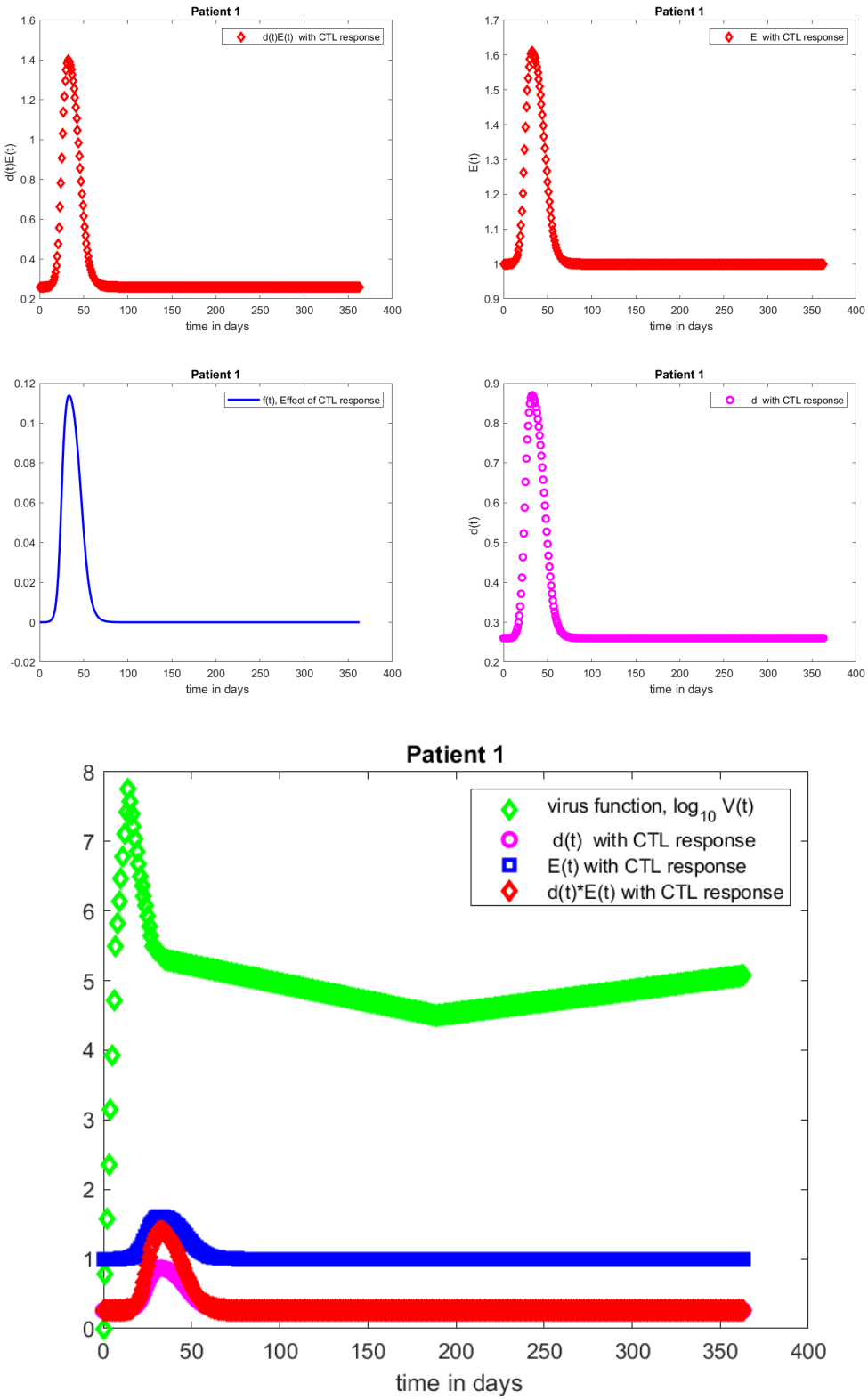


Figure 4.13: Results for Patient 1 using Test 2. The parameters for this test are $t_1 = 1$, $t_2 = 0$, $\Delta T_1 = 2.5$, and $\Delta T_2 = 5$. Each plot demonstrates the CTL response's effect on viral dynamics: (Top left) $d(t)E(t)$, (Top right) $E(t)$, (Middle left) $f(t)$, (Middle right) $d(t)$, and (Bottom) overall viral dynamics $V(t)$, $d(t)$, $E(t)$, and $d(t)E(t)$.

4.1.2.1 Dynamics of Virus Function for Test 2, Patient 1

The figures below illustrate the dynamics of the virus function for Patient 1 in Test 2 under different initial values of E. The parameters for this test are $t_1 = 1$, $t_2 = 0$, $\Delta T_1 = 2.5$, and $\Delta T_2 = 5$. The objective is to determine the E value that best aligns the computed $V(t)$ with the observed $g_3(t)$.

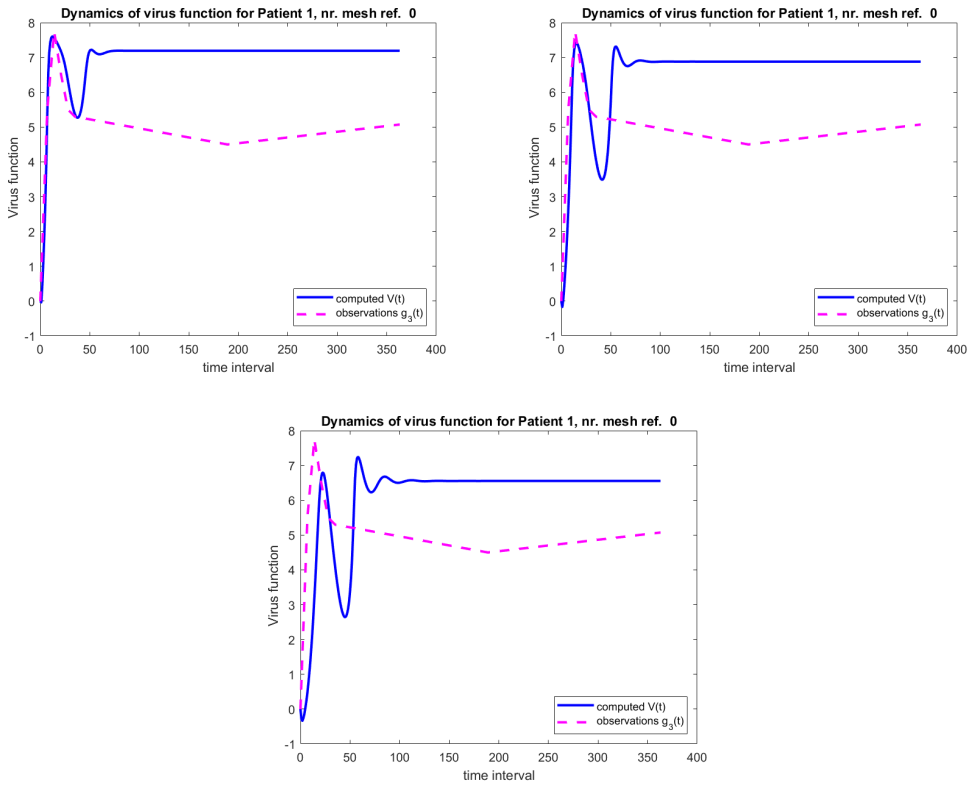


Figure 4.14: Dynamics of the virus function for Patient 1 in Test 2 with varying E. (Top left) $E = 1$, (Top right) $E = 2$, (Bottom) $E = 4$.

From these plots, we observe that the computed $V(t)$ matches the observed g_3 most closely during the first 50 days for $E = 1$, suggesting it provides the best initialization for the dynamics in Test 2 for Patient 1. To further support this, Figure 4.15 shows how the residual, the difference between the model and the data, changes before and after optimization for different values of E.

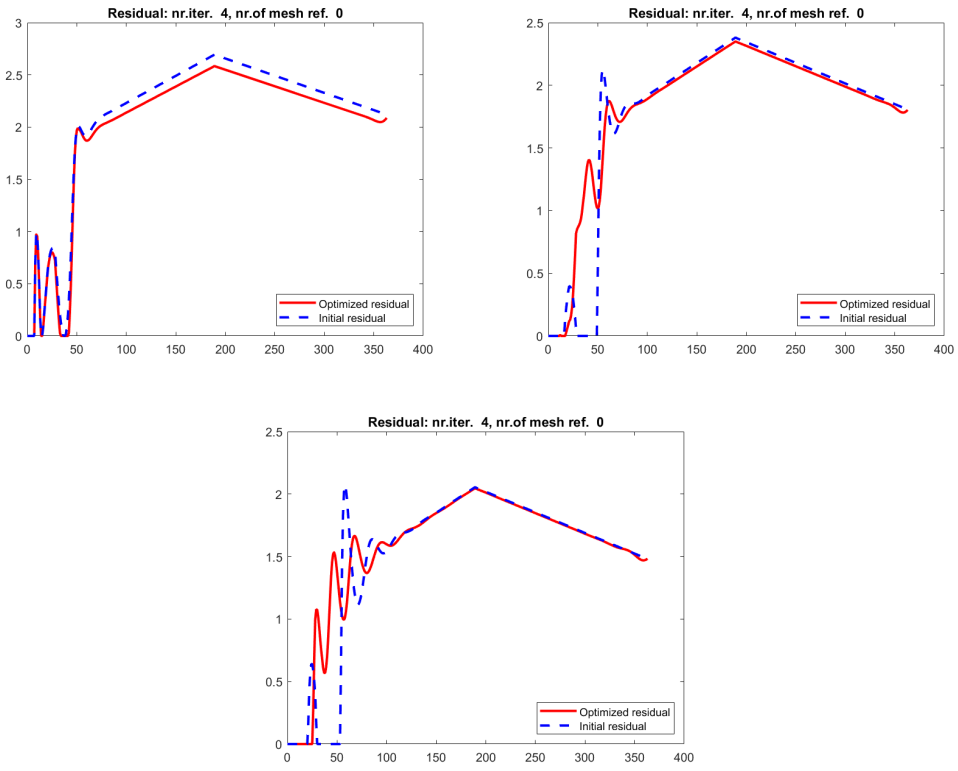


Figure 4.15: Initial and optimized residuals in Test 2 for different values of E. (Top left) $E = 1$, (Top right) $E = 2$, (Bottom) $E = 4$. The blue dashed line shows the residual before optimization, and the red solid line shows the residual after optimization.

For $E = 1$, the red line is clearly lower than the blue one, especially during the first 50 days. This shows that the optimization worked well and improved the model's fit to the data. When $E = 2$, the improvement is smaller, but we can still see that the red line is slightly lower than the blue one after day 50. In the case of $E = 4$, the blue and red lines are already very close to each other, which means that the optimization has little effect.
Overall, $E = 1$ gives the best result, as it leads to the greatest improvement in the model's agreement with the data—particularly during the early stage of the infection, which is a critical period in this analysis.

4.1.2.2 Analysis of Parameter Effects on Virus Dynamics

The effect of changing ΔT_1 and ΔT_2 is shown in the top left, top right, and middle left plots of Figure 4.16. When ΔT_2 is increased to 15 (top right), the computed $V(t)$ deviates more from the observed data $g_3(t)$. When ΔT_2 is decreased to 3.5 (middle left), $V(t)$ shows more variation. The baseline case with $\Delta T_1 = 2.5$ and $\Delta T_2 = 5$ (top left) provides better agreement with the observations.
The influence of t_1 and t_2 is seen in the top left, middle right, and bottom plots. When t_1 is increased to 50 (middle right) or t_2 to 50 (bottom), the computed $V(t)$ diverges from $g_3(t)$, especially at early times. The best match occurs in the baseline

case (top left) with $t_1 = 1$ and $t_2 = 0$.
Overall, the baseline parameters $\Delta T_1 = 2.5$, $\Delta T_2 = 5$, $t_1 = 1$, and $t_2 = 0$ give the best fit between $V(t)$ and $g_3(t)$.

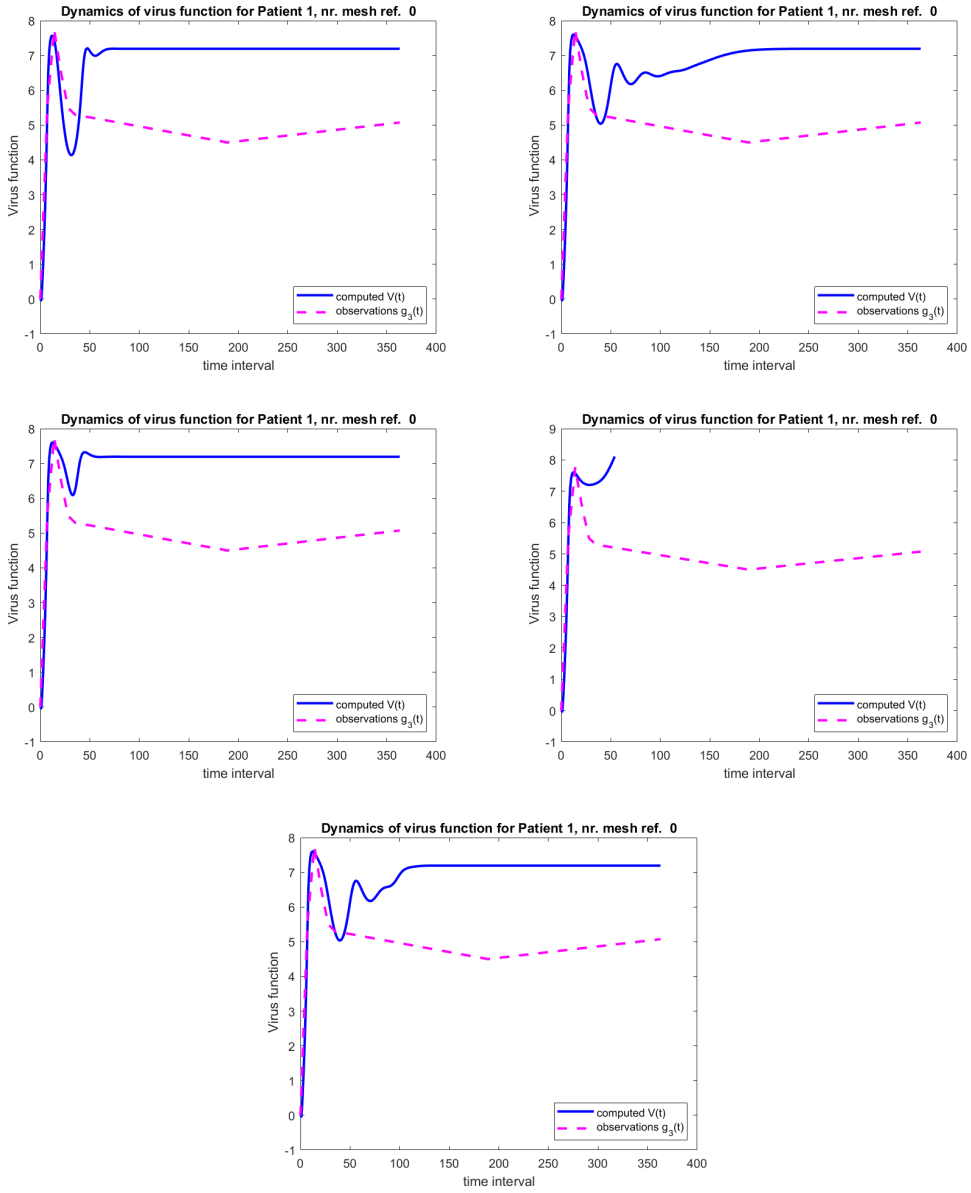


Figure 4.16: Impact of varying ΔT_1, ΔT_2, t_1, and t_2 on the dynamics of the virus function. (Top left) Baseline: $\Delta T_1 = 2.5$, $\Delta T_2 = 5$, $t_1 = 1$, $t_2 = 0$. (Top right) Increasing ΔT_2 to 15. (Middle left) Decreasing ΔT_2 to 3.5. (Middle right) Increasing t_1 to 50. (Bottom) Increasing t_2 to 50.

4.1.2.3 Analysis of Test 2 for Patient 2

The dynamics of the virus function for different values of E are shown in Figure 4.17. For $E = 2$ (top left), the computed virus function $V(t)$ is somewhat close to the observed data $g_3(t)$, but there are visible differences, especially after day 50. When $E = 2.4$ (top right), the match between $V(t)$ and $g_3(t)$ is clearly better, showing

the best agreement among all three cases. For $E = 3$ (bottom), the computed $V(t)$ diverges more from $g_3(t)$, especially early in the simulation.

In conclusion, the best fit is observed for $E = 2.4$. The case $E = 2$ gives a reasonable approximation, while $E = 3$ does not align well with the observed data.

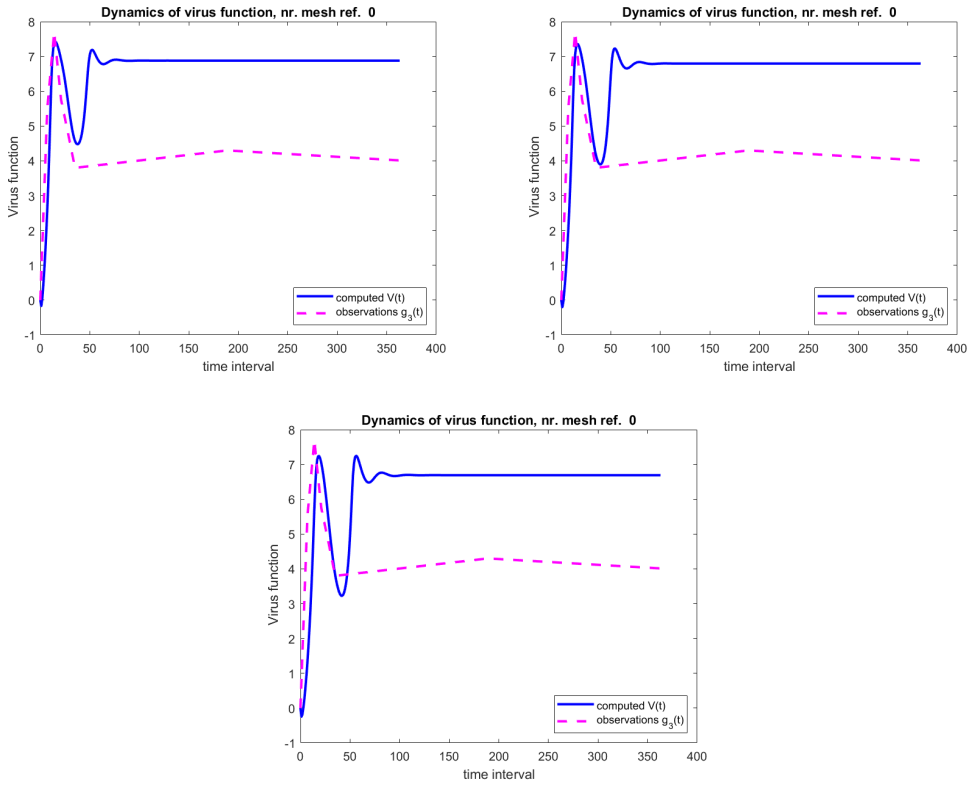


Figure 4.17: Dynamics of the virus function for Patient 2 in Test 2 with varying E. (Top left) $E = 2$, (Top right) $E = 2.4$, (Bottom) $E = 3$.

4.1.2.4 Analysis of Test 2 for Patient 3

The virus dynamics for different values of E are shown in Figure 4.18. For $E = 1.5$ (top left), the computed virus function $V(t)$ is partially aligned with the observed data $g_3(t)$, but differences appear in the later phase. When $E = 2.5$ (top right), the match between $V(t)$ and $g_3(t)$ is much better, with the best overall agreement. For $E = 3.5$ (bottom), the computed $V(t)$ deviates from $g_3(t)$, especially during the early and middle parts.

In conclusion, the best result for Patient 3 is obtained when $E = 2.5$. The case $E = 1.5$ gives a partial fit, while $E = 3.5$ shows poor agreement with the observed data.

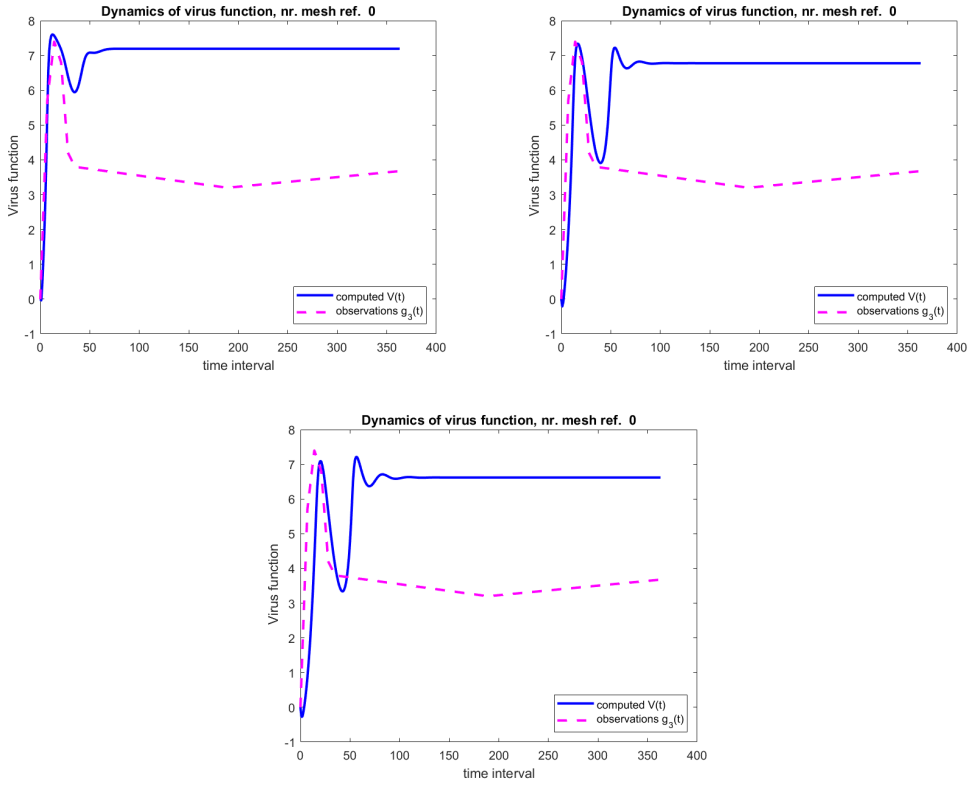


Figure 4.18: Dynamics of the virus function for Patient 3 in Test 2 with varying E. (Top left) $E = 1.5$, (Top right) $E = 2.5$, (Bottom) $E = 3.5$.

4.1.2.5 Analysis of Test 2 for Patient 4

The results for Patient 4 are shown in Figure 4.19. Among the tested values, $E = 3$ with $\Delta T_2 = 5$ (top right) gives the best match between the computed virus function $V(t)$ and the observed data $g_3(t)$. Adjusting ΔT_2 slightly to 5.5 (bottom left) further improves the fit. Other choices, such as $E = 1.5$ (top left) and $E = 4.5$ (bottom right), show less agreement with the observations.

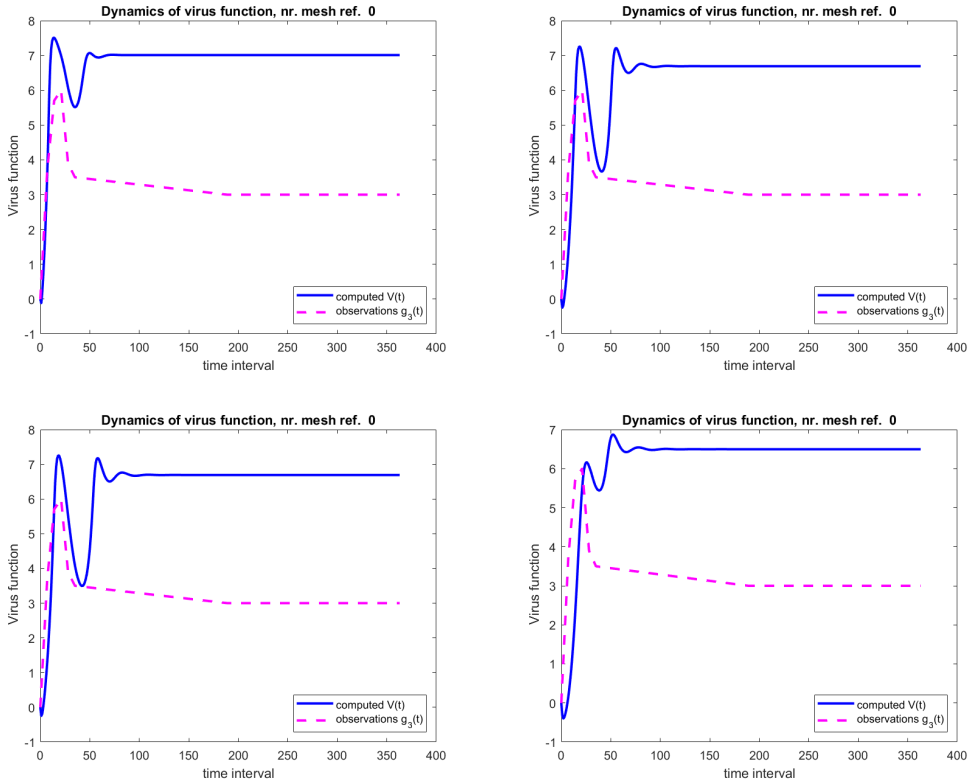


Figure 4.19: Dynamics of the virus function for Patient 4 in Test 2 with varying E and ΔT_2. (Top left) $E = 1.5, \Delta T_2 = 5$, (Top right) $E = 3, \Delta T_2 = 5$, (Bottom left) $E = 3, \Delta T_2 = 5.5$, (Bottom right) $E = 4.5, \Delta T_2 = 5$.

4.2 Forward Problem Solution and Reconstruction of $V(t)$ and $\Sigma(t)$ in the Second Phase

4.2.1 Reconstruction and Optimization Results for Patient 1

4.2.1.1 Results with Initial Guess $E(t) = 1$

At the beginning (iteration 0), the computed virus function $V(t)$ was quite different from the observed data $g_3(t)$, especially during the first 50 days, as shown in the top left plot of Figure 4.20. The computed $V(t)$ follows the general trend of the observations by increasing to a peak, decreasing to an intersection point, and then rising again, but the early dynamics were not well captured.
After 10 iterations of the CGA algorithm, the computed $V(t)$ aligns much better with the observed data, as shown in the top right and bottom plots of Figure 4.20. The sharp rise and early drop are now accurately captured, and the curve remains close to the observations afterward.
For $\Sigma(t)$, the computed values were already close to the observed data before optimization and did not change significantly after it. The main improvement from the optimization is clearly visible in the virus function $V(t)$, especially during the first

50 days.

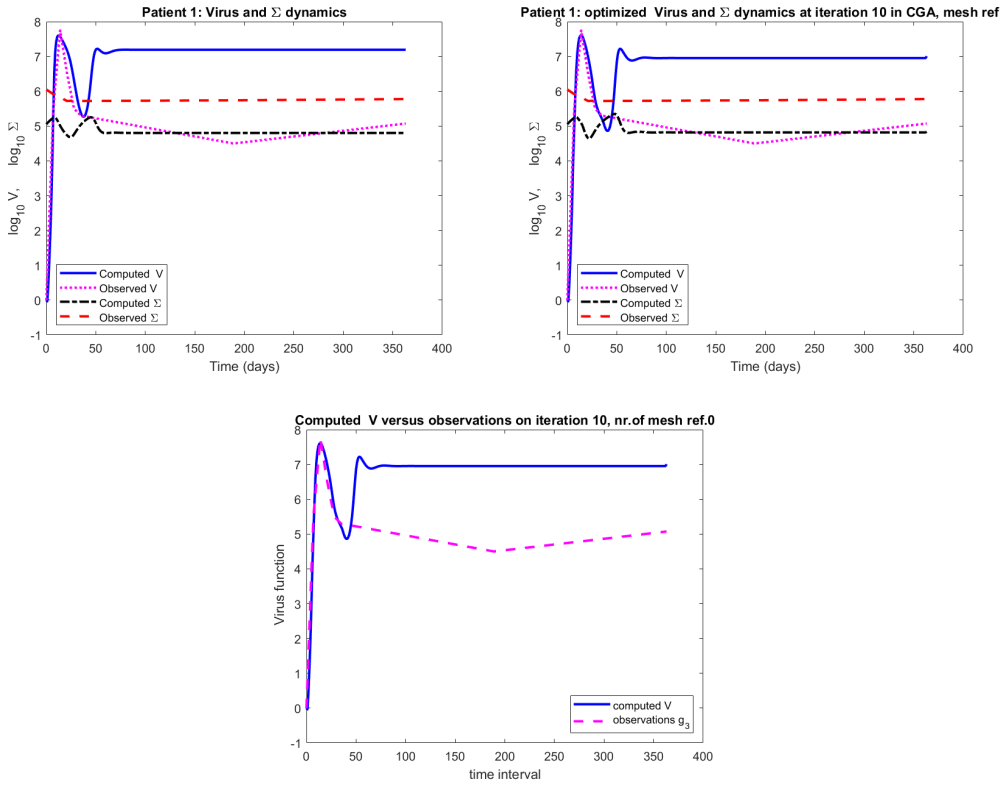


Figure 4.20: Patient 1: Comparison of virus $V(t)$ and immune response $\Sigma(t)$. (Top left) Computed $V(t)$ and $\Sigma(t)$ for the initial guess $E(t) = 1$ (iteration 0). (Top right) Optimized $V(t)$ and $\Sigma(t)$ after 10 CGA iterations. (Bottom) Optimized $V(t)$ compared to observed viral load $g_3(t)$ at iteration 10.

Figure 4.21 shows the behavior of the optimization process for Patient 1 with initial guess $E(t) = 1$.

In the top left plot, we see the convergence of the gradient norm over 10 iterations. The norm decreases steadily, showing that the optimization is progressing and improving the solution step by step.

The top right plot shows the norm of the relative error in the computed $E(t)$. The error drops sharply after the first iteration and continues to decrease slowly, which means the computed $E(t)$ becomes close to the true solution early in the process.

The bottom plot compares the initial and optimized residuals. After optimization, the residual is clearly reduced, meaning that the computed solution better matches the observed data.

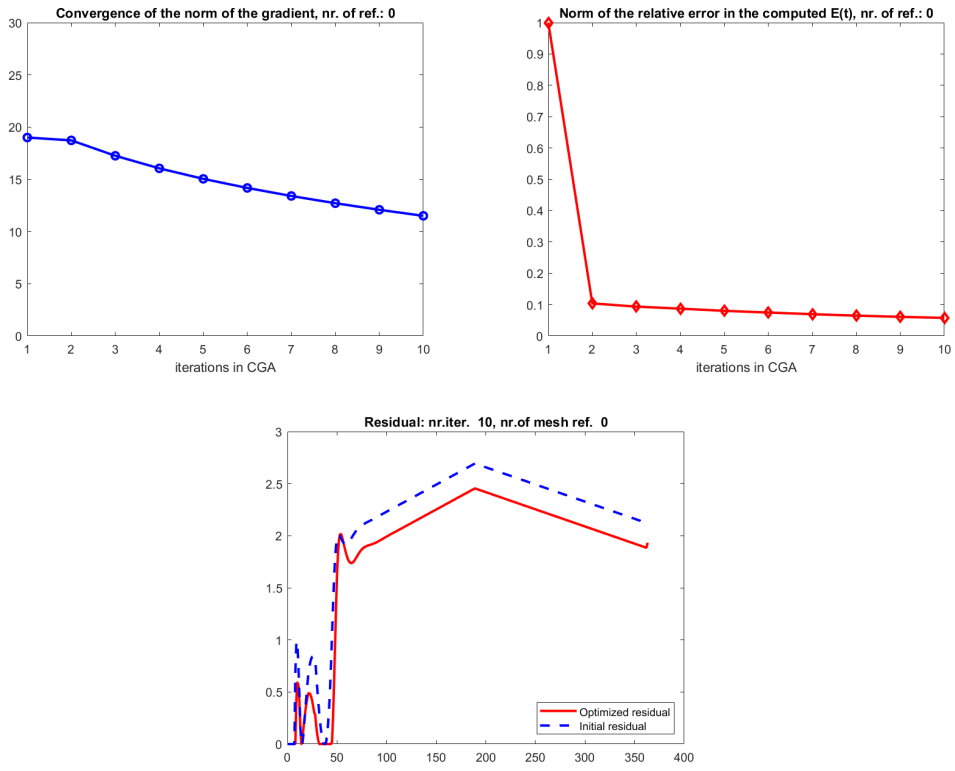


Figure 4.21: Optimization results for Patient 1 with initial guess $E(t) = 1$. (Top left) Convergence of the norm of the gradient. (Top right) Norm of the relative error in computed $E(t)$. (Bottom) Initial and optimized residuals.

In Figure 4.22, we compare three curves for $E(t)$:

- The **blue dashed curve** shows the initial guess.
- The **pink dashed curve** shows the computed $E(t)$ after CGA optimization.
- The **red solid curve** shows the polynomial fitting using least squares (LS) applied to the optimized solution.

At the beginning, we can see that the initial guess is not very accurate. It rises and then stays low without capturing the correct behavior.

After running CGA, the computed solution (pink curve) fits the expected shape much better, showing a rise to a peak around day 40 and then a smoother behavior afterward. However, there are still some small oscillations.

The red curve (LS fitting) gives a smoother version of the CGA result. After day 50, the red and pink curves are very close, showing that the CGA optimization already provided a good reconstruction, and the LS fitting just removed small oscillations.

In summary, CGA improved the guess significantly, and LS fitting smoothed the final $E(t)$ curve.

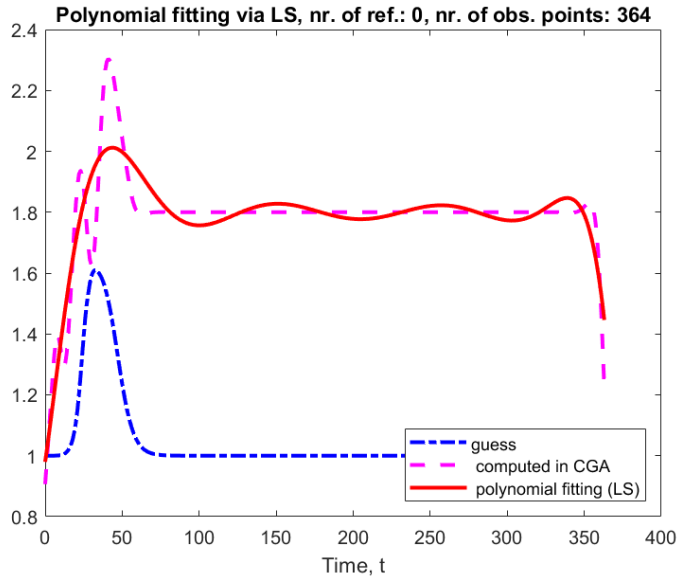


Figure 4.22: Polynomial fitting of the optimized $E(t)$ for Patient 1. Blue: initial guess, pink: CGA result, red: least squares fitting.

4.2.1.2 Results for Patient 1 with Initial Guess $E(t) = 1.6$

Figures 4.23 show the results for $E(t) = 1.6$. At the beginning (Top left in Figure 4.23), the computed virus function $V(t)$ already followed the observations fairly well compared to $E = 1$. However, after intersecting the observed curve, the computed $V(t)$ continues decreasing too much, which is not realistic.

After 10 CGA iterations, the top right and bottom plots in Figure 4.23 show that the computed virus curve improved and better follows the observed data, especially during the first 50 days. The algorithm corrects the extra drop and keeps the computed curve close to the observations. $\Sigma(t)$ was already close to observations and stayed stable.

In summary, even if the initial guess is not perfect, the CGA algorithm improves the accuracy of the result.

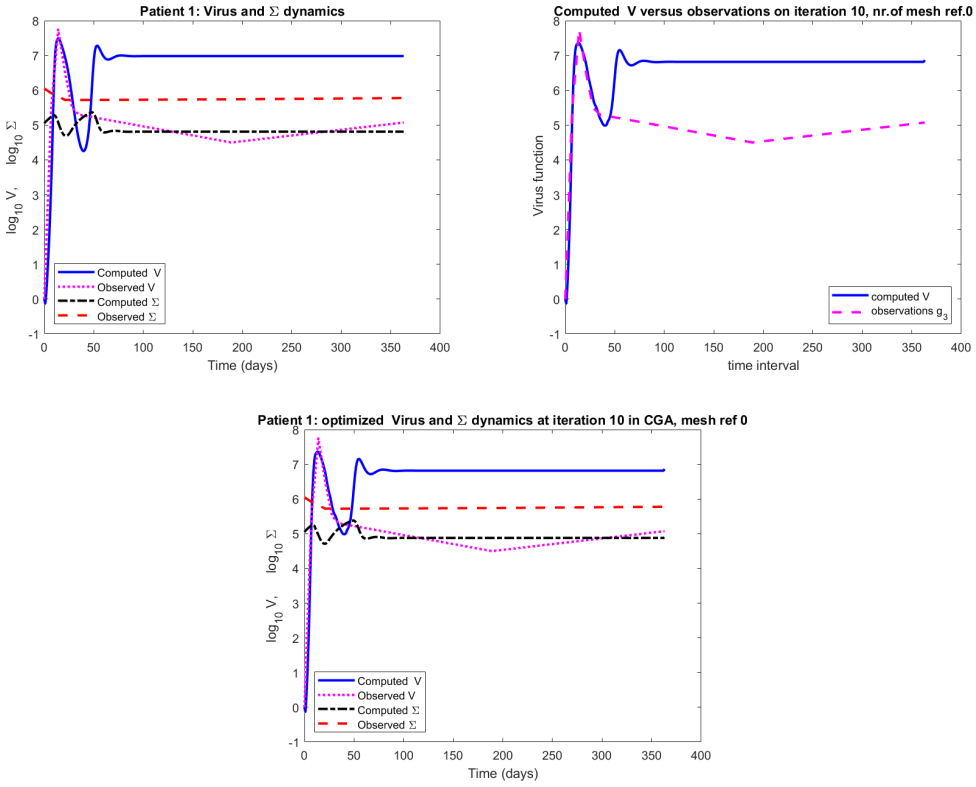


Figure 4.23: Patient 1, initial guess $E(t) = 1.6$. (Top left) Computed $V(t)$ and $\Sigma(t)$ compared to observations. (Top right) Computed $V(t)$ vs observations after optimization. (Bottom) Optimized $V(t)$ and $\Sigma(t)$ after 10 CGA iterations.

Figure 4.24 provides additional evidence that the CGA algorithm works effectively, even when the initial guess $E(t) = 1.6$ is not ideal. The convergence of the gradient norm shows that the optimization progresses steadily, improving the solution step by step. The relative error in the computed $E(t)$ drops sharply after the first iteration and continues to decrease slightly, indicating that the solution becomes more accurate over time. The polynomial fitting plot illustrates that the optimized $E(t)$ closely follows the least squares curve, while the initial guess was significantly different. In the residual plot, we observe that the residual values become smaller after optimization, which means the model fits the observed data better. Overall, these plots confirm that the CGA algorithm improves both the accuracy of $E(t)$ and the overall model performance.

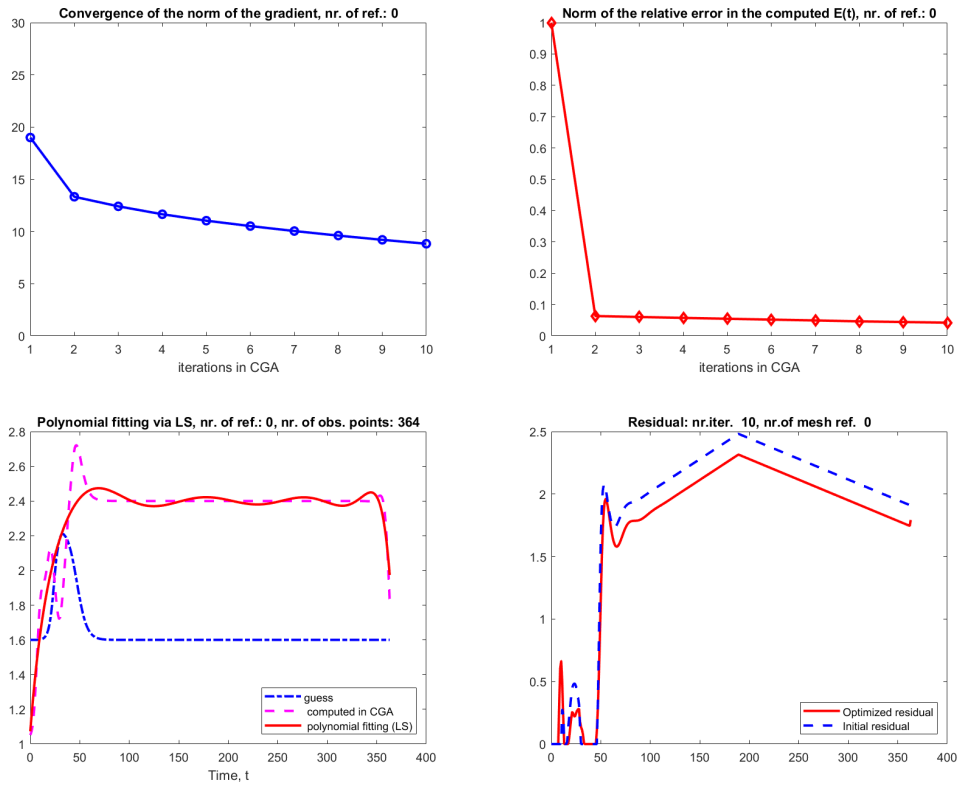


Figure 4.24: Patient 1 with initial guess $E(t) = 1.6$. (Top left) Convergence of the norm of the gradient. (Top right) Norm of the relative error in the computed $E(t)$. (Bottom left) Polynomial fitting via least squares compared to CGA result and initial guess. (Bottom right) Initial and optimized residuals.

4.2.2 Reconstruction and Optimization Results for Patient 2

Figure 4.25 shows the results for Patient 2 with initial guess $E(t) = 1.5$. At the beginning, the computed virus $V(t)$ and $\Sigma(t)$ were far from the observed data, especially during the first 50 days. The computed virus curve dropped too much after the peak and did not follow the observed values well.

After running 10 iterations of the CGA algorithm, the results improved clearly. The computed virus $V(t)$ fits the observed data much better, especially in the early stage. The sharp rise, drop, and later stable values are now captured correctly. Also, the computed $\Sigma(t)$ matches the observed $\Sigma(t)$ more closely after optimization.

These results show that even if the initial guess is not very accurate, the CGA algorithm can correct it and improve the solution. In this case, choosing $E = 1.5$ gave very good results after the optimization process.

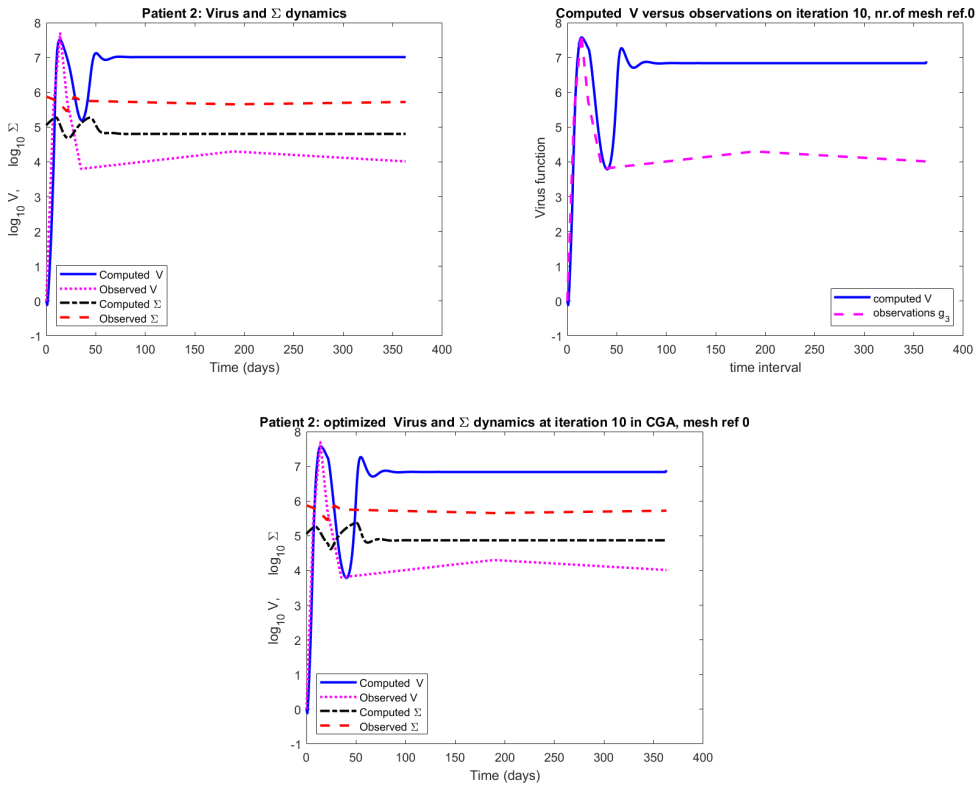


Figure 4.25: Patient 2, initial guess $E(t) = 1.5$. (Top left) Computed $V(t)$ and $\Sigma(t)$ compared to observations. (Top right) Computed $V(t)$ vs observations after optimization. (Bottom) Optimized $V(t)$ and $\Sigma(t)$ after 10 CGA iterations.

Figure 4.26 shows a 3D plot of the gradient values during the 10 iterations of the CGA algorithm for Patient 2. This view helps us better see how the gradient behaves over time.

We can clearly see that the largest gradient values are found at the beginning of the time interval (around the first 50 days). This means that most of the important updates in the optimization happen early, which is expected since the virus dynamics change quickly at the start.

As the algorithm continues, the surface becomes flatter, showing that the gradient values get smaller. This means that the solution is improving and getting closer to the correct answer. The algorithm does not need to make large changes in the later time steps.

This plot confirms that the CGA algorithm focuses more on the early time region where the data is most active, and then smoothly converges as the result becomes more accurate.

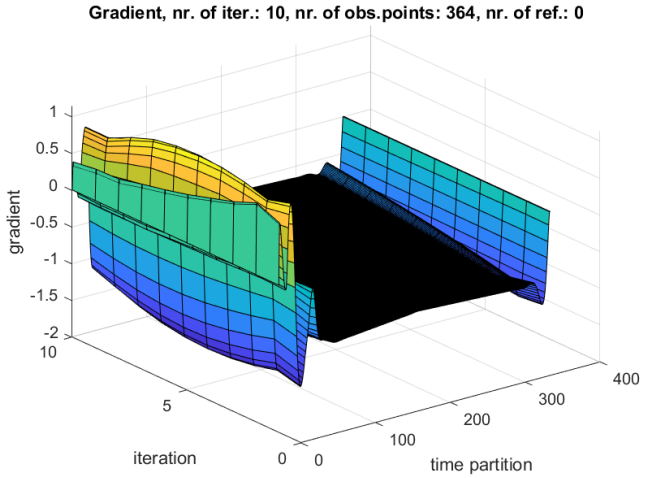


Figure 4.26: 3D plot of the gradient over time and iterations for Patient 2. The gradient is large in the early time steps and becomes smaller as the algorithm converges.

Here also, Figures 4.27 clearly show how the CGA algorithm improves the results for Patient 2 with $E(t) = 1.5$. The norm of the gradient becomes smaller after each iteration, and the relative error drops quickly. The computed control $E(t)$ becomes very close to the least squares fit, and the residual is lower after optimization. These plots confirm that our method works well, even when the initial guess is not perfect.

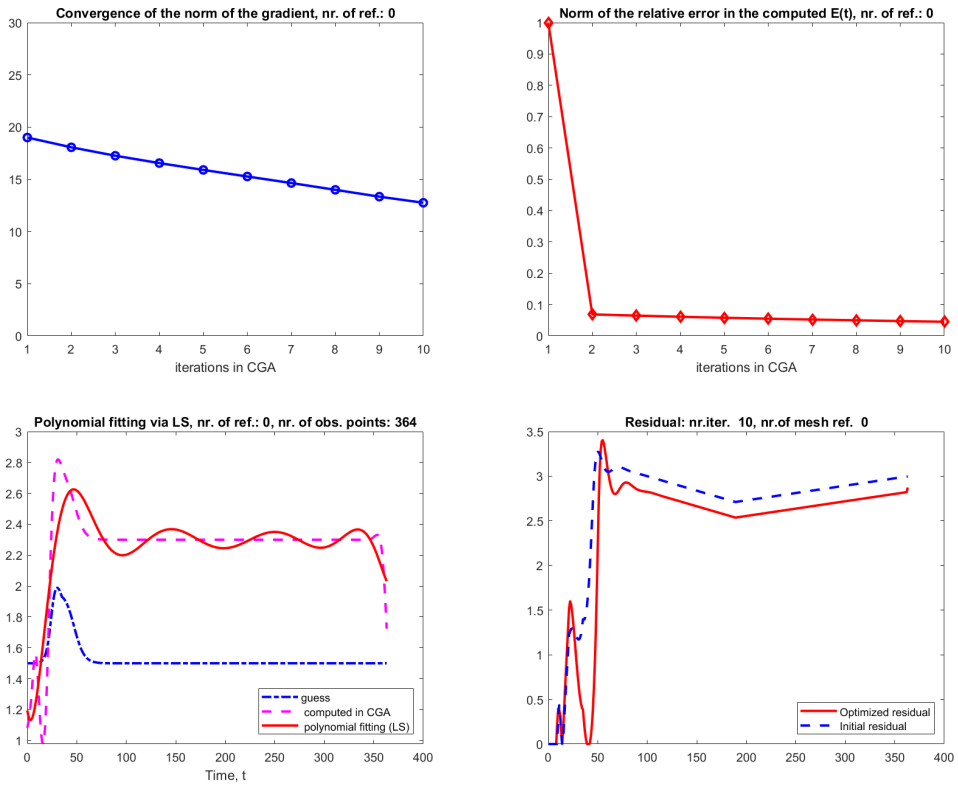


Figure 4.27: Patient 2 with initial guess $E(t) = 1.5$. (Top left) Convergence of the gradient norm. (Top right) Decrease of the relative error. (Bottom left) Polynomial fitting shows improved computed $E(t)$. (Bottom right) Residual reduced after optimization.

4.2.3 Reconstruction and Optimization Results for Patient 3

Figures 4.28 and 4.29 show the results for Patient 3 when we use the initial guess $E(t) = 1.6$. As shown in Figures 4.28, the optimization also worked well for Patient 3. Before optimization, the computed virus $V(t)$ and $\Sigma(t)$ were far from the observed data. After 10 CGA iterations, the computed values closely matched the observations. This confirms that even when starting from a non-optimal guess like $E = 1.6$, the algorithm improves the solution and gives accurate results.

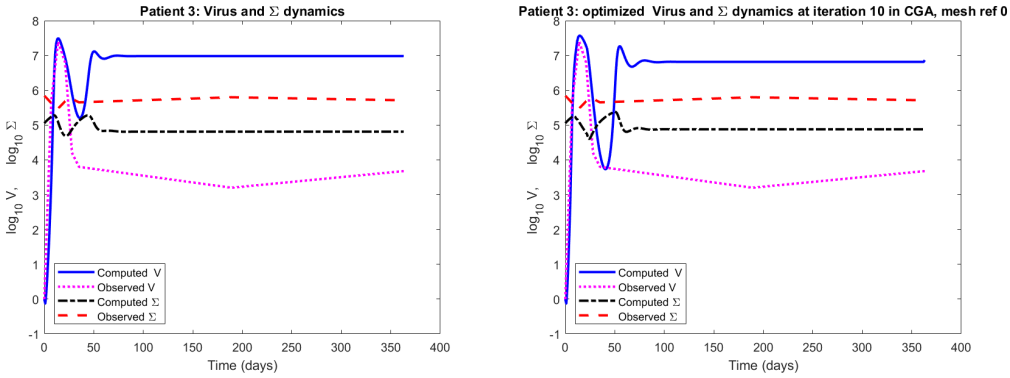


Figure 4.28: Patient 3, reconstruction with initial guess $E(t) = 1.6$. (Left) Initial computed $V(t)$ and $\Sigma(t)$ vs observations. (Right) Optimized solution after 10 CGA iterations.

To further support this, Figure 4.29 shows several useful plots. The gradient norm decreases over the iterations, which means the algorithm is improving the solution. The relative error in $E(t)$ also drops sharply. The new computed $E(t)$ (pink) matches closely with the least squares fit (red), while the initial guess (blue) was far off. Finally, the residual values after optimization are smaller than the initial ones. This confirms that the CGA algorithm worked well for Patient 3.

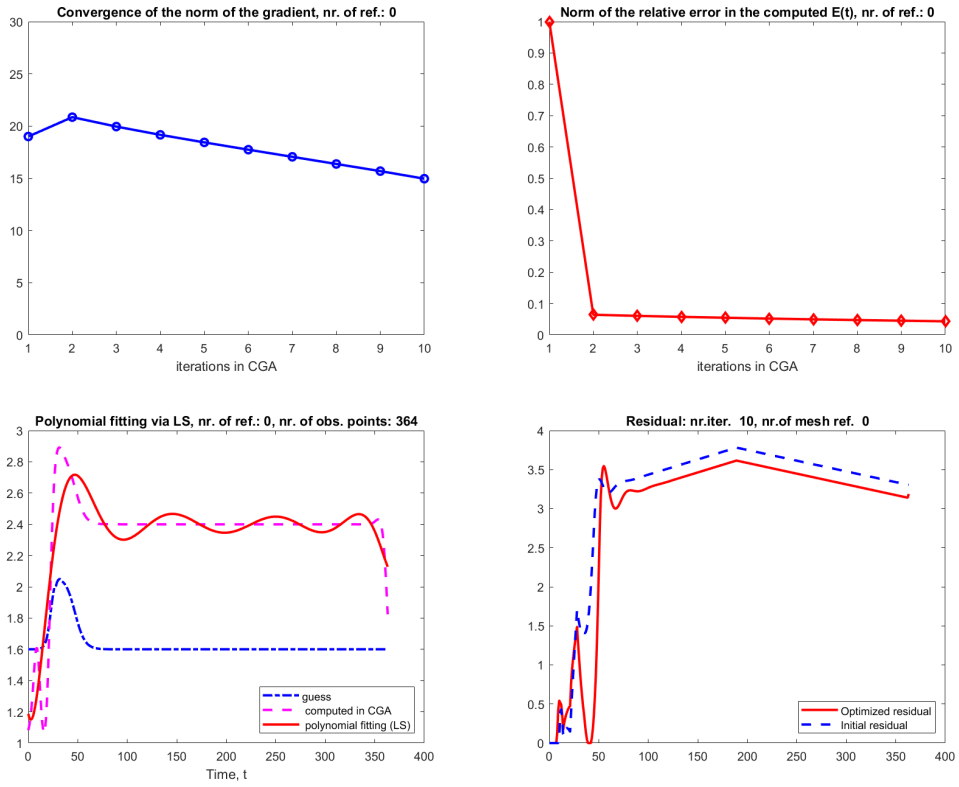


Figure 4.29: Supporting plots for Patient 3, initial guess $E(t) = 1.6$. Gradient norm and relative error decrease during optimization. The computed $E(t)$ becomes close to the LS fit, and residuals improve.

4.2.4 Reconstruction and Optimization Results for Patient 4

Here also, Figures 4.30 show how the CGA algorithm improves the results for Patient 4 using the initial guess $E(t) = 1.4$. The initial computed virus and $\Sigma(t)$ curves are far from the observations. After 15 CGA iterations, the computed curves move much closer to the observed data. This is especially important since this case is more difficult, and still the algorithm finds a good fitting. We can say that this is the best result possible for Patient 4.

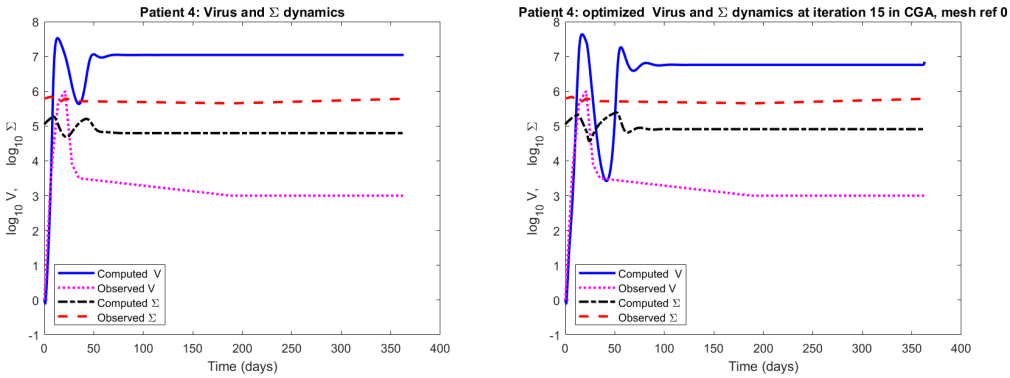


Figure 4.30: Patient 4, initial guess $E(t) = 1.4$. Left: Computed $V(t)$ and $\Sigma(t)$ compared to observations before optimization. Right: Optimized curves after 15 CGA iterations.

Figure 4.30 shows that even for Patient 4, where finding a good fit is more difficult, the CGA algorithm still works well. The initial guess $E(t) = 1.4$ gives a poor fit at the beginning, but after 15 iterations, the computed virus $V(t)$ and $\Sigma(t)$ follow the observed data more closely. The supporting plots in Figure 4.31 confirm the improvement: the gradient norm becomes smaller, the error in $E(t)$ drops fast, and the optimized $E(t)$ matches the polynomial fit well. The residuals are also lower after optimization. This confirms that the CGA algorithm is able to improve the result even in a challenging case like this.

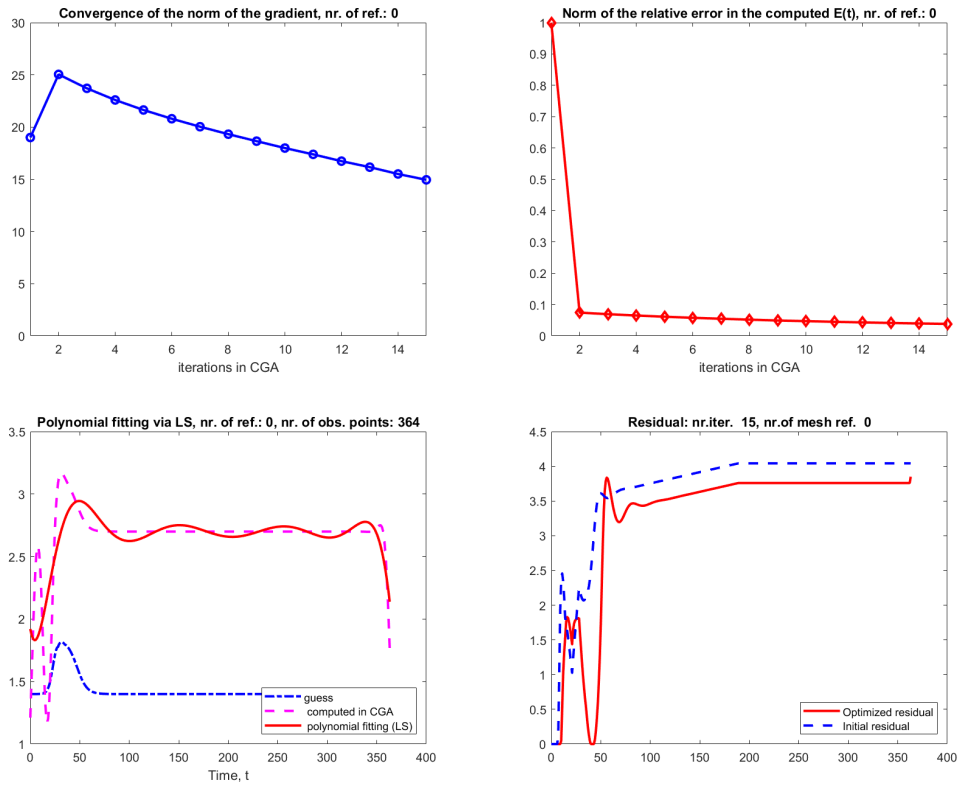


Figure 4.31: Supporting plots for Patient 4 with $E(t) = 1.4$. Top: Gradient norm and relative error show improvement during CGA iterations. Bottom: Optimized $E(t)$ fits the polynomial well, and the residuals are reduced.

5

Conclusion

5.1 Conclusion of the First Phase

In the first phase, we focused on modeling the viral load $V(t)$ using a system of three ordinary differential equations (ODEs). Since the model does not include immune response terms, we mainly focused on getting a good fit during the first 50 days, where the virus dynamics are most active.
In Test 1, the fitting results were mixed: we saw underfitting for Patient 1, okay fitting for Patient 4, and good fitting for Patients 2 and 3. In Test 2, the fitting improved a lot and gave good results for all four patients. This shows that Test 2 works better overall.
The fitting also depended a lot on the choice of parameters. A good initial guess for the immune function $E(t)$ made a big difference in both tests. The time delays ΔT_1 and ΔT_2 also played an important role. If ΔT_2 is too large or too small compared to ΔT_1, it can cause underfitting or even unstable behavior. Choosing them carefully helps balance early fitting and smooth dynamics.
We also found that smaller values of t_1 and t_2 gave better fitting, with t_1 having a stronger effect. Overall, while not all results were perfect, the method was able to capture the main virus behavior and showed stable performance. These results support using this approach in the second phase with more advanced modeling.

Conclusion of Phase 2 – Full Model

In this phase, we extended the model by incorporating the immune response through the term $\Sigma(t) = T(t) + I(t)$, and applied the Conjugate Gradient Algorithm (CGA) for parameter optimization. The resulting model demonstrated a good fit for both viral load $V(t)$ and the immune response $\Sigma(t)$, particularly during the acute phase of infection. The optimization process showed stability, as indicated by the decreasing gradient norm, consistent relative errors, and improved residuals across iterations.
Using only viral load data $V(t)$ in the optimization typically resulted in fast convergence, but often failed to achieve an accurate fit. In contrast, including both $V(t)$ and $\Sigma(t)$ led to significantly better fitting results. The CGA method proved robust, yielding improvements even when starting from poor initial guesses. Adding immune response data not only stabilized the optimization but also improved the accuracy of fitting during the early phase of infection.
Patient 4 remained the most difficult to fit, yet the inclusion of $\Sigma(t)$ still led to improvements without overfitting. This suggests that further enhancement may be

possible by adopting an adaptive version of the CGA method, such as the Adaptive CGA (ACGA).

Limitation: Despite the observed improvements, accurately modeling the latent phase after acute infection remains challenging. This limitation might be overcome by extending the current model to a full system of four ODEs, including an equation for latent T cells.

Bibliography

[1] @miscWHO2025, author = World Health Organization, title = Data on the Size of the HIV Epidemic, year = 2025, url = https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/data-on-the-size-of-the-hiv-aids-epidemic, note = Accessed: March 14, 2025

[2] @miscHIVgov2025, author = HIV.gov, title = Global HIV Statistics, year = 2025, url = https://www.hiv.gov/hiv-basics/overview/data-and-trends/global-statistics, note = Updated: February 7, 2025. Accessed: March 14, 2025.

[3] L. Beilina, M. Eriksson, I. Gainova, Time-adaptive determination of drug efficacy in mathematical model of HIV infection, *Differential Equations and Dynamical Systems*, 32(1), 313–347, 2024. `https://doi.org/10.1007/s12591-021-00572-w`

[4] P. K. Srivastava, M. Banerjee, and P. Chandra, Modeling the drug therapy for HIV infection, *Journal of Biological Systems*, 17(2), 213–223, 2009.

[5] R. J. De Boer, R. M. Ribeiro, A. S. Perelson, Current Estimates for HIV-1 Production Imply Rapid Viral Clearance in Lymphoid Tissues, *PLoS Computational Biology*, 6(9), 2010.

[6] S. W. Kazer, T. P. Aicher, D. M. Muema, et al., Integrated single-cell analysis of multicellular immune dynamics during hyperacute HIV-1 infection, *Nature Medicine*, 26:511-518, 2020.

[7] A. B. Bakushinskii and M. Yu. Kokurin, *Iterative Methods for Approximate Solution of Inverse Problems*, Springer, New York, 2004.

[8] J. Albersmeyer, Adjoint Based Algorithms and Numerical Methods for Sensitivity Generation and Optimization of Large Scale Dynamic Systems, Ph.D. thesis, Heidelberg University, 2010.

[9] H. T. Banks and K. Kunisch, *Estimation techniques for distributed parameter systems*, Birkhäuser 1989. ISBN 978-1-4612-3700-6.

[10] L. Beilina and I. Gainova, Time-adaptive FEM for distributed parameter identification in biological models, *Applied Inverse Problems*, Springer Proceedings in Mathematics & Statistics, 48, pp.37–50, 2013.

[11] L. Beilina and I. Gainova, Time-adaptive FEM for distributed parameter identification in mathematical model of HIV infection with drug therapy, *Inverse Problems and Applications*, Springer Proceedings in Mathematics & Statistics, 120, pp.111–124, 2015.
[8], [12]

[12] E. Polak, *Optimisation. Algorithms and consistent approximations*, AMS, V.124, Springer.

[13] A. N. Tikhonov, On the stability of inverse problems (in Russian), Doklady of the USSR Academy of Science 39 (1943), 195–198.

[14] A. N. Tikhonov and V. Ya. Arsenin, *Solutions of Ill-Posed Problems*, Winston and Sons, Washington, DC, 1977.

[15] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov and A. G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*, London: Kluwer, London, 1995.

A

Appendix 1

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY