Mathematical modelling of baculovirus infection process: Kinetic parameter estimation

Suraj Kumar Singh

(CH16MTECH11014)

A Dissertation Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद

Indian Institute of Technology Hyderabad

Department of Chemical Engineering

June, 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Suraj Kumar Singh

Suraj Kumar Singh
(CH16MTECH11014)

Approval Sheet

This thesis entitled **Mathematical Modelling of baculovirus infection process: kinetic parameter Estimation** by Suraj Kumar Singh is approved for the degree of Master of Technology from IIT Hyderabad.

Kishalay Mitra

(Dr. Kishlay Mitra) Examiner
Dept. of Chemical Engineering
IIT-Hyderabad

(Dr. Aravind Kumar Rengan) Examiner
Dept. of Biomedical Engineering
IIT-Hyderabad

Balaji

(Dr. Balaji Iyer) Examiner
Dept. of Chemical Engineering
IIT-Hyderabad

Lopamudra Giri

(Dr. Lopamudra Giri) Adviser
Dept. of Chemical Engineering
IIT-Hyderabad

Acknowledgements

I thank almighty whose blessing has enabled me to accomplish my project work successfully.

It is my pride privilege to express my sincere thanks and deep sense of gratitude towards **Dr. Lopamudra Giri**, for her advice, supervision and patience through which this work was able to take a shape in which it has been presented. It was her valuable discussions and endless endeavours through which I gained a lot. Her constant encouragement has always been a moral support for me. Again I am very thankful to her for her time and engagement, and for sharing her knowledge with me.

A special word of thanks to my thesis evaluation committee Dr Kishlay Mitra, Dr. Aravind Kumar Rengan and Dr. Balaji Iyer for their valuable suggestions.

It is hard for me to express enough gratitude to my seniors **Mr. Soumitri Srinivas** (Ph.D. Scholar) and **Mrs. Abha Saxena** (Ph.D. Scholar) who have always been a source of inspiration for me and stood by my side at the toughest times.

On a more personal note, the whole credit of my achievements goes to my parents, who were always there for me in my difficulties. It was their unshakable faith in me that has always helped me to proceed further.

Suraj Kumar Singh

Dedicated to

My parents

Abstract

Although there are several mathematical models present for baculovirus infection, the specific functions for insect cell growth and cell death during infection processes remain unknown. Specifically, it is challenging to identify the most suitable model from a large set of plausible models and estimate the kinetic parameters to account for the day to day variability present in the infection experiments. In this context, identification of an unstructured model that can predict the day to day variability in cell growth and cell viability can be useful in determining the optimal operating conditions in fermenters at industrial scale. The major objectives of the present work were to develop a model screening framework that can be used to select the best model and identify the growth and death mechanisms during viral infection through non-linear programming. We then constructed a series of plausible models based on system of ordinary differential equations and performed the model selection using experimental data obtained from shaker flasks. The proposed scheme was tested for selecting the model for uninfected cell growth profiles. The objective function used was the root mean square error between the predicted values and experimental data points obtained from triplicate dataset. The computational scheme was validated using two types of virus, the WT AcMNPV and stabilized AcMNPV. Additionally, we propose a numerical scheme to simulate the cell growth and cell viability during viral passaging. The kinetic parameters were estimated in case of growth of uninfected cells, cells infected with WT virus as well as stabilized AcMNPV. The result shows that Monods equation fits the best for insect cell growth without infection and infection with WT AcMNPV. Whereas, the Contois model fits the best for the stabilized virus. The simulated results also indicate that the day to day variability in cell growth and cell viability profile can be explained through the variation in the specific growth rate and the death rate. The estimated kinetic parameters indicate that the growth and death parameters undergo specific modifications during the passaging of viruses associated to infection process. Additionally, we propose an integrated model for the infection process that simulates the DNA replication, mRNA and protein expression as well as polyhedra production. Specifically, we present the comparison between the unstructured model and the structured integrated model with respect to accuracy and computation time. Current study provides a predictive framework that has a potential application for large scale production of baculovirus.

Contents

Declaration 2

Approval Sheet 3

Acknowledgements 4

Abstract 6

1 Introduction 8

1.1 Baculovirus Production Experiment 8

2 Literature Review 11

2.1 Mathematical modelling for baculovirus infection process 12

2.2 Optimization Methods 18

3 Modelling of baculovirus infection process 22

3.1 Introduction 22

3.2 Unstructure model 23

3.3 Parameter Estimation 26

3.4 Multiple model formulation 27

3.5 Sensitivity analysis 27

3.6 Analysis of the proposed models and comparison with other possible mechanisms 28

3.7 Results 28

4 Integrated model for baculovirus infection process 43

4.1 Detailed modelling of baculovirus infection process 43

4.2 Basic assumptions of infection process 43

4.3 Model formulation 43

4.4 parameter estimation 44

4.5 Sensitivity analysis 51

4.6 Results 52

7 Conclusions & Discussions 60

Nomenclature 61

References 62

Appendix 63

Chapter 1

1.1 Introduction

Baculoviruses are known to infect the invertebrates and studied widely as biopesticides in crop fields. Baculoviruses contain circular double-stranded genome ranging from 80 to 180 kbp [1]. Economical large-scale production of baculovirus has the significant impact in its industrial applications such as the production of vaccines biopesticides and recombinant proteins1 [2]. Mathematical models are useful in quantification of growth and product formation kinetics. Such robust models also find applications in design, optimization, and scale-up of bioreactors. Development of such predictive model has potential applications in industries for the optimal production of baculovirus in fermenters. Specifically, the requirement of large experimental time and expensive resources such as media, sterilization and cell maintenance schedules further motivates the construction of unstructured models which can be simulated in no time [3].

The existing models available for insect cell growth and baculovirus infection shows the prediction profiles for cell growth but simulation results were not validated with adequate experimental data. The absence of detailed experimental data and lack of mechanistic information on baculovirus infection process makes it challenging to identify simple unstructured model capable of simultaneous emulation of insect cell growth, cell death, budded virus and polyhedral formation. In this study, we investigated the cell growth pattern, and cell death pattern using light microscopy. Moreover, we also investigate the formation of polyhedra with passaging of viruses in insect cells through the polyhedra counting using light microscopy. Further, we constructed a model containing substrate consumption, oxygen consumption and carbon-di-oxide formation and percentage of viable cells with different carbon sources for uninfected cell growth as well as infected cell growth. We then focused on unstructured modeling and parameter estimation for cell growth without viral infection and with the viral infection.

We developed a novel computational strategy to select the optimal model from a series of mathematical constructs. Parameter estimation was performed using fmincon which is a function included in MATLAB's optimization toolbox that seeks the minimizer of a scalar function having multiple variables, within a region specified by linear constraints and bounds. The final model was validated with a triplicate experimental dataset and was further used to investigate the specific differences between the growths of infected cells as passaging of viruses was performed. A comprehensive comparison between several existing growth models was performed to choose the best growth model for cell growth.

The details of failure to match the experimental and simulated data and the corresponding physical reasons are also included. The novelty of the work lies in the combination of the experimental and computational study of baculovirus production in insect cells to identify the best-unstructured model for insect cell growth [3]. Specifically, this work is the first attempt to construct a model for simulation of virus passaging based on experimental data obtained from infection in suspension cell culture in shaker flasks.

1.2 Existing Challenges

Identification of a general mathematical structure and estimation of corresponding kinetic parameters can be used to predict the growth and death profile as infection experiments are costly and time-consuming. Hence the specific challenges in analysis of baculovirus infection processes can be described as follows: (1) identification of general mathematical model based on system of nonlinear ODEs that will match with experiments performed in different days (complex growth and death pattern were observed for some of the experiments) (2) identification of the difference between wild-type virus and genetically modified virus (3)estimation of kinetic parameters that can be used for experiments performed in different days.

For biological models, the estimation of kinetic parameters can often be formulated as an optimization problem. Generally, the objective function is the difference between simulated models using kinetic parameters and the respective experimental measurements. The optimization methods will then try to find the output of the mathematical model formed using kinetic parameters which closely fits the experimental measurements.

The objective of the current thesis is to construct a mathematical model to simulate the cell growth pattern, substrate consumption, oxygen consumption, carbon dioxide production and percentage of viable cells for growth of insect cells in shaker flasks. Similarly, we provide a model consisting of a system of ordinary differential equations for the growth of insect cells after baculovirus infection. Additionally, we have constructed a numerical scheme to simulate the cell density and viable cells after passaging of viruses in shaker flasks. Experimental data obtained from the shaker flask experiments were used to estimate the kinetic parameter for the case of infected and uninfected cell growth. The proposed strategy can be used to construct model, select model and estimate kinetic parameter for various viral infections.

Chapter 2

Literature Review

Mathematical modelling can be used to understand a large numbers of interacting components involved in complex processes associated with biological systems. In this study, experimental data was used to investigate the cell growth pattern, substrate consumption, oxygen consumption and percentage viability for uninfected and infected insect cell growth. The main objective was to identify the unstructured model and corresponding kinetic parameters for cell growth with and without viral infection. Here we developed a novel computational strategy to select the optimal model from a series of several combinatorial mathematical constructs. Parameter estimation was performed using fmincon function in MATLAB optimization toolbox which seeks the minimizer of a scalar function of multiple variables, within a region specified by linear constraints and bounds. The final model was validated with a triplicate experimental dataset and was further used to investigate the specific differences between the growth and death of infected cells with passaging of viruses in shaker flasks. A comprehensive comparison between several existing growth models were performed to choose the best model for insect cell growth. The details of failure to match certain data points and corresponding physical reasons are also included. The novelty of the work lies in combination of experimental and computational study of baculovirus production and identification of best unstructured model from various growth models**.**

2.1 Mathematical models for baculovirus infection process

2.1.1 Modeling rotavirus-like particles production in a baculovirus expression vector system (BEVs): Infection kinetics, baculovirus DNA replication, mRNA synthesis and protein production

Rotavirus-like particles (Rota VLPs) are excellent vaccine candidates against rotavirus infection, since they are non-infectious, highly immunogenic, amenable to large-scale production and safer to produce than those based on attenuated viruses. One of the work on mathematical modeling for the production of rotavirus like particles in BEVs focuses on analysis of the major events taking place inside *Sf-9* cells infected by recombinant baculovirus. The timeframe for vDNA, mRNA and VP synthesis was found to be reduced through increasing multiplicity of infection (MOI) due to the metabolic burden effect. The model exhibits acceptable prediction power of the dynamics of intracellular vDNA replication, mRNA synthesis and viral proteins production for the three proteins involved [4].

The depletion of extracellular virus due to binding to cell surface is

$$\frac{dV_j}{dt} = -k_a(N_i + N_u) * V_j \qquad [1]$$

The attachment rate is defined as

$$k_a = k_f(\alpha R) \qquad [2]$$

Where , V_j = concentration of extracellular Virus

k_a = is attachment rate

$N_i \& N_u$ = are concentration of infected and uninfected cell respectively

α = no. of attachment protein per virus

R = no. of surface receptors

The rate of change of infected cell

$$\frac{dN_i}{dt} = k_a N_u V_t \left(\frac{1}{MOI}\right) - k_d N_i \qquad [3]$$

The cell death rate k_d has two terms

$if\ t < \delta_D \quad k_d = k_{d_1} \qquad\qquad if(DNA_t) \leq 10 \quad k_{d_2} = k^*$

$if\ t \geq \delta_D \quad k_d = k_{d_1} + k_{d_2} \qquad if(DNA_t) > 10 \quad k_{d_2} = k^{*}\log(DNA_t)$

Where, k_{d_1} = intrinsic cell death rate = 0.008 hr-1

k_{d_2} = cell death rate due to infection

δ_D = time instant when the cell death increases

V_t= total extracelluler virus concentration

k* = increase in cell death rate

Also, the healthy cell population is given by,

$$\frac{dN_u}{dt} = -k_a N_u V_t \left(\frac{1}{MOI}\right) - k_{d_1} N_u \qquad [4]$$

And the vDNA replication was given by the following equation,

$$\frac{dDNA_j^{nuc}}{dt} = \eta_{traf} k_a V_j(t - \tau_{traf}) \left(1 + \frac{N_u}{N_i}\right) + k_{RDNA} DNA_j^{nuc} f_{DNA,rep}(t, \delta_{DNA,low}, \delta_{DNA,high}) \qquad [5]$$

$f_{DNA,rep}(t, \delta_{DNA,low}, \delta_{DNA,high}) =$

$if(t < \delta_{DNA_{min}}) \qquad f_{DNA} = 0$

$if(\delta_{DNA_{min}} < t < \delta_{DNA_{max}}) \qquad f_{DNA} = 1 - \frac{t - \delta_{DNA_{min}}}{\delta_{DNA_{max}} - \delta_{DNA_{min}}}$

$if(\delta_{DNA_{max}} < t) \qquad f_{DNA} = 0$

Total number of vDNA & *m*RNA copies inside the cell was given by the following equation,

$$\frac{dDNA_j^T}{dt} = \eta_{traf} k_a V_j \left(1 + \frac{N_u}{N_i}\right) + k_{RDNA} DNA_j^{nuc} f_{DNA,rep}(t, \delta_{DNA,low}, \delta_{DNA,high}) \qquad [6]$$

$$DNA_{total} = \sum DNA_j^T$$

$$\frac{dRNA_j}{dt} = k_{SRNA} DNA_j^{nuc} f_{VP}(t, \delta_{VP,low}, \delta_{VP,high}) - k_{DRNA,j} RNA_j \qquad [7]$$

The rate of change of VP was as follows

$$\frac{dVP_j}{dt} = k_{VP,j}\left(\frac{RNA_j}{RNA_j + K_{RNA}}\right) f_{VP}\left(t, \delta_{VP,low}, \delta_{VP,high}\right) N_i \qquad [8]$$

The intracellular protein content was given by,

$$\frac{dVP_j^{int}}{dt} = k_{VP,j}\left(\frac{RNA_j}{RNA_j + K_{RNA}}\right) f_{VP}\left(t, \delta_{VP,low}, \delta_{VP,high}\right) \qquad [9]$$

$if\left(t < \delta_{Vp,low}\right)$ $\qquad f_{VP} = 0$

$if\left(\delta_{VP,low} < t < \delta_{VP,high}\right)$ $\qquad f_{VP} = 1 - \frac{t - \delta_{Vp,low}}{\delta_{VP,high} - \delta_{Vp,low}}$

$if\left(\delta_{VP,high} < t\right)$ $\qquad f_{VP} = 0$

Where, $k_{VP,j}$ = maximum VPj synthesis rate

K_{RNA} = half-saturation constant for intracellular mRNA.

$k_{VP} = k_{VP,j}^{*}$ $\qquad DNA_{total} \leq 10$

$k_{VP} = k_{VP,j}^{*} \log(DNA_{total})$ $\qquad DNA_{total} > 10$

Time instant relation with intracellular copies of DNA

$\delta_{VP,high} = \delta_{DNA,high} = \delta_D = \delta^* \quad if\ DNA_{total} \leq 10$

$\delta_{VP,high} = \delta_{DNA,high} = \delta_D = \delta^* \quad if\ DNA_{total} > 10$

2.1.2 A Mathematical Model of the Trafficking of Acid-Dependent Enveloped Viruses: Application to the Binding, Uptake, and Nuclear Accumulation of Baculovirus

Since a quantitative understanding of virus trafficking would be useful in treating viral-mediated diseases, developing protocols for viral gene therapy, designing infection regimens for viral expression systems, and optimizing vaccine and recombinant protein production, there has been attempts to construct models for virus internalization. Here we discuss a classical model for the attachment, internalization, endosomal fusion, lysosomal routing, and nuclear accumulation of baculovirus in SF21 insect cells. The model accounts for multivalent bond formation of the virus with cell surface receptors. The model mimics accurately the experimental trafficking dynamics of the virus at both low and high virion to cell ratios, and estimates a receptor number of 11,000 per cell [5]. The equations and the variables can be defined as follows:

$$\frac{dV_{ex}}{dt} = -\left(\alpha k_f C\right) V_{ex} R_{sf} + k_r V_1 \qquad [1]$$

$$\frac{dV_{int}}{dt} = k_{ev} V_s \qquad [2]$$

$$\frac{dV_s}{dt} = \left(\alpha k_f C\right) V_{ex} R_{sf} - k_r V_1 - k_{ev} * V_s \qquad [3]$$

Where, V_{ex} = No. of Extracellular Virus per cell, a = No. of attachment possible

k_f = 3D forward rate constant, C = Celluler Constant, R_{sf}= No. of free Receptors

k_r = 3D dissociation rate constant, V_1 = No. of Viruses per cell, V_s= Total cell surface virus

V_{int} = No. of internalized Virus, k_{ev} = Endocytosis rate constant.

Multivalent bond formation was modeled as follows,

$$\frac{dV_1}{dt} = \left(\alpha k_f C\right) V_{ex} R_{sf} - k_r V_1 - (j-1) k_x V_1 R_{sf} + 2k_{-x} V_2 - k_{ev} * V_s \qquad [4]$$

$$\frac{dV_i}{dt} = (j-i+1) k_x V_{-1} R_{sf} - (j-1) k_x V_i R_{sf} - i k_{-x} V_i + (i+1) k_{-x} V_{i+1} - k_{ev} * V \qquad [5]$$

$$\frac{dV_j}{dt} = k_x V_{j-1} R_{sf} - j k_{-x} V_j - k_{ev} * V_j \qquad [6]$$

Where, V_i= Number of Viruses with i bound receptors per virus

j = the maximum no of receptors found

k_x = 2D forward rate constant

k_{-x} = 2D reverse rate constant

Endosomal diffusion and transport to nucleus was described as follows,

$$\frac{dV_{endosome}}{dt} = k_{ev} V_s - k_{fus} * V_{endosome} - k_{trans,v} * V_{endosome} \qquad [7]$$

The rate equation for the release of virus into cytosol and accumulation in Nucleus were given as

[8]

[9]

$$\frac{dV_{cytosol}}{dt} = k_{fus} * V_{endosome} - k_n V_{cytosol}$$

$$\frac{dV_{nucleus}}{dt} = k_n V_{cytosol}$$

The balance of Intracellular Free Receptors

$$\frac{dR_{if}}{dt} = k_{er} R_{sf} - k_{rec} R_{if} - k_{tran,r} R_{if} \qquad [10]$$

The number of free surface receptors are

$$\frac{dR_{sf}}{st} = -(\alpha k_f C) V_{ex} R_{sf} + k_r V_1 - k_x R_{sf} \sum_{i=1}^{j} (j-1) V_i + k_{-x} \sum_{i=2}^{j} i V_i - k_{er} R_{sf} + k_{rec} R_{if} + k_s R_{so} \qquad [11]$$

Where, k_{er} = rate constant for endocytosis

$k_{tran,r}$, k_{rec}= transport and recycle rate constant

R_{so} = initial no. of free receptors

R_{if} = no. of free recptors

R_{sf} = No. of Bound receptors

2.1.3 Cycles, chaos, and evolution in virus cultures: A model of defective interfering particles

Defective interfering particles (DIP) are spontaneous deletion mutants of viruses that replicate at the expense of the parent virus. DIPs have complex effects on the growth of viruses *in-vitro*, including the establishment of persistent infection, cyclical variation in virus titer, eradication of replicating virus, and rapid evolution of the virus. The mathematical model based on experimental observations can be used to explain the major effects of DIP on the population dynamics of virus growth [6].

The number of uninfected cells were expressed as follows,

$$\frac{d(CU)}{dt} = \mathrm{\mu}_2 * CU - a * CU * (VS + VD) \qquad [1]$$

Number of uninfected cell infected by DIPs virus:

[2]

$$\frac{d(CD)}{dt} = \mu_2 * CD + a * CU * VD - a * VS * CD$$

Equation for formation of cell infected by standard virus:

$$\frac{d(CS_1)}{dt} = a * VS * CU - \frac{CS_1}{\tau} - p_1 * CS_1 - a * CS * VD * z_1 \quad [3]$$

$$\frac{d(CS_i)}{dt} = \frac{CS_{i-1} - CS_i}{\tau} - CS_i * p_i - a * CS_i * VD * z_i \quad [4]$$

$$1 < i < n$$

$$\frac{d(CS_n)}{dt} = \frac{CS_{(n-1)}}{\tau} - CS_n * p_n - a * CS_n * VD * z_n \quad [5]$$

Similarly, the equations for formation of cell infected by both standard and DIPs virus,

$$\frac{d(CB_1)}{dt} = a * VD * \Sigma CS_i + a * CD * VS - \frac{CB_1}{\tau} - CB_1 * p_1 \quad [6]$$

$$\frac{d(CB_i)}{dt} = \frac{\left(CB_{(i-1)} - CB_i\right)}{\tau} - CB_i * p_i \quad [7]$$

$$1 < i < n$$

$$\frac{d(CB_i)}{dt} = \frac{CB_{(n-1)}}{\tau} - CB_n * p_n \quad [10]$$

The virus dynamics was expressed as follows,

$$\frac{d(VS)}{dt} = b * \Sigma CS_i * p_i - a * VS * (sum(CS) + sum(CB) + CD + CU) \quad [11]$$

And the dynamics of DIPS were expressed as,

$$\frac{d(VD)}{dt} = b * \Sigma CB_i * p_i - a * VD * (sum(CS) + sum(CB) + CD + CU) \quad [12]$$

The above models are efficient in explaining the mechanism of virus internalization, DNA replication, mRNA expression as well as standard virus formation and DIPs formation. However, there is no simple unstructured model present to explain the cell growth, cell death dynamics. Moreover, there is no model for predicting the cell growth, cell death and polyhedral production for passaging of viruses for suspension cell culture. Hence we propose a model building scheme and parameter estimation that can be used for shaker flask experiments or fermenter operation.

2.2 Optimization method: non-linear constrained optimization

$$\text{min F(X) subject to: } A * X \leq B,\ A_{eq} * X = B_{eq} \quad \text{(linear constraints)}$$

$$C(X) \leq 0\ ,\ C_{eq}(X) = 0 \qquad \text{(nonlinear constraints)}$$

$$LB \leq X \leq UB \qquad \text{(bounds)}$$

fmincon function in MATLAB implements four different algorithms: interior point, SQP, active set, and trust region reflective. Interior point method the default algorithm to solve the optimization problem.

Syntax of fmincon in MATLAB

$x = fmincon(f(x), x_0, A, B, A_{eq}, B_{eq}, LB, UB, nonlcon)$ defines a set of lower and upper bounds on the design variables, X, so that a solution is found in the range LB <= x <= UB. The function NONLCON accepts X and returns the vectors C and Ceq, representing the nonlinear inequalities and equalities respectively. fmincon minimizes FUN such that C(X) <= 0 and Ceq(X) = 0.

Where, x_0 in the initial guess provided for solving the problem.

Example of solving nonlinear constrained optimization

The objective function need to be minimizaed such that they should satisfy the equality and inequality condition. Now, converting nonlinear Constrained optimization problem to standard form,

$$\begin{array}{ll} minimize & f(x) \\ \text{s. t.} & g(x) \geq b \\ & h(x) = 0 \end{array} \quad \longrightarrow \quad \begin{array}{ll} minimize & f(x) \\ \text{s. t.} & \left.\begin{array}{r} g(x) - b - s = 0 \\ h(x) = 0 \end{array}\right\} C(x) = 0 \end{array}$$

$$s = 0$$

$$x = 0$$

Converting this standard form into Barrier function

$$\begin{array}{lll} \textit{minimize} & f(x) & \\ \text{s. t.} & C(x) = 0 & \longrightarrow \\ & x \geq 0 & \end{array} \qquad \begin{array}{ll} \textit{minimize} & f(x) - \mu \sum_{i=1}^{n} \ln(x_i) \\ \text{s. t.} & C(x) = 0 \end{array}$$

Karush-Kuhn-Tucker (KKT) conditions for Barrier problem

$$\begin{array}{ll} \textit{minimize} & f(x) - \mu \sum_{i=1}^{n} \ln(x_i) \\ \text{s. t.} & C(x) = 0 \end{array} \quad \longrightarrow \quad \begin{array}{ll} \multicolumn{2}{l}{\nabla f(x) + \nabla c(x)\lambda \ - \ \mu \displaystyle\sum_{i=1}^{n} \frac{1}{x_i} = 0} \\ \text{s. t.} & C(x) = 0 \end{array}$$

Define $z_i = \mu / x_i$ and solving the modified versions of KKT conditions

$$\nabla f(x) + \nabla c(x)\lambda \ \text{-}\ z = 0$$

$$C(x) = 0$$

$$XZe - \mu e = 0$$

Where, e is column matrix of ones

Next, finding the KKT solutions using Newton-Raphson Method

$$\nabla f(x) + \nabla c(x)\lambda \ \text{-}\ z = 0$$

$$\longrightarrow$$

$$XZe - \mu e = 0$$

$C(x) = 0$ $$\begin{bmatrix} W_k & \nabla c(x_k) & -I \\ \nabla c(x_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k \text{ - } z_k \\ C(x_k) \\ X_k Z_k e - \mu_j e \end{pmatrix}$$

$$W_k = \nabla_{xx}^2 L(x_k, \lambda_k, z_k) = \nabla_{xx}^2 (f(x_k) + c(x_k)^T \lambda_k \text{ - } z_k$$

$$z_k = \begin{bmatrix} z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & z_n \end{bmatrix} \qquad x_k = \begin{bmatrix} x_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & x_n \end{bmatrix}$$

Rearranging into symmetric linear system,

$$\begin{bmatrix} W_k + \Sigma_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ C(x_k) \end{pmatrix} \qquad \Sigma_k = X_k^{-1} Z_k$$

Solving for d_k^z after the linear solution to d_k^x and d_k^λ with explicit solution

$$d_k^z = \mu_k X_k^{-1} e - z_k - \Sigma_k d_k^x$$

Step Size(α)

Two objective in evaluating progress

i. Minimize objective

ii. Minimize constraint violations

Two popular approaches

i. Decrease the merit function $merit = f(x) + \nu\Sigma|c(x)|$

ii. Filter methods

Cut back size until improvement

$x_{k+1} = x_k + \alpha_k d_k^x$

$\lambda_{k+1} = \lambda_k + \alpha_k d_k^\lambda$

$z_{k+1} = z_k + \alpha_k d_k^z$

Convergence criteria

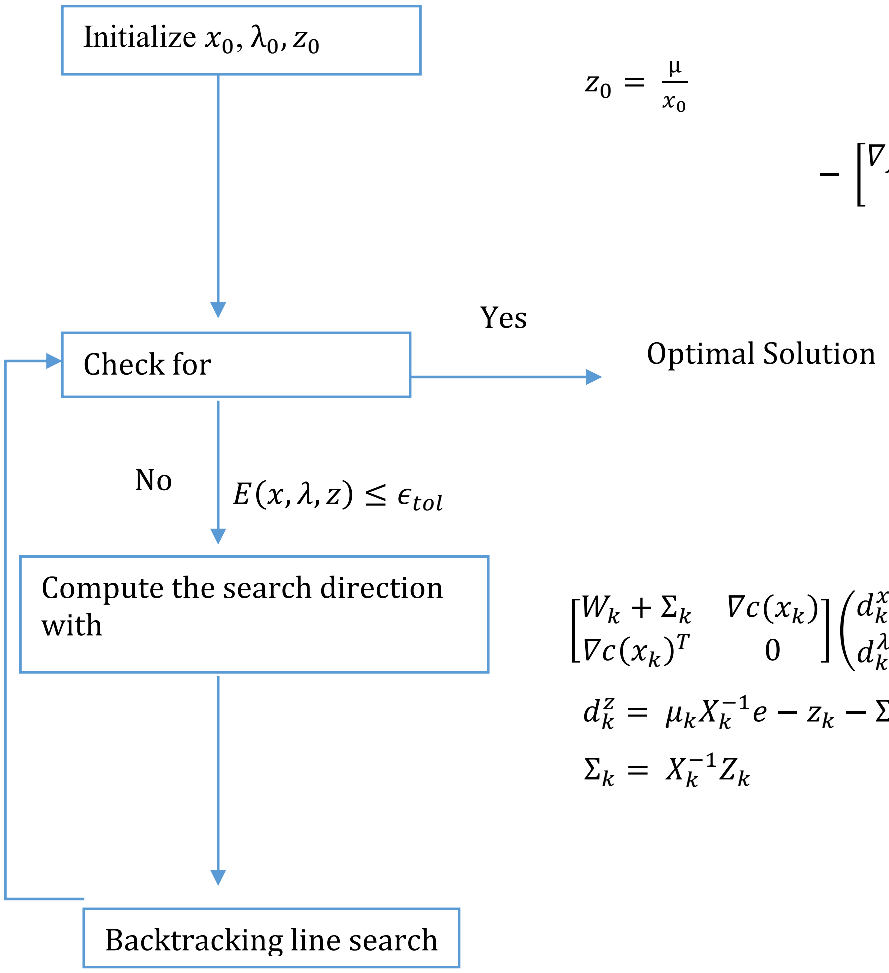
Convergence when KKT conditions are satisfied with a tolerance, (Tolerance for constraints violations maybe more restrictive

$$max|\nabla f(x) + \nabla c(x)\lambda\ \text{- z}| \leq\ \epsilon_{tol}$$

$$max|c(x)| \leq \epsilon_{tol}$$

$$max|XZe -\ \mu e| \leq\ \epsilon_{tol}$$

Interior point method Overview


Initialize x_0, λ_0, z_0
Check for
Yes
Optimal Solution
No
$E(x, \lambda, z) \leq \epsilon_{tol}$
Compute the search direction with
Backtracking line search

$$x_0 = feasilble \qquad Initialize\ \lambda$$

$$z_0 = \frac{\mu}{x_0} \qquad \begin{bmatrix} I & \nabla c(x_0) \\ \nabla c(x_0)^T & 0 \end{bmatrix} \begin{pmatrix} W \\ \lambda_0 \end{pmatrix} = -\begin{bmatrix} \nabla f(x_0) - z_{l,0} - z_{u,0} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} W_k + \Sigma_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ C(x_k) \end{pmatrix}$$

$$d_k^z = \mu_k X_k^{-1} e - z_k - \Sigma_k d_k^x$$

$$\Sigma_k = X_k^{-1} Z_k$$

$$x_{k+1} = x_k +\ \alpha_k d_k^x$$

$$\lambda_{k+1} = \lambda_k +\ \alpha_k d_k^\lambda$$

$$z_{k+1} = z_k +\ \alpha_k d_k^z$$

Determine α by decrease in merit

function or with filter Methods

Chapter 3

Modeling of baculovirus infection process

3.1 Introduction

Construction of unstructured model consists of major two steps of building a model database and testing the models with multiple experimental dataset. For selection of best model, we estimated the parameters that can accurately describe the experimental dataset minimization of the error value. The error function was defined as the sum of the square of the differences between the model predictions and experimental data. Parameter estimation is critical because the infection process is highly non-linear and stochastic.

Experimental set up: Insect cells (Sf-9 cells) were grown for two cases, where one contains cells without any viral infection and the other one containing cells with AcMNPV and Ac-FPm are of two types. A recombinant baculovirus denoted Ac-FPm was constructed to remove the 13 TTAA transposon target sites in the AcMNPV *fp25k* gene such that the amino acid sequence of the FP25K protein remained unchanged. AcMNPV (wild type) and Ac-FPm (Stabilized type)**.** In case 1, cells will grow with substrate and oxygen Figure 3.1 but in case of case 2 with viral infection, cells will grow along with baculovirus formation Figure 3.2. Additionally, the passaging experiments were performed in shaker flasks as mentioned in Figure 3.3. The infection process mechanism is explained with detail in Figure 3.4 Thus, in case 1, the model variables are: cell mass, substrate, oxygen and carbon di-oxide while in case2, the model variables are: cell mass, virus concentration, oxygen, carbon di-oxide and baculovirus as product

3.2 Unstructured model: Model selection strategy

The model for the infection process was constructed using a set of coupled ordinary differential equations initial value problems (ODE-IVPs) with various kinetic parameters such as cell growth, substrate consumption, cell death, oxygen consumption rate and carbon-di-oxide formation. Various kinetic parameters include cell growth rate, maintenance coefficient, product degradation rate and other kinetic parameters. The measurable states were cell mass concentration, cell viability and baculovirus concentration. Since large number of mechanisms can be constructed using the combinations of various cell growth, death, baculovirus formation and degradation models, we explored a set of total 24 mechanisms. Here we provide a comprehensive comparison of various models for the cell growth. The simulation of the temporal profiles for cell, substrate and product in case of all models were performed by solving ODE-IVPs by the fourth-order Runge-Kutta method. The initial value problems (ODE-IVPs) was formulated using various kinetic parameters comprising of mainly rate constants describing the constants such as specific growth rate, substrate consumption rate, death rate of cells, oxygen consumption rate, carbon-di-oxide formation rate, maintenance coefficient, product degradation rate.

We have assumed that growth of uninfected cell is depending on oxygen, substrate and carbon dioxide but infected cell growth does not depend on these variables and we are also assuming first order death rate for certain period of time and then it is depending on number of virus in the cell. also, in case of infected cell growth, we were having triplicates of data and for every experimental set we were having different initial conditions which were following different growth patterns. So, we have changed growth and death rate for each sets of data and we are calculating RMSE from all three sets of data. The initial condition for uninfected cell model is shown in Table 3.1 and in case of infection in Table 3.2 and Table 3.3 for wild type virus and stabilized virus respectively.

Based on the model comparison study, we proposed the specific mechanism that fits the experimental data with least root mean square error value (RMSE). The following subsection describes the method for parameter estimation used in details.

3.2.1 Model formulation for cell growth and cell viability

A computational strategy has been implemented to compare among various unstructured models and identify the model that fits best the experimental data. The selected models corresponding to the cell growth without viral infection and with viral infection are presented below. The cell growth was modeled as the term ($[Cell\ mass]$), where μ is the specific growth rate given by equation 1 for both with and without viral infection. The details of the various models for cell growth are presented in term T11 in Table 3.4.

The detailed model formulation and the explanation of various terms are given below.

$$\mu = \mu_{max} * \left(\frac{Substrate}{K_s + Substrate}\right) \qquad [1]$$

$$\mu_2 = \mu * \left(\frac{Oxygen}{Oxygen\ + K_{o_2}}\right) * \left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K}_{\mathrm{CO}_2}}\right) + 1}\right) \qquad [2]$$

$$\frac{d(Uninfected\ cell)}{dt} = \mu_2 * Uninfected\ cell\ - a * Uninfected\ cell * Virus$$
$$-k_{d_1} * Uninfected \qquad [3]$$

$$\frac{d(Infected\ cell)}{dt} = a * Uninfected\ cell * Virus - k_d * Infected\ cell \qquad [4]$$

$$\frac{d(Oxygen)}{dt} = kla * ({O_2}^* - Oxygen)$$
$$- \frac{Uninfected\ cell + Infected\ cell}{Y_o} \qquad [5]$$

$$\frac{d(Carbon\ di - oxide)}{dt} = \frac{Uninfected\ cell + Infected\ cell}{Y_c} \qquad [6]$$

$$\frac{d(Dead\ cell)}{dt} = k_{d_1} * Uninfected\ cell + k_d * Infected\ cell \qquad [7]$$

$$\frac{d(Virus)}{dt} = b * Infected\ cell - a * Uninfected\ cell * Virus \qquad [8]$$

$$\frac{d(Substrate)}{dt} = -\mu * \left(\frac{Uninfected\ cell + Infected\ cell}{Ys}\right) \qquad [9]$$

Equation 1 describes the dependence of specific growth rate on substrate [7] whereas equation 2 describes the dependence of specific growth rate on oxygen and carbon di-oxide [9]. Equation 3 describes dynamics of uninfected cell mass concentration. In this equation, the first term represents the growth of uninfected cell mass, second term represents the infection rate and third term represents death of the uninfected cells. Equation 4 describes the dynamics of infected cell mass concentration [5]. In this equation first term represents growth of the infected cell from infection and second term represents death of the infected cell [8] [10]. Equation 5 and 6 describes the dynamics of oxygen and carbon di-oxide concentration [10]. Equation 7 describes the dynamics of dead cell concentration whereas equation 8 describes the rate of change of virus concentration [6]. Equation 9 describes the rate of substrate depletion. First term in the equation is the amount of substrate [7].

$$\frac{d[uninfected\ cell]}{dt} = T_{21} + T_{22} + T_{23}$$

$$\frac{d(Infected\ cell)}{dt} = T_{31} + T_{32}$$

$$\frac{d(oxygen)}{dt} = T_{41} - T_{42}$$

$$\frac{d(carbon\ dioxide)}{dt} = T_{51}$$

$$\frac{d(dead\ cells)}{dt} = k_d(T_{61} + T_{62})$$

$$\frac{d(virus)}{dt} = T_{71} - T_{72}$$

$$\frac{d(Substrate)}{dt} = T_{81}$$

However, for constructing a model for uninfected cell, we followed the similar procedure but without the variables such as infected cell and virus concentration. For example, term T_{22} and T_{42} were considered as zero as no infection takes place in case of un-infected cell growth.

3.3 Parameter Estimation

A non-linear programming (NLP) problem was formulated with an error function as the objective function. The error function is described in equation 10. The rate constants were set as the decision variables whose upper bounds and lower bounds were determined carefully by performing sensitivity analysis with respect to each parameter in the model. Sensitivity analysis was carried out by perturbing one parameter at a time till the limit within which the model converged, while other parameters were fixed at some base values. This ensured the convergence of ODE- IVP model for the candidates of parameters generated by the optimizer. The NLP formulation is given below.

Minimize Error

Such that,

$$0 \leq K_p \leq K_p^{UL}\ \forall\ p = 1\ to\ N$$

where,

$$Error = \sum_{k=1}^{3}\sum_{j=1}^{2}\sqrt{\frac{1}{N_i}\left(\sum_{i}^{N_j}\left(y_i^j - \hat{y}_i^j\right)^2\right)} \qquad [10]$$

$\hat{y}_i^j = f(K_p, y_0^j)$ Where, f is the proposed ODE – IVP model and y_0^j is initial conditions

y_i^j is the i^{th} experimental value (concentration) of j^{th} component.

$\hat{y}_i^j$ is the i^{th} simulated value (concentration) of j^{th} component.

$\forall\ i = 1\ to\ N_j$

N_j is number of experimental observations

$\forall\ j = 1\ to\ 2$ | 1: Cell Density, 2: Cell viability

N is number of parameters in the proposed model.

$\forall\ k = 1\ to\ 3$ is for three data set

Fmincon is an optimization technique which implements interior point method to find the local optima. However, one of the disadvantage of using this optimizer is to reach the local minima rather than global minimum. The flowchart for the model selection and parameter estimation algorithm is depicted in figure. Here, we started with i = 1 where i corresponds to i^{th} model to be formulated from the possible model structures. The objective function was calculated through the computation of root mean square values obtained from the triplicate dataset. The day to day variability in experiments were attributed to the variation in cell growth rate and death rate constant across infection experiments performed on different days.

3.4 Multiple model formation

In order to justify the efficiency of the proposed model and corresponding mechanisms, we compared them with other models that are potential to qualify as possible unstructured models. The other models are also capable of emulating the trend in cell growth dynamics. The general model structure for the two cases are as shown below.

$$\mu_2 = \mu_{max}(T_{11})(T_{12})(T_{13})$$

We selected twelve cell growth models by varying the specific growth rate using different models as shown in table 3.4.

3.5 Sensitivity analysis

Sensitivity analysis is an important tool that is used to identify the parameters that affect the response variable. This technique is useful for finding out the parameter range in which the parameter estimation is to be conducted. From this analysis, we obtained the desired parameter range for choosing the initial guess values. We performed the sensitivity analysis with +100% change in each of the parameter and observed the effect on two variables, cell density and percentage of viable cells.

3.6 Analysis of the proposed models and comparison with other possible mechanisms

A structure representing series of prominent models considered are presented for un-infected and infected cells in figures respectively. Table 3.5 shows comparison between simulated values for various growth models. The results clearly show that Monod model is best compared to other models for un-infected cell growth. The value of objective function RMSE is 22700 which is less than other possible models. Table 3.7 shows the comparison between simulated values for various growth models for infection with wild type virus and stabilized virus. Monod model is best compared to other models in the case of wild type virus infection and contois model is best compared to other models in case of stabilized virus infection. This was evident through the lowest value of the objective function obtained for this case when compared to the other models RMSE of 33400 for wild type and 43400 for stabilized type viral infection. Out of the multiple models we have sorted the RMSE and provided the least RMSEs of 9 models for un-infected cell and 9 models for infected cell. Of all the models considered, the best model is chosen based on the least value of RMSE and presented in Tables.

3.7 Results

In order to identify the most sensitive parameters and their ranges to be used, first we performed the sensitivity analysis. Figure 3.5 shows the 3-D representation of the parametric sensitivity analysis for the proposed model. Where X axis is the kinetic parameters, Y axis is model variables and Z axis is the percentage variation of the response variables. The result shows that the cell density is sensitive to specific growth rate, saturation constant (CO_2), and cell viability is sensitive to specific growth rate, infection rate constant and virus production rate.

3.8.1 Multiple model formulation for uninfected and infected cells

In order to identify the best growth model for uninfected insect cell growth, we constructed multiple models and estimated kinetic parameters for all of them. For all the models present in Table 3.4, we estimated parameters using NLP formulation according to the workflow shown in Figure 3.6. The comparison of simulated results and experimental data for uninfected cells are presented in Figure 3.7. The final results for the best model for growth of uninfected cells are shown in Figure 3.8. The RMSE corresponding to each of the models tested are presented in Table 3.5 and estimated parameter values are listed in Table 3.6 for uninfected cell.

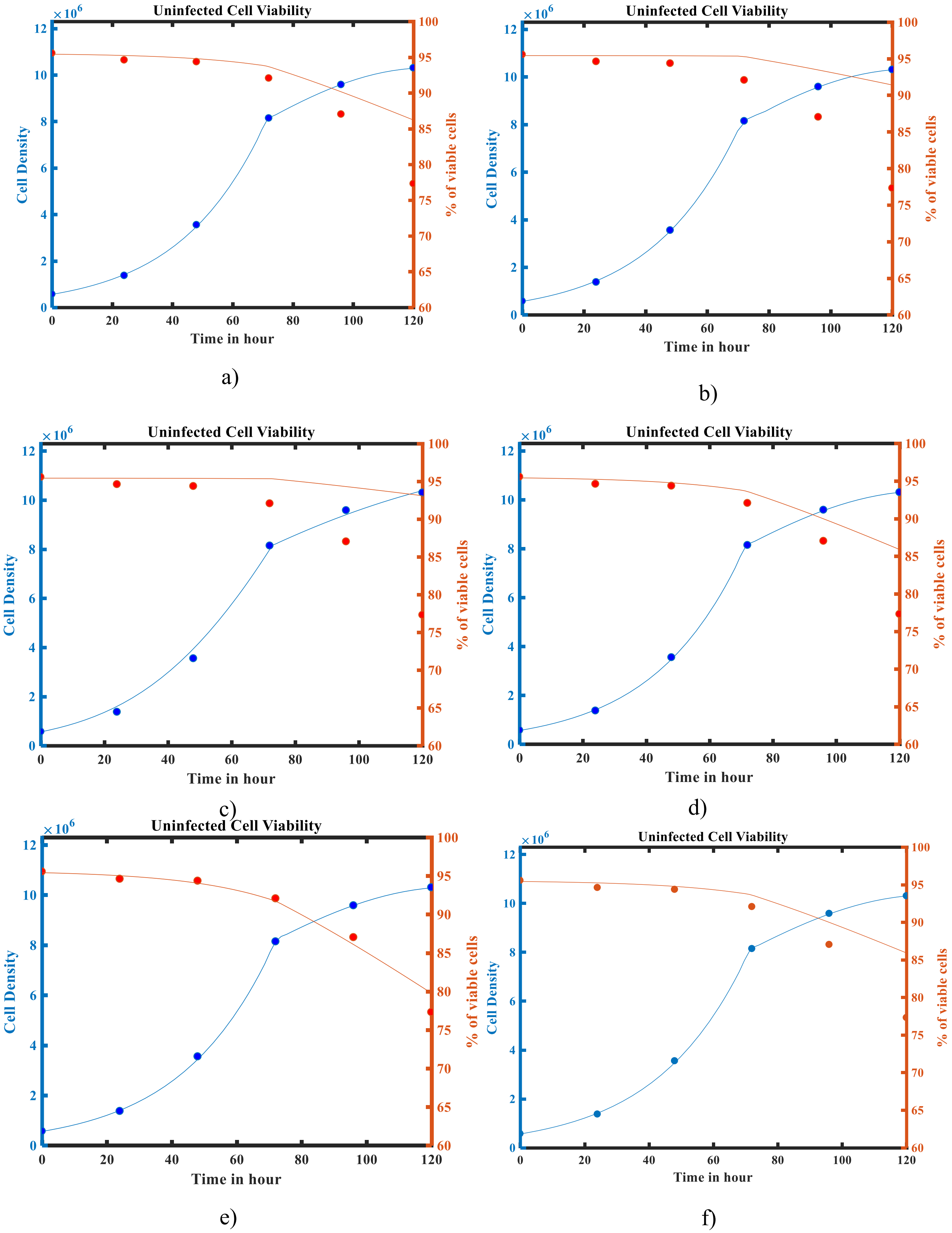
Similarly, in case of infected cell growth we formulated multiple models followed by parameter estimation for each of the model. We have considered the model having lowest RMSE as the best model for cell growth. In order to account for the day to day variability in growth profile and viable cell density profile, we assumed different death rate and specific growth rate. Table 3.4 shows the different growth models considered for growth of infected cells. The specific cell growth was assumed as a function of substrate, carbon dioxide and oxygen concentration. The comparison between simulated results and experimental data on cell density of WT AcMNPV are shown in Figure 3.9. Similarly, the comparison of simulated results and experimental data for cell viability of WT AcMNPV is shown in Figure 3.10. The results show the comparison between various plausible models and Monod's model is identified as the best fit having lowest RMSE. Figure 3.11 and Figure 3.12 shows the similar comparison between simulated results and experimental data for cell density and cell viability for the stabilized AcMNPV. In contrast to the WT AcMNPV, the model comparison results show that Contois model is the best fit to capture the growth of cells infected by stabilized virus.

Table 3.7 shows the comparison of RMSEs corresponding to each of the model after estimating the parameters for wild type and stabilized virus. The comparison between the experiment and simulation for the best model for WT and stabilized are shown in Figure 3.13 and Figure 3.14. Table 3.8 shows the comparison of estimated kinetic parameters for the best model having lowest RMSE for WT and stabilized virus. The result indicates that the specific growth rate and death rate may vary with experiments performed on different days with some deviations (Table 3.8).

Also, the result shows that the parameters corresponding to the growth kinetics and death kinetics are different in case of cell growth with viral infection and growth without viral infection (Table

3.6 and Table 3.8). Similarly, the results presented in Table 8 depicts that there is a difference between the kinetic parameters estimated for the case of WT and stabilized virus.

Figure 3.15 and Figure 3.16 shows the comparison between the experimental and simulated results for the cell viability and cell density (at 48 hr pi) after passaging for wild type virus and and stabilized virus respectively. In this case we have used the best model chosen from the previous analysis. The results show that a variation in specific growth rate at each passage may account for the variation in cell density and cell viability (Table 3.9).



a)

b)

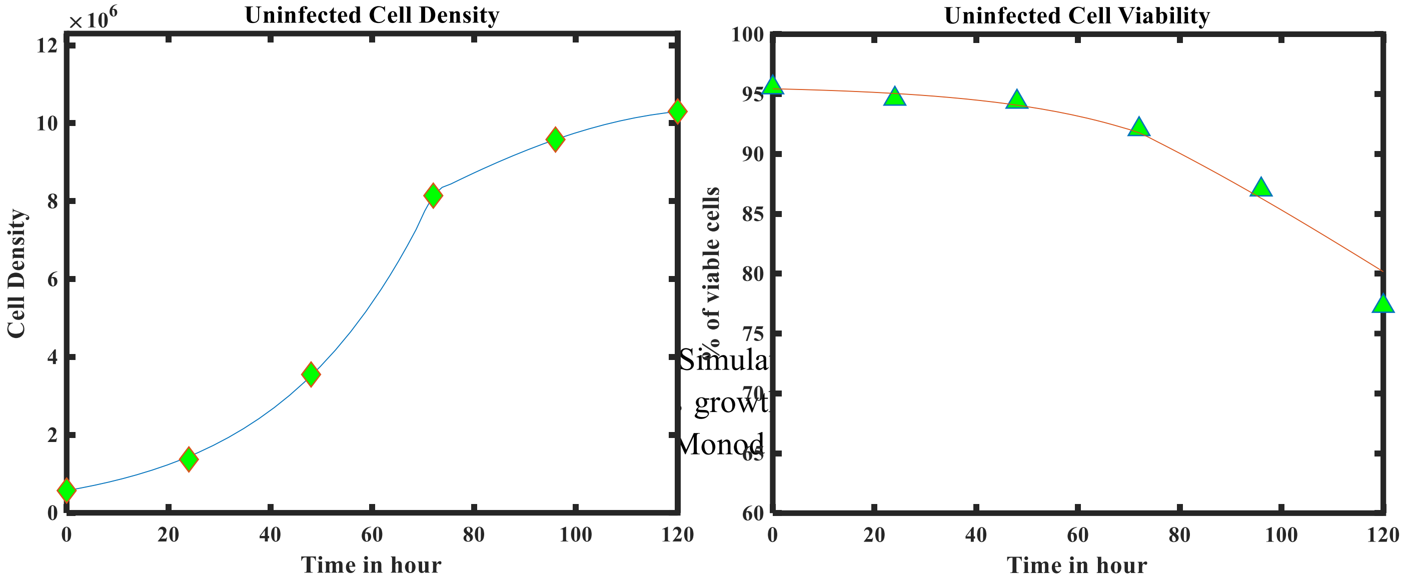
c)

d)

e)

f)

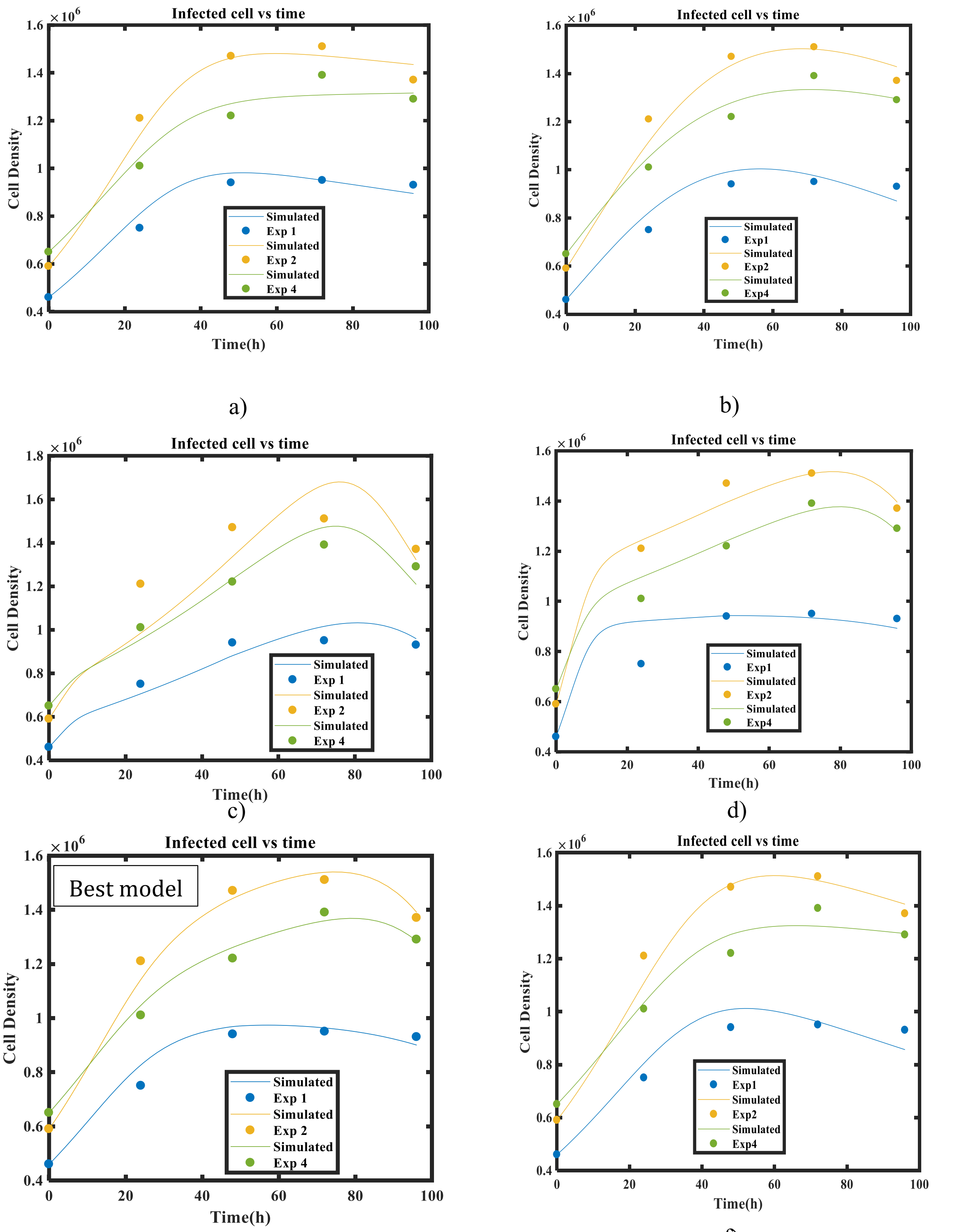
Figure 3.7 Comparison of experimental and simulated data for cell density and cell viability of uninfected cell growth experiment by various growth kinetic models a) Aiba model b) Andrews model c) Contois model d) Haldane model e) Monod model and f) Tessier model



a)

b)

Figure 3.8 Simulated and experimental results for the proposed model for uninfected cell growth experiment a) Uninfected cell density (number of cells/ milliliter) vs. time and b) percentage of viable cell vs. time



a)

b)

c)

d)

e)

f)

Figure 3.9 Comparison of experimental data and simulated results for cell density (number of cells/ milliliter) of infected cells for Wild type virus (AcMNPV) by various kinetic models: a) Aiba model b) Andrews model c) Contois model d) Haldane model e) Monod model f) Tessier

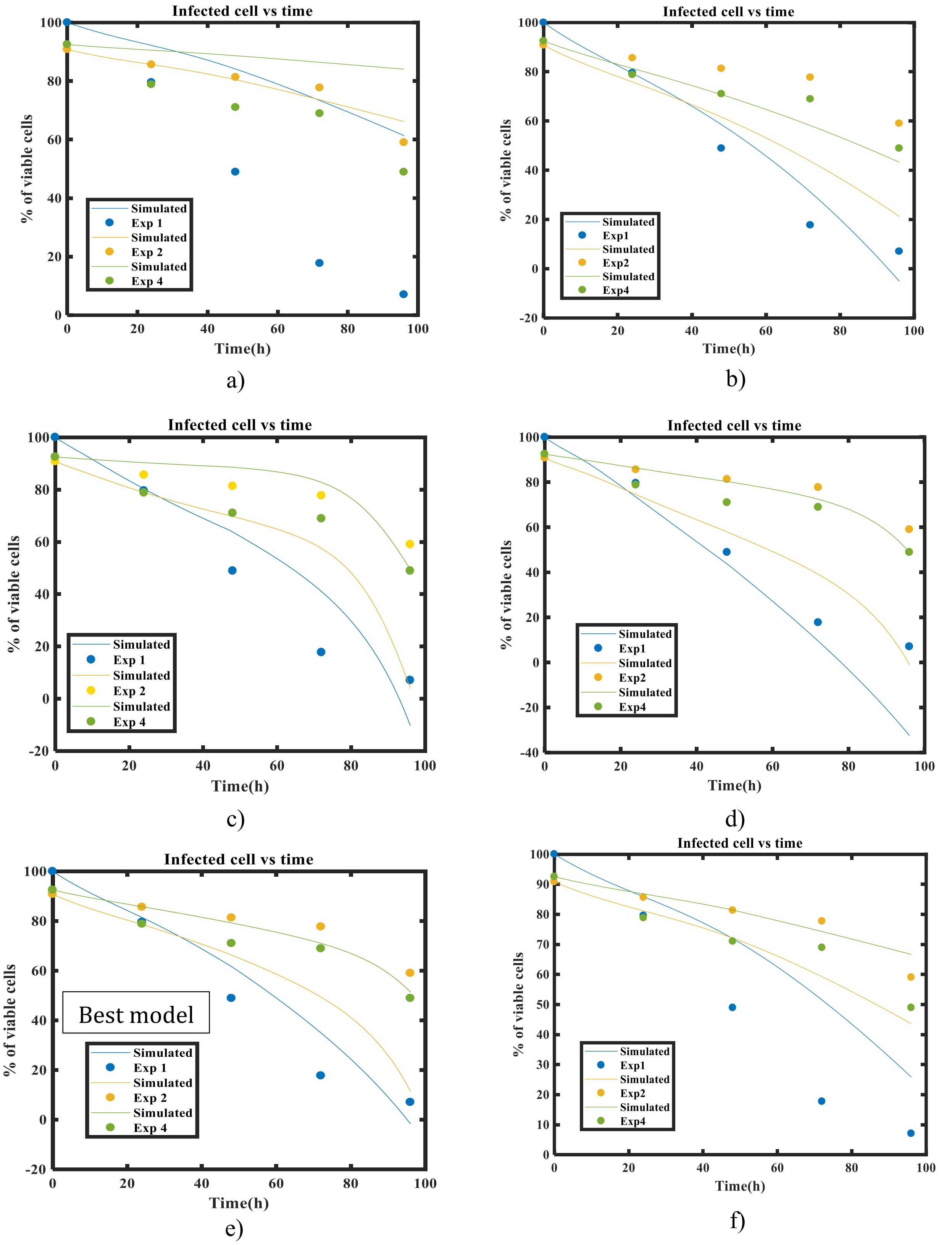
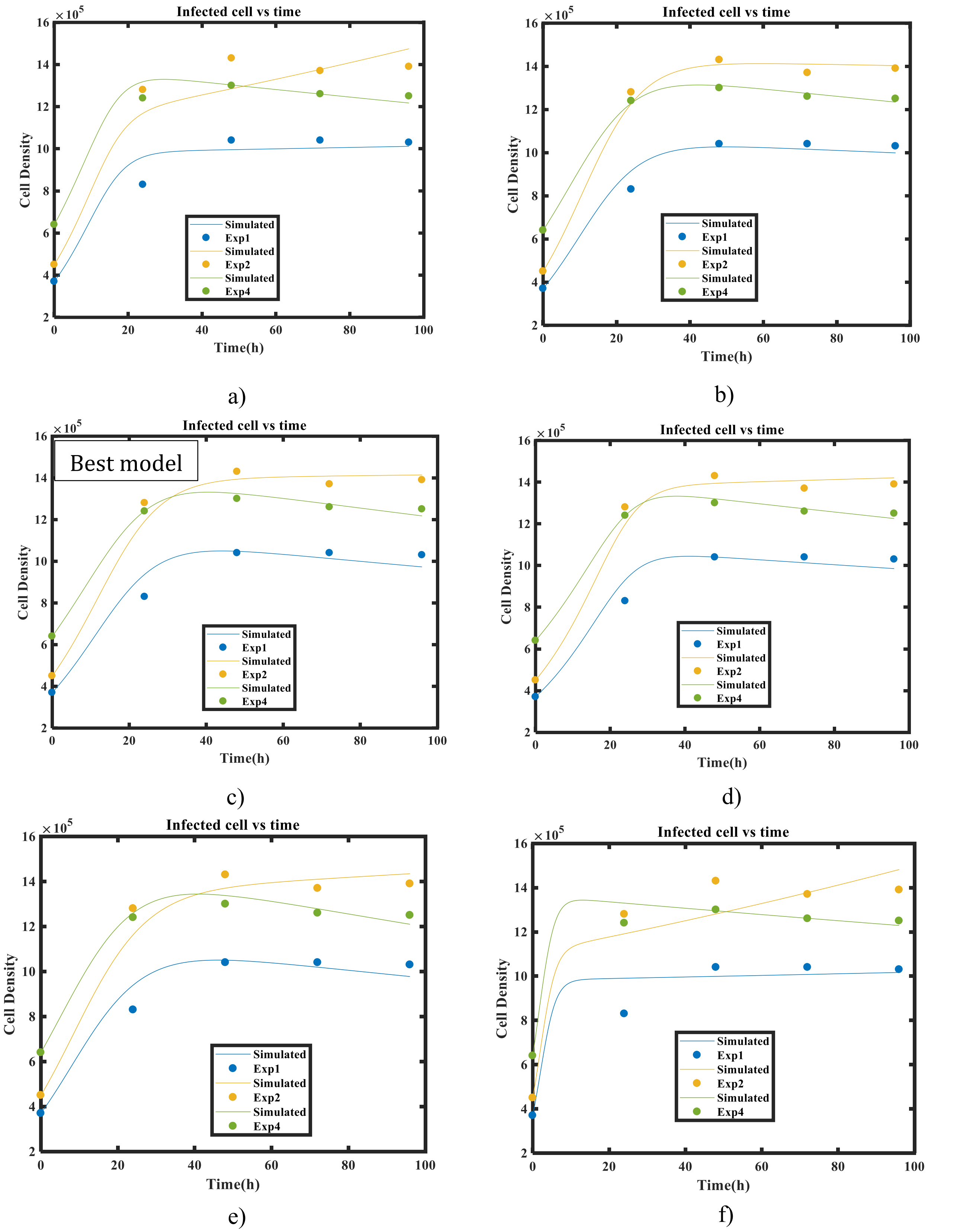


Figure 3.10 Comparison of experimental data and simulated results for cell viability of infected cells for Wild type virus (AcMNPV) by various kinetic models: a) Aiba model b) Andrews model c) Contois model d) Haldane model e) Monod model f) Tessier Model



a)

b)

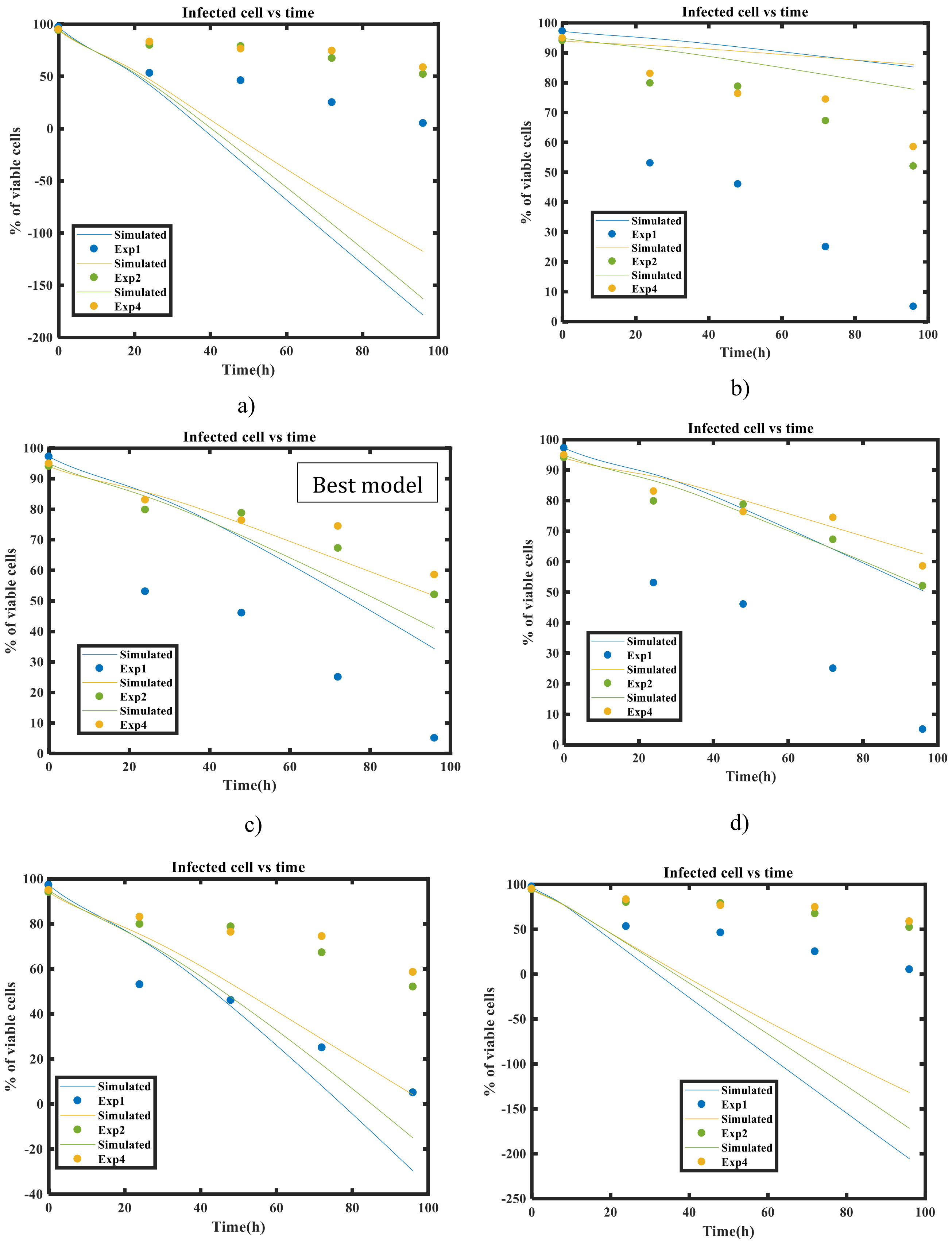
c)

d)

e)

f)

Figure 3.11 Comparison of experimental data and simulated results for cell density (number of cells/ milliliter) of infected cells for stabilized virus (Ac-FPm) by various kinetic models: a) Aiba model b) Andrews model c) Contois model d) Haldane model e) Monod model f) Tessier Model



a)

b)

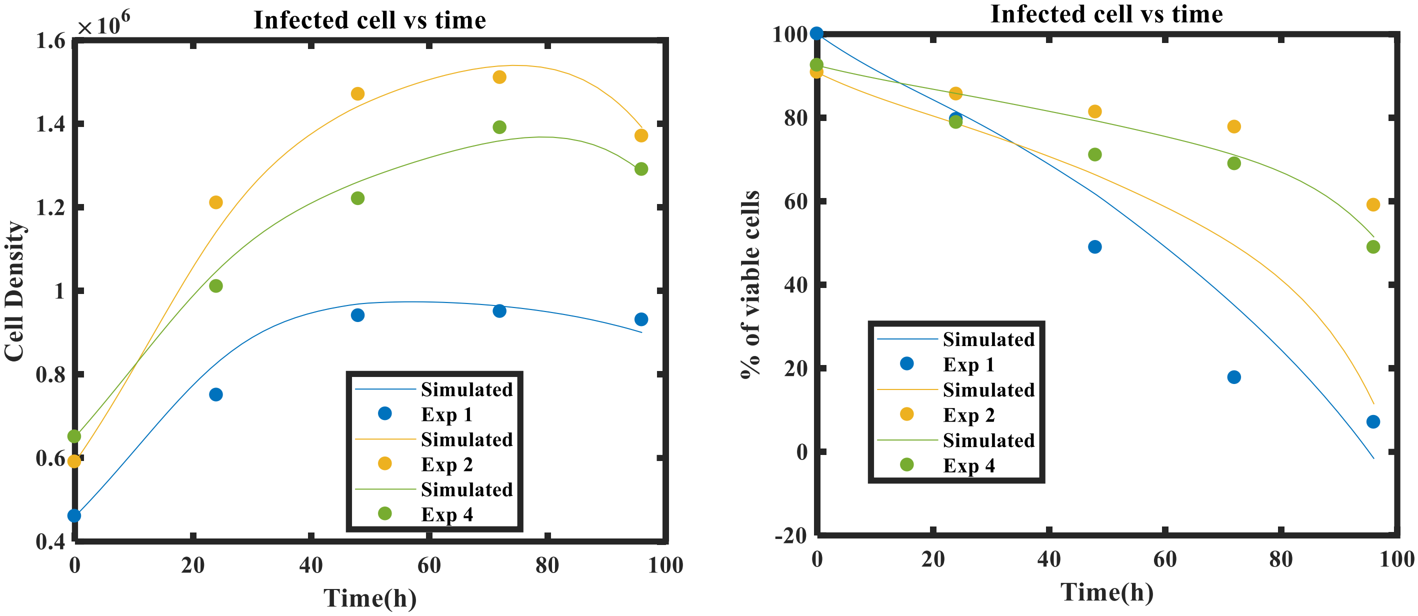
c)

d)

e)

f)

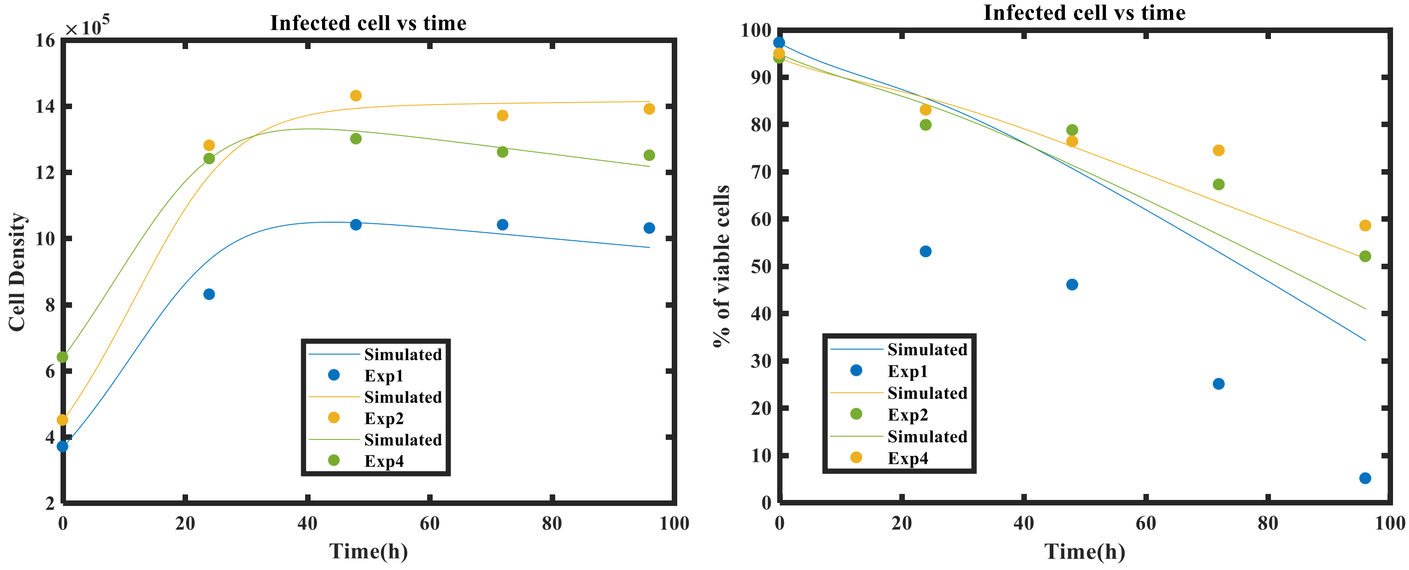
Figure 3.12 Comparison of experimental data and simulated results for the cell viability of infected cells for stabilized virus (Ac-FPm), a) Aiba model b) Andrews model c) Contois model d) Haldane model e) Monod model f) Tessier Model.



a)

b)

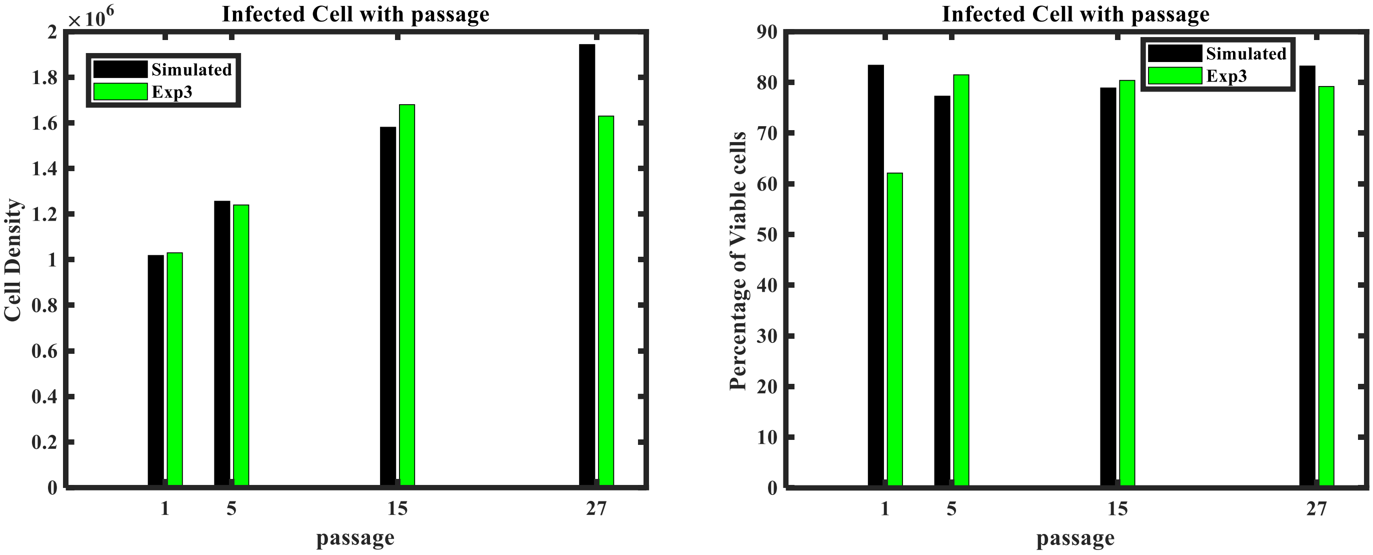
Figure 3.13 Simulated and experimental results for the proposed model for infected cell growth experiment for WT virus a) infected cell density (number of cells/ milliliter) with time and b) percentage of viable cell with respect to time



a)

b)

Figure 3.14 Simulated and experimental results for the proposed model for infected cell growth experiment for stabilized virus a) infected cell density (number of cells/ milliliter) with time and b) percentage of viable cell with respect to time



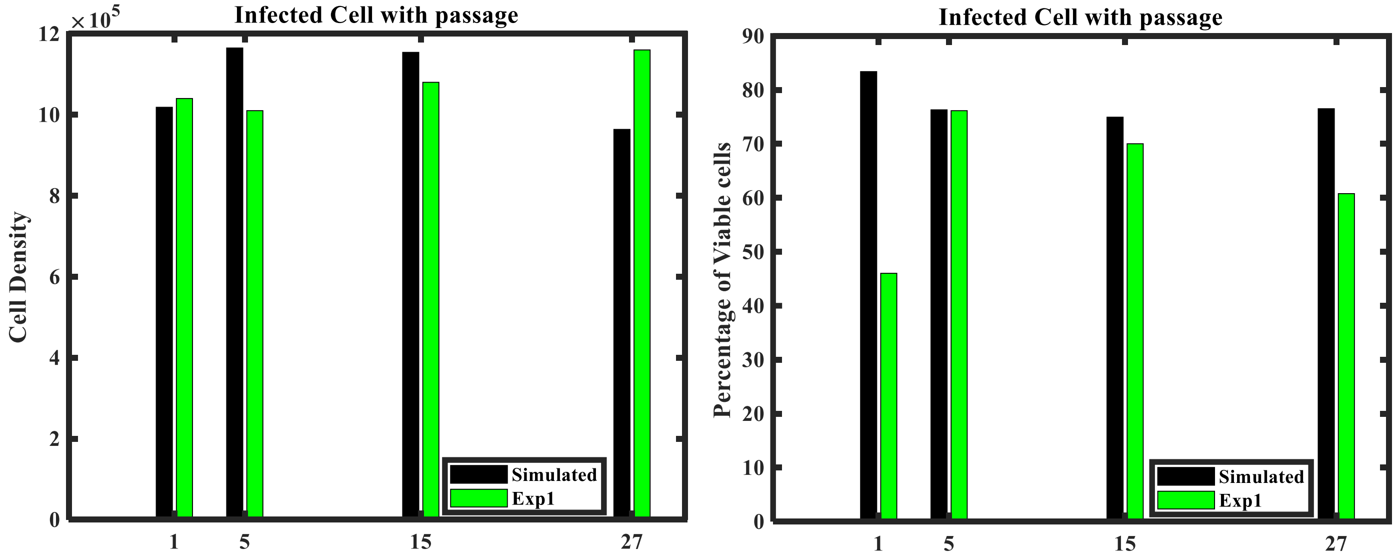
Passage number

Passage number

a)

b)

Figure 3.15 Comparison of Simulated and experimental data for wild type viral infection at different passage number for one set of data a) Cell density (number of cells/ milliliter) with respect to passage b) percentage of viable cells with respect to passage



Passage number

Passage number

a)

b)

Figure 3.16 Comparison of Simulated and experimental data for stabilized viral infection at different passage for one set of data a) Cell density (number of cells/ milliliter) with respect to passage b) percentage of viable cells

Table 3.1 Initial Conditions: uninfected cell

Variable	Uninfected cell	Substrate	Oxygen	Carbon dioxide	Dead cell
Initial Condition	570000	1000	1000	0	0

Table 3.2 Initial Conditions: infected cell, wild type virus

Variable (in case of wild type virus)	Uninfected cell	Substrate	Oxygen	Carbon dioxide	Dead cell	Virus	Infected cell
Initial Condition (data 1)	460000	1000	1000	0	0	5150	0
Initial Condition (data 1)	590000	1000	1000	0	0	5150	0
Initial Condition(data 1)	650000	1000	1000	0	0	5150	0

Table 3.3 Initial Conditions: infected cell, Stabilized virus

Variable (in case of stabilized virus)	Uninfected cell	Substrate	Oxygen	Carbon dioxide	Dead cell	Virus	Infected cell
Initial Condition (data 1)	370000	1000	1000	0	0	5150	0
Initial Condition (data 1)	450000	1000	1000	0	0	5150	0
Initial Condition(data 1)	640000	1000	1000	0	0	5150	0

Table 3.4 List of growth models for growth of infected and uninfected cells

Model No	Growth kinetic Model	T11	T12	T13
1	Aiba	$\frac{S}{S+Ks} * \exp\left(-\frac{S}{Ksi}\right)$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
2	Andrews	$1/\left(1+\frac{Ks}{s}\right)\left(1+\frac{S}{Ksi}\right)$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
3	Contois	$\frac{S}{K_s * CU + S}$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
4	Haldane	$\frac{S}{K_s + S + \frac{S^2}{Ksi}}$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
5	Monod	$\frac{S}{K_s+S}$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
6	Tessier	$1 - \exp\left(-\frac{S}{Ksi}\right)$	$\frac{O_2}{O_2+K_{O_2}}$	$\left(\frac{K}{\exp\left(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}}\right)+1}\right)$
7		$\frac{S}{K_s+S}$	0	0
8		$\frac{S}{K_s+S}$	$\frac{O_2}{O_2+K_{O_2}}$	0
9		$\frac{S}{K_s+S}$	0	$\left(\frac{K}{\exp(\frac{\text{carbon dioxide}}{\mathrm{K_{co2}}})+1}\right)$

Table 3.5 RMSE for different models used to identify the best model for uninfected cell

Model name & number	RMSE
1. Aiba	192000
2. Andrews	67400
3. Contois	47400
4. Haldane	32100
5. Monod	22700
6. Tessier	218000
7	3230000
8	327000
9	6790000

Table 3.6 Estimated parameter for the proposed model: uninfected cell growth

Parameters and their units	Parameter name	value
$\mu(hour^{-1})$	Specific growth rate	0.227
$k_{d_1}(hour^{-1})$	Intrinsic death rate constant	0.055
$k_{d_2}(hour^{-1})$	Death rate constant	0.082
$Y_s(g/g)$	Yield of cell/ substrate	$4 * 10^6$
$K_{o_2}(gL^{-1})$	Half-Velocity constant	128.37
$Y_c(g/g)$	Yield of cell/ carbon dioxide	$10\ X\ 10^6$
$K_{co_2}(gL^{-1})$	Half-Velocity constant	324.42
$K_s(gL^{-1})$	Half-Velocity constant	0.0115
$K(gL^{-1})$	Equilibrium constant	0.925
$Y_o(g/g)$	Yield of cell/ oxygen	$10\ X\ 10^7$

Table 3.7 RMSE for different models used to identify the best model for infected cell growth

Model no	RMSE (wild type virus)	RMSE(stabilized virus)
1. Aiba	43400	156000
2. Andrews	69500	78500
3. Contois	99900	43400
4. Haldane	52400	60900
5. Monod	33400	59900
6. Tessier	46300	243000
	71600	81400
	214000	387000
	241000	401000

Table 3.8 Estimated parameter for the proposed model: infected cell growth

Parameters and their units	Values (wild virus) Exp 1	Values (wild virus) Exp 2	Values (wild virus) Exp 3	Mean and standard deviation	Values (stabilized virus) Exp1	Values (stabilized virus) Exp2	Values (stabilized virus) Exp 3	Mean and standard deviation
$\mu_{max}(hour^{-1})$	0.0064	0.0099	0.0099	0.0087 ±0.00196	0.005	0.0069	0.0069	0.006 ±0.001
$k_{d_1}(hour^{-1})$	1	0.095	0.01	0.368 ±0.548	0.00514	0.005	0.00596	0.0054 ±0.005
$k^*(hour^{-1})$	0.1991	0.1991	0.1991		0.1565	0.1565	0.1565	
$Y_s(g/g)$	$10\ X\ 10^5$	$10X10^5$	$10\ X\ 10^5$		$10\ X\ 10^5$	$10\ X\ 10^5$	$10\ X\ 10^5$	
$Y_{o_2}(g/g)$	$10\ X\ 10^7$	$10X10^7$	$10\ X\ 10^7$		$10\ X\ 10^7$	$10\ X\ 10^7$	$10\ X\ 10^7$	
$K_{o_2}(gL^{-1})$	69.35	69.35	69.35		86.20	86.20	86.20	
$Y_{co_2}(g/g)$	$4\ X\ 10^8$	$4\ X\ 10^8$	$4\ X\ 10^8$		$4\ X\ 10^8$	$4\ X\ 10^8$	$4\ X\ 10^8$	
$kla(gL^{-1})$	0.112	0.112	0.112		0.134	0.134	0.134	
$O^*(gL^{-1})$	21.33	21.33	21.33		7.13	7.13	7.13	
$K_s(gL^{-1})$	79.31	79.31	79.31		1.42	1.42	1.42	
$K(gL^{-1})$	9.05	9.05	9.05		19.73	19.73	19.73	
$K_{co_2}(gL^{-1})$	1999.73	1999.73	1999.73		2116.91	2116.91	2116.91	
$a(gL^{-1})$	$1\ X\ 10^{-7}$	$1\ X\ 10^{-7}$	$1\ X\ 10^{-7}$		$9.8\ X\ 10^{-7}$	$9.8\ X\ 10^{-7}$	$9.8\ X\ 10^{-7}$	
$b(gL^{-1})$	0.165	0.165	0.165		0.002	0.002	0.002	

Table 3.9 Comparison of parameters for various passages in WT and Stabilized virus

Parameter (Wild type virus)	Passage1	Passage 5	Passage 15	Passage 27	Mean ± standard deviation
$\mu_{max}(hour^{-1})$	0.007	0.00773	0.00941	0.01195	0.0011 ±0.0022
Parameter (stabilized virus)	Passage1	Passage 5	Passage 15	Passage 27	Mean ± standard deviation
$\mu_{max}(hour^{-1})$	0.005	0.00562	0.00762	0.00943	0.0069±0.002

Chapter 4

Integrated model for baculovirus infection

4.1 Detailed modeling of the baculovirus infection process

The major objective of this chapter is to formulate a detailed model that integrates the dynamics of infected cell, uninfected cell, standard virus, defective interfering particles (DIPs) along with the DNA, RNA, protein expression and polyhedral production. In contrast to the unstructured model present in Chapter 3, here we present a structured model that includes the kinetics of DNA, RNA, proteins and DIPs formation. Specifically, the model variables are cell mass, oxygen, carbon di-oxide, DNA, RNA, proteins and baculovirus along with DIPs as product.

4.2 Basic assumptions on the infection process

The basic assumptions underlying the double stranded DNA virus are as follows, (1) the budded virus is internalized in the cells and it releases its viral gene (DNA) into that cell, (2) the viral DNA is transported from the cytosol to the surface of the nucleus known as trafficking, (3) the viral DNA is replicated and their copies are formed by using host machinery, (4) RNA is formed to produce DNA, (5) Early, late and very late proteins, FP25K, GP64 are expressed by RNAs, (6) Nucleocapsids are formed followed by ODV formation from neucleocapsids, (7) Polyhedrin proteins are transferred to the nucleus by FP25K protein resulting in the formation of Nuclear polyhedrosis virus (NPV), (8) The cells are then exploded and the viruses the virus comes out of the cell (9) Budded viruses are budded along wth the surface protein GP64, and (10) Occluded virused are formed through envelopment in ODVs and embedded in polyhedra. Figure 4.1 shows schematic of viral internalization, DNA replication, protein synthesis and virus step by step.

4.3 Model formulation

Dynamics of various components of the whole infection process shown above were represented by a set of coupled ordinary differential equations initial value problems (ODE-IVPs) with various kinetic parameters such as specific growth rate, death rate oxygen consumption rate, carbon-di-oxide formation rate, maintenance coefficient, product degradation rate, rate of DNA replication, rate of RNA formation, rate of protein formation etc. The simulation of the temporal profiles for the model variables were performed by solving ODE-IVPs by the fourth-order Runge-Kutta method.

4.4 Parameter Estimation

Extreme nonlinear dynamics of the model (ODE- IVPs) generally lead to difficulty in estimation of the rate constants present in the proposed models. A non-linear programming (NLP) problem was formulated with an error function as the objective function. The rate constants were set as the decision variables whose upper bounds and lower bounds were determined carefully by performing sensitivity analysis with respect to each parameter in the model. Sensitivity analysis was carried out by perturbing one parameter at a time till the limit within which the model converged, while other parameters were fixed at some base values. This ensured the convergence of ODE- IVP model for the candidates of parameters generated by the optimizer. The NLP formulation is given below.

Minimize Error

such that,

$$0 \leq K_p \leq K_p^{UL} \; \forall \; p = 1 \; to \; N$$

Where the error function is described in equation.

$$Error = \sum_{k=1}^{3}\sum_{j=1}^{2}\sqrt{\frac{1}{N_i}\left(\sum_{i}^{N_j}\left(y_i^j - \hat{y}_i^j\right)^2\right)}$$

$\hat{y}_i^j = f(K_p, y_0^j)$ Where, f is the proposed ODE – IVP model and y_0^j is initial conditions

y_i^j is the i^{th} experimental value (concentration) of j^{th} component.

$\hat{y}_i^j$ is the i^{th} simulated value (concentration) of j^{th} component.

$\forall \; i = 1 \; to \; N_j$

N_j is number of experimental observations

$\forall \; j = 1 \; to \; 2$ | 1: Cell Density, 2: Cell viability

N is number of parameters in the proposed model.

$\forall\ k = 1\ to\ 3$ is for three data set

4.4.1 Integrated model: description of the variables

The model has nine essentials components of the system, (1) the host cells (2) the standard (wild-type) virion, and (3) the defective interfering particles (DIPs), (4) oxygen (5) carbon di-oxide (6) substrate (7) DNAs (8) mRNA and (9) dead cells. The number of free DIP and standard virus particles were denoted as *VD* and *VS* respectively. The number of uninfected cell, concentration of oxygen, carbon dioxide and substrate was denoted by *CU,* O_2, CO_2 and S respectively. The initial state was specified by initial values of *CU, VD, VS*, O_2, CO_2, and S which is given in Table 4.1 for wild type viral infection and in Table 4.2 for stabilized viral infection.

The basic underlying assumptions are as follows, (1) infection of the cell occurs through random contact between virus particles and cells, (b) a is constant defined as the infection rate per cell-virion pair (3) CD cell is formed by infection of CU cell with DIPs virus, (4) CS cell is formed by infection of CU cell with standard (wild-type) virus (5) CB cells are formed by infection of DIP virus with CS cell or infection of standard virus with CD cell or co-infection of both the virus with uninfected cell, (5) cell growth is dependent on substrate, oxygen and carbon-di oxide concentrations, (6) CS cell will burst out at the end of the virus generation time, where the virus generation time is not fixed but we are assuming it is distributed around an appropriate mean generation time, (7) production of CB cells form CS cell occurs when CS cell is co-infected with DIP. But, this conversion of a CS cell to a CB cell occurs if the CS cell is co-infected with a DIP before a cutoff time because after that cutoff time, the CS cell will go through bursting which results in releasing of standard virus. Conversion from CD cell to CB cell occurs when CD is gets confected cell standard virus. The CB cell also bursts after virus generation time which results in production of DIPs and standard virus.

4.4.2 Model equations

$$\mu = \mu_{max} * \left(\frac{S}{K_s + S}\right)$$

$$\mathrm{\mu}_2 = \mathrm{\mu} * \left(\frac{O_2}{O_2 \ + K_{o_2}}\right) * \left(\frac{K}{\mathrm{CO_2 + K_{CO_2}}}\right)$$

$$\frac{d(CU)}{dt} = \mathrm{\mu}_2 * CU \ - a * CU * (VS + VD) - k_{d_1} * CU$$

$$\frac{d(CD)}{dt} = \mathrm{\mu}_2 * CD + a * CU * VD - a * VS * CD - k_D * CD$$

In equation 3, the first term on the right hand side of the equation represents cell growth, the second term represents loss of the cell through infection and the third term represents death of the cell. In equation 4, the first term represents cell growth, the second term represents gain through infection and, the third term represents cell loss due to coinfection and the last term represents the death of the cell.

The age of the cells is defined for the CS and CB cells and for this the number of subclasses of CS and CB were defined where τ is a constant representing the duration of a subclass. The maximum time that can elapse before a CS and CB cells releases the virus was set to be *T*. The number of subclasses CS and CB cells was divided as $n = \frac{T}{\tau}$ .Where, CS_i and CB_i denote the i^{th}subclass. The equations were written as

$$\frac{d(CS_1)}{dt} = \ a * VS * CU + k_{trans} * poly \ - \frac{CS_1}{\tau} - p_1 * CS_1 \quad [5]$$
$$- a * CS * VD * z_1 - k_D * CS_1$$

$$\frac{d(CS_i)}{dt} = \frac{CS_{i-1} - CS_i}{\tau} + k_{trans} * poly - CS_i * p_i \ - a * CS_i * VD * z_i - k_D * CS_i \quad [6]$$

$$1 < i < n$$

$$\frac{d(CS_n)}{dt} = \frac{CS_{(n-1)}}{\tau} + k_{trans} * poly - CS_n * p_n \ - a * CS_n * VD * z_n - k_D * CS_n \quad [7]$$

The first term in equation 5 represents the growth of the cell, the second term represents the gain of cells through infection, the third term is production of CS cells based on polyhedrin protein, the fourth term is the rate of transfer of cells from CS_1 to CS_2 through aging, the fifth term represents the rate of loss of cell undergoing virus release, and the last term is the rate of cell apoptosis. The variable p_i $(i = 1,2\ldots, n)$ is basically the death rates of cells in the i^{th} subclass undergoing virus release. The quantities z_i indicates the possibility of coinfection with DIP. We assume that the conversion is possible $(z_i = 1)$ before a certain time $i = i^*$. Until this time standard virus generation takes place after which conversion is impossible $(z_i = 0)$. The equation for CB cells are almost similar to those for the CS cells except the fact that the production of CB cells is possible in two ways. One way is to infect CD cells by standard virus and the other way is to infect CS cells by DIPs. These equations are:

$$\frac{d(CB_1)}{dt} = a * VD * \Sigma CS_i * z_j + a * CD * VS + k_{trans} * poly - \frac{CB_1}{\tau} - CB_1 * p_1 - k_D * CB_1 \qquad [8]$$

$$\frac{d(CB_i)}{dt} = k_{trans} * poly + \frac{\left(CB_{(i-1)} - CB_i\right)}{\tau} - CB_i * p_i - k_D * CB_i \qquad [9]$$

$$1 < i < n$$

$$\frac{d(CB_i)}{dt} = k_{trans} * poly + \frac{CB_{(n-1)}}{\tau} - CB_n * p_n - k_D * CB_n \qquad [10]$$

Where, ΣCS_i is the summation of all the subclasses of CS cell. Now, the rate of formation of DIPs and standard virus were obtained by summing the rates of virus release from the subclasses of CS and CB and subtracting the rates of virus loss through the infection process.

$$\frac{d(VS)}{dt} = b * \Sigma CS_i * p_i * gp64 - a * VS * (sum(CS) + sum(CB) + CD + CU) \qquad [11]$$

$$\frac{d(VD)}{dt} = b * \Sigma CB_i * p_i * gp64 - a * VD * (sum(CS) + sum(CB) + CD + CU) \qquad [12]$$

Generally, the production of DNA takes place from standard virus in between a specific time period (f_{DNA}). We assumed that DNA production does not take place before and after that time interval.

$$\frac{d(DNA)}{dt} = k_{DNA} * f_{DNA} * VS + f_{DNA} * k_{rep} * DNA \; - \; k_D * DNA \qquad [13]$$

$$\boxed{\begin{array}{ll} if\left(t < \delta_{DNA_{min}}\right) & f_{DNA} = 0 \\ if\left(\delta_{DNA_{min}} < t < \delta_{DNA_{max}}\right) & f_{DNA} = 1 - \dfrac{t - \delta_{DNA_{min}}}{\delta_{DNA_{max}} - \delta_{DNA_{min}}} \\ if\left(\delta_{DNA_{max}} < t\right) & f_{DNA} = 0 \end{array}}$$

The first term in equation 13 represents the DNA production from standard viruses, the second term represents the replication of DNA, and the last term represents the degradation of DNA.

Production of mRNA was modeled as follows (equation 14, 15 and 16), where RNA are of three types: (1) early RNA (2) late RNA and (3) very late RNA. As the production of early RNA starts before DNA replication, the production early RNA is mainly dependent on viruses. In contrast, the late and very late RNA is mainly dependent on amount of DNA present in the nucleus. The time varying functions are included to take care of the RNA production during specific phases.

$$\frac{d\left(RNA_{early}\right)}{dt} = f_{Early} * k_{RNA} * VS \; - k_2 * RNA_{early} \qquad [14]$$

$$\frac{d(RNA_{late})}{dt} = f_{late} * k_{RNA} * DNA \; - k_2 * RNA_{late} \qquad [15]$$

$$\frac{d\left(RNA_{verylate}\right)}{dt} = f_{verylate} * k_{RNA} * DNA \; - k_2 * RNA_{verylate} \qquad [16]$$

$$
\begin{array}{ll}
if\left(t < \delta_{RNAe_{min}}\right) & f_{early} = 0 \\
if\left(\delta_{RNAe_{min}} < t < \delta_{RNAe_{max}}\right) & f_{early} = 1 - \frac{t \ -\delta_{RNAe_{min}}}{\delta_{RNAe_{max}} - \delta_{RNAe_{min}}} \\
if\left(\delta_{RNAe_{max}} < t\right) & f_{early} = 0
\end{array}
$$

$$
\begin{array}{ll}
if\left(t < \delta_{RNAl_{min}}\right) & f_{late} = 0 \\
if\left(\delta_{RNAl_{min}} < t < \delta_{RNAl_{max}}\right) & f_{late} = 1 \text{ - } \frac{t - \delta_{RNAl_{min}}}{\delta_{RNAl_{max}} - \delta_{RNAl_{min}}} \\
if\left(15 < \delta_{RNAl_{max}}\right) & f_{late} = 0
\end{array}
$$

$$
\begin{array}{ll}
if\left(t < \delta_{RNAvl_{min}}\right) & f_{verylate} = 0 \\
if\left(\delta_{RNAvl_{min}} < t < \delta_{RNAvl_{max}}\right) & f_{verylate} = 1 \ - \frac{t - \delta_{RNAvl_{min}}}{\delta_{RNAvl_{max}} - \delta_{RNAvl_{min}}} \\
if\left(\delta_{RNAvl_{max}} < t\right) & f_{verylate} = 0
\end{array}
$$

In equation 14, the first term represents the production of RNA early amount, and the second term represents the degradation of RNAs. Similarly, we formulated the equations for the production of late and very late RNA (equation 15 and 16).

$$\frac{d(fp25k)}{dt} = k_{pe} * \left(\frac{RNA}{RNA + KRNA}\right) * f_{late} * sum(CB + CD + CS) - k_2 * fp25k \qquad [17]$$

[18]

$$\frac{d(gp64)}{dt} = k_{pe} * \left(\frac{RNA}{RNA + KRNA}\right) * f_{early} * sum(CB + CD + CS) * \left(\frac{1}{fp25k + k_{fp}}\right)$$

$$-k_2 * gp64$$

[19]

$$\frac{d(Polyhedrin)}{dt} = k_{pe} * \left(\frac{RNA}{RNA + KRNA}\right) * f_{verylate} * sum(CB + CD + CS) * \left(\frac{fp25k}{fp25k + k_{ODV}}\right) - k_2 * polyhedrin$$

In equation 17, the first term represents the production of GP64 protein amount, and the second term represents the degradation of the protein. Similarly, we formulated the equations for late and very late proteins FP25K and polyhedrin (equation 18 and 19). In these case, it is assumed that amount of GP64 protein in inversely proportional to FP25k protein and amount of polyhedrin is directly proportional to FP25k protein.

$$\frac{d(O_2)}{dt} = kla * ({O_2}^* - O_2) - \frac{(CU + \Sigma CS_i + \Sigma CB_i + CD)}{Y_o} \qquad [20]$$

$$\frac{d(Carbon\ di - oxide)}{dt} = \frac{(CU + \Sigma CS_i + \Sigma CB_i + CD)}{Y_c} \qquad [21]$$

$$\frac{d(Dead\ cell)}{dt} = k_{d_1} * CU + k_d * (\Sigma CS_i + \Sigma CB_i + CD) \qquad [22]$$

$$\frac{d(Substrate)}{dt} = -\mu * \left(\frac{CU + \Sigma CS_i + \Sigma CB_i + CD}{Y_S}\right) \qquad [23]$$

In equation 20, the first term represents mass transfer rate of oxygen into insect cell and the second term represents the oxygen consumption rate. In equation 22, the first term represents the death of uninfected cell and the second term represents the death of infected cells.

4.4.3 Virus release times

The distribution of virus release time for the CS and CB cells is governed by the variable p. We assumed that the p value will be zero before 18 h p.i as no virus release takes place before this time, and after this time, the virus release will follow an exponential function.

$$\boxed{\begin{array}{ll} p \ = 0 & if(t < 18) \\ p \ = \alpha . e^{\beta(i-i')} & Otherwise \end{array}}$$

Where, i' is the minimum number of subclasses through which a CS or CB cell must pass before it first exposed to the risk of cell lysis. Rate of release of virus are controlled by the values of α, β and i'. Also, we are assuming that co-infection does not take place before a certain time.

$$\boxed{\begin{array}{ll} z_i \ = 1 & if(t < 10) \\ z_i \ = 0 & Otherwise \end{array}}$$

4.5 Sensitivity analysis

Sensitivity analysis is an important tool that is used to identify the parameters that affect the response variable. This technique is useful for finding out the parameter range in which the parameter estimation is to be conducted. From this analysis, we obtained the desired parameter range for choosing the initial guess values. We performed the sensitivity analysis with ± 100% change in each of the parameter and observed the effect on two variables, cell density and percentage of viable cells.

4.5 Results

First we performed the parametric sensitivity analysis for the complex model and identified the parameter range to be used for matching the simulated results and experimental data. Figure 4.2 shows the 3-D representation of the parametric sensitivity analysis for the proposed integrated model. Where X axis shows all the parameters, Y axis shows all the model variables and Z axis shows the change in model variables with respect to parameters. The result shows that the cell density is sensitive to specific growth rate, transport of protein rate constant and rate of protein synthesis. On the other hand, the percent of polyhedra per cell is sensitive to specific growth rate, α, β. Also, cell viability is sensitive to specific growth rate and DNA transcription rate.

Figure 4.3 and Figure 4.4 shows the comparison of simulated results obtained from the integrated model and experimental data. The result shows that the complex model is able to predict the trends for experiment 2 and 3 better than the trend presents in experiment 1. Table 4.3 show the parameters estimated for the complex model for WT and stabilized AcMNPV using NLP. In contrast, the simple model gives a better result as the case of very low growth rate and high death rate present in experiment 1 is not captured by the complex model. Since the RMSE for the complex model is lower than the simple model (Table 4.5), it can be concluded that the detailed model with a higher number of parameter can capture the non-linearity better than the simple model. Another specific advantage of the complex model is that the model is able to predict the percentage of polyhedra along with the cell density and cell viability. Table 4.3 shows that many of the parameters show significant changes for the genetically modified stabilized virus, whereas some of the parameters remain similar for the two viruses. Table 4.4 shows the comparison of average specific growth rate and death rate for the cells. Additionally, we present a detailed comparison of simple model and complex model with respect to computation time, RMSE and number of parameters and variables.

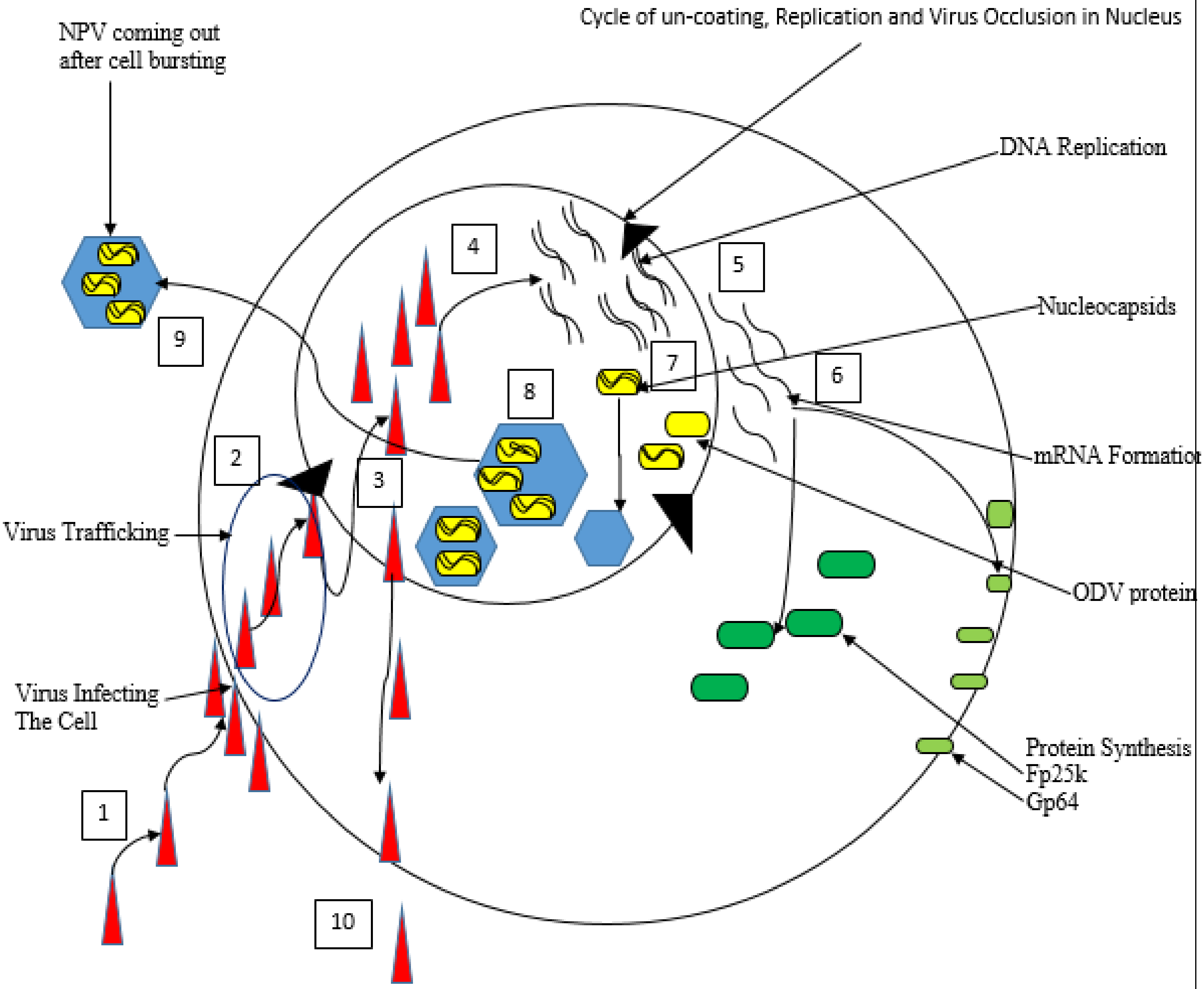

Cycle of un-coating, Replication and Virus Occlusion in Nucleus
NPV coming out
after cell bursting
DNA Replication
Nucleocapsids
mRNA Formatio
ODV protein
Protein Synthesis
Fp25k
Gp64
Virus Trafficking
Virus Infecting
The Cell
1
2
3
4
5
6
7
8
9
10

Figure 4.1: Schematic of viral internalization, DNA replication, protein synthesis and virus packaging (budded virus, occlusion derived virus and polyhedra)

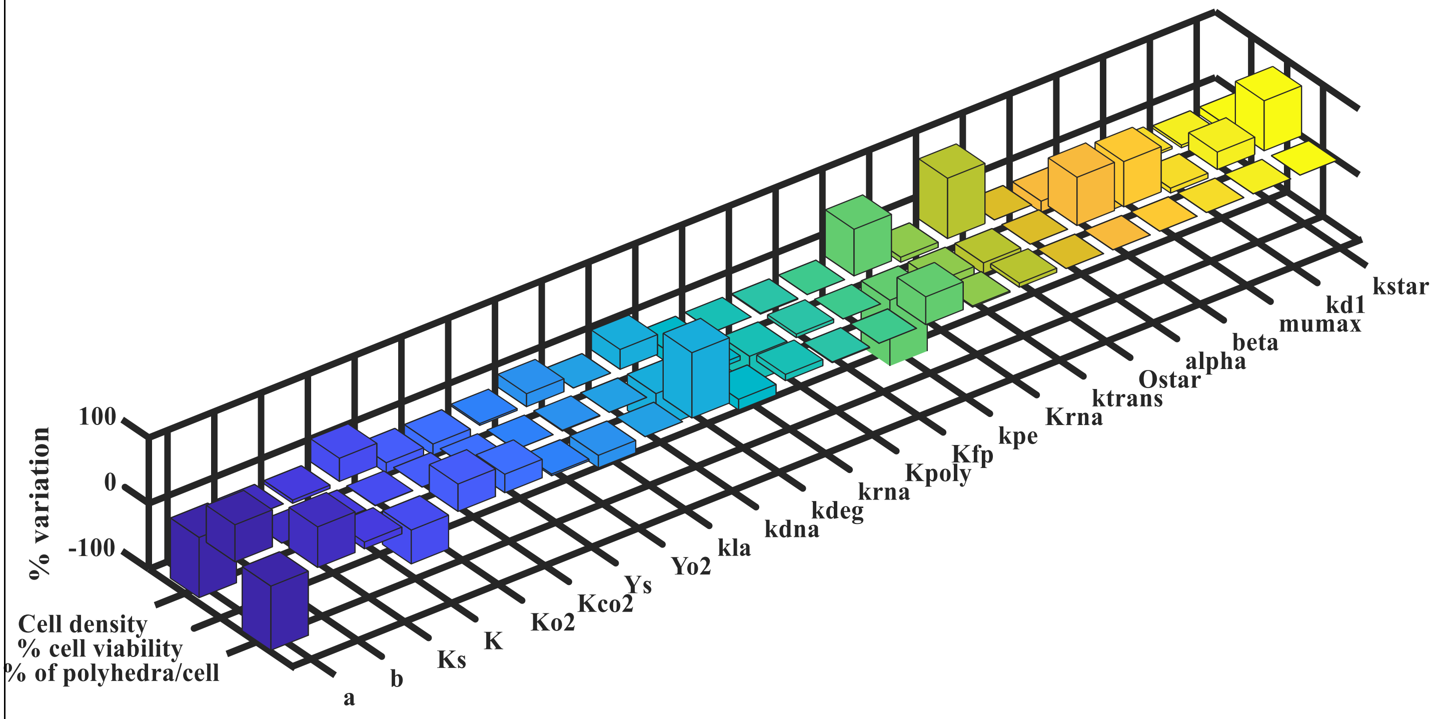
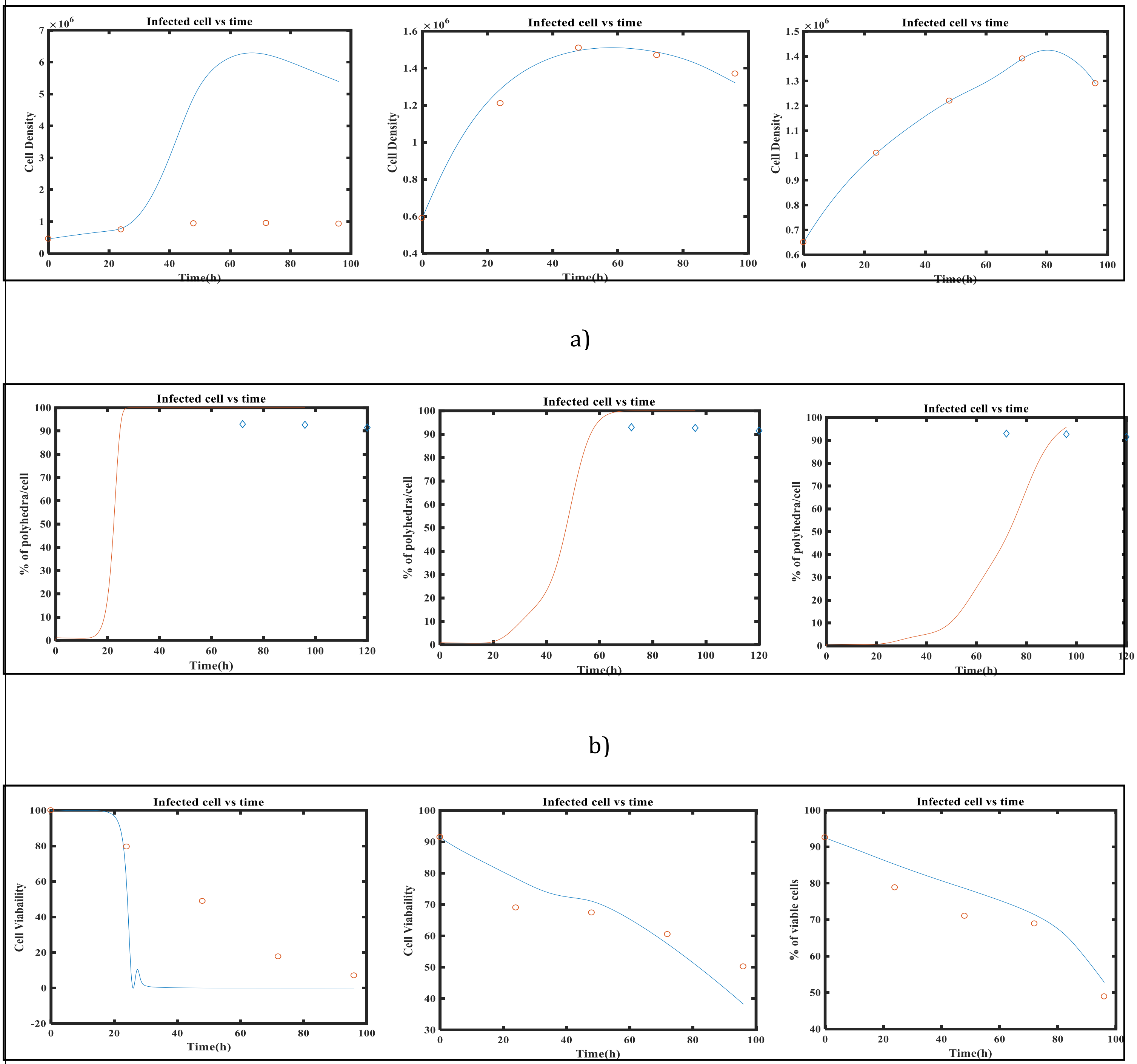


Figure 4.2 Sensitivity analysis of complex model. X axis shows all the parameters in the model and Y axis shows the variables and Z axis shows the percentage variation of the variables with parameters

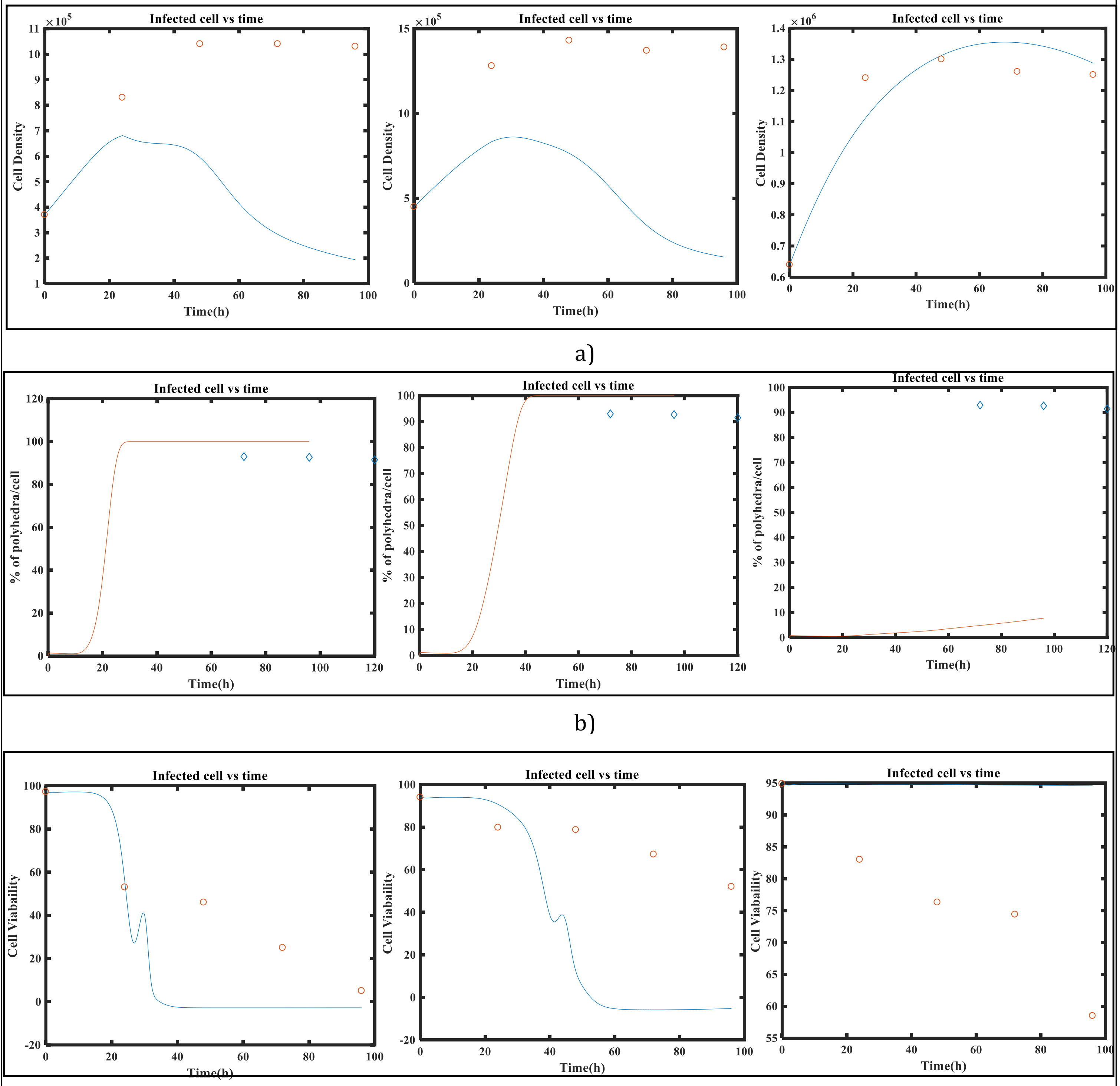


a)

b)

c)

Figure 4.3 Simulated and experimental results for the proposed model for infected cell growth experiment for wild type virus and triplicate data set a) infected cell density with time and b) percentage of polyhedra per cell with respect to time and c) percentage of viable cells with time



a)

b)

c)

Figure 4.4 Simulated and experimental results for the proposed model for complex infected cell growth for stabilized virus and triplicate data set a) cell density with time and b) percentage of polyhedra/cell with respect to time and c) % of viable cells with time

Table 4.1 Initial Conditions for infected cell for wild type viral infection for complex model

Variable (in case of wild type virus)	Uninfected cell	Substrate	Oxygen	Carbon dioxide	Dead cell	Virus	Infected cell	DNA	RNA	proteins
Initial Condition (data 1)	460000	1000	1000	0	0	5150	0	0	0	0
Initial Condition (data 1)	590000	1000	1000	0	0	5150	0	0	0	0
Initial Condition(data 1)	650000	1000	1000	0	0	5150	0	0	0	0

Table 4.2 Initial Conditions for infected cell for stabilized viral infection for complex model

Variable (in case of stabilized virus)	Uninfected cell	Substrate	Oxygen	Carbon dioxide	Dead cell	Virus	Infected cell	DNA	RNA	proteins
Initial Condition (data 1)	370000	1000	1000	0	0	5150	0	0	0	0
Initial Condition (data 1)	450000	1000	1000	0	0	5150	0	0	0	0
Initial Condition(data 1)	640000	1000	1000	0	0	5150	0	0	0	0

Table 4.3 Estimated parameters for complex infection model

Parameters	Parameter name	Wild type Virus Exp1	Wild type Virus Exp 2	Wild type Virus Exp 3	Stabilized virus Exp 1	Stabilized virus Exp 2	Stabilized virus Exp 3
$a(gL^{-1})$	Infection rate constant	1.94 $X\ 10^{-5}$	1.94 $X\ 10^{-5}$	1.94 $X\ 10^{-5}$	1.82 $X\ 10^{-5}$	1.82 $X\ 10^{-5}$	1.82 $X\ 10^{-5}$
$b(gL^{-1})$	Virus production rate	21.57	21.57	21.57	23.14	23.14	23.14
$K_s(gL^{-1})$	Half-Velocity constant	50.00	50.00	50.00	106.42	106.42	106.42
$K(gL^{-1})$	Saturation constant	0.628	0.628	0.628	0.971	0.971	0.971
$K_{o_2}(gL^{-1})$	Half-Velocity constant	430.379	430.379	430.379	493.601	493.601	493.601
$K_{co_2}(gL^{-1})$	Half-Velocity constant	14.913	14.913	14.913	14.702	14.702	14.702
$Y_s(g/g)$	Yield of cell / substrate	1.20 $X\ 10^5$	1.20 $X\ 10^5$	1.20 $X\ 10^5$	1.12 $X\ 10^5$	1.12 $X\ 10^5$	1.12 $X\ 10^5$
$Y_{o_2}(g/g)$	Yield of cell / oxygen	$8\ X\ 10^5$	$8\ X\ 10^5$	$8\ X\ 10^5$	$8\ X\ 10^5$	$8\ X\ 10^5$	$8\ X\ 10^5$
$kla(gL^{-1})$	Mass transfer rate constant	$1\ X\ 10^{-5}$	$1\ X\ 10^{-5}$	$1\ X\ 10^{-5}$	5.94 $X\ 10^{-5}$	5.94 $X\ 10^{-5}$	5.94 $X\ 10^{-5}$
$k_{DNA}(gL^{-1})$	DNA replication constant	1.95	1.95	1.95	3.80	3.80	3.80
$k_{deg}(gL^{-1})$	Degradation rate constant	0.00728	0.00728	0.00728	0.0132	0.0132	0.0132
$O^*(gL^{-1})$	Equilibrium oxygen concentration	2.699	2.699	2.699	1.073	1.073	1.073
$k_{RNA}(gL^{-1})$	Transcription rate constant	5.766	5.766	5.766	3.505	3.505	3.505
$K_{poly}(gL^{-1})$	Equilibrium constant for fp25k	0.500	0.500	0.500	1.179	1.179	1.179
$K_{fp}(gL^{-1})$	Equilibrium constant for polyhedrin	0.008	0.008	0.008	0.008	0.008	0.008
$k_{pe}(gL^{-1})$	Protein synthesis rate constant	0.088	0.088	0.088	0.0768	0.0768	0.0768
$K_{RNA}(gL^{-1})$	half-saturation constant for protein	16.05	16.05	16.05	27.37	27.37	27.37
$k_{trans}(gL^{-1})$	DNA transfer constant	0.00898	0.00898	0.00898	0.0006	0.0006	0.0006

α(----)	Virus release rate constant	0.001	0.001	0.001	0.001	0.001	0.001
$\beta(hour^{-1})$	Bursting rate constant	0.501	0.501	0.501	0.307	0.307	0.307
$\mu_{max}(hour^{-1})$	Maximum specific growth rate	0.87	1.084	1.088	0.93	1.09	1.013
$k_{d_1}(hour^{-1})$	Intrinsic death rate	0.15	0.001	0.0014	0.03	0.0072	0.13
$k^*(hour^{-1})$	Death rate due to infection	0.0049	0.0049	0.0049	0.001	0.001	0.001

Table 4.4 parameter values for three different simulations with mean and standard deviation.

Parameter (Wild type virus)	Exp 1	Exp 2	Exp 3	Mean and standard deviation
$\mu_{max}(hour^{-1})$	0.87	1.084	1.088	1.014 $\pm$0.124
$k_{d_1}(hour^{-1})$	0.15	0.001	0.0014	0.0508$\pm$0.049
Parameter (stabilized virus)	Exp 1	Exp 2	Exp 3	Mean and standard deviation
$\mu_{max}(hour^{-1})$	0.93	1.09	1.013	1.011$\pm$0.0462
$k_{d_1}(hour^{-1})$	0.03	0.0072	0.13	0.0557$\pm$0.03777

Table 4.5 Difference between simple model and complex model

	RMSE	Number of parameters	Number of variables	Computational time(seconds)
Simple model	33400	14	7	146
Complex model	12700	23	75	1567

Chapter 5

Conclusions and future work

The present work provides a framework for finding the most suitable model corresponding to wild type stabilized baculovirus from a list of models. Moreover, it gives a framework for selecting the growth model for uninfected cells as well as infected cells. Also, we conclude that model having Monod growth model can capture the growth of uninfected cell better compared to other growth models. The results show that while the infected cell growth follows Monod's equation in case of wild type virus whereas that for the stabilized virus follows Contois equation. Also, we conclude that the growth rate constant and death rate constant may change from experiment to experiment. Specifically, it is challenging to model the death dynamics since the mechanism for apoptosis after viral infection is less investigated. We propose a future project focusing on modeling the cell death kinetics where time varying features of cell viability can be captured. Specifically, dependency of death rate constant on virus present and carbon di oxide concentration may be included to improve on the match between experiment and simulations. The parameter estimation result shows that the infection process is statistical and the death rate and growth rate may follow a distribution during an infection process. Also, death rate constant can be dependent on the number of viral DNA present in the nucleus of the cell. Apart from the infection, we also show a proof of concept for setting up a simulation platform for passaging of viruses in shaker flasks. The result shows that that a change cell growth and cell death rate in each passage is able to capture the trends during passaging. Additionally, the analysis on complex model shows that the DNA profile and other mechanisms can capture the cell density profile better than the simple model. But it still remains challenging to obtain a good fit for three datasets and the cell viability. One of the reason behind this could be the incomplete model for cell death mechanisms based on apoptotic genes. In order to obtain the global optimum, we can use genetic algorithm. However, the current optimization technique can be used for getting the convergence in shorter duration.

Based on the current computational framework, further projects can be formulated as follows: (1) identification of the specific mechanism causing nonlinear viability profile and testing the time varying models for cell death (2) Study of death rate constant as a function of oxygen, substrate, carbon dioxide and viral DNA concentration to obtain a better match between simulation and experiment (3) Implementation of the complex model features to capture the cell death (4) Setting up a simulation platform for predicting the cell growth and cell death profile for passaging of viruses in shaker flask and estimation of parameters based on an objective function considering triplicate dataset. In order to improve on parameter estimation, genetic algorithm or other nature inspired optimization methods can be used.

Nomenclature

WT	Wild type
AcMNPV	Autographa californica nuclear polyhedrosis virus
ODE	Ordinary differential equation
DIP	Defective interfering particles
SQP	Sequential quadratic programming
IVP	Initial value problem
RMSE	Root mean square error
NLP	Non-linear programming
Sf	Spodoptera frugiperda

Passage number Passage number

References

[1] Hofmann, C.; Sandig, V.; Jennings, G.; Rudolph, M.; Schlag, P.; Strauss, M. (1995). "Efficient Gene Transfer into Human Hepatocytes by Baculovirus Vectors". *Proceedings of the National Academy of Sciences*. **92** (22): 10099–10103. doi:10.1073/pnas.92.22.10099. PMC 40743

[2] Demain, Arnold L., and Preeti Vaishnav. "Production of recombinant proteins by microbes and higher organisms." *Biotechnology advances* 27.3 (2009): 297-306.

[3] Singh, Ranjana, et al. "Identification of unstructured model for subtilin production through Bacillus subtilis using hybrid genetic algorithm." *Process Biochemistry* 60 (2017): 1-12.

[4] Roldao, António, et al. "Modeling rotavirus-like particles production in a baculovirus expression vector system: Infection kinetics, baculovirus DNA replication, mRNA synthesis and protein production." *Journal of biotechnology*128.4 (2007): 875-894.

[5] Dee, Kennie U., and Michael L. Shuler. "A mathematical model of the trafficking of acid-dependent enveloped viruses: Application to the binding, uptake, and nuclear accumulation of baculovirus." *Biotechnology and bioengineering* 54.5 (1997): 468-490.

[6] Kirkwood, T. B., and C. R. Bangham. "Cycles, chaos, and evolution in virus cultures: a model of defective interfering particles." *Proceedings of the National Academy of Sciences*91.18 (1994): 8685-8689.

[7] Ghovvati, Mahsa, et al. "Comparison across growth kinetic models of alkaline protease production in batch and fed-batch fermentation using hybrid genetic algorithm and particle swarm optimization." *Biotechnology & Biotechnological Equipment* 29.6 (2015): 1216-1225.

[8] Kumar, Arun, and Michael L. Shuler. "Model of a split-flow airlift bioreactor for attachment-dependent, baculovirus-infected insect cells." *Biotechnology progress* 11.4 (1995): 412-419.

[9] Vajrala, Sucheta G., and David W. Murhammer. "Effect of CO 2 on uninfected S f-9 cell growth and metabolism." *Biotechnology progress* 32.2 (2016): 465-469.

[10] Kim, Sun U., Daniel C. Kim, and Prasad Dhurjati. "Mathematical modeling for mixed culture growth of two bacterial populations with opposite substrate preferences." *Biotechnology and bioengineering* 31.2 (1988): 144-159.

Appendix

Experimental data

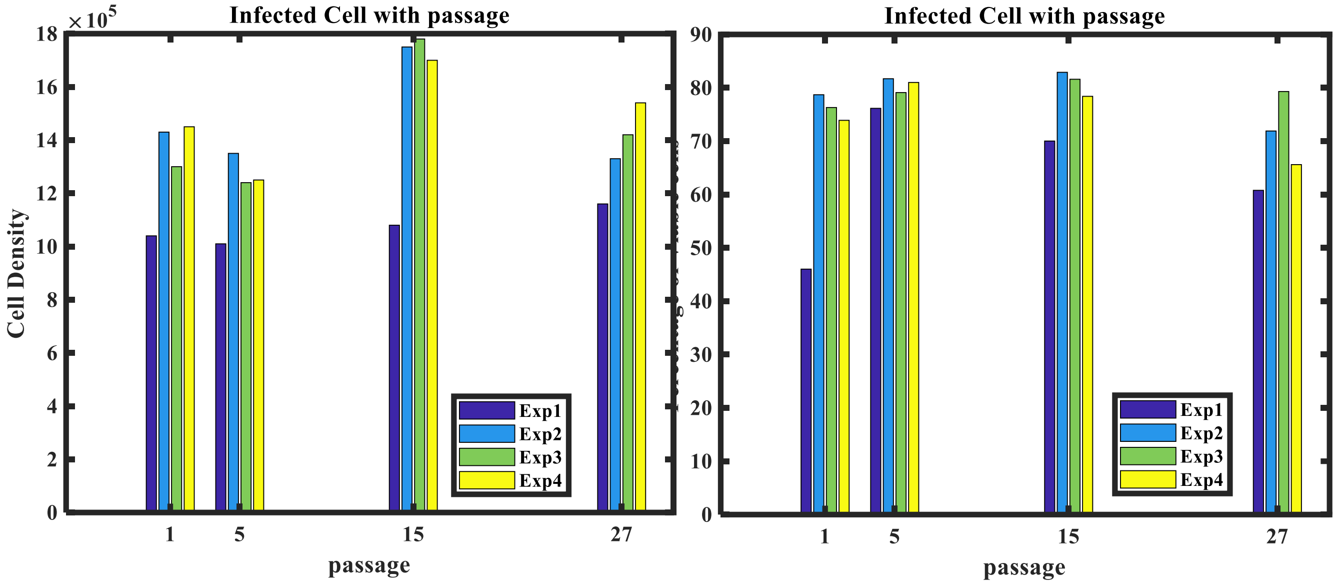


Figure S1. Infected cell with passage in case of stabilized viral infection for four different experiment a) Cell density with respect to passage b) percentage of viable cells with respect to

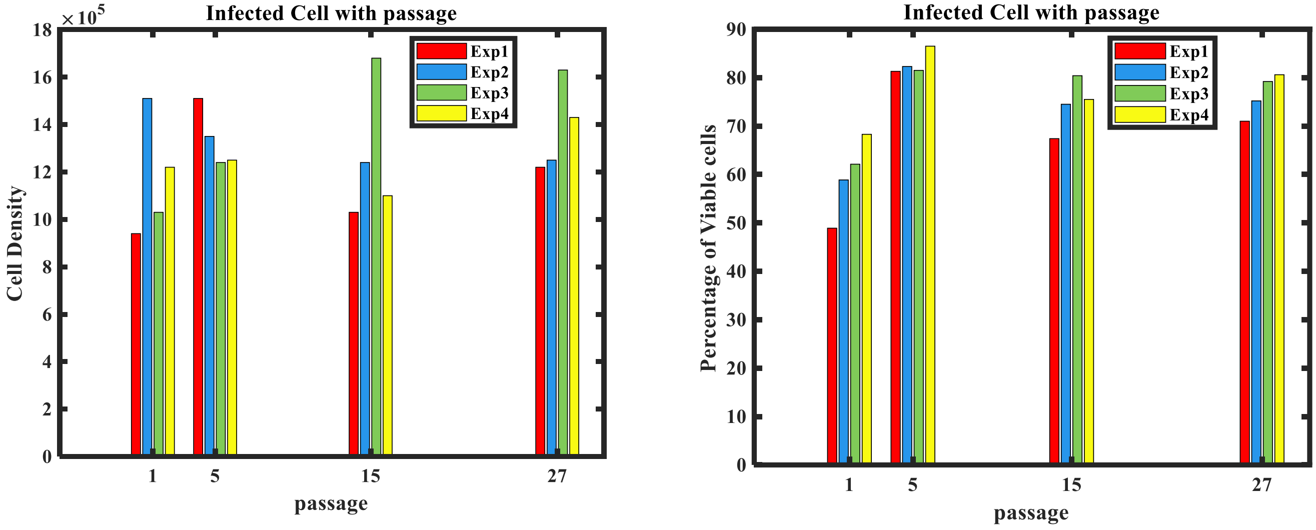


Figure S2. Infected cell with passage for in case of wild type viral infection four different experiment a) Cell density with respect to passage b) percentage of viable cells with respect to passage

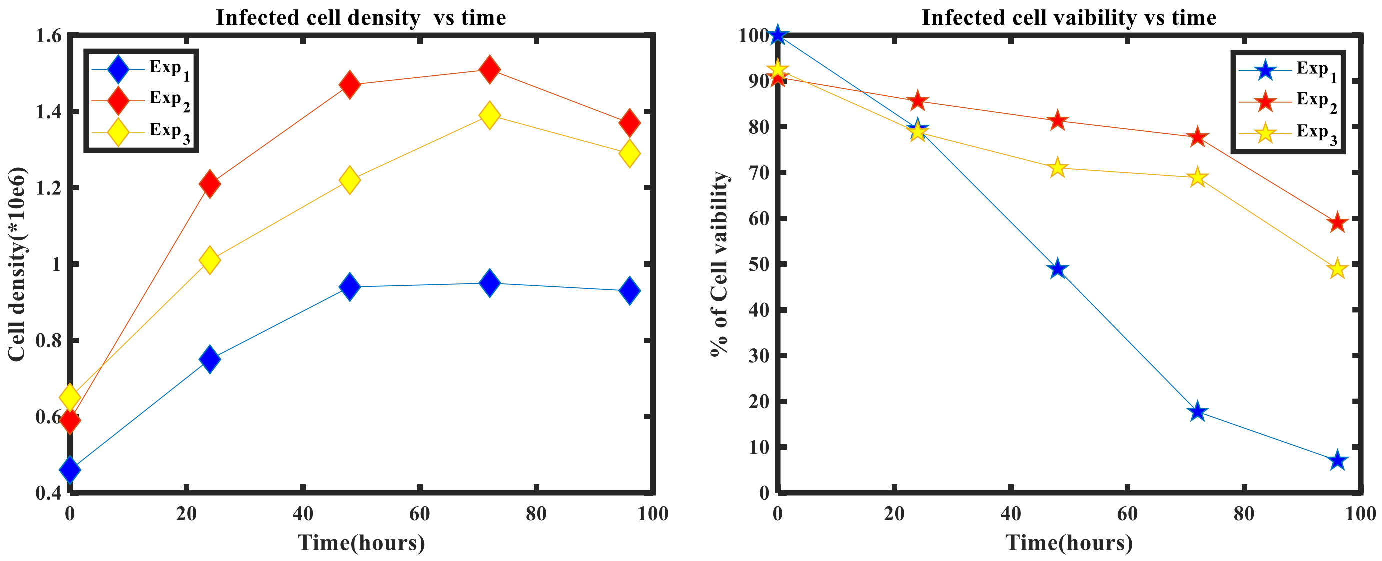


Figure S3. Infected cell experimental results for three sets of data

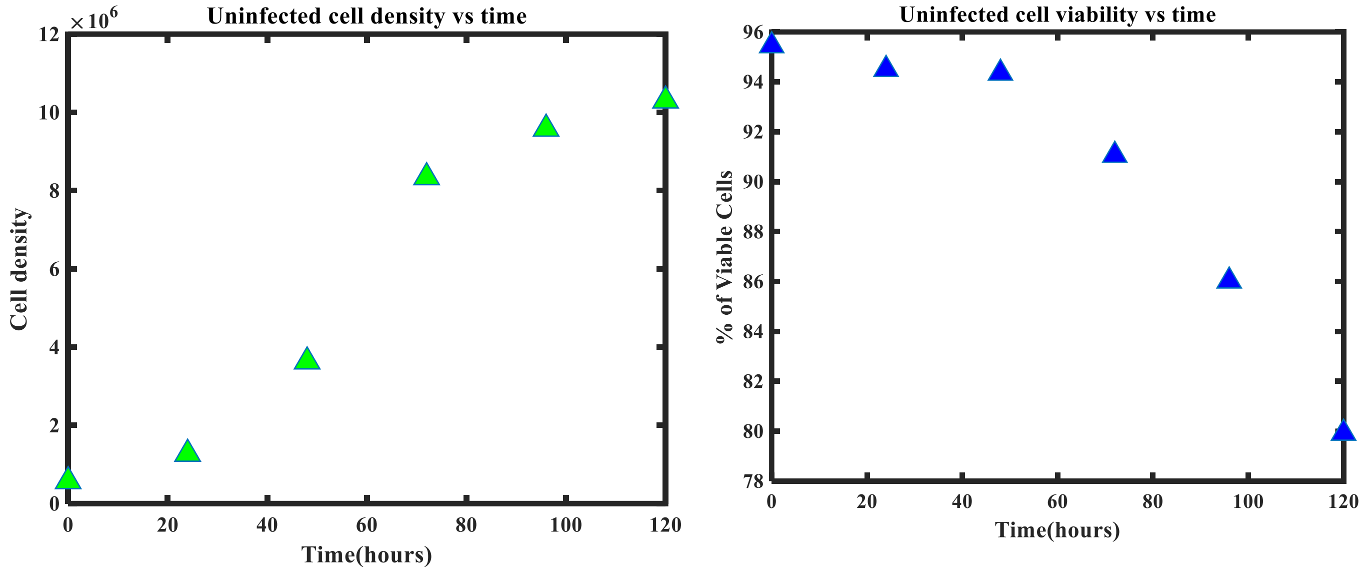


Figure S4. Uninfected cell experimental results for three sets of data

Uninfected cell growth Main file

```
clc;
close all;
clear all;
global t
param0 = [0.26 0.107 400000 100 0.0001 10000000 10.27 200 50 1.1 0.09
10000000 1000 1000];
lb=zeros(1,10);
tspan = [0 120];
p = fmincon(@obj_fn_uninfected,param0,[],[],[],[],[0.001 0.0001 1000 0.01
0.000001 1000 0.01 0.01 0.01 0.1 0.001 100000 10 10],[10 10 100000000 1000000
10 5000000000 2000 1000000 500000 2000 10 1000000000 10000 10000]);

y0(1) = 570000;
y0(2) = 1000;
y0(3) = 1000;
y0(4) = 0;
y0(5) = 0.0455*y0(1);
[t,y] = ode45(@(t,y)ODEset_Uninfected(t,y,p),tspan,y0);
vd(:,1) = 95.45 - (y(:,5)/sum(y(:,1)))*100 ;
figure(1)
uc = [570000 1370000 3550000 8140000 9580000 10300000];
ti = [0 24 48 72 96 120];
figure(1)
plot(t,y(:,1))
hold on;
plot(ti,uc,'d','Markersize',10,'MarkerFaceColor','g')
axis([0,120,0,12300000])
title('Uninfected Cell Density')
ylabel('Cell Density')
xlabel('Time in hour')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)

figure(2)
plot(t,y(:,2),t,y(:,3),t,y(:,4))
legend('Substrate','O2','CO2')
% axis([0,120,0,1200])
% yyaxis right
figure(3)
vde = [95.55 94.6 94.35 92.06 87.02 77.3];
plot(ti,vde,'^','Markersize',10,'MarkerFaceColor','g')
axis([0,120,60,100])
hold on;plot(t,vd(:,1))
title('Uninfected Cell Viability')
ylabel('% of viable cells')
xlabel('Time in hour')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
% figure(3)
% plot(ti,y(:,2),ti,y(:,3),ti,y(:,4));
% legend('Substrate','Oxygen','Carbon di-oxide')
% set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
% ylabel('Concentration')
% xlabel('Time in hour')
```

Uninfected cell growth objective function

```
function [rmse] = obj_fn_uninfected(param)
tspan = [0 24 48 72 96 120];
y0(1) = 570000;
y0(2) = 1000;
y0(3) = 1000;
y0(4) = 0;
y0(5) = 0.0455*y0(1);
[t,y] = ode45(@(t,y)ODEset_Uninfected(t,y,param),tspan,y0);
vd(:,1) = 95.45 - (y(:,5)/sum(y(:,1)))*100 ;
% figure(1)
uc = [570000 1370000 3550000 8140000 9580000 10300000];
% ti = [0 24 48 72 96 120];
% plot(t,y(:,1))
% hold on;
% plot(ti,uc,'d','Markersize',10,'MarkerFaceColor','g')
% axis([0,120,0,12300000])
% % figure(2)
% % plot(t,y(:,2),t,y(:,3),t,y(:,4))
% % legend('Substrate','O2','CO2')
% % axis([0,120,0,1200])
% % figure(3)
vde = [95.55 94.6 94.35 92.06 87.02 77.3];
% % plot(ti,vde,'^','Markersize',10,'MarkerFaceColor','g')
% % axis([0,120,0,120])
% % hold on;
% % plot(t,vd(:,1))
% error = 0;
sum1 = 0;
sum2 = 0;
for i=1:6
    sum1= (sum1 + (uc(i)-y(i,1))^2);
    Rsq1 = 1 - sum((uc(i)-y(i,1)).^2)/sum((uc(i)-mean(uc)).^2);
end
for i=1:6
    sum2= (sum2 + (vde(i)-vd(i,1))^2);
end
% [(uc' - y(:,1)); (vde' - vd)]
[rmse] = sqrt((sum1+sum2)/6)
% tot_err = rmse/(10000000)*100;
% error = [error tot_err];
disp(Rsq1);
% figure(2)
% bar(error);
% ylim([0 ,100]);
% ylabel('% RMSE')
% xlabel('Interations')
% set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
end
```

1) Uninfected cell growth ODE set

```
function dmdt = ODEset_Uninfected(t,y,param)
mumax = param(1);%0.155;
kd1 = param(11);
kd2 = param(2);%0.022;
% kd1 = 0.6;
Ys = param(3);%400000;
Ko2 = param(4);%100;
kla = param(5);%0.008;
Yc = param(6);%5000000;
Ostar = param(7);%10.27;
Kco2 = param(8);%200;
Ks = param(9);%50;
frac = param(10);%0.6;
Yo = param(12);
mu1 = mumax*(y(2)/(y(2) + Ks));
O2 = (y(3)/(y(3) + Ko2));
CO2 = (frac/(exp(y(4)/Kco2)+ 1));
mu = mu1*O2*CO2 ;
% q_d = kd2*y(1)/log(y(2));
if t<73
    kd = kd1;
else
    kd = kd2;
end
dydt(1) =  mu*y(1) - kd*y(1);
dydt(2) = -mu1*(y(1)/Ys);
dydt(3) =  - y(1)/Yo  ;
dydt(4) = y(1)/Yc ;
dydt(5) = kd*y(1);
dmdt = dydt';

end
```

Infected cell growth Main file

```
% function Infection_paper_model()
clc;
close all;
clear all;
global t
p0=[5150 1000 1000 1000000 1000000000 10 400000000 0.144 11 12 11 2000
0.000001 0.02 0.01 0.01 0.01 0.02 0.02 0.02 0.002 0.002 0.002 50];
% p0 = [0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
0.01];
% p0 = [0.1099 0.000000058286 0.0568 1.3535 0.1413 0.8474 0.0548 1.0902
0.00000003054 0.2222 0.00026388 0.000054323 0.000000003054 0.4734];
lb=[515 10 10 100000 100000000 1 40000000 0.0144 1 1 1 200 0.0000001 0.002
0.0002 0.0002 0.0002 0.001 0.001 0.05 0.000266 0.000266 0.000266 10];
ub = [515000 10000 10000 10000000 1000000000 100 4000000000 0.544 110 120 110
20000 0.00001 0.2 0.2 0.2 0.2 0.1 0.1 1 0.2 0.2 0.2 100];
p = fmincon(@my_obj_infected,p0,[],[],[],[],lb,ub);
tspan = [0 96];
UC = [460000 590000 650000];
y0(2) = p(1);
y0(3) = p(2);
y0(4) = 0;
y0(5) = 0;
y0(6) = p(3);
y0(7) = 0;
% Ys = p(4);%1000000;
% Yo2 = p(5);%1000000000;
% Ko2 = p(6);%10;
% Yco2 = p(7);%400000000;
% kla = p(8);%0.144;
% Ostar = p(9);%11;
% Ks = p(10);%12;
% K = p(11);%11;
% Kco2 = p(12);%2000;% frac = 0.3;
% a1 = p(13);%0.000001;
% ka = p(14);%0.02;
%
% mumax = p(15:17);
% kd1 = p(18:20);
% kstar = p(21:23);

cell_den = zeros(4,5);
cell_via = [];

for j=1:3
    y0(1) = UC(j);
[t,y] = ode45(@(t,y)ODEset_infected(t,y,[p(1,1:14) p(15+(j-1))
p(15+size(4,1)+(j-1)) p(15+2*size(4,1)+(j-1))]),tspan,y0);
ti = [0 24 48 72 96];
% uc1 =  [460000 650000 940000 950000 970000];
% uc2 =  [590000 1210000 1510000 1470000 1370000];
% uc3 =  [440000 1040000 1030000 1070000 1270000];
% uc4 =  [650000 1010000 1220000 1390000 1290000];
% cell_den(:,j) = (y(:,1)+y(:,7));
```

```
uc =  [460000 750000 940000 950000 930000;
        590000 1210000 1470000 1510000 1370000;
        650000 1010000 1220000 1390000 1290000];

vde = [100 79.55 48.9 17.71 7.03;
        90.8 85.6 81.3 77.7 59;
        92.5 78.8 71 68.9 48.9];
VCI = [100 90.8 92.5];
dead = (y(:,5)./(y(:,1)+y(:,7)))*100 ;
vd = VCI(j) - dead;
% cell_via(:,j) = vd(:,1);

%stabalizes virus
% uc5 = [515000 1165000 1305000 1315000 1242500];
% uc1 = [370000 830000 1040000 1040000 1030000];
% uc2 = [450000 1280000 1430000 1370000 1390000];
% uc3 = [640000 1240000 1300000 1260000 1250000];
% uc4 = [600000 1310000 1450000 1590000 1300000];
% % yyaxis left
% error_1 = [uc1(1) uc2(1) uc3(1) uc4(1)];
% error_2 = [uc1(2) uc2(2) uc3(2) uc4(2)];
% error_3 = [uc1(3) uc2(3) uc3(3) uc4(3)];
% error_4 = [uc1(4) uc2(4) uc3(4) uc4(4)];
% error_5 = [uc1(5) uc2(5) uc3(5) uc4(5)];
%
% sdd = [std(error_1) std(error_2) std(error_3) std(error_4) std(error_5)];
figure(j)
plot(t,(y(:,1)+y(:,7)))
hold on;
plot(ti,uc(j,:),'o')
% hold on;
% plot(ti,uc2,'d','Markersize',10,'MarkerFaceColor','r')
% hold on;
% plot(ti,uc3,'d','Markersize',10,'MarkerFaceColor','y')
% hold on;
% plot(ti,uc4,'d','Markersize',10,'MarkerFaceColor','g')
% axis([0,100,400000,1430000])
ylabel('Cell Density')
xlabel('Time(h)')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
title('Infected cell vs time')

% figure(9)
% plot(t,y(:,2),t,y(:,3),t,y(:,4))
% axis([0,120,0,1200])
% legend('Substrate','Oxygen','Carbon di-oxide')
% % yyaxis right
% figure(2)
% vde1 = [100 79.55 48.9 17.71 7.03];
% vde2 = [90.8 85.6 81.3 77.7 59];
% vde3 = [91.5 69 67.4 60.5 50.2];
% vde4 = [92.5 78.8 71 68.9 48.9];
% %stabalized virus
% vde5 = [94.15 79.1 76.4 69.175 55.05];
% vde1 = [97.2 53.03 46.01 25 5.04];
% vde2 = [94 79.8 78.7 67.2 52];
% vde3 = [94.9 83 76.3 74.4 58.5];
% vde4 = [90.5 74.5 73.9 68.3 54.7];
% err_1 = [vde1(1) vde2(1) vde3(1) vde4(1)];
% err_2 = [vde1(2) vde2(2) vde3(2) vde4(2)];
% err_3 = [vde1(3) vde2(3) vde3(3) vde4(3)];
% err_4 = [vde1(4) vde2(4) vde3(4) vde4(4)];
% err_5 = [vde1(5) vde2(5) vde3(5) vde4(5)];
%
% sde = [std(err_1) std(err_2) std(err_3) std(err_4) std(err_5)];
figure(j+4)
plot(ti,vde(j,:),'o')
% hold on;
% plot(ti,vde2,'^','Markersize',10,'MarkerFaceColor','b')
% hold on;
% plot(ti,vde3,'^','Markersize',10,'MarkerFaceColor','g')
% hold on;
% plot(ti,vde4,'^','Markersize',10,'MarkerFaceColor','y')
hold on;
plot(t,vd)
ylabel('% of viable cells')
xlabel('Time(h)')
title('Infected cell vs time')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
% axis([0,100,20,100])
% legend('Exp1','Exp2','Exp3','Exp4','Simulated')
% disp(p)
% figure(3)
% plot(ti,y(:,2),ti,y(:,3),ti,y(:,4));
% legend('Substrate','Oxygen','Carbon di-oxide')
% set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
% ylabel('Concentration')
% xlabel('Time(h)');
% figure(4)
% plot(ti,y(:,6),ti,y(:,7))
% legend('Virus Conc','Infected CD')
% set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
% ylabel('Concentration')
% xlabel('Time(h)');
end
```

Infected cell Objective function

```
function [rmse] = my_obj_infected(param)
tspan = [0 24 48 72 96];
UC = [460000 590000 650000];
y0(2) = param(1);
y0(3) = param(2);
y0(4) = 0;
y0(5) = 0;
y0(6) = param(3);
y0(7) = 0;
Error = 0;

for j=1:3
    y0(1) = UC(j);
[t,y] = ode15s(@(t,y)ODEset_infected(t,y,[param(1,1:14) param(15+(j-1))
param(15+size(4,1)+(j-1)) param(15+2*size(4,1)+(j-1))]),tspan,y0);
VCI = [100 90.8 92.5];
dead = (y(:,5)./(y(:,1)+y(:,7)))*100 ;
vd = VCI(j) - dead;
uc =  [460000 750000 940000 970000 950000;
        590000 1210000 1510000 1470000 1370000;
        650000 1010000 1220000 1390000 1290000];

vde = [100 79.55 48.9 17.71 7.03;
        90.8 85.6 81.3 77.7 59;
        92.5 78.8 71 68.9 48.9];
inf(:,1) = y(:,1) + y(:,7);

sum1=0;
sum2=0;
% error = 0;
for i=1:5
    sum1= (sum1 +(uc(j,i)-inf(i,1)).^2);
%     Rsq1 = 1 - sum((uc(j,i)-inf(i,1)).^2)/sum((uc(j,i)-mean(uc)).^2);
end
for i=1:5
    sum2= (sum2 +(vde(j,i)-vd(i,1)).^2)*param(24);
end
Error = Error + sum1 + sum2;
end
[rmse] = sqrt(Error/15)
% set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
end
```

Infected cell ODE set

```
function dmdt = ODEset_infected(t,y,param)
Ys = param(4);%1000000;
Yo2 = param(5);%1000000000;
Ko2 = param(6);%10;
Yco2 = param(7);%400000000;
kla = param(8);%0.144;
Ostar = param(9);%11;
Ks = param(10);%12;
K = param(11);%11;
Kco2 = param(12);%2000;% frac = 0.3;
a1 = param(13);%0.000001;
ka = param(14);%0.02;

mumax = param(15);%0.01;
kd1 = param(16);%0.0266;
kstar =  param(17);%0.0002;

mu = mumax*(y(2)/(y(2)+Ks))*(y(3)/(y(3) + Ko2))*(K/(exp(y(4)/Kco2)+1));
mu1 = mumax*(y(2)/(y(2) + Ks));
kd3 = kstar*log(y(6));
if t< 48
    kd = kd1;
else
    kd = kd1+kd3;
end

dydt(1) =  mu*y(1) - a1*y(6)*y(1) - kd1*y(1);
dydt(2) = -(mu1)*((y(1)+y(7))/Ys) ;
dydt(3) = kla*(Ostar - y(3)) - (y(1)+y(7))/Yo2;
dydt(4) = (y(1)+y(7))/Yco2 ;
dydt(5) = kd1*y(1) + kd*y(7);
dydt(6) = ka*y(7) - a1*y(1)*y(6);
dydt(7) = a1*(y(1)*y(6)) - kd*y(7);
dmdt = dydt';
end
```

Infection growth with passage

```
function Infected_passage
close all;
clear all;
clc;
y0(1) = 535000;%--cell
y0(2) = 1000;%---substrate
y0(3) = 1000;%---oxygen
y0(4) = 0;%---co2
y0(5) = 0.063*y0(1);%---dead cells
y0(6) = 53500;%----virus
y0(7) = 0;%----inf cells
xfE = [];  xfB = [];
passages = 27; time = 0; t0 = 0; tf = 48;
param(1) = 0.007;
param(2) = 0.0034;
param(3) = 0.2;
for i = 1:passages
[~,y] = ode45(@(t,y)ODEset_infected(t,y,param),[t0 tf],y0);
% % y(end,:)
param(1) = 1.02*param(1);
% param(2) = 1.02*param(2);
% param(3) = 1.01*param(3);
if i<5
    y0(1) = 535000;
    y0(5) = 0.066*y0(1);
elseif i>= 5 && i<15
    y0(1) = 523330;
    y0(5) = 0.096*y0(1);
else
    y0(1) = 430000;
    y0(5) = 0.07*y0(1);
end
y0 = [y0(1) 1000 1000 0 y0(5) y0(1)/10 0] + 0.1*y(end,:) ;
% y0 = [(535000+0.1*y(end,:)) 1000, 1000, 0 (0.1*y(end,5)) (0.1*y(end,6))
0.1*y(end,7)];
t0 = tf; tf = tf + 48;
time = time + 48;
% xcumE = [xcumE [x(1:end,1)]'];
% xcumB = [xcumB [x(1:end,2)]'];
vd = 100 - (y(end,5)./(y(end,1)+y(end,7)))*100 ;
xfE = [xfE (y(end,1)+y(end,7))];
xfB = [xfB vd];
% sz = size(xfE);
% disp(sz);
end
% inf_pass = [1213330 1750000 1670000 1510000];
% via_pass = [56 66.2 66 57];
pass = [1 5 15 27];
Bar_y = [940000 1510000 1030000 1220000
         1510000 1350000 1240000 1250000
         1030000 1240000 1680000 1100000
         1220000 1250000 1630000 1430000];
Bar_y = [xfE(1,1) 1030000
         xfE(1,5) 1240000
         xfE(1,15) 1680000
         xfE(1,27) 1630000 ];
% Bar_y = [xfE(1,1) 1040000 1430000 1300000 1450000
%          xfE(1,5) 1010000 1350000 1240000 1250000
%          xfE(1,15) 1080000 1750000 1780000 1700000
%          xfE(1,27) 1160000 1330000 1420000 1540000];

bAr_y = [48.9 58.87 62.12 68.29
         81.3 82.3 81.5 86.5
         67.4 74.5 80.4 75.5
         71 75.2 79.2 80.6];
bAr_y = [xfB(1,1) 62.12
         xfB(1,5) 81.5
         xfB(1,15) 80.4
         xfB(1,27) 79.2];
% bAr_y = [xfB(1,1) 46.01 78.7 76.3 73.9
%          xfB(1,5) 76.15 81.7 79.1 81
%          xfB(1,15) 70.02 82.9 81.6 78.4
%          xfB(1,27) 60.78 71.9 79.3 65.6];


figure(1)
bb = bar(pass,Bar_y,'g');
bb(1).FaceColor = 'k';
hold on;
% plot(pass,inf_pass);
legend('Simulated','Exp3','Exp2','Exp3','Exp4','location','south')
xlabel('passage');title('Infected Cell with passage')
ylabel('Cell Density')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)

% hold on;
% axis([0,28,0,2000000])

figure(2)
bb2 = bar(pass,bAr_y,'g');
bb2(1).FaceColor = 'k';
hold on;
% bar(pass,via_pass);
xlabel('passage');title('Infected Cell with passage')
ylabel('Percentage of Viable cells')
legend('Simulated','Exp3','Exp2','Exp3','Exp4','location','south')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)

% hold on;
% axis([0,28,0,100])
end


function dmdt = ODEset_infected(t,y,param)
Ys = 1000000;
Yo2 = 1000000000;
Ko2 = 69.36;
```

```
Yco2 = 400000000;
kla = 0.1129;
Ostar = 21.3356;
Ks = 79.31;
K = 9;
Kco2 = 1999.9;
a1 = 0.0000001;
ka = 0.1659;

mumax = param(1);
kd1 = param(2);%0.0266;
kstar =  param(3);%0.0002;

mu = mumax*(y(2)/(y(2)+Ks))*(y(3)/(y(3) + Ko2))*(K/(exp(y(4)/Kco2)+1));
mu1 = mumax*(y(2)/(y(2) + Ks));
kd3 = kstar*log(y(6));
if t< 48
    kd = kd1;
else
    kd = kd1+kd3;
end

dydt(1) =  mu*y(1) - a1*y(6)*y(1) - kd1*y(1);
dydt(2) = -(mu1)*((y(1)+y(7))/Ys) ;
dydt(3) = kla*(Ostar - y(3)) - (y(1)+y(7))/Yo2;
dydt(4) = (y(1)+y(7))/Yco2 ;
dydt(5) = kd1*y(1) + kd*y(7);
dydt(6) = ka*y(7) - a1*y(1)*y(6);
dydt(7) = mu*y(7) + a1*(y(1)*y(6)) - kd*y(7);
dmdt = dydt';
end
```

Complex model: main file

```
clc;
close all;
clear all;
global t
% lb=zeros(1,14);
p0 = [5350 535 1000 1200 0.00002 30 100 0.912 500 15 120000 800000 0.00004
2.5 0.009263 4.3 1 0.01 0.076 15 0.01 1.27 0.001 0.5 10 18 1.09 0.002 0.001];
lb = [1000 100 500 600 0.00001 20 50 0.5 200 10 100000 500000 0.00001 1.5
0.005263 1.43 0.5 0.008 0.03 5 0.0006 0.5 0.0008 0.3 8 15 1 0.001 0.0008];
ub = [53500 5350 5000 2000 0.00005 100 300 1.5 1000 30 400000 1000000 0.00008
5 0.02 10 2 0.05 0.1 40 0.05 2.7 0.002 1 12 20 1.11 0.005 0.005];
p = fmincon(@Obj_Complex,p0,[],[],[],[],lb,ub);
tspan = [0 96];
UC = [460000 590000 650000];
y0(2:62) = 0;
y0(63) = p(1);
y0(64) = p(2);
y0(65) = 0;
y0(66) = p(3);
y0(67) = p(4);
y0(68:75) = 0;
VCI = [100 90.8 92.5];
for j=1:3
    y0(1) = UC(j);
[t,y] = ode23(@(t,y)ODEset_complex(t,y,p),tspan,y0);
% vd(:,1) = 100 - (y(:,65)./(y(:,1)+y(:,2)+f+g))*100 ;
ti = [0 24 48 72 96];
f = 0;
g = 0;
 for i = 1:30
 d = y(:,i+2);
 f = f + d;
 e = y(:,i+32);
 g = g + e;
 end
poly = ((f + g)./(f+g+y(:,1)+y(:,2)))*100;
figure(j)
plot(t,(y(:,1)+y(:,2)+f+g));%,t,y(:,1),t,y(:,2),t,f,t,g);
% legend('Cell D','CU','CD','CS','CB')
% legends_a{j} = char(['ws=',num2str(ws)]); %,' max: ',num2str(Max_c)]);
% legend(char(legends_a),'location','best')
% hold on;
% disp(j);
% end
hold on;
uc =  [460000 750000 940000 950000 930000;
        590000 1210000 1470000 1510000 1370000;
        650000 1010000 1220000 1390000 1290000];

vde = [100 79.55 48.9 17.71 7.03;
        90.8 85.6 81.3 77.7 59;
        92.5 78.8 71 68.9 48.9];
dead = (y(:,5)./(y(:,1)+y(:,7)))*100 ;
vd = VCI(j) - dead;
```

```
% cell =  [535000 970000 1175000 1220000 1225000];
% uc1 =  [535000 970000 1175000 1240000 1225000];
% uc2 =  [523330 1190000 1280000 1270000 1410000];
% uc3 =  [560000 1086670 1253330 1310000 1310000];
% uc4 =  [630000 1000000 1430000 1340000 1510000];
% plot(ti, cell,'O');
plot(ti,uc(j,:),'o')
ylabel('Cell Density')
xlabel('Time(h)')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
title('Infected cell vs time')
% hold on;
% plot(ti,uc2,'d','Markersize',10,'MarkerFaceColor','r')
% hold on;
% plot(ti,uc3,'d','Markersize',10,'MarkerFaceColor','y')
% hold on;
% plot(ti,uc4,'d','Markersize',10,'MarkerFaceColor','g')
figure(j+5)
c_poly_E1 =[92.95 92.66 91.48];
ti2 =  [72 96 120];
plot(ti2,c_poly_E1,'d');
hold on;
plot(t,poly);
hold on;
ylabel('% of polyhedra/cell')
xlabel('Time(h)')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
title('Infected cell vs time')

figure(j+10)
plot(t,vd);
hold on;
plot(ti,vde(j,:),'o')
ylabel('Cell Viabaility')
xlabel('Time(h)')
set(gca,'FontName','Times New
Roman','Fontsize',12,'FontWeight','bold','linewidth',3)
title('Infected cell vs time')
end
% figure(4)
% plot(t,y(:,66),t,y(:,67),t,y(:,68))
% legend('Substrate','O2','CO2')
% axis([0,120,0,1200])
%
% figure(5)
% plot(t,y(:,69),t,y(:,70),t,y(:,71),t,y(:,72))
% legend('DNA','RNAe','RNAl','RNAvl')
```

Infected cell Objective function

```
function  [rmse] = Obj_Complex(param)
tspan = [0 24 48 72 96];
UC = [460000 590000 650000];
y0(2:62) = 0;
y0(63) = param(1);
y0(64) = param(2);
y0(65) = 0;
y0(66) = param(3);
y0(67) = param(4);
y0(68:75) = 0;
VCI = [100 90.8 92.5];
Error = 0;
for j=1:3
    y0(1) = UC(j);
[t,y] = ode23(@(t,y)ODEset_complex(t,y,param), tspan, y0);
f = 0;
g = 0;
 for i = 1:30
 d = y(:,i+2);
 f = f + d;
 e = y(:,i+32);
 g = g + e;
 end
poly(:,1) = ((f + g)./(f+g+y(:,1)+y(:,2)))*100;
dead = (y(:,5)./(y(:,1)+y(:,7)))*100 ;
vd = VCI(j) - dead;
uc =  [460000 750000 940000 970000 950000;
        590000 1210000 1510000 1470000 1370000;
        650000 1010000 1220000 1390000 1290000];

vde = [100 79.55 48.9 17.71 7.03;
        90.8 85.6 81.3 77.7 59;
        92.5 78.8 71 68.9 48.9];
% c_poly_E1 =[73.64 76.53 69.10];
% c_poly_E1 = [92.95 92.66 91.48];
inf(:,1) = y(:,1) + y(:,2) + f + g;
sum1=0;
sum2=0;
% error = 0;
for i=1:5
    sum1= sum1 + (uc(j,i)-inf(i,1))^2;
end
for i=1:5
    sum2= (sum2 + (vde(j,i)-vd(i,1))^2);
end
Error = sum1+sum2;
end
[rmse] = sqrt((Error)/15)

End
```

Complex model ODE set

```
function dmdt = ODEset_complex(t,y,param)
a= param(5);%0.00002;
b= param(6);%30;
Ks = param(7);%100;
K = param(8);%0.912;
Ko2 = param(9);%500;
Kco2 = param(10);%15;
Ys = param(11);%120000;
Yo2 = param(12);%800000;
kla = param(13);%0.00004;
kdna = param(14);%2.5;
kdeg = param(15);%0.009263;
krna = param(16);%4.3;
Kpoly = param(17);%1;
Kfp = param(18);%0.01;
kpe = param(19);%0.076;
Krna = param(20);%15;
ktrans = param(21);%0.01;
Ostar = param(22);%1.27;
alpha = param(23);%0.0001;
beta = param(24);%0.5;

mumax = param(27);%1.1;
kd1 = param(28);%0.002;
kstar = param(29);%0.001;

mu1 = mumax*(y(66)/(y(66) + Ks));
O2 = (y(67)/(y(67) + Ko2));
CO2 = (K/(y(68)+ Kco2));
mu = mu1*O2*CO2;
for i = 1:30
    if i <=param(25)
        z(i) = 1;
    else
        z(i) = 0;
    end
end
for i = 1:30
    if i <=param(26)
        p(i) = 0;
    else
        p(i) = alpha*exp(beta*(i-param(26)));
    end
end
kd3 = kstar*log(y(69));
if t< 24
    kd = kd1;
else
    kd = kd1+kd3;
end
dydt(1) = mu*y(1) - a*(y(63)+y(64))*y(1) - kd1*y(1);
dydt(2) = mu*y(2) + a*y(64)*y(1) -a*y(63)*y(2) - kd*y(2);
dydt(3) = a*y(63)*y(1) + ktrans*y(75) - y(3) - y(3)*p(1) -a*y(3)*y(64)*z(1)-
kd*y(3);
for i = 4:31
    dydt(i) = y(i-1) + ktrans*y(75) - y(i) -y(i)*p(i-2) - a*y(i)*y(64)*z(i-2)
- kd*y(i);
end
dydt(32) = y(31)+ ktrans*y(75) - y(32)*p(30) - a*y(32)*y(64)*z(30) -
kd*y(32);
sumes = 0;
for i = 1:30
v = y(i+2)*z(i);
sumes = sumes + v;
end
dydt(33) = ktrans*y(75)+ a*y(64)*sumes +a*y(63)*y(2)- y(33) - y(33)*p(1) -
kd*y(33);
for i = 34:61
    dydt(i) =  ktrans*y(75)+ (y(i-1) - y(i)) - y(i)*p(i-32) - kd*y(i);
end
dydt(62) = ktrans*y(75) + y(61) - y(62)*p(30)- kd*y(62);
r =0;
s= 0;
for i = 1:30
    v = y(i+2)*p(i);
    r = r + v;
    x = y(i+32)*p(i);
    s = s + x;
end

dydt(63) = b*r*y(73) - a*y(63)*sum(y(1:62));
dydt(64) = b*s*y(73) - a*y(64)*sum(y(1:62));
dydt(65) = kd1*y(1) + kd*sum(y(2:62));
dydt(66) = -mu1*((sum(y(1:62))/Ys));
dydt(67) = kla*(Ostar - y(67)) - ((sum(y(1:62))/Yo2)) ;
dydt(68) = ((sum(y(1:62))/Yo2));
     if  6< t && t <= 20
         f1 = 1 - ((t-6)/14);
     else
         f1 = 0;
     end
     if 1<t  && t <= 6
         f2 = 1 -((t-1)/5);
     else
         f2 = 0;
     end
     if 6<t  && t <= 15
         f3 = 1 -((t-6)/9);
     else
         f3 = 0;
     end
     if 15<t  && t <= 48
         f4 = 1 -((t-15)/33);
     else
         f4 = 0;
     end
dydt(69) = f1*a*y(63) + kdna*y(69)*f1 - kdeg*y(69);
dydt(70) = f2*krna*y(63) - kdeg*y(70);                          %RNA early
dydt(71) = f3*krna*y(69) - kdeg*y(71);                           %RNA late
dydt(72) = f4*krna*y(69) - kdeg*y(72);
dydt(73) = f2*kpe*(y(70)/(y(70) + Krna))*sum(y(2:62)) - kdeg*y(73);
%gp64 protein
dydt(74) = f3*kpe*(y(71)/(y(71) + Krna))*(1/(Kfp + y(74)))*sum(y(2:62)) -
kdeg*y(74);     %fp25k protein
dydt(75) = f4*kpe*(y(72)/(y(72) + Krna))*(y(74)/(y(74)+Kpoly))*sum(y(2:62))-
kdeg*y(75); %ODE protein
dmdt = dydt';
end
```