

William E. Schiesser

Virus Host Cell Genetic Material Transport

Computational ODE/PDE Modeling
with R



Springer

Virus Host Cell Genetic Material Transport

William E. Schiesser

Virus Host Cell Genetic Material Transport

Computational ODE/PDE Modeling with R

 Springer

William E. Schiesser
Lehigh University
Bethlehem, PA, USA

ISBN 978-3-030-68864-6 ISBN 978-3-030-68865-3 (eBook)
<https://doi.org/10.1007/978-3-030-68865-3>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express, or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The reproduction and spread (dispersion, diffusion) of a virus during an epidemic proceeds when the virus attaches (adheres, binds, fuses) to a host cell and viral genetic material (VGM) (protein, DNA, RNA) enters (invades, penetrates) the cell, then replicates, and perhaps mutates, in the cell. This process of a virus host cell interaction is described in [1]:

It has been known for decades that once a virus gets inside a cell, it hijacks the cellular processes to produce virally encoded protein that will replicate the virus's genetic material. Viral mechanisms are capable of translocating proteins and genetic material from the cell and assembling them into new virus particles.

The virus replication within the host cell is described in [1]:

An individual viral particle, called a virion, is a far simpler structure than a bacterium. It has often been questioned whether a virus is alive. It is certainly not living in the everyday sense of the word. Virions consist of genetic material—DNA or RNA enclosed in a protein coating. Many viruses, called enveloped viruses, have an additional outer membrane that encloses the protein coat. This membrane envelope is material co-opted from the cell's own membrane. As the new virion buds out from an infected host cell, it is wrapped by the cell's bilayer membrane and carries with it any protein that happens to be embedded in the membrane at the budding site. Enveloped viruses are then free to begin a new cycle of infection by fusing their cell-derived envelope with the cellular membrane of an uninfected cell.

The movement of the VGM across the host cell outer membrane is a spatiotemporal dynamic process that is modeled in this book as a system of ordinary and partial differential equations (ODE/PDEs). First, a ODE/PDE model is developed for a single protein transmitted from the virus through the cell membrane into the cell interior where it replicates, and possibly mutates.

The movement of the virus protein through the cell membrane is modeled as a diffusion process expressed by the diffusion PDE (Fick's second law) with dependent variable $V_1(x, t)$, and independent variables x , the position in the membrane, and t , time. Within the cell, the time variation of the VGM is modeled as an ODE with dependent variable $C_1(t)$. The subscript 1 denotes the first virus protein considered.

The single protein model is then extended to the replication, and possibly mutation, of the virus protein within the cell interior to produce additional virus proteins that diffuse out through the cell membrane where they can infect other host cells.

The model time scale in minutes is $0 \leq t \leq 240$ (four hours), and the evolution of the dependent variables is computed by the numerical integration of the ODE/PDEs from zero initial conditions (IC) $V_1(x, t = 0) = C_1(t = 0) = 0$. The departure of the dependent variables from zero is in response to the virus protein concentration at the outer membrane surface (the point at which the virus binds to the host cell).

The numerical integration of the ODE/PDE model equations is performed with routines coded (programmed) in R, a quality, open-source scientific computing system that is readily available from the Internet. Formal mathematics is minimized, e.g., no theorems and proofs. Rather, the presentation is through detailed examples that the reader/researcher/analyst can execute on modest computers. The ODE dependent variables are plotted against t and the PDE dependent variables are plotted against x and t with basic R plotting utilities. The solution is by the numerical method of lines (MOL), an established general algorithm for ODE/PDE systems.

As extensions of the ODE/PDE models,

- The flux of virus proteins at the outer cell membrane boundary is computed and plotted against t . Also, a hypothesized vaccine is included by variations in the entering flux at the cell membrane outer surface.
- A therapeutic drug is hypothesized to vary the rate of VGM production within the cell.
- The RHS terms and LHS derivatives in t are computed and plotted as an indication of the origin of the solution properties.
- Cross diffusion between two virus proteins in the cell membrane is implemented within the MOL framework.

The R routines are available from a download link so that the example models can be executed without having to first study numerical methods and computer coding. The routines can then be applied to variations and extensions of the ODE/PDE model, such as changes in the parameters and the form of the model equations.

The author would welcome comments/suggestions concerning this approach to the analysis of the virus host cell dynamics (directed to wes1@lehigh.edu).

[1] Cohen, F.S. (2016), How Viruses Invade Cells, *Biophysical Journal*, **110**, pp 1026–1032

Contents

1	Virus Protein ODE/PDE Models	1
1.1	ODE/PDE model for a single virus protein	2
1.2	ODE/PDE model for a second virus protein	3
	References	4
2	Implementation of the ODE/PDE Models	5
2.1	R routines for the ODE/PDE models	5
2.1.1	Main program for one protein	6
2.1.2	ODE/MOL routine	11
2.1.3	Numerical, graphical output	14
2.2.1	Main program for two proteins	18
2.2.2	ODE/MOL routine	24
2.2.3	Numerical, graphical output	27
2.3	Summary and conclusions	32
	Reference	33
3	Host Cell Proteins with Cross Diffusion	35
3.1	R routines for an ODE/PDE model with cross diffusion	35
3.1.1	Main program	37
3.1.2	ODE/MOL routine	46
3.1.3	Numerical, graphical output	52
3.2	Summary and conclusions	60
	Reference	60
4	Postulated Vaccine and Therapeutic	61
4.1	ODE/PDE model for a postulated vaccine	61
4.1.1	Main program	62
4.1.2	ODE/PDE routine	62
4.1.3	Numerical, graphical output	63
4.2	ODE/PDE model for a postulated therapeutic	71
4.2.1	Main program	71

4.2.2	ODE/PDE routine.....	73
4.2.3	Numerical, graphical output.....	73
4.3	Summary and conclusions.....	81
	Reference.....	81
5	Detailed ODE/PDE Model Analysis.....	83
5.1	Detailed analysis of ODE/PDE model.....	83
5.1.1	Main program for ODE analysis.....	85
5.1.2	ODE/PDE routine.....	92
5.1.3	Numerical, graphical output.....	93
5.2.1	Main program for PDE analysis.....	98
5.2.2	ODE/PDE routine.....	105
5.2.3	Numerical, graphical output.....	105
5.3	Summary and conclusions.....	114
	Reference.....	115
A1.1	dss004 listing.....	117
A1.2	Test of dss004.....	119
A2.1	Test of dss044.....	135
A2.1.1	Main program.....	135
A2.1.2	dss044 Listing.....	150
A2.2	p refinement.....	153
A2.2.1	Main program.....	153
A2.2.2	dss006, dss046 listings.....	153
A2.2.3	Graphical, numerical output.....	156
A2.3	Summary and conclusions.....	164
	Reference.....	165
	Index.....	167

Chapter 1

Virus Protein ODE/PDE Models



Abstract The modeling of virus host cell interactions starts with a virus protein transmitted from the virus through the cell membrane into the cell interior where it replicates, and possibly mutates.

The movement of the virus protein through the cell membrane is modeled as a diffusion process expressed as a partial differential equation (PDE). Within the cell, the time variation of the virus protein is modeled with an ordinary differential equation (ODE).

The ODE/PDE model is then extended to two proteins. The second protein is produced within the cell and diffuses back out of the cell as a virion that can then go on to infect other cells. This process describes the spread of the virus.

Keywords virus host cell interaction · genetic material transport · mathematical model · partial differential equation (PDE) · initial condition (IC) · boundary condition (BC) · ordinary differential equation (ODE) · R coding · method of lines (MOL)

Introduction

The modeling of virus host cell interactions starts with a virus protein¹ transmitted from the virus through the cell membrane into the cell interior where it replicates, and possibly mutates. This process of virus-cell/adhesion-invasion is briefly described as [3]

A virus causes disease by entering cells in the human body and hijacking their genetic machinery so as to reproduce itself again and again: It turns its hosts into viral factories.

A more detailed introduction to virus cell interactions is given in [1, 2, 4].

The movement of the virus protein through the cell membrane is modeled as a diffusion process expressed as a partial differential equation (PDE). Within the

¹*virus protein* also termed *viral protein* or just *protein*. *host cell* also termed just *cell*.

cell, the time variation of the virus protein is modeled with an ordinary differential equations (ODE).

The statement of this ODE/PDE model follows, including the initial conditions (ICs), boundary conditions (BCs) and parameters.

1.1 ODE/PDE model for a single virus protein

The ODE/PDE variables are listed in Table 1.1.

Table 1.1 ODE/PDE model variables

$V_1(x, t)$	virus protein concentration in the cell membrane
$C_1(t)$	virus protein concentration in the cell interior
x	position within the membrane
t	time

The movement of the virus protein in the cell membrane is modeled with the diffusion equation

$$\frac{\partial V_1(x, t)}{\partial t} = D_{V1} \frac{\partial^2 V_1(x, t)}{\partial x^2} \quad (1.1-1)$$

where D_{V1} is a diffusivity.

Eq. (1.1-2) is second order in x and therefore requires two boundary conditions (BCs).

$$D_{V1} \frac{\partial V_1(x = x_u, t)}{\partial x} = k_{1u}(V_{1s}(t) - V_1(x = x_u, t)) \quad (1.1-2)$$

BC (1.1-2) equates the rate of diffusion of the protein at the membrane outer boundary $x = x_u$, $-D_{V1} \frac{\partial V_1(x = x_u, t)}{\partial x}$ (Fick's first law), to the rate of mass transfer at the outer boundary, $k_{1u}(V_{1s}(t) - V_1(x = x_u, t))$, where k_{1u} is a mass transfer coefficient to be specified. $V_{1s}(t)$ is the concentration of viral genetic material outside the cell.

$$-D_{V1} \frac{\partial V_1(x = x_l, t)}{\partial x} = k_{1l}(C_1(t) - V_1(x = x_l, t)) \quad (1.1-3)$$

BC (1.1-3) equates the rate of diffusion of the protein at the membrane inner boundary $x = x_l$, $-D_{V1} \frac{\partial V_1(x = x_l, t)}{\partial x}$ (Fick's first law), to the rate of mass transfer at the inner boundary, $k_{1l}(C_1(t) - V_1(x = x_l, t))$, where k_{1l} is a mass transfer coefficient to be specified.

Eq. (1.1-1) is first order in t and requires one initial condition (IC)

$$V_1(x, t = 0) = V_{10}(x) \quad (1.1-4)$$

where $V_{10}(x)$ is a function to be specified, and is generally taken as the zero function, that is, no virus protein initially in the cell membrane.

The concentration of the protein in the cell interior, $C_1(t)$, is modeled with an ODE

$$\frac{dC_1(t)}{dt} = -k_{1l}(C_1(t) - V_1(x = x_l, t)) \quad (1.1-5)$$

that equates the temporal derivative $\frac{dC_1(t)}{dt}$ to the rate of transfer of the protein from the inner boundary of the membrane, $-k_{1l}(C_1(t) - V_1(x = x_l, t))$, to the cell interior.

The IC for eq. (1.1-6) is

$$C_1(t = 0) = C_{10} \quad (1.1-6)$$

where C_{10} is a prescribed constant.

Eqs. (1.1) constitute the ODE/PDE model for a single virus protein. The model is next extended so that the cell interior produces a protein that then diffuses out of the cell.

1.2 ODE/PDE model for a second virus protein

The variables and parameters for the second protein are designated with subscript 2. The diffusion equation is a direct analog of eq. (1.1-1).

$$\frac{\partial V_2(x, t)}{\partial t} = D_{V2} \frac{\partial^2 V_2(x, t)}{\partial x^2} \quad (1.2-1)$$

The BC (1.2-2) at the outer boundary $x = x_u = 1$ equates the rate of diffusion, $D_{V2} \frac{\partial V_2(x = x_l, t)}{\partial x}$, to the mass transfer rate in response to an ambient protein concentration V_{2a} , $k_{2u}(V_{2a} - V_2(x = x_u, t))$, where k_{2u} is a mass transfer coefficient for the second protein.

$$D_{V2} \frac{\partial V_2(x = x_u, t)}{\partial x} = k_{2u}(V_{2a} - V_2(x = x_u, t)) \quad (1.2-2)$$

BC (1.2-3) equates the rate of diffusion of the protein at the membrane inner boundary $x = x_l$, $D_{V2} \frac{\partial V_2(x = x_l, t)}{\partial x}$, to the rate of mass transfer at the inner boundary, $-k_{2l}(C_2(t) - V_2(x = x_l, t))$, where k_{2l} is a mass transfer coefficient for the second protein.

$$D_{V2} \frac{\partial V_2(x = x_l, t)}{\partial x} = -k_{2l}(C_2(t) - V_2(x = x_l, t)) \quad (1.2-3)$$

The initial condition for $V_2(x, t)$ is

$$V_2(x, t = 0) = V_{20}(x) \quad (1.2-4)$$

where $V_{20}(x)$ is a function to be specified (usually taken as the zero function).

The concentration of the second protein in the cell interior is modeled with an ODE

$$\frac{dC_2(t)}{dt} = -k_{2l}(C_2(t) - V_2(x = x_l, t)) + k_{r2}C_1^{n_2} \quad (1.2-5)$$

that equates the temporal derivative $\frac{dC_2(t)}{dt}$ to the sum of (1) the rate of transfer of the second protein from the cell interior to the membrane inner boundary, $-k_{2l}(C_2(t) - V_2(x = x_l, t))$, and (2) the rate of production of the second protein from the first protein, $+k_{r2}C_1^{n_2}$. For (2), an n_2^{th} order reaction is assumed, but this rate can be modified to reflect another rate of production of the second protein.

The IC for eq. (1.2-5) is

$$C_2(t = 0) = C_{20} \quad (1.2-6)$$

where C_{10} is a prescribed constant.

Eqs. (1.1), (1.2) constitute the ODE/PDE models implemented in the R routines discussed in Chapter 2.

References

1. Cohen, F.S. (2016), How Viruses Invade Cells, *Biophysical Journal*, **110**, pp 1026–1032
2. Graw, F., and A.S. Perelson (2016), Modeling Viral Spread, *Annual Reviews Virology*, **3**, pp 555–572
3. Iwasaki, A., and R. Medzhitov (2020), Scared That Covid-19 Immunity Won't Last? Don't Be, *New York Times*, July 31, 2020, p1
4. Maginnis, M.S. (2018), Virus-Receptor Interactions: The Key to Cellular Invasion *Journal Molecular Biology*, **430**, pp 2590–2611

Chapter 2

Implementation of the ODE/PDE Models



Abstract The ODE/PDE model of Chapter 1 for a single virus protein is implemented in R routines consisting of a main program and an ODE/MOL subordinate routine called by the main program. The two protein model of Chapter 1 is then programmed as a main program and subordinate ODE/MOL routine.

The numerical and graphical output for the one and two protein models is displayed with standard R utilities. The formation of the second protein within the host cell can result in the transmission of the resulting virions to other host cells, and thus is the basis for virus transmission.

Keywords virus host cell interaction · viral genetic material (VGM) transport · VGM multiplication · VGM mutation · mathematical model · partial differential equation (PDE) · initial condition (IC) · boundary condition (BC) · ordinary differential equation (ODE) · R coding · method of lines (MOL)

Introduction

The ODE/PDE models of Chapter 1 for one protein (eqs. (1.1)) and two proteins (eqs. (1.2)) are implemented in R routines discussed in this chapter.

2.1 R routines for the ODE/PDE models

Eqs. (1.1) constitute the ODE/PDE model for a single virus protein that is implemented with the following R routines, starting with a main program.

2.1.1 Main program for one protein

The main program for eqs. (1.1) follows.

```

#
# One ODE, one PDE model
#
# Delete previous workspaces
  rm(list=ls(all=TRUE))
#
# Access ODE integrator
  library("deSolve");
#
# Access functions for numerical solution
  setwd("f:/vci/chap2");
  source("pdela.R");
  source("dss004.R");
#
# Parameters
  Dv1=1.0e-02;
  V1s=1;
  k1l=0.1;
  k1u=0.1;
  V10=0;
  C10=0;
#
# Spatial grid (in x)
  nx=21;xl=0;xu=1;
  x=seq(from=xl,to=xu,by=(xu-xl)/(nx-1));
#
# Independent variable for ODE integration
  t0=0;tf=240;nout=21;
  tout=seq(from=t0,to=tf,by=(tf-t0)/(nout-1));
#
# Initial condition (t=0)
  u0=rep(0,nx+1);
  for(i in 1:nx){
    u0[i]=V10;
  }
  u0[nx+1]=C10;
  ncall=0;
#
# ODE integration
  out=lsodes(y=u0,times=tout,func=pdela,
            sparsetype="sparseint",rtol=1e-6,

```

```

        atol=1e-6,maxord=5);
nrow(out)
ncol(out)
#
# Arrays for plotting numerical solution
V1=matrix(0,nrow=nx,ncol=nout);
C1=rep(0,nout);
for(it in 1:nout){
  for(i in 1:nx){
    V1[i,it]=out[it,i+1];
  }
  C1[it]=out[it,2+nx];
}
#
# Display numerical solution
iv=seq(from=1,to=nout,by=4);
for(it in iv){
  cat(sprintf("\n t          C1(t)\n"));
  cat(sprintf("%6.1f%18.3e\n",tout[it],C1[it]));
  cat(sprintf(" t          x          V1(t)\n"));
  iv=seq(from=1,to=nx,by=4);
  for(i in iv){
    cat(sprintf("%6.1f%6.1f%12.3e\n",
              tout[it],x[i],V1[i,it]));
  }
}
#
# Calls to ODE routine
cat(sprintf("\n\n ncall = %5d\n\n",ncall));
#
# Plot ODE/PDE solutions
#
# V1
par(mfrow=c(1,1));
matplot(x=x,y=V1,type="l",xlab="x",ylab="V1(x,t)",
        xlim=c(xl,xu),lty=1,main="",lwd=2,col="black");
#
# C1
par(mfrow=c(1,1));
plot(x=tout,y=C1,type="l",xlab="t (min)",ylab="C1(t)",
     xlim=c(t0,tf+10),lty=1,main="",lwd=2,col="black");

```

Listing 2.1 Main program for eqs. (1.1)

We can note the following details about Listing 2.1.

- Previous workspaces are deleted.

```
#
# One ODE, one PDE model
#
# Delete previous workspaces
rm(list=ls(all=TRUE))
```

- The R ODE integrator library `deSolve` is accessed [1]. Then the directory with the files for the solution of eqs. (1.1) is designated. Note that `setwd` (set working directory) uses `/` rather than the usual `\`.

```
#
# Access ODE integrator
library("deSolve");
#
# Access functions for numerical solution
setwd("f:/vci/chap2");
source("pdela.R");
source("dss004.R");
```

- The model parameters are specified numerically.

```
#
# Parameters
Dv1=1.0e-02;
V1s=1;
k1l=0.1;
k1u=0.1;
V10=0;
C10=0;
```

The parameter values were selected to give a time scale of 240 minutes (min) as explained subsequently. Specifically, numerical values are assigned to the

- Diffusivity D_{V1} in eqs. (1.1-1), (1.1-2), (1.1-3).
 - Source term in eq. (1.1-2), $V_{1s}(t)$.
 - Mass transfer coefficient k_{1l} in eqs. (1.1-3), (1.1-5).
 - Mass transfer coefficient k_{1u} in eq. (1.1-2).
 - ICs of eqs. (1.1-4), (1.1-6), $V_{10}(x)$, C_{10} .
- A spatial grid for eq. (1.1-1) is defined with 21 points so that $x = 0, 0.05, \dots, 1$. The membrane length is a normalized value, $x = x_u = 1$.

```
#
# Spatial grid (in x)
nx=21;xl=0;xu=1;
x=seq(from=xl,to=xu,by=(xu-xl)/(nx-1));
```


- An interval in t is defined for 21 output points, so that $t_{out}=0, 240/20=12, \dots, 240$ (min).

```
#
# Independent variable for ODE integration
t0=0;tf=240;nout=21;
tout=seq(from=t0,to=tf,by=(tf-t0)/(nout-1));
```

- ICs (1.1-4), (1.1-6) are implemented. $V_1(x, t = 0)$ is placed in a vector $u0[i]$ of length nx , then $C_1(t = 0)$ is added as an element to the end of the vector, $u0[nx+1]$

```
#
# Initial condition (t=0)
u0=rep(0,nx+1);
for(i in 1:nx){
  u0[i]=V10;
}
u0[nx+1]=C10;
ncall=0;
```

Also, the counter for the calls to `ode1a` is initialized.

- The system of $21 + 1$ ODEs is integrated by the library integrator `lsodes` (available in `deSolve`, [1]). As expected, the inputs to `lsodes` are the ODE function, `pde1a`, the IC vector $u0$, and the vector of output values of t , t_{out} . The length of $u0$ (22) informs `lsodes` how many ODEs are to be integrated. `func`, `y`, `times` are reserved names.

```
#
# ODE integration
out=lsodes(y=u0,times=tout,func=pde1a,
           sparsetype="sparseint",rtol=1e-6,
           atol=1e-6,maxord=5);
nrow(out)
ncol(out)
```

`nrow`, `ncol` confirm the dimensions of `out`.

- $V_1(x, t)$, $C_1(t)$ are placed in a matrix and vector for subsequent plotting.

```
#
# Arrays for plotting numerical solution
V1=matrix(0,nrow=nx,ncol=nout);
C1=rep(0,nout);
for(it in 1:nout){
  for(i in 1:nx){
    V1[i,it]=out[it,i+1];
  }
  C1[it]=out[it,2+nx];
}
```

The offset +1 is required because the first element of the solution vectors in `out` is the value of t and the 2 to 23 elements are the 21 values of V_1 and single value of C_1 . These dimensions from the preceding calls to `nrow`, `ncol` are confirmed in the subsequent output.

- The numerical values of $C_1(t)$, $V_1(x, t)$ returned by `lsodes` are displayed. Every fourth value in t and x appear from `by=4`.

```
#
# Display numerical solution
iv=seq(from=1,to=nout,by=4);
for(it in iv){
  cat(sprintf("\n t           C1(t)\n"));
  cat(sprintf("%6.1f%18.3e\n",tout[it],C1[it]));
  cat(sprintf(" t           x           V1(t)\n"));
  iv=seq(from=1,to=nx,by=4);
  for(i in iv){
    cat(sprintf("%6.1f%6.1f%12.3e\n",
               tout[it],x[i],V1[i,it]));
  }
}
```

- The number of calls to `pde1a` is displayed at the end of the solution.

```
#
# Calls to ODE routine
cat(sprintf("\n\n ncall = %5d\n\n",ncall));
```

- $V_1(x, t)$ is plotted against x and parametrically in t with the R utility `matplot`. `par(mfrow=c(1,1))` specifies a 1×1 matrix of plots, that is, one plot on a page.

```
#
# V1
par(mfrow=c(1,1));
matplot(x=x,y=V1,type="l",xlab="x",ylab="V1(x,t)",
        xlim=c(xl,xu),lty=1,main="",lwd=2,col="black");
```

- $C_1(t)$ is plotted against t with the R utility `plot`.

```
#
# C1
par(mfrow=c(1,1));
plot(x=tout,y=C1,type="l",xlab="t (min)",ylab="C1(t)",
     xlim=c(t0,tf+10),lty=1,main="",lwd=2,col="black");
```

This completes the discussion of the main program for eqs. (1.1). The ODE/MOL routine `pde1a` called by `lsodes` from the main program for the numerical MOL integration of eqs. (1.1) is next.