TRANSMISSION RATE IN PARTIAL DIFFERENTIAL EQUATION IN EPIDEMIC MODELS

A thesis submitted to

the Graduate College of

Marshall University

In partial fulfillment of

the requirements for the degree of

Master of Arts

in

Mathematics

by

Alaa Elkadry

Approved by

Dr. Anna Mummert, Committee Chairperson
Dr. Bonita Lawrence
Dr. Scott Sarra

Marshall University
May 2013

ACKNOWLEDGMENTS

I would like to thank and express my sincere gratitude to my thesis advisor, Dr. Anna Mummert, for her help and for providing the expertise, guidance, encouragement, and for her patience throughout the period of completing this thesis which made this project possible.

I would also like to thank my thesis committee, Dr. Bonita Lawrence and Dr. Scott Sarra who were always willing to help. Additionally, I am very grateful for the graduate research opportunities which have been provided by Department of Mathematics at Marshall University, and I would like to thank all my professors for teaching me and being my second family.

Finally, I would like to thank all members of my family who presented huge psychological or financial support helped me to achieve one of my goals in this life.

CONTENTS

LIST OF FIGURES

ABSTRACT

TRANSMISSION RATE IN PARTIAL DIFFERENTIAL EQUATION IN EPIDEMIC MODELS

Alaa Elkadry

The rate at which susceptible individuals become infected is called the transmission rate. It is important to know this rate in order to study the spread and the effect of an infectious disease in a population. This study aims at providing an understanding of estimating the transmission rate from mathematical models representing the population dynamics of an infectious diseases using two different methods. Throughout, it is assumed that the number of infected individuals is known. In the first chapter, it includes historical background for infectious diseases and epidemic models and some terminology needed to understand the problems. Specifically, the partial differential equations SIR model is presented which represents a disease assuming that it varies with respect to time and a one dimensional space. Later, in the second chapter, it presents some processes for recovering the transmission rate from some different SIR models in the ordinary differential equation case, and from the PDE-SIR model using some similar techniques. Later, in the third chapter, it includes some terminology needed to understand "inverse problems" and Tikhonov regularization, and the process followed to recover the transmission rate using the Tikhonov regularization in the non-linear case. And finally, in the fourth chapter, it has an introduction to an optimal control method followed to use Tikhonov regularization to recover the transmission rate.

Chapter 1

INTRODUCTION

Infectious diseases have been responsible for the death of millions of peoples all around the world [9]. Today, infectious diseases are still a major causes of mortality. This is why people are concerned and want to know more about each infectious disease. Epidemiology is the study of health and disease in a particular population in order to control associated health problems. In this paper we use mathematical models to study the transmission rate of some infectious diseases. We start by introducing terminology and describing some models.

1.1 INFECTIOUS DISEASES

Infectious diseases, also known as transmissible diseases or communicable diseases, are a great human concern. By the first wave of the bubonic plague pandemic, the Black Death, between 1348 and 1350, at least a third of the population of Europe had died, between 75 million and 200 million people [9].A lot of people die every year because of diseases such as AIDS, malaria and measles Millions of others suffer from the infection.

Infectious diseases are caused by pathogenic microorganisms or simply "germs." Germs are tiny living things that can be found almost everywhere such as in the air or in water. An infectious disease is an illness which arises through transmission of the infectious agent. You can get infected by touching, eating, drinking or breathing something that contains a germ. Infectious diseases are spread through both direct (Figure 1.1) and indirect contact

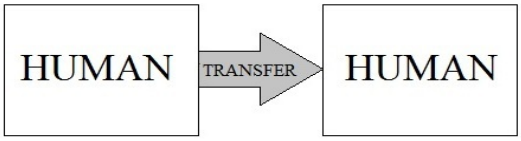


Figure 1.1: Disease transmission schematic with direct contact

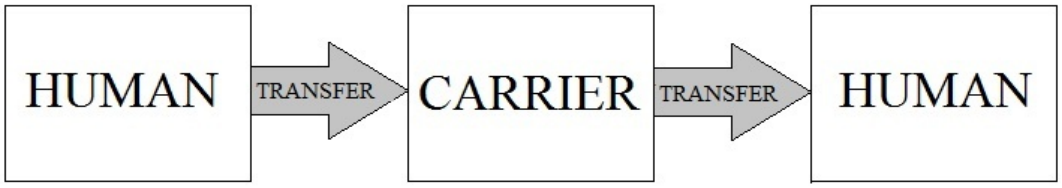


Figure 1.2: Disease transmission schematic with indirect contact

(Figure 1.2) [5].

Direct contact transmission occurs when there is physical contact between an infected person or animal and a susceptible person. Germs can pass from one person to another through physical contact, coughing, through a blood transfusion, or even through the exchange of body fluids from sexual contact. Also, a pregnant woman may pass germs that cause infectious diseases to her unborn baby. AIDS, Anthrax and Plagues are some examples of disease that spread by direct contact.

Indirect contact transmission occurs when there is no direct contact between the holder of the disease and the one that this disease is transmitted to. Germs can be found on inanimate objects such as doorknobs. When you touch a doorknob after someone with the flu has touched it, you can pick up the germs he or she left behind, and you may become infected. Also, a disease can spread by airborne dispersal, like droplet transmission, when someone is sick and he sneezes, he expels droplets into the air around him; the droplets he expels contain the germ that caused his illness. A disease can also spread through food

contamination or insect bites. Some germs use insects such as mosquitoes, fleas, lice or ticks to move from host to host [5]. Malaria and Tapeworm are some examples of disease that spread by indirect contact.

Table 1.1: Infectious Diseases and How They Are Spread

Disease	Transmission
AIDS	Contact of body fluid with that of an infected person. Sexual contact and sharing of unclean paraphernalia for intravenous drugs.
Blastomycosis	Inhaling contaminated dust
Botulism	Consuming contaminated food
Hepatitis	Direct or indirect contact with infected person
Common cold	Direct or indirect contact with infected person
Encephalitis	Mosquito bite
Gonorrhea	Sexual contact
Malaria	Mosquito bite
Measles	Direct or indirect contact with infected person
Tapeworm	Consuming infected meat or fish
Typhus	Lice, flea, tick bite

Infectious diseases are classified by frequency of occurrence; an infectious disease can be an epidemic, endemic or pandemic. An endemic is an infectious disease that is constantly present in a population. For example, malaria is endemic in Brazil. A disease that quickly and severely affects a large number of people and then subsides is an epidemic. For epidemics we have many cases of infection in a given area in short period. Seasonal influenza is an example of an epidemic. A pandemic disease is a world-wide epidemic that may affect entire continents or even the world. Influenza is occasionally pandemic, such as the pandemic of 1918. Throughout the Middle Ages, successive pandemics of the plague killed millions. Thus, from an epidemiologists point of view, the Black Death in Europe and AIDS in sub-Saharan Africa are pandemic rather than epidemics.

It is true that the health system has been improved, but now, people could be in contact easily with many others due to the development in the transportation system [3]. This why the risk of pandemics is still high. Though we know more about diseases, we are much more exposed to the risk.

1.2 MATHEMATICAL MODELING

Mathematical modeling is an activity of translating a real world situation into the abstract language of mathematics for subsequent analysis of this problem. A mathematical model is a mathematical description of the situation, typically in the form of a system of equations. The systems are solved to get solutions and then compared to experimental results.

Each involvement in developing other sciences is a success for Mathematics, and, for mathematicians, biology opens up many new branches of study. Biology provides interesting topics and mathematics provides models to help in understanding these topics. Infectious diseases are one of the interesting topics. Mathematical modeling helps in the understanding of the spread of an infectious disease and provides a platform to study how to control the spread and thus control health problems.

Mathematical models help in the study of patterns of infections, and therefore they are valuable tools in order to detect and prevent a disease. Mathematical modeling in epidemiology provides an understanding of the underlying mechanisms that influence the spread of disease and, in the process, it suggests control strategies. The disease transmission could be studied from different perspective such as animal to human transmission, but let us fix our attention on human to human transmission.

In order to study an infectious disease, many factors should be considered. Some of these factors are related to the population, such as age, sex, genetics and immunity status. Other factors are more related to the disease itself, such as the type of this disease. Finally, some other factors are related to the environment, such as pollution and temperature. Each factor could affect the speed and the spread of a disease in a different way. As with all modeling, mathematical modeling requires simplifying assumptions. It is hard to include all factors in the model because it will make the model more complicated and harder to solve and all factors may be unknown. On the other hand, if the model is too simple it may fail to provide a reasonable platform for testing hypotheses.

Mathematical models have been used to study the dynamics of infectious diseases at

the population level. Most models are in the form of ordinary differential equation(ODEs) [16] due to their simplicity and the limited computational tools that were available in the past. But as mentioned before, most of the time a more realistic description requires a more complicated system of equations to be analyzed mathematically. Advances in computational tools have opened the door for mathematical analysis of more complicated systems.

In order to make a model for a disease in a population, we divide the population into different classes and we study the variation in numbers in these classes with respect to time. The choice of these classes is related to the disease studied. The most used classes are, S, the number of susceptible individuals, E, the number of exposed individuals, I, the number of infected individuals and R, the number of recovered individuals. For example, if a model considers "flu" as disease, this model would be an SIS model because individuals recover with no immunity to the disease; that is, individuals are immediately susceptible once they have recovered. Some of the well studied epidemic models are: SI, SIR, SIS, $SEIR$, $SEIS$,...[See table 1.2]

Table 1.2: Different epidemiology models

Model name	Special Property	Example of a disease
SI	No recovery	AIDS
SIR	Permanent immunity after recovery	Influenza
SIS	No immunity after recovery	H1N5
$SEIR$	Latent period where the infected is not infectious yet	Measles
$SEIS$	SIS + latent period	Malaria

1.3 SIR MODEL

When we model a disease we need to understand the way this disease spreads, be able to predict the future of this disease and understand how we may control the spread of this disease. This why any model should describe the behavior of the disease in a correct way. Different researchers provided different models that describe different cases; in this paper, we will mention or describe some of those models.

In 1927 Kermack and McKendrick [8] provided an epidemic model that was considered a generalized model at the time. In this model, the total population is assumed to be constant and divided into three classes. Susceptible class contains individuals who have no immunity to the infectious agent; any member of the susceptible class could become infected. Infectious class contains individuals who are currently infected and can transmit the infection to susceptible individuals who they contact. Recovered class contains individuals who have returned to a normal state of health after being infected and those individuals have gained permanent immunity. This model is called the SIR model. In this model we assume that:

1. The way a person can leave the susceptible group, S, is to become infected. The way a person can leave the infected group, I, is to recover from the disease.

2. The recovery rate r and the transmission rate β are the same for all individuals and are supposed positive.

3. There is homogeneous mixing, which means that individuals of the population make contact at random and do not mix mostly in a smaller subgroup.

4. The disease is novel, so no vaccination is available.

5. The population size, N, is constant and large.

6. Any recovered person in R has permanent immunity.

Given these assumptions, Kermack and McKendrick presented the following system:

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \tag{1.1}$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - rI(t) \tag{1.2}$$

$$\frac{dR(t)}{dt} = rI(t) \tag{1.3}$$

with the initial conditions:

$$\begin{cases} S(0) = S_0 > 0 \\ I(0) = I_0 > 0 \\ R(0) = 0 \end{cases} \tag{1.4}$$

$S(t)$ is the number of susceptible individuals at time t, $I(t)$ is the number of infected individuals at time t, $R(t)$ is the number of removed individuals at time t. β is the transmission rate and r is the recovery rate [See table 1.3]. This model represents an epidemic which quickly infects many people then subsides. Therefore it is reasonable to assume that the class of susceptible individuals, S, does not include new incoming people from births or recovered individuals who lost their immunity to the disease. The term $\beta S(t)I(t)$ is the incidence rate or the rate of new infections; it is a bilinear incidence which is considered as a good model for disease spread in urban settings.

The total population $N(t) = S(t) + I(t) + R(t)$ is assumed to be constant. It is clear that $\frac{dN}{dt} = \frac{d(S+I+R)}{dt} = 0$. Therefore it is true that $N(t)$ is a constant, N.

Table 1.3: Units for the SIR ODE model

Variable	description	unit
β	Transmission rate	$\frac{1}{\text{people}\times\text{days}}$
r	Recovery rate	$\frac{1}{\text{days}}$
t	Time	days
S	Number of susceptible people	people
I	Number of infected people	people
R	Number of recovered people	people
N	Total number of people	people

The SIR model is a simple model that does not take in consideration age, sex, spatial position or any other factors. But it opened the door to the development of other extended models that describe various types of epidemics.

1.4 SIR MODEL INCLUDING SPATIAL POSITION

Webb [17] proposed and analyzed a disease model structured by spatial position. He assumed that spatial mobility is governed by random diffusion coefficients respectively D_S, D_I and D_R for the susceptible, infected and recovered classes.

We assume here that the spatial position is a bounded one-dimensional environment, $[0, L]$, with $L > 0$.

Let $S = S(t, x)$, $I = I(t, x)$; and $R = R(t, x)$ represent, respectively, the number of susceptible, infected, and recovered individuals at location x at time t. The transmission rate β and the recovery rate r are supposed positive [See table 1.4]. We get the system:

$$\frac{dS}{dt} = -\beta SI + D_S S_{xx} \tag{1.5}$$

$$\frac{dI}{dt} = \beta SI - rI + D_R I_{xx} \tag{1.6}$$

$$\frac{dR}{dt} = rI + D_R R_{xx} \tag{1.7}$$

We assume homogeneous Neumann boundary conditions representing a closed environment where individuals are not allowed to leave or enter the studied environment:

$$S_x(t,0) = S_x(t,L) = I_x(t,0) = I_x(t,L) = R_x(t,0) = R_x(t,L) = 0$$

with the initial conditions:

$$\begin{cases} S(0) = S_0 > 0 \\ I(0) = I_0 > 0 \\ R(0) = 0 \end{cases} \tag{1.8}$$

Table 1.4: Units for the SIR PDE case

Variable	description	unit
β	Transmission rate	$\frac{1}{\text{people}\times\text{days}}$
t	Time	days
r	Recovery rate	$\frac{1}{\text{days}}$
$S(t,x)$	Number of susceptible people at time t and space x	people
$I(t,x)$	Number of infected people at time t and space x	people
$R(t,x)$	Number of recovered people at time t and space x	people
$N(t,x)$	The population at time t and space x	people
D_S, D_I, D_R	Diffusion rates	$\frac{people^2}{days}$

The spatial factor, x, can be spatially discrete or spatially continuous. In either case,

the spatial factor is used to describe the mobility of the population. This mobility can be due to travel and migration, and it could be between cities, towns or even countries, depending on the studied case.

1.5 TRANSMISSION RATE

An infectious disease is transmitted from some source to some susceptible individual. The rate at which susceptibles become infected is called the transmission rate. The transmission rate of many infectious diseases varies in time. A good example of this variation is the transmission rate of influenza, which is thought to vary seasonally due to changing seasonal humidity. A disease transmission from infected individuals to susceptible individuals defines the dynamic of an infectious disease. Transmission rate is defined as the product of the total contact rate by the probability of transmission occurring. The transmission rate is impossible to measure for most infections, but we need to know this rate in order to study the changes in a population due to an infectious disease.

One of the principal challenges in epidemiological modeling is to parameterize models with realistic estimates for all parameters including the transmission rates. More accurate parameter values will lead to more realistic models which allow for better analysis of strategies for control and predictions of disease outcomes. The recovery rate can be measured in a laboratory, for example if people, on average, stay sick for two days the recovery rate is $r = \frac{1}{2}$. But the transmission rate is hard to predict.

During a disease outbreak we could count people getting infected such as those in hospitals, we can assume that we know $I(t)$. In this paper we assume that the number of infected $I(t, x)$ is known and we want to find the spatiotime-dependent transmission rate function $\beta(x, t)$.

1.6 LATENT PERIOD

For some diseases, when a person becomes infected, he does not become infectious until the virus or bacteri develops and gets stronger. In this case, the infected will be a disease holder who cannot transmit it to any other person until the infection matures. The period that starts when the person becomes infected, until the person becomes infectious is called the latent period.

If there is a latent period then we should divide our population into four classes:
Susceptible class contains individuals who have no immunity to the infectious agent, so might become infected if exposed. Exposed or Latent class contains individuals who are infected but cannot infect other individuals. Infectious class contains individuals who are currently infected and can transmit the infection to susceptible individuals who they contact. Recovered class contains individuals who returned to a normal state of health after being infected. The model is called the $SEIR$ model.
In our case we will assume that the individual becomes infectious as soon as he get infected, the SIR model.

1.7 DESCRIPTION OF THE SITUATION

Consider an infectious disease with no latent period (no temporal delay in infection) and no available vaccination. The total population is divided into three classes: susceptible class "S", infected class "I", and removed class "R." Assume that the infected class has full immunity after recovery which means that as soon as an infected individual recovers from the infection he will be immune, he will not be susceptible or infected again. Therefore this individual will move from the infectious class, I, to the recovered class, R.
Assume that we know the number of births, also, we assume that we know the death rate. Also we will assume that newborns are not protected from the disease and they could become infected; they are susceptibles.
This system describes an endemic i.e. the disease is present in the population for a long period of time where the susceptible class, S, includes new income from births.

We have an SIR disease model in a spatially continuous environment. We consider the one dimensional space $\Omega = [-1, 1]$.

Consider the notations:

Let $x \in \Omega$.

$S = S(t, x)$, the number of susceptible people at time t and in position x,

$I = I(t, x)$, the number of infected people at time t and in position x,

$R = R(t, x)$, the number of recovered people at time t and in position x.

$N(t, x)$ the total number of people at time t and in position x.

$S(t) = \int_\Omega S(t, x)\ dx$, number of susceptible people at time t,

$I(t) = \int_\Omega I(t, x)\ dx$, the number of infected people at time t,

$R(t) = \int_\Omega R(t, x)\ dx$, the number of recovered people at time t,

$N(t) = \int_\Omega S(t, x) + I(t, x) + R(t, x)\ dx$, the number of people at time t.

The partial derivative of the population $N(t, x)$ with respect to time, t, will be given by

$$\frac{\partial N(t,x)}{\partial t} = B(t,x) + D_S \frac{\partial^2 S(t,x)}{\partial x^2} + D_I \frac{\partial^2 I(t,x)}{\partial x^2} + D_R \frac{\partial^2 R(t,x)}{\partial x^2} - dN(t,x) - \gamma I(t,x) \quad (1.9)$$

where $N(t, x)$ is the population at time t in space x,

$B(t, x)$ is the number of new born in space x at time t,

D_S is the diffusion rate for the susceptible class, which is constant for all susceptible people,

D_I is the diffusion rate for the infected class, which is constant for all infected people,

D_R is the diffusion rate for the recovered class, which is constant for all recovered people,

d is the death rate,

γ is the extra death rate due to the disease.

1.8 SIR MODEL IN THIS SITUATION

Figure 1.3 describes our situation.

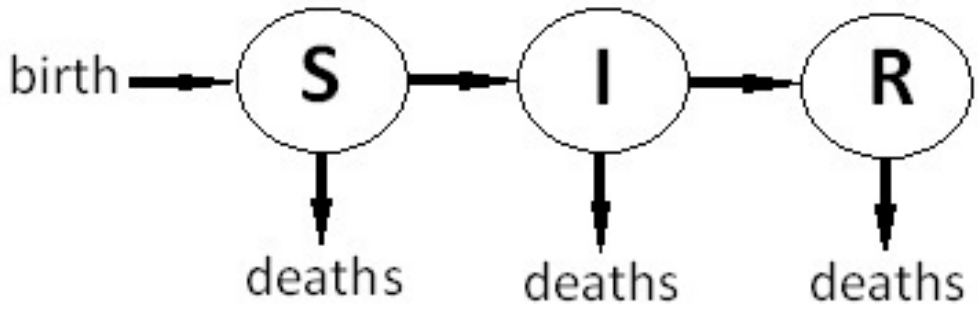


Figure 1.3: SIR-model figure with population demographics

The assumptions lead us to a set of partial differential equations.

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{1.10}$$

$$\frac{\partial I(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma)I(t,x) - rI(t,x) + D_I I_{xx}(t,x) \tag{1.11}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) - dR(t,x) + D_R R_{xx}(t,x) \tag{1.12}$$

We will consider a closed environment. All individuals are not allowed to escape or leave out of the studied environment; i.e, for this model we have homogeneous Neumann boundary conditions:

$$S_x(t,0) = S_x(t,1) = I_x(t,0) = I_x(t,1) = R_x(t,0) = R_x(t,1) = 0$$

The expected duration of the disease is $\frac{1}{r}$ time units . The parameters d, r, γ are greater or equal to zero.

1.9 NUMERICAL SIMULATION - SOLUTION TO THE SIR-PDE

The SIR-PDE system has three time-dependent partial differential equations. In this section we will assume that the transmission rate β is known and show that the SIR-PDE system can be solved. And as we know, we can solve time-dependent partial differential equations numerically using different methods. In this paper we will use a Runge-Kutta

Table 1.5: Units for the SIR PDE model with demographics

Variable	description	unit
β	Transmission rate	$\frac{1}{\text{people}\times\text{days}}$
γ	Extra death rate due to the infection	$\frac{1}{\text{days}}$
t	Time	Days
r	Recovery rate	$\frac{1}{\text{days}}$
$S(t,x)$	Number of susceptible people at time t and space x	people
$I(t,x)$	Number of infected people at time t and space x	people
$R(t,x)$	Number of recovered people at time t and space x	people
$B(t,x)$	Number of birth at time t and space x	$\frac{\text{people}}{\text{days}}$
$N(t,x)$	The population at time t and space x	people
D_S, D_I, D_R	Diffusion rates	$\frac{\text{location}^2}{\text{days}}$

method to solve our SIR-PDE system. The space derivatives of the system are discretized with a finite difference method and then the resulting system of ODEs is advanced in time with a fourth order Runge-Kutta method.

We will consider different situations and show the graphical solution in each situation. In all the situations we will assume that:

1. All the diffusion rates are equal and constant($D_S = D_I = D_R = 0.01$).
2. The recovery rate is constant(r=0.3).
3. Initially, the number of infected people in each space is one($I(0,x) = 1 \; for \; all \; x \in \Omega$).
4. We will consider the time interval $[0, 5]$.
5. Initially the population distribution has a sine shape.
6. The death rate is constant $d = 0.001$.
7. The extra death rate due to the infection is constant.

(For more information about the code refer to A.1)

1.9.1 SITUATION 1

Here we consider a disease that does not cause death. Then the extra death rate due to the infection, γ, is equal to zero.

Figure 1.4(a) shows the initial condition in this situation. Figures 1.4(b), 1.4(c) and 1.4(d) refer respectively to $t = 1,\ 3,\ 4.5$.

1.9.2 SITUATION 2

Here we consider a disease that does cause death. Then the extra death rate due to the infection, γ, is not equal to zero. We will assume that γ is constant, $\gamma = 0.3$.
Figure 1.5(a) shows the initial condition in this situation. Figures 1.5(b), 1.5(c) and 1.5(d) refer respectively to $t = 1,\ 3,\ 4.5$.

Note

1. The effect of the diffusion is clear; the population distribution is changing from the sine shape (Figures 1.4(a) and 1.5(a)) to be more straight (Figures 1.4(d) and 1.5(d)).

2. The number of susceptible people is decreasing, while the number of recovered and infected people is increasing.

1.10 $SEIR$ MODEL IN THIS SITUATION

As we can see from its name, the SEIR model contains one more compartment, the exposed compartment 'E'. An SEIR model is a general case of SIR model. For some diseases, the virus needs some time ($\frac{1}{\nu}$, where ν is the rate at which people moves from exposed to infectious class) to develop and become stronger. During this time, the infected person is not infectious, this why we separate those two groups.

We make the following assumptions

1. The SEIR model represents infection dynamics in a population of size N and includes the spatial position, with a natural death rate and a birth rate.

2. $N(t, x) = S(t, x) + E(t, x) + I(t, x) + R(t, x)$, not constant.

3. The infection may cause mortality. We have an extra death rate due to the disease.

4. Individuals are moved directly into the susceptible class at birth.

5. We have homogeneous mixing, i.e., individuals of the population make contact at random and do not mix mostly in a smaller subgroup.

6. The model assumes that recovered individuals are immune from infection.

7. Infected individuals move into the Exposed class and after an average latent period $(\frac{1}{\nu})$, if they escape mortality, they move to the infectious class.

8. We assume that spatial mobility is governed by random diffusion coefficients, D_S, D_E, D_I and D_R for the susceptible, exposed, infectious and recovered classes respectively. The spatial position is a bounded one-dimensional environment, $[0, 1]$

9. We are going to consider that the disease is new or simply no vaccination is available.

All individuals are not allowed to escape or leave the studied environment, i.e. for this model we have homogeneous Neumann boundary conditions representing a closed environment:
$S_x(t,0) = S_x(t,1) = E_x(t,0) = E_x(t,1) = I_x(t,0) = I_x(t,1) = R_x(t,0) = R_x(t,1) = 0$
with the initial conditions:

$$\begin{cases} S(0) = S_0 > 0 \\ E(0) = 0 \\ I(0) = I_0 > 0 \\ R(0) = 0 \end{cases} \tag{1.13}$$

Under the assumptions we get:

Table 1.6: Units

Variable	description	unit
β	Transmission rate	$\frac{1}{people \times days}$
γ	Extra death rate due to the infection	$\frac{1}{days}$
ν	Rate at which people moves from exposed to infectious class	$\frac{1}{days}$
t	Time	Days
r	Recovery rate	$\frac{1}{days}$
$S(t,x)$	Number of susceptible people at time t and space x	people
$E(t,x)$	Number of exposed people at time t and space x	people
$I(t,x)$	Number of infectious people at time t and space x	people
$R(t,x)$	Number of recovered people at time t and space x	people
$N(t,x)$	The population at time t and space x	people
D_S, D_E, D_I, D_R	Diffusion rates	$\frac{location^2}{days}$

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{1.14}$$

$$\frac{\partial E(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma)E(t,x) - \nu E(t,x) + D_E E_{xx}(t,x) \tag{1.15}$$

$$\frac{\partial I(t,x)}{\partial t} = -dI(t,x) - (r+\gamma)I(t,x) + D_I I_{xx}(t,x) + \nu E(t,x) \tag{1.16}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) - dR(t,x) + D_R R_{xx}(t,x) \tag{1.17}$$

Figure 1.6 represents an $SEIR$ model.

In this paper we will concentrate on SIR model.

1.11 FORWARD PROBLEM

The forward problem is the mathematical process of predicting data based on some mathematical model with a given set of model parameters. In general , we have the following equation

$$Ax = y, \tag{1.18}$$

where A is an operator between two spaces X and Y, $A : X \to Y$ such that $x \in X$ and $y \in Y$

'x' is the set of known model parameters,

'y' the estimated data,

the operator 'A' represents the model.

The forward problem is the estimation or computation of 'y' given the model parameter 'x' and the model 'A' in the equation $Ax = y$

Figure 1.7 describes the forward problem.

Note

In the SIR-PDE model case, we have the forward problem of knowing all the parameter values ($\beta,\ d,\ \gamma,\ B,\ D_S, \ldots$) and using that to get the values of S, I and R.

1.12 HADAMARD'S PRINCIPLE

Given the mapping $A : X \longrightarrow Y$, with X and Y two given spaces, the equation $Ax = y$ is said to be well-posed if:

1. A solution exists, for each $y \in Y, \exists\ x \in X$, such that $Ax = y$.

2. The solution is unique; '$Ax_1 = Ax_2 \Longrightarrow x_1 = x_2$ '.

3. The solution is stable, which means that $x \in X$ depends continuously on $y \in Y$.

These conditions are known as Hadamard principles.

An equation is ill-posed if it is not well-posed, i.e, "one at least of the conditions is not satisfied." In the past, a problem that failed to satisfy Hadamard principles was classified by mathematicians under the title of "not correctly set" problem. With time progress, it was clear that some important problems failed at least one of Hadamard principles like the Cauchy problem for the heat equation in reversed time.

1.13 DEFINITIONS

In this section we have some definitions of some mathematical terms that we will use in this paper, especially in chapter three.

Definition 1.13.1. Let X be a vector space over the $\Re$. X is a normed space if there exist a function $\|.\|$:$X \longrightarrow \Re$ called norm, and satisfy the following:

1. $\|f\| \geq 0$, $\|f\| = 0$ if and only if $f = 0$.(positive definite)
2. $\|\alpha f\| = |\alpha|\|f\|$ for any real α.(homogeneity)
3. $\|f + g\| \leq \|f\| + \|g\|$. (the triangle inequality)

Definition 1.13.2. Let X be a vector space over the field $\Re$. The inner product is a mapping $< ., . >$: $X \times X \longrightarrow \Re$ with the following properties:

1. $< x + y, z >=< x, z > + < y, z >$ for all $x, y, z \in X$
2. $< \alpha x, y >= \alpha < x, y >$ for all $x, y \in X$ and $\alpha \in \Re$
3. $< x, y >=< y, x >$ for all $x, y \in X$
4. $< x, x >\in \Re$ and $< x, x >\geq 0$ for all $x \in X$
5. $< x, y + z >=< x, y > + < x, z >$ for all $x, y, z \in X$
6. $< x, \alpha y >= \alpha < x, y >$ for all $x, y \in X$ and $\alpha \in \Re$

Definition 1.13.3. A vector space X over $\Re$ with inner product $< ., . >$ is called a pre-Hilbert space over $\Re$.

Definition 1.13.4. A normed space X over $\Re$ is called complete or a Banach space if every Cauchy sequence converges in X.

Definition 1.13.5. A is a compact operator if it is a linear operator from a Banach space X to a Banach space Y such that the image under A for any bounded subset of X is a relatively compact set (its closure is compact).

Definition 1.13.6. A complete pre-Hilbert space is called a Hilbert space.

Definition 1.13.7. A sequence of vectors in an inner product space X is called weakly convergent to a vector x in X, $x_n \rightharpoonup x$, if: $< x_n, y >\longrightarrow< x, y >$ for all $y \in X$

Definition 1.13.8. $F : X \longrightarrow Y$ is weakly sequentially closed if for a sequence $x_n \in D(F)$ the domain of F, $x_n \rightharpoonup x$ (weak convergence) and $F(x_n) \rightharpoonup y$ in Y imply $x \in D(F)$ and

$F(x) = y$

Definition 1.13.9. Let V and W be Banach spaces, and $U \subset V$ be an open subset of V. A function $f : U \to W$ is called Fréchet differentiable at $x \in U$ if there exists a bounded linear operator $A_x : V \to W$ such that

$$\lim_{h \to 0} \frac{\|f(x+h) - f(x) + A_x(h)\|_W}{\|h\|_V} = 0$$

If the limit exists, we write $Df(x) = A_x$ and call it the Fréchet derivative of f at x.

Definition 1.13.10. Suppose H is a Hilbert space, with inner product $< ., . >$. Consider a continuous linear operator $A : H \to H$. There exists a unique continuous linear operator $A^* : H \to H$ with the following property:

$$< Ax, y >=< x, A^*y >$$

for all $x, y \in H$. This operator A^* is the adjoint of A.

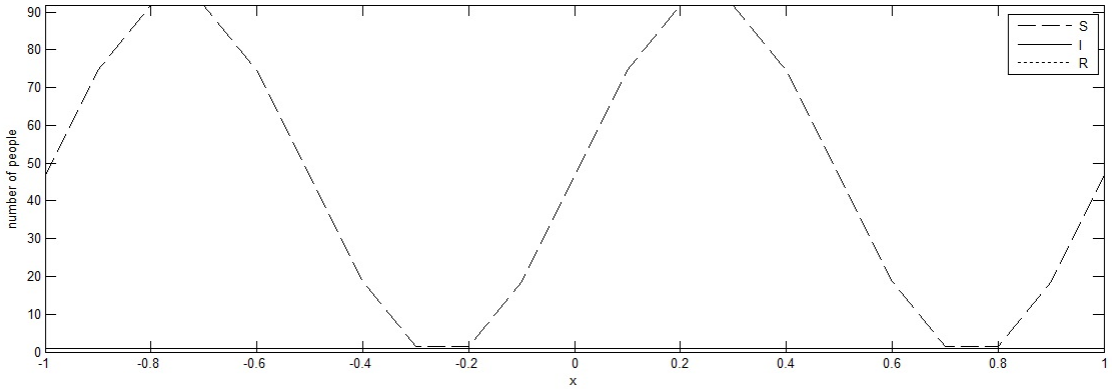
1.14 PLAN

The transmission rate $\beta(t, x)$ is an important factor in the study of the spread of any disease. In this paper we want to recover the transmission rate $\beta(t, x)$ using different method.

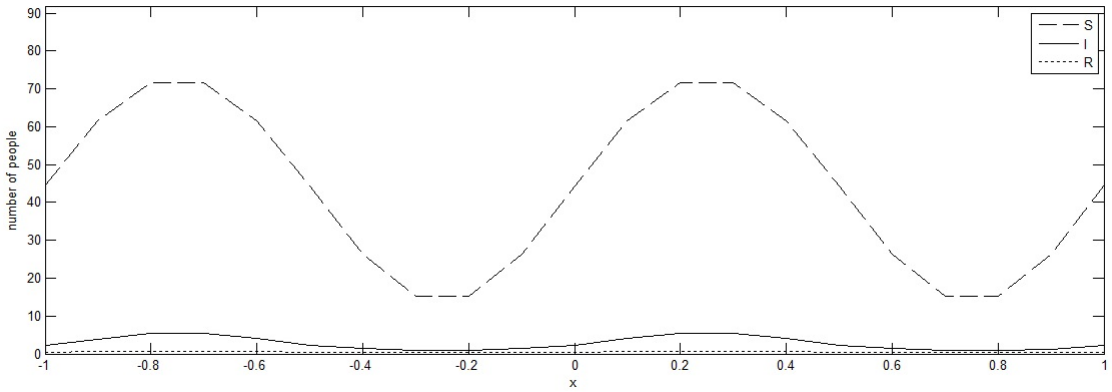
In Chapter 2 we will review the transmission rate recovery procedure for the ordinary differential equation case and try to identify the transmission rate in the partial differential equations system using similar procedures.

In Chapter 3 we will discuss the inverse problem procedure and the Tikhonov regularization and apply it to our system in order to find an approximation of the transmission rate $\beta(t, x)$.

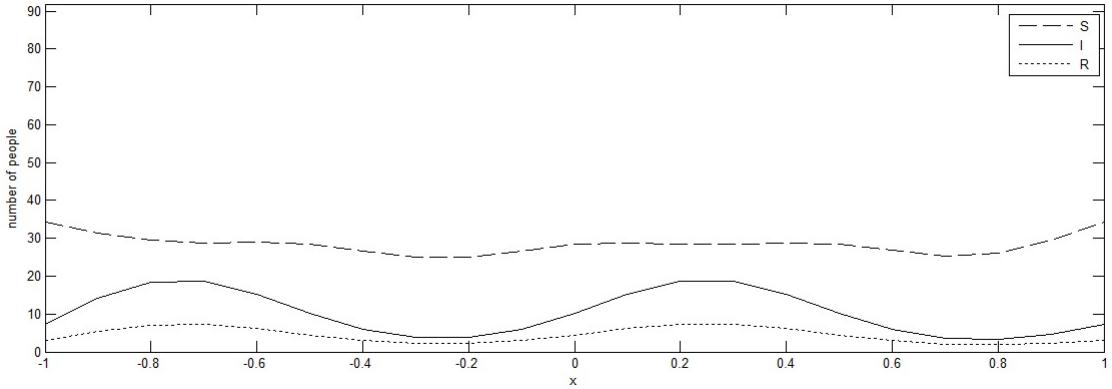
In Chapter 4 we will introduce the Optimal Control Theory and apply it to our system in order to find some numerical approximation of the transmission rate $\beta(t, x)$.



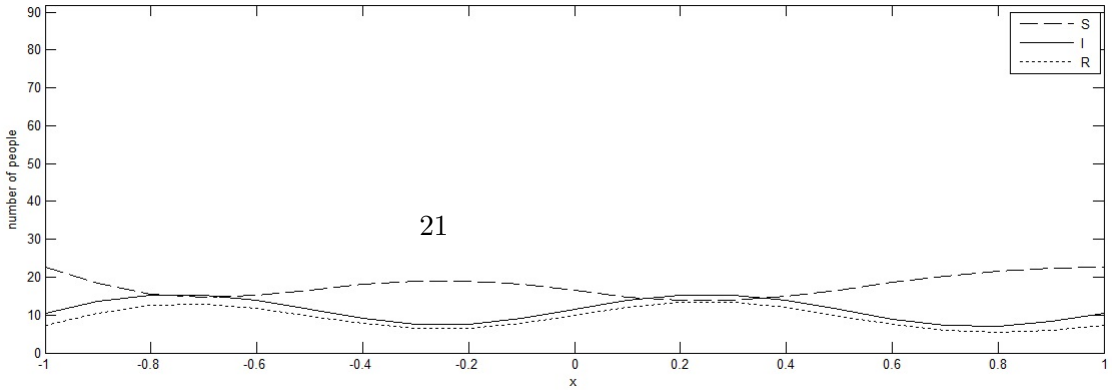
(a) Initial condition when $\gamma = 0.3$

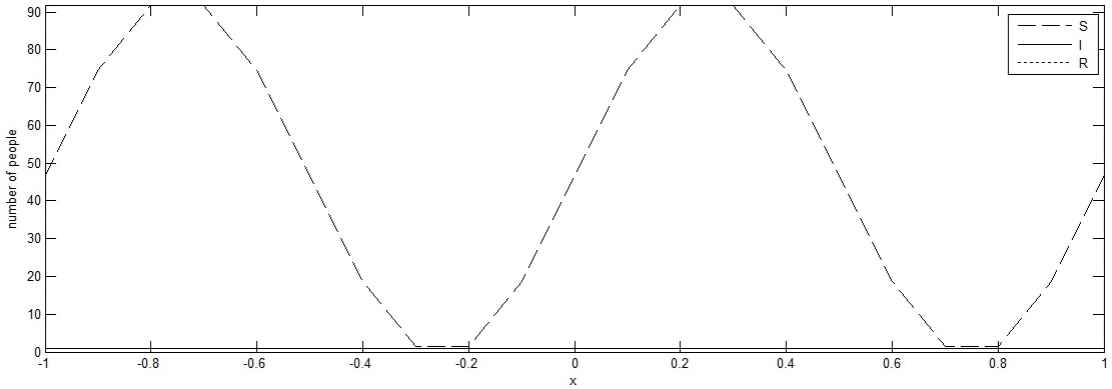


(b) $t = 1$ when $\gamma = 0.3$

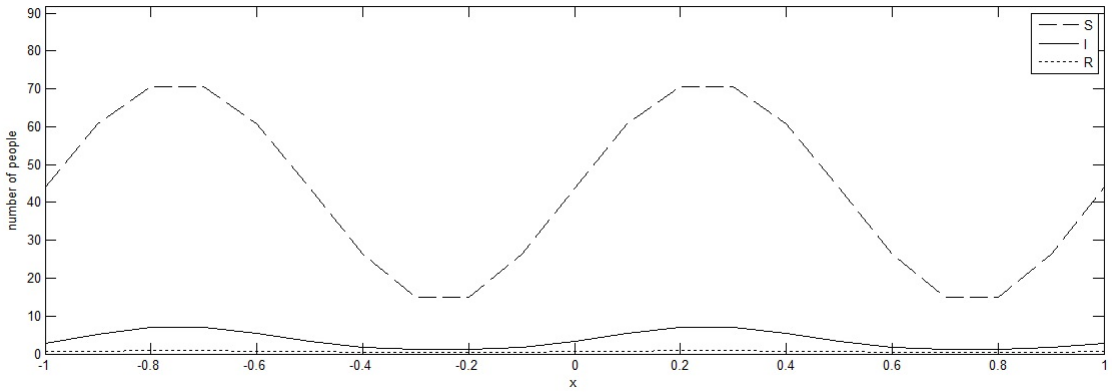


(c) $t = 3$ when $\gamma = 0.3$

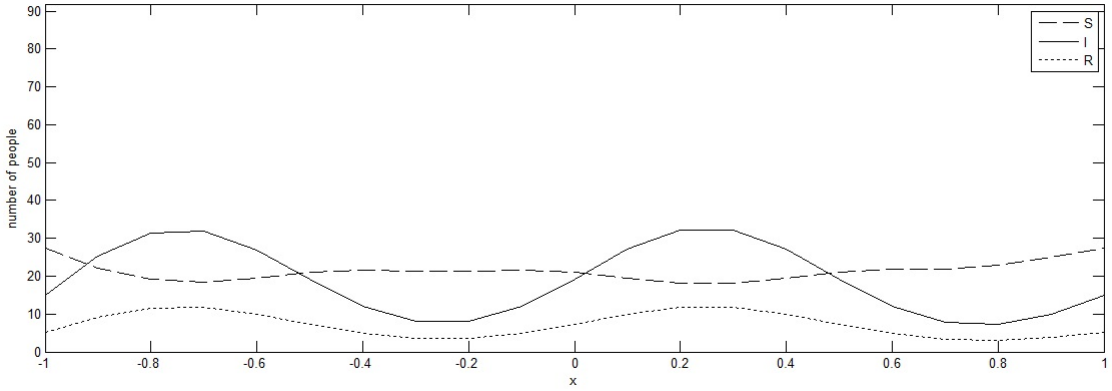




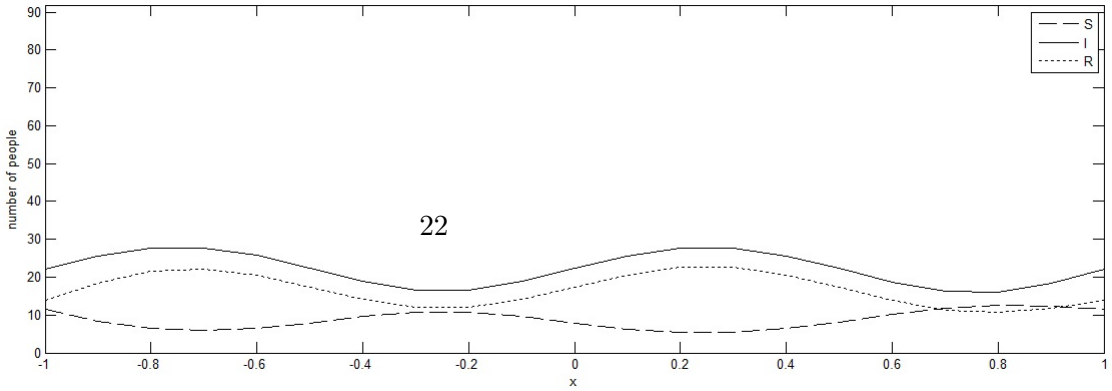
(a) Initial condition when $\gamma = 0.3$



(b) $t = 1$ when $\gamma = 0.3$



(c) $t = 3$ when $\gamma = 0.3$



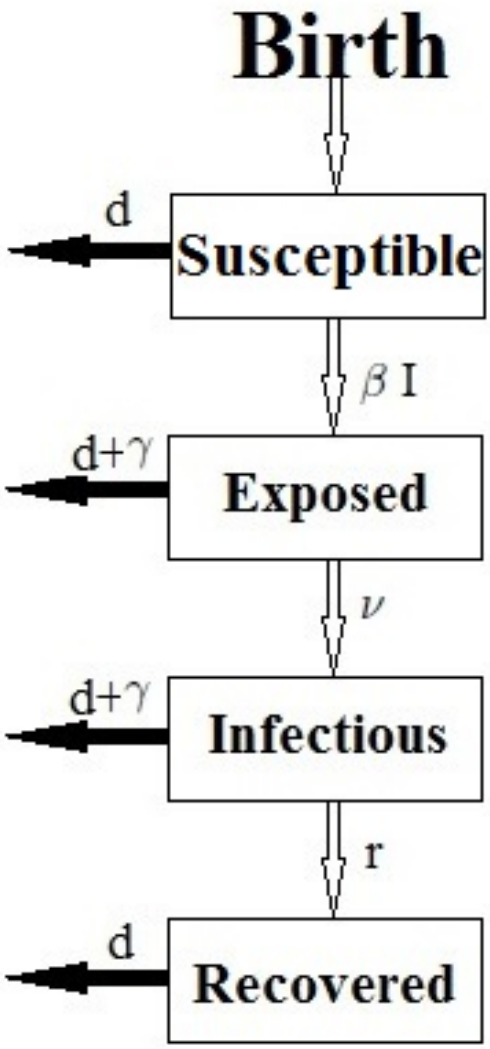


Figure 1.6: $SEIR$-model figure

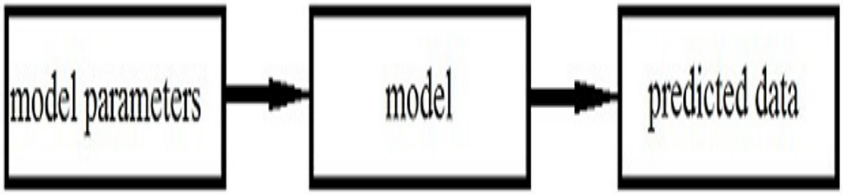


Figure 1.7: Forward Problem

Chapter 2

DIRECT TRANSMISSION RATE IDENTIFICATION PROCEDURES

The transmission rate parameter depends on the total contact rate and the probability of transmission occurring during a contact. The transmission rate is impossible to measure due to the challenges of measuring both the total contact rate and the probability of transmission, but we need to know this parameter in order to effectively study infectious diseases.

The transmission rate of infectious diseases vary in time, for example, due to seasonal contact rates among school children. It is also true that this rate may vary from one place to another. For example, it may change due to different humidities in different places. In this study, we assume that the transmission rate varies in both time and space and we want to identify the rate given infection data.

We will start this chapter by discussing some time-dependent procedures for recovering the transmission rate for the classical ordinary differential equation SIR model. Then we extend these methods to the SIR partial differential equation system.

2.1 TIME DEPENDENT TRANSMISSION RATE FUNCTION

Starting with the classical ordinary differential equation SIR model and assuming the transmission rate varies in time, Pollicot, Wang, and Weiss [14] describe a procedure to extract the time dependent transmission rate, $\beta(t)$, given information on the number of

infected people $I(t)$.

2.1.1 EXTRACTING THE TIME DEPENDENT TRANSMISSION RATE

Pollicott, Wang, and Weiss [14] used Kermack and McKendrick [8] SIR model (mentioned in Section 1.3) and allowed the transmission rate to be a time-dependent function $\beta(t)$ and derived a procedure to recover this $\beta(t)$ (see also [4, 12]). Data on the numbers infected by many diseases are collected by the United States Centers for Disease Control and Prevention, 'CDC'. Thus, the authors considered $I(t)$ as known. The extraction procedure in the ODE case begins with finding an expression for $S(t)$ using the differential equation for $I(t)$ [1.2]. Then take the derivative of this equation and substitute those expressions in the differential equation for $S(t)$. The resulting equation is a non linear Bernoulli differential equation for β. Use the change of variable $\frac{1}{\beta(t)}$ to get a linear Bernoulli differential equation and solve it using the method of integrating factors to get an expression for $\beta(t)$.

The $\beta(t)$ recovery algorithm has four steps and requires two conditions

Step1

Generate a smooth function $f(t)$ using the infection data.

Check Condition 1

$$\frac{f'(t)}{f(t)} > -r,$$

where r is the recovery rate.

Step2

Compute

$$p(t) = \frac{f''(t)f(t) - f'(t)^2}{f(t)(f'(t) + rf(t))}$$

where f' refers to $\frac{df}{dt}$ and f'' refers to $\frac{d^2f}{dt^2}$.

Condition 1 guarantees that the denominator in $p(t)$ is not equal to zero,and therefore $\beta(t)$ is well defined.

Step3

Compute $P(t) = \int_0^t p(\tau)d\tau$

Check Condition 2:

$$\beta(0) < \frac{1}{\int_0^T e^{p(s)} f(s) ds},$$

where T is the time length of the infection data.

A priori $\beta(0)$ is unkown, so choose $\beta(0)$ sufficiently small to satisfy condition 2.

Step4

Apply the formula

$$\beta(t) = \frac{1}{\frac{e^{-P(t)}}{\beta(0)} - e^{-P(t)} \int_0^t \ e^{P(s)} f(s) \ ds}$$

to compute $\beta(t)$ on the given interval $[0, T]$. Condition 2 guarantees that $\beta(t)$ is greater than zero because it does not make any biological sense to have a negative transmission rate.

A second method for extracting the transmission rate is provided by Hadeler [4]. Hadeler's method leads to a formula for $\beta(t)$ that is equivalent to the formula that Pollicott, Wang, and Weiss [4] described . Hadeler also used Kermack and McKendrick [8] SIR system mentioned in Section 1.3 with normalized total population size as $S(t) + I(t) + R(t) = 1$.

This second extraction method begins with adding the first two equations to get a differential equation for $S + I$, integrating both sides gives an equation for $S(t) + I(t)$ in which all but $S(t)$ is assumed to be known, then solving for $S(t)$. In the $I'(t)$ equation of the system, solve for $S(t)$. Equate the two expressions of $S(t)$ and solve for $\beta(t)$. The following simplified expression for $\beta(t)$ is:

$$\beta(t) = \frac{\frac{I'(t)}{I(t)} + r}{S(0) + I(0) - I(t) - r \int_0^t I(s) ds} \tag{2.1}$$

where I' refers to $\frac{dI}{dt}$.

2.1.2 EXTENDING THE RECOVERY PROCEDURE TO OTHER RESULTS

In their paper, Pollicott, Wang, and Weiss [14] showed that analogous results and inversion formulae hold for some extended models. One of those extended models is the SIR

model with demographic rates.

$$\frac{dS(t)}{dt} = d - \beta S(t)I(t) - dS \tag{2.2}$$
$$\frac{dI(t)}{dt} = \beta S(t)I(t) - rI(t) - dI \tag{2.3}$$
$$\frac{dR(t)}{dt} = rI(t) - dR \tag{2.4}$$

where d is the birth and death rate.
The necessary and sufficient condition for recovering $\beta(t)$ given d and r is $\frac{f'(t)}{f(t)} > -(d+r)$

Also, in his paper, Hadeler [4] shows that the parameter estimation method works also for the model with demographic replacement (when the birth rate is not equal to the death rate).

In 2012 Mummert [12] extend the recovery procedures to heterogeneous populations with two population classes. Extending the recovery procedure to more than two populations can be done in a similar way (see also Hadeler [7]).
For two population classes the model has six equations:

$$\frac{dS_1(t)}{dt} = -\beta_{11}S_1(t)I_1(t) - \beta_{12}S_1(t)I_2(t) \tag{2.5}$$
$$\frac{dI_1(t)}{dt} = \beta_{11}S_1(t)I_1(t) + \beta_{12}S_1(t)I_2(t) - \gamma_1 I_1(t) \tag{2.6}$$
$$\frac{dR_1(t)}{dt} = \gamma_1 I_1(t) \tag{2.7}$$
$$\frac{dS_2(t)}{dt} = -\beta_{21}S_1(t)I_1(t) - \beta_{22}S_1(t)I_2(t) \tag{2.8}$$
$$\frac{dI_2(t)}{dt} = \beta_{21}S_1(t)I_1(t) + \beta_{22}S_1(t)I_2(t) - \gamma_2 I_2(t) \tag{2.9}$$
$$\frac{dR_2(t)}{dt} = \gamma_2 I_2(t) \tag{2.10}$$

The model has four transmission rates: $\beta_{11}(t)$ the time dependent transmission rate function within the first class, $\beta_{22}(t)$ the time dependent transmission rate function within

the second class, $\beta_{12}(t)$ the time dependent transmission rate function between the first and the second classes, $\beta_{21}(t)$ the time dependent transmission rate function between the second and the first classes.

Mummert solved for the transmission rates in two different cases. The first case considers the situation when the transmission rate within the first class and the transmission rate between the first and the second classes are equal to the same time dependent transmission rate $\beta_1(t) = \beta_{11}(t) = \beta_{12}(t) = \beta_{21}(t)$, but those are different from the transmission rate within the second class $\beta_2(t) = \beta_{22}$. The second case considers when the transmission rate within the first class and the transmission rate between the first and the second classes are equal, $\beta_{11} = \beta_{12}$, and the transmission rate between the second and the first classes and the transmission rate within the second class are equals $\beta_{21} = \beta_{22}(t)$. Mummert got the solutions in those two cases and found that in order to have those solutions we have four conditions:

Condition 1:

$$\frac{dI_1(t)}{dt} + \gamma_1 I(t) > 0, \text{ for all } t.$$

Condition 2:

$$\frac{dI_2(t)}{dt} + \gamma_2 I(t) > 0, \text{ for all } t.$$

Condition 3:

$$S_1(0) - \int_0^t g_1(s)ds > 0$$

for all t where

$$g_1(t) = \frac{dI_1(t)}{dt} + \gamma_1 I_1(t)$$

Condition 4:

$$S_2(0) - \int_0^t g_2(s)ds > 0$$

for all t where

$$g_2(t) = \frac{dI_2(t)}{dt} + \gamma_2 I_2(t)$$

2.2 TRANSMISSION RATE IN SPATIAL-TIME SYSTEM

People travel by foot, road or air between cities and even rural areas, so diseases can be spread between distant places. It is obvious that disease parameters may vary spatially. We want to study a model that includes the diffusion of people in one dimension and extract the transmission rate in this case.

In this model we will assume that $I(t,x)$ is known for all $x \in \Omega$, and for all t in the time interval $[0,1]$. We also assume that the prevalence data function $I(t,x)$ is sufficiently differentiable. We will follow the procedure described by Pollicott, Wang, and Weiss [14]. We will use the fact that $I(t,x)$ is known to solve for $S(t,x)$. Then substitute this expression and its derivatives into the equation of $S(t,x)$. The result will be a partial differential equation for the function β. Solving this equation gives the transmission function. Recall, we are working with the following system of partial differential equations:

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{2.11}$$

$$\frac{\partial I(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma+r)I(t,x) + D_I I_{xx}(t,x) \tag{2.12}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) - dR(t,x) + D_R R_{xx}(t,x) \tag{2.13}$$

Theorem For the partial differential equations SIR model given by Equations 2.11 - 2.13, there exists $t_* > 0$, such that, for all $t \in [0, t_*)$ the transmission function satisfies the following equation:

$$(D_S g)z_{xx} + (2D_S g_x)z_x - gz_t + (D_S g_{xx} - dg - g_t)z + B - g = 0 \tag{2.14}$$

where $z = \dfrac{1}{\beta I}$ and $g = g(t,x) = I_t + (d+\gamma+r)I - D_I I_{xx}$
if and only if

$$I_t + (d+\gamma+r)I - D_I I_{xx} > 0. \tag{2.15}$$

Proof.

Because $\beta(0,x) > 0$ we know that there exists a $t_* > 0$ such that $\beta(t,x) > 0$ for $0 \leq t < t_*$ and $\beta(t_*,x) = 0$. The transmission rate determined by solving 2.20 will be valid on the interval $[0,t_*)$. The rate will fail to be the desired rate for $t \geq t_*$. Which corresponds to Condition 2 in classical ODE SIR model.

Solve Equation 2.12 for S to find that

$$S(t,x) = \frac{I_t + (d+\gamma+r)I - D_I I_{xx}}{\beta I}. \tag{2.16}$$

S represents the number of susceptible people, then $S(t,x)$ should be positive for all x, and all t.

Since $I(t,x) > 0$ for all x, and all t and same for β, then we need to have

$I_t + (d+\gamma+r)I - D_I I_{xx} > 0$ for all x, and all t, to ensure that $S(t,x)$ is positive for all x, and all t.

Set the numerator and denominator in 2.16 to be functions $g(t,x)$ and $h(t,x)$, respectively:

$$g = g(t,x) = I_t + (d+\gamma+r)I - D_I I_{xx} \tag{2.17}$$

$$h = h(t,x) = \beta I \tag{2.18}$$

Differentiate $S(t,x) = \dfrac{g(t,x)}{h(t,x)}$ with respect to t, and replace in 2.11 :

$$\frac{\partial S(t)}{\partial t} = \frac{\partial}{\partial t}\left(\frac{g(t,x)}{h(t,x)}\right) = \frac{g_t h - h_t g}{h^2}.$$

Replace also βSI by $g(t,x)$ since

$$S(t,x) = \frac{g}{h} = \frac{g}{\beta(t,x)I(t,x)}$$

Implies

$$g = \beta(t,x)S(t,x)I(t,x).$$

Then we get

$$\frac{g_t h - h_t g}{h^2} = B(t,x) - d\frac{g}{h} - g + D_S(\frac{g}{h})_{xx}.$$

Simplify to find that

$$g_t \frac{1}{h} - g\frac{h_t}{h^2} = B(t,x) - dg\frac{1}{h} - g + D_S(\frac{g}{h})_{xx}.$$

Making the substitution $z(t,x) = \dfrac{1}{h(t,x)}$, yields the equation

$$g_t z + g z_t = B(t,x) - dgz - g + D_S(gz)_{xx}.$$

Differentiate (gz) with respect to x twice and simplify to find that β satisfies the partial differential equation

$$(D_S g)z_{xx} + (2D_S g_x)z_x - gz_t + (D_S g_{xx} - dg - g_t)z + B(t,x) - g = 0 \tag{2.19}$$

or

$$z_t = \frac{(D_S g)z_{xx} + (2D_S g_x)z_x + (D_S g_{xx} - dg - g_t)z + B(t,x) - g}{g} \tag{2.20}$$

Note

2.15 corresponds to Condition 1 in classical ODE SIR model.

We assume that we know $\beta(0,x) = \beta_0 > 0$, and since we know I, we know I_0, then we know z_0.

2.3 CONSTANT POPULATION SIZE

Call $N(t)$ the total population in the universe Ω at time t. The position of people in the universe does not affect the the total population in the universe. As we know

$$N(t,x) = S(t,x) + I(t,x) + R(t,x).$$

The population size at time t is given by

$$N(t) = \int_{\Omega} N(t,x)dx. \tag{2.21}$$

Now let us suppose that the population at each $x \in \Omega$ is constant and does not change with time. Then

$$N(t,x) = S(t,x) + I(t,x) + R(t,x) = N(x)$$

is independent of time, and

$$N(t) = \int_{\Omega} N(t,x)dx = N,$$

with N a constant.

In this case we have

$$\frac{\partial N(t,x)}{\partial t} = \frac{\partial S(t,x)}{\partial t} + \frac{\partial I(t,x)}{\partial t} + \frac{\partial R(t,x)}{\partial t} = \frac{\partial N(x)}{\partial t} = 0. \tag{2.22}$$

Then $N(x) - (S(t,x) + I(t,x)) = R(t,x)$

In 2.13 replace $R(t,x)$ by $N(x) - (S(t,x) + I(t,x))$

$\frac{\partial(N(x) - (S(t,x) + I(t,x)))}{\partial t}$

$= rI(t,x) + D_R(N(x) - (S(t,x) + I(t,x)))_{xx} - d[N(x) - (S(t,x) + I(t,x))]$

After simplifying we get $\frac{\partial(S(t,x) + I(t,x)))}{\partial t}$

$= -rI(t,x) - D_R(N(x) - (S(t,x) + I(t,x)))_{xx} + d[N(x) - (S(t,x) + I(t,x))]$

Also we could get another expression for $\frac{\partial(S(t,x) + I(t,x))}{\partial t}$ by adding 2.11 and 2.12 together:

$$\frac{\partial(S(t,x) + I(t,x)))}{\partial t} = B(t,x) - dS(t,x) + D_S S_{xx}(t,x) - (d+\gamma+r)I(t,x) + D_I I_{xx}(t,x) \tag{2.23}$$

Combining the two expressions for $\frac{\partial(S+I)}{\partial t}$ yields:

$$-rI(t,x) - D_R(N(x) - (S(t,x) + I(t,x)))_{xx} + d[N(x) - (S(t,x) + I(t,x))]$$

$$= B(t,x) - dS(t,x) + D_S S_{xx}(t,x) - (d+\gamma+r)I(t,x) + D_I I_{xx}(t,x).$$

After simplification, we are left with

$$(D_S - D_R)S_{xx}(t,x) = (D_R - D_I)I_{xx}(t,x) + dN(x) - B(t,x) - D_R N_{xx}(x) + \gamma I(t,x), \quad (2.24)$$

which is equivalent to

$$S_{xx}(t,x) = \frac{(D_R - D_I)I_{xx}(t,x) + dN(x) - B(t,x) - D_R N_{xx}(x)}{D_S - D_R} \quad (2.25)$$

Which is true when $D_S \neq D_R$. This expression is known and can be computed because all the terms of the second side are known. Let $T(t,x) = S_{xx}(t,x)$
Replacing S_{xx} in equation 2.14 of the previous section by this value will leave us with

$$(g_t + dg)z + gz_t - B(t,x) + g - D_S T(t,x) = 0. \quad (2.26)$$

2.26 is an equation for z, and it is easy to solve since we just have z and z_t.

2.4 NOTE

1. The method of section 2.3 is similar to Hadeler's[4] method.
2. Finding β from the data is an example of inverse problem.
3. Finding β from the data in this way requires a lot of derivative, which leads us to a solution that is not exact.

2.4.1 SUMMARY

In this case, recovering the transmission rate, $\beta(t,x)$, requires a lot of derivatives of the data function. This is why we have to think about some other methods to recover the transmission rate, $\beta(t,x)$. Because recovering the transmission rate, $\beta(t,x)$, is an ill-posed inverse problem, we will discuss inverse problems in the next Chapter and apply Tikhonov regularization to recover the transmission rate, $\beta(t,x)$.

Chapter 3

INVERSE PROBLEM PROCEDURE

An inverse problem begins with the solution or observational data and attempts to estimate the parameters present in the model. Recently, a remarkable number of partial differential equation problems have been solved using the inverse problem procedure [6], taking advantage of the high speed processors allows computation of solutions to high scales inverse problems. We will use inverse problems procedure to solve for the transmission rate.

3.1 INVERSE PROBLEM

In the past, ill-posed problems were considered unnatural and impossible to be solved. Today such problems exist in many fields and technology allows us to solve them. We will consider the following mathematical model form:

$$Ax = y \tag{3.1}$$

with A is a bounded linear operator, given two Hilbert spaces X and Y, $A : X \to Y$ with $x \in X$ and $y \in Y$. The model is the mathematical relationship between model parameters and the data; it may be linear or nonlinear. The direct problem is to find the value of the measurable parameters (data), y, given the value of the model parameters, x. An inverse problem is to determine the model parameters having instead the data.

Figure 3.1 describes the Inverse Problem.

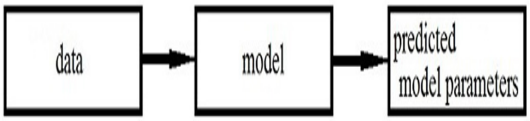


Figure 3.1: Inverse Problem

When the problem is well-posed, the operator A has a continuous inverse A^{-1} that we can find. And for $y \in Y$, we can find the unique $x \in X$ using the fact that $x = A^{-1}y$. Our aim is to approximate a solution 'x' given $y \in Y$ numerically when the problem is ill-posed, if A^{-1} does not exist or it is not continuous. In real life, x cannot take all possible values; x may be subject to some restriction such as $x \geq 0$, to ensure that the estimated value of x has a physical meaning and to ensure that the operator A is well defined [1, 6]. This why we consider the set $M \in X$. When the problem is ill-posed, and since we know 'y', we need to consider all the possible $x \in X$ that makes the problem well defined and has a physical meaning such that $Ax = y$; we will suppose that all the possible x are in M. We need to find $x_1^* \in M$ such that

$$\|Ax_1^* - y\|_Y = \inf_{x \in M} \|Ax - y\|_Y, \tag{3.2}$$

where $\|.\|_Y$ is the norm over the space Y.

Ill-posedness causes that a very small errors, such as those caused by rounding, can lead to a large deviations from the exact solution. To address this problem we need to use regularization.

3.1.1 NOISY DATA

Inverse problems are often unstable, i.e., a small error in one of the data values could be responsible for a big variation in the model. Therefore, in practice we need to worry about the fact that most of the time we do not have the exact data "y." Instead, usually we have some discrete noisy measurement y_δ such that $\|y - y_\delta\| \leq \delta$, where δ is called noise level.

In this paper we avoid this added complication and assume that the data are perfect.

3.2 TIKHONOV REGULARIZATION FOR LINEAR ILL-POSED PROBLEMS

In order to solve an ill-posed inverse problem in a stable manner, we need to construct a regularized solution. Tikhonov Regularization is probably the most successful regularization technique of all time. This technique is used to solve many problems in different fields of study [2, 10]. When the operator A is linear and the problem is ill-posed in the sense that the inverse operator A^{-1} of A exists but it is not continuous [13], we say that we have a linear ill-posed problem . In order to find a solution in stable manner, we will use Tikhonov regularization and instead of solving 3.2 we need to find

$$x_\alpha = \min_{x \in X} J_\alpha(x) = \min_{x \in X}(\|Ax - y_\delta\|_Y^2 + \alpha\|x - x^*\|_X) \tag{3.3}$$

where $\|.\|_X$ is the norm over the space X, x^* is our initial guess for the solution and α is the regularization parameter. In case we do not have any guess we consider $x^* = 0$.

It can be shown that for every positive parameter α there exists a unique $x_\alpha \in X$ for which the functional J_α attains its minimum [18].

3.3 TIKHONOV REGULARIZATION FOR NON-LINEAR ILL-POSED PROBLEMS

Nonlinear ill-posed inverse problems can be represented by the form

$$F(x) = y, \tag{3.4}$$

where F is a nonlinear compact and continuous operator that acts between two Hilbert spaces X and Y. The basic assumptions are that F is continuous and is weakly sequentially closed. In order to solve this problem we use Tikhonov regularization [15]. Tikhonov

regularization consists of finding a solution x_α^δ of the following minimization problem:

$$x_\alpha = arg \min_{x \in M} J_\alpha(x) \tag{3.5}$$

where

$$J_\alpha(x) = \|F(x) - y\|_Y^2 + \alpha \|x - x^*\|_X \tag{3.6}$$

with $x^* \in X$ an initial guess for a solution of 3.4.

The existence and uniqueness of a minimizer in the nonlinear case is not as clear as in the linear case. For a positive regularization parameter α, minimizers exist under the assumptions that F is weakly closed, continuous and Frechet differentiable with convex M, but need not to be unique due to the nonlinearity of the operator F. The element x^* is an initial guess of x_α and it plays the role of telling us which solution we want to find.

3.4 NUMERICAL SIMULATION

In this section we will be using Matlab to recover $\beta(t, x)$ numerically.

3.4.1 TIKHONOV REGULARIZATION

In order to use Tikhonov regularization we will use the 'RegTools' package (byPer Christian Hansen 16 Apr 1998(Updated 18 Mar 2008)). We will apply Tikhonov regularization in two ways: first we will recover S, βS and R, and get the transmission rate β by dividing βS by S. After, and to confirm the first result, we will add the first two equations in our SIR system, and recover S and use this S in the second equation, 1.11, to get β.

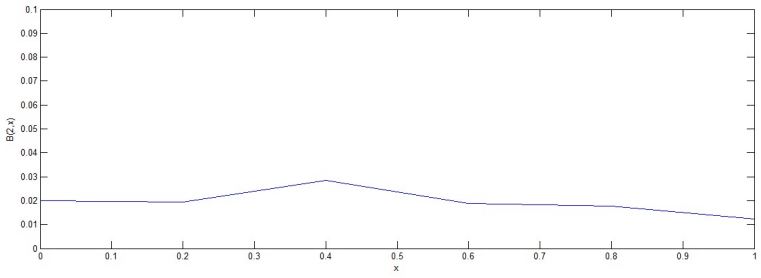


Figure 3.2: β for $t = 2$

RECOVERING S, βS AND R

Our system is

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{3.7}$$

$$\frac{\partial I(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma)I(t,x) - rI(t,x) + D_I I_{xx}(t,x) \tag{3.8}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) + D_R R_{xx}(t,x) - dR(t,x) \tag{3.9}$$

We will rewrite this system by isolating the terms that we consider that we know

$$B(t,x) = \frac{\partial S(t,x)}{\partial t} + dS(t,x) + \beta(t,x)S(t,x)I(t,x) - D_S S_{xx}(t,x) \tag{3.10}$$

$$g(t,x) = \beta(t,x)S(t,x)I(t,x) \tag{3.11}$$

$$rI(t,x) = \frac{\partial R(t,x)}{\partial t} - D_R R_{xx}(t,x) + dR(t,x) \tag{3.12}$$

with

$$g(t,x) = \frac{\partial I(t,x)}{\partial t} + (d+\gamma)I(t,x) + rI(t,x) - D_I I_{xx}(t,x)$$

By applying the Tikhonov regularization(for more information about the code, refer to A.2 to this system we will get $S, \beta S$ and R,

and by dividing βS by S we will get β.

For some created data, Figure 3.2 shows the result for $t = 2$, and Figure 3.3 shows the result for $t = 3$.

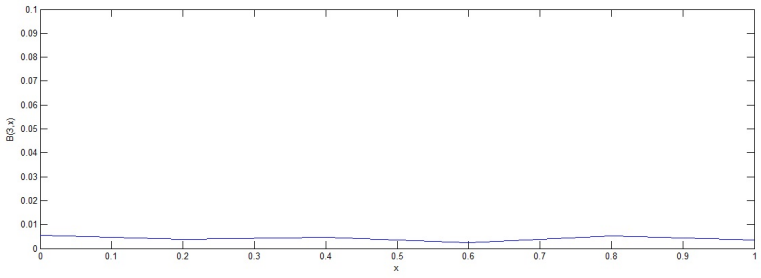


Figure 3.3: β for $t = 3$

RECOVERING S

We will add the first two equations in the system:

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{3.13}$$

$$\frac{\partial I(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma)I(t,x) - rI(t,x) + D_I I_{xx}(t,x) \tag{3.14}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) + D_R R_{xx}(t,x) - dR(t,x) \tag{3.15}$$

We get

$$\frac{\partial S(t,x)}{\partial t} + \frac{\partial I(t,x)}{\partial t}$$
$$= B(t,x) - dS(t,x) + D_S S_{xx}(t,x) - (d+\gamma)I(t,x) - rI(t,x) + D_I I_{xx}(t,x) \text{ or}$$

$$\frac{\partial S(t,x)}{\partial t} + dS(t,x) - D_S S_{xx}(t,x) = B(t,x) - g(t,x) \tag{3.16}$$

with

$$g(t,x) = \frac{\partial I(t,x)}{\partial t} + (d+\gamma)I(t,x) + rI(t,x) - D_I I_{xx}(t,x).$$

Also we will apply the Thikhonov regularization to this equation and recover S.

The second equation in the system will give us

$$\beta(t,x)S(t,x)I(t,x) = g(t,x). \tag{3.17}$$

Therefore

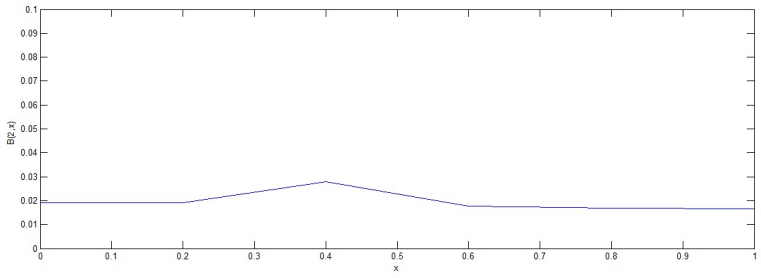


Figure 3.4: β for $t = 2$

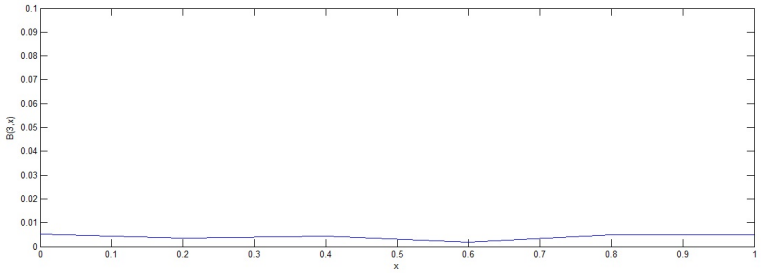


Figure 3.5: β for $t = 3$

$$\beta(t,x) = \frac{g(t,x)}{S(t,x)I(t,x)} \tag{3.18}$$

For the same data used before, Figure 3.4 shows the result for $t = 2$, and Figure 3.5 shows the result for $t = 3$.

Note

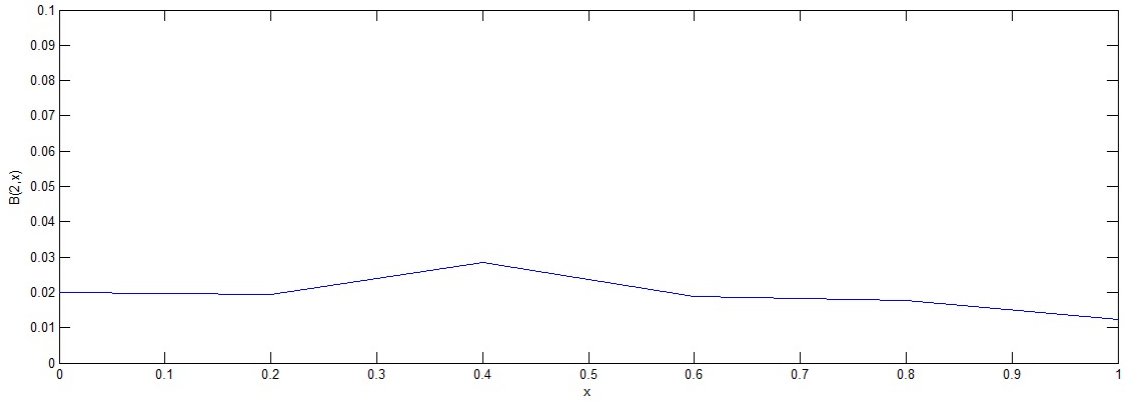
Now let us compare both methods.

Comparing results from both method (Figure 3.4.1) we could say that both methods are giving some similar results; results are the same except on the boundaries, where the error is of order 10^{-5} (For more information about the code refer to A.2)

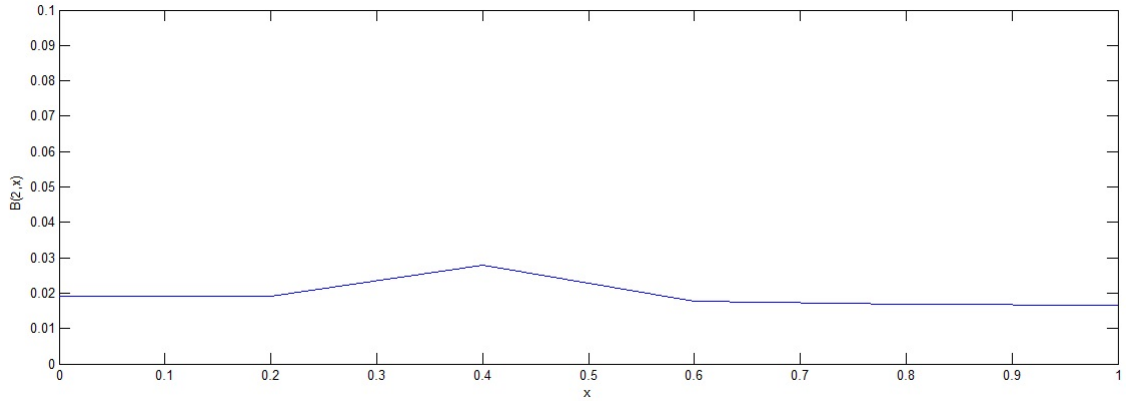
3.5 SUMMARY

In this chapter, we recovered the transmission rate, $\beta(t,x)$, but not directly. We recovered S and R and then we found the transmission rate, $\beta(t,x)$. This is why we we need to

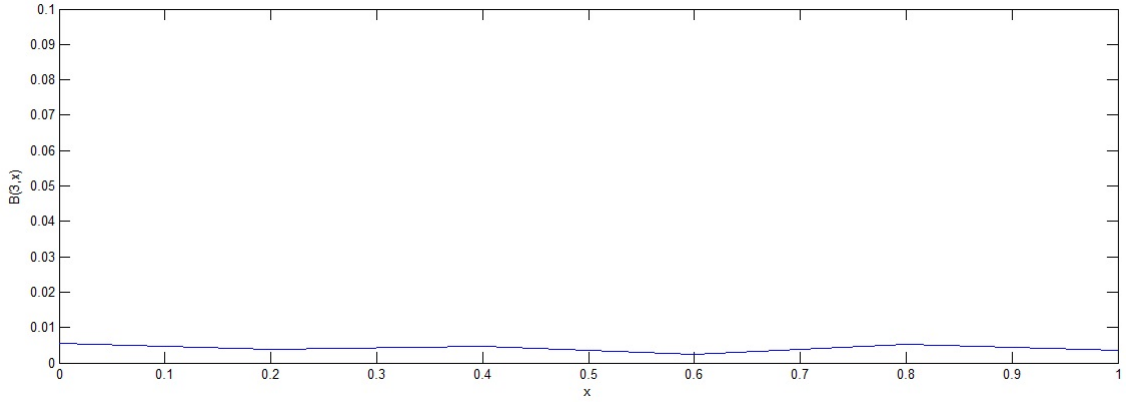
think about some method to recover the transmission rate, $\beta(t, x)$, directly. This is why, in the next chapter, we will introduce the optimal control method and recover the transmission rate, $\beta(t, x)$, using Tikhonov regularization.



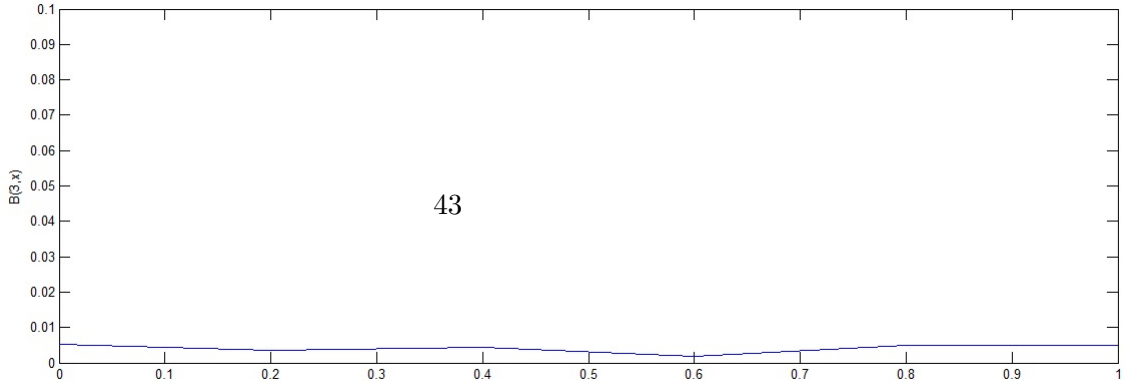
(a) $t = 2$ first method



(b) $t = 2$ second method



(c) $t = 3$ first method



Chapter 4

OPTIMAL CONTROL METHOD

Optimal control theory is an extension of the calculus of variations, a field of mathematical analysis that deals with maximizing or minimizing functionals. In optimal control theory the goal is to determine the control law for a given system that optimizes a stated criterion.

In this paper we are interested in optimal control of partial differential equations which was developed by J.L. Lions in the 1970's [11]. We continue to work with the SIR-PDE 2.11, 2.12 and 2.13, and we consider $\beta(t, x)$ the control.

4.1 OVERVIEW

Optimal control theory was developed first for the ordinary differential equations case. The optimal control in the ordinary differential equations case can be derived using Pontryagin's maximum principle which gives a necessary condition for the control [11]. But there is no complete generalization of the Pontryagin's maximum principle to partial differential equations.

Let Ω be an open, connected subset of $\Re^n$, where x is the space variable such that $x \in \Omega$. For u and v, two integrable functions (in the Lebesgue sense) on Ω, we say v is the weak x_i-derivative of u if

$$\int_\Omega u\phi_{x_i} dx = -\int_\Omega v\phi dx \tag{4.1}$$

for all ϕ in $C_c^\infty(\Omega)$, the set of all infinitely differentiable functions on Ω with compact support.

The general idea of optimal control of partial differential equations starts with a partial differential equation with state solution w and control u. Let A be a partial differential operator

$$Aw = f(w, u) \tag{4.2}$$

in $\Omega \times [0, T]$, with appropriate initial conditions and boundary conditions, with space variable x, and time variable t. We want to find the optimal control u^* such that

$$J(u^*) = \min_u J(u) \tag{4.3}$$

with

$$J(u) = \int_0^T \int_\Omega g(t, x, w(t, x), u(t, x))dt\ dx \tag{4.4}$$

for an appropriate chosen function g.

For a given control u, there exists a state solution w depending on u. Therefore we define the sensitivity of the state on the control as

$$\psi = \lim_{\epsilon \to 0} \frac{w(u + \epsilon l) - w(u)}{\epsilon} \tag{4.5}$$

This sensitivity is the directional (Gateaux) derivative of w with respect to u in the directional l. The sensitivity ψ solves a partial differential equation which is a linearized version of the state partial differential equation

$$L\psi = F(w, l, u) \tag{4.6}$$

with appropriate initial conditions and boundary conditions.
The linear operator L comes from linearizing the state partial differential equation operator A.

The PDE that we will use to find the optimal control is formed using the adjoint operator of L. The operator L and the adjoint operator L^* are related by

$$< \lambda, L\psi >=< L^*\lambda, \psi >, \tag{4.7}$$

where $< ., . >$ is the inner product as defined in chapter1.

The adjoint PDE is

$$L^*\lambda = \frac{\partial(g)}{\partial w}, \tag{4.8}$$

where g is from equation 4.4.

If u^* is the desired minimum control, then we have

$$0 \leq \lim_{\epsilon \to 0^+} \frac{J(u^* + \epsilon l) - J(u^*)}{\epsilon}. \tag{4.9}$$

The boundary and initial conditions are chosen to have desirable properties (see Section 4.2). Assuming u^* minimize $J(u)$, starting from 4.9 and using the property 4.7 of the adjoint, we will get an expression for u^* which will be called "the optimal characterization."

4.2 USE THE OPTIMAL CONTROL THEORY TO RECOVER β

Let us recall our SIR system of partial differential equations

$$\frac{\partial S(t,x)}{\partial t} = B(t,x) - dS(t,x) - \beta(t,x)S(t,x)I(t,x) + D_S S_{xx}(t,x) \tag{4.10}$$

$$\frac{\partial I(t,x)}{\partial t} = \beta(t,x)S(t,x)I(t,x) - (d+\gamma)I(t,x) - rI(t,x) + D_I I_{xx}(t,x) \tag{4.11}$$

$$\frac{\partial R(t,x)}{\partial t} = rI(t,x) + D_R R_{xx}(t,x) - dR(t,x) \tag{4.12}$$

Because $N(t,x) = S(t,x) + R(t,x) + I(t,x)$ and $B(t,x) = \mu N(t,x)$, with μ the birth rate in the population, we will rewrite our system as

$$S_t = \mu(S + R + I) - dS - \beta SI + D_S S_{xx} \tag{4.13}$$

$$I_t = \beta SI - dI - \gamma I - rI + D_I I_{xx} \tag{4.14}$$

$$R_t = rI + D_R R_{xx} - dR \tag{4.15}$$

We will consider the space $\Omega = [0, 1]$, the time $[0, T]$, and set $Q = \Omega \times [0, T]$. The initial conditions are $S(0, x) = S_0(x)$, $I(0, x) = I_0(x)$, $R(0, x) = R_0(x)$. The boundary conditions are Neumann conditions

$$S_x(t, 0) = S_x(t, L) = I_x(t, 0) = I_x(t, L) = R_x(t, 0) = R_x(t, L) = 0.$$

We are going to assume that we know the infection data $Z(x, t)$ for $(x, t) \in Q$. We want to find $\beta(t, x)$ so that the state output matches the data, and such that $\beta(t, x) > 0$

$$\min_{\beta} \int_Q (I(t, x) - Z(t, x))^2 dtdx, \tag{4.16}$$

which is an ill-posed problem. Therefore we need to apply Tikhonov regularization, and this why we will minimize

$$J_\alpha(\beta) = \min_{\beta} \left[\frac{1}{2} \int_Q (I(t, x) - Z(t, x))^2 dtdx + \frac{\alpha}{2} \int_Q \beta^2 dtdx \right] \tag{4.17}$$

with h the regularization parameter.

Now let us apply the optimal control theory to find β.

The derivative of the states with respect to β are

$$\lim_{\epsilon \to 0} \frac{S(\beta + \epsilon k) - S(\beta)}{\epsilon} = \psi_1 \tag{4.18}$$

$$\lim_{\epsilon \to 0} \frac{I(\beta + \epsilon k) - I(\beta)}{\epsilon} = \psi_2 \tag{4.19}$$

$$\lim_{\epsilon \to 0} \frac{R(\beta + \epsilon k) - R(\beta)}{\epsilon} = \psi_3 \tag{4.20}$$

and ψ_1, ψ_2, ψ_3, satisfy the linearized state partial differential equations

$$(\psi_1)_t = \mu\psi_1 + \mu\psi_2 + \mu\psi_3 - d\psi_1 - \beta(S\psi_1 + \psi_1 I) - kSI + D_S(\psi_1)_{xx} \tag{4.21}$$

$$(\psi_2)_t = \beta(S\psi_2 + \psi_1 I) + kSI - d\psi_1 - \gamma\psi_2 - r\psi_2 + D_I(\psi_2)_{xx} \tag{4.22}$$

$$(\psi_3)_t = r\psi_2 + D_R(\psi_3)_{xx} - d\psi_3 \tag{4.23}$$

with the initial conditions:

$$\psi_1(0,x) = 0,\ \psi_1(0,x) = 0,\ \psi_3(0,x) = 0$$

and the boundary conditions

$$\frac{\partial\psi_1}{\partial x}(t,0) = \frac{\partial\psi_1}{\partial x}(t,1) = \frac{\partial\psi_2}{\partial x}(t,0) = \frac{\partial\psi_2}{\partial x}(t,1) = \frac{\partial\psi_3}{\partial x}(t,0) = \frac{\partial\psi_3}{\partial x}(t,1) = 0.$$

In the operator form, the linearized system becomes

$$L\begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\end{pmatrix} = \begin{pmatrix}-kSI\\ kSI\\ 0\end{pmatrix} \tag{4.24}$$

where

$$L = (\widehat{L} + \eta) \tag{4.25}$$

$$\widehat{L} = \frac{\partial}{\partial t} - D\frac{\partial^2}{\partial x \partial x} \tag{4.26}$$

$$\eta = \begin{pmatrix}-d - \beta I + \mu & -\beta S + \mu & \mu\\ \beta I & \beta S - d - r - \gamma & 0\\ 0 & r & -d\end{pmatrix} \tag{4.27}$$

We want to find the adjoint, such that

$$< L\psi, \lambda >= \langle \psi, L^*\lambda \rangle \tag{4.28}$$

or

$$\int_Q (L\psi)\lambda dtdx = \int_Q \psi(L^*\lambda)dtdx \tag{4.29}$$

It will lead us to an integral, and by integration by part we get the result. We assume that $\lambda_i(T) = 0$ so that the state equations and the adjoint equations match. The state equations and the adjoint equations have opposite time orientations.

The adjoint operator is

$$\widehat{L}^* = -\frac{\partial}{\partial t} - D\frac{\partial^2}{\partial x \partial x} \tag{4.30}$$

$$\eta^* = \eta^T = \begin{pmatrix} -d - \beta I + \mu & \beta I & 0 \\ -\beta S + \mu & \beta S - d - r - \gamma & r \\ \mu & 0 & -d \end{pmatrix} \tag{4.31}$$

and

$$L^* = \widehat{L}^* + \eta^T \tag{4.32}$$

The adjoint partial differential equations becomes

$$L^* \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} J(\beta)_S \\ J(\beta)_I \\ J(\beta)_R \end{pmatrix} = \begin{pmatrix} 0 \\ I - Z \\ 0 \end{pmatrix} \tag{4.33}$$

with $\lambda_i(x, T) = 0$ and $\frac{\partial \lambda_i}{\partial x}(0, t) = frac\partial\lambda_i\partial x(1, t) = 0$ for $i = 1,\ 2$ and 3.

Now let us assume that β^* minimize $J_\alpha(\beta)$ (4.17). Then for any direction k (We choose

the same arbitrary direction as in the derivatives 4.18, 4.19 and 4.20)

$$0 \leq \lim_{\epsilon \to 0^+} \frac{J(\beta^* + \epsilon k) - J(\beta^*)}{\epsilon} \tag{4.34}$$

$$= \lim_{\epsilon \to 0^+} \frac{1}{\epsilon} [\frac{1}{2} \int_Q [I(\beta^* + \epsilon k) - Z]^2 - [I(\beta^*) - Z]^2 \; dxdt \tag{4.35}$$

$$+ \frac{\alpha}{2} \int_Q (\beta^* + \epsilon k)^2 - (\beta^*)^2 \; dxdt] \tag{4.36}$$

$$= \int_Q \psi_2 (I(\beta^*) - Z) \; dxdt + \alpha \int_Q \beta^* k \; dxdt \tag{4.37}$$

$$= \int_Q \begin{pmatrix} \psi_1 & \psi_2 & \psi_3 \end{pmatrix} L^* \begin{pmatrix} 0 \\ I - Z \\ 0 \end{pmatrix} dxdt + \alpha \int_Q \beta^* k \; dxdt \tag{4.38}$$

$$= \int_Q \begin{pmatrix} \psi_1 & \psi_2 & \psi_3 \end{pmatrix} L^* \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} dxdt + \alpha \int_Q \beta^* k \; dxdt \tag{4.39}$$

$$= \int_Q \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} L \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} dxdt + \alpha \int_Q \beta^* k \; dxdt \tag{4.40}$$

$$= \int_Q \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} \begin{pmatrix} -kSI \\ kSI \\ 0 \end{pmatrix} dxdt + \alpha \int_Q \beta^* k \; dxdt \tag{4.41}$$

$$= \int_Q k(-\lambda_1 SI + \lambda_2 SI + \alpha\beta^*) dxdt \tag{4.42}$$

Then we have

$$0 \leq \int_Q k(-\lambda_1 SI + \lambda_2 SI + \alpha\beta^*) dxdt \tag{4.43}$$

and it holds for every function k. Because k can be positive, negative, zero or any combination of those, for the inequality to hold we must have that the integrand is zero. So

$$-\lambda_1 SI + \lambda_2 SI + \alpha\beta^* = 0 \tag{4.44}$$

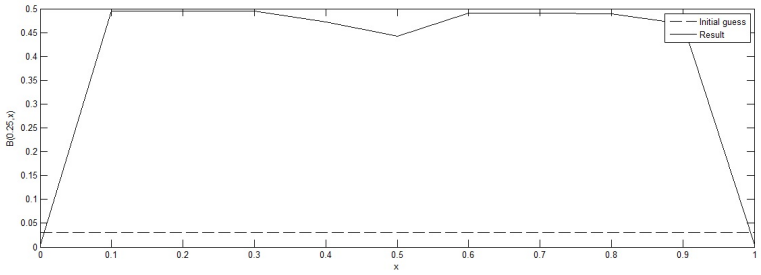


Figure 4.1: β for $t = 0.25$

This leads us to the optimal control characterization

$$\beta^* = \frac{1}{\alpha}(\lambda_1 - \lambda_2)SI. \tag{4.45}$$

4.3 NUMERICAL SIMULATION

In Matlab, we code the following: for a fixed α

1. interpolate the known data Z to a function $Z(x,t)$ on all of Q
2. choose initial guess for $\beta(x,t)$ (Here we choose the constant function 0.03)
3. solve state PDE system forward in time
4. solve adjoint PDE system backward in time
5. update β using the optimal control characterization.
6. Repeat (3), (4) and (5) until state, adjoint, and β are closed to previous iteration.

(For more information about the code refer to A.3) Figure 4.1 shows the result for $t = 0.25$.

Figure 4.2 shows the result for $t = 0.75$.

4.4 CONCLUSION

We have different methods of recovering the transmission rate in the PDE case. Numerical method are good but not as accurate as we want. In order to get better results, we

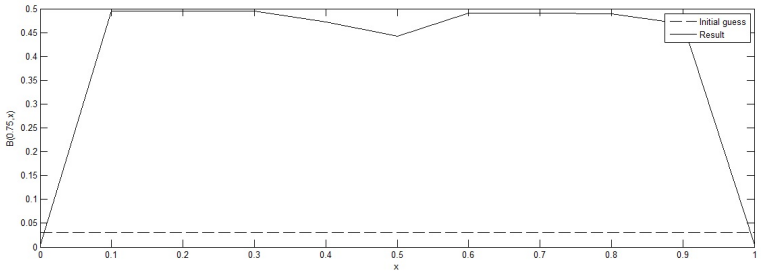


Figure 4.2: β for $t = 0.75$

may think of different methods; for example, instead of using RK4, use some more accurate method; and instead of using the finite difference method, we may think of the Chebyshev pseudo-spectral or any more accurate method.

Appendix A

CODE EXPLANATION

A.1 PDE-SIR CODE

We want to solve the PDE-SIR system. We have a time derivative, which is why we will use the Runge-Kutta 4 to solve this PDE system. Also, because we have space derivatives, we need to use the finite difference method to get them. We used the following:

1. For the population N we used a sine function.
2. $t \in [0, 5]$.
3. $x \in [-1, 1]$.
4. The number of infected people at time $t = 0$ is 1, $I(0, x) = 1$ for all $x \in [-1, 1]$.
5. The number of recovered people at time $t = 0$ is 0, $R(0, x) = 0$ for all $x \in [-1, 1]$.
6. The birth rate is the same as the death rate is $\mu = 0.001$.
7. The transmission rate is $\beta = 0.03$.
8. Diffusions are $D_S = D_I = D_R = 0.01$.
9. The recovery rate is $r = 0.3$.
10. The extra death rate due to infection, when it is used, is $\gamma = 0.3$.

We use a loop because of the diffusion terms which has a second order space derivative. The PDE is solved by fixing a space location and using RK4 to solve forward in time. For the Runge-Kutta method we only need one starting value, and since we have S_0, I_0 and R_0 we could solve this system.

The Matlab code used is

```
%In this file the Beta Identification Formula
%is tested using a known beta(x,t).
%This is joint work with Alaa Elkadry.

%Anna Mummert February 2013

function SIR_PDE_Beta_02282013_v1
clc %clear command window
format longG %displays numbers fully

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SOLVE THE PDE WITH KNOWN BETA

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the model parameter values %%%%%%%%%%%%%%%%%%

t0 = 0;
tf = 5;
dt = 0.01;
time = t0 : dt : tf;

x0 = -1;
xf = 1;
dx = 0.1;
space = x0 : dx : xf;

N = 1000;
PopTemp = N*sin(space* (2*pi))+N;
% PopTemp = N;
S = sum(PopTemp);
PopInitial = N / S *PopTemp;

InfInitial = ones(size(space)) ;
SusInitial = PopInitial - InfInitial;
RecInitial = zeros(size(space));

mu = 0.001;
beta = 0.03;
diffusion = 0.01;
recovery = 0.3;
infdeath = 0.0;

Beta = beta * ones(length(space),length(time));
Death = mu * ones(length(space),length(time));
SusDiff = diffusion * ones(length(space),length(time));
InfDiff = SusDiff;
RecDiff = SusDiff;
Recovery = recovery * ones(length(space),length(time));
InfDeath = infdeath * ones(length(space),length(time));


State = zeros(3,length(space),length(time));
State(:,:,1) = [ SusInitial; InfInitial; RecInitial ];
```

```
SIRxx = zeros(3,length(space),length(time));

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% an iterated loop to solve the pde.  The loop is
% required because the diffusion terms SIRxx are
% estimated.  The estimation improves after each
% iteration.

% The pde is solved by fixing a space location
% and using RK4 to solve forward in time.

iter = 1;
maxiter = 500;
while iter < maxiter
    oldState = State;

for j = 1 : length(space)
for i = 1 : length(time) - 1

    if j ~= 1 && j ~= length(space)
            SIRxx(:,j,i) = (State(:,j+1,i) - 2 * State(:,j,i) + State(:,j-
1,i)) / (dx ^ 2);
    end % end creation of SIRxx

    State(:,j,i+1) = rk4(State(:,j,i),SIRxx(:,j,i));

end % end time loop
end % end space loop

err = max(max(max(abs(oldState - State))));
if err < 0.01
    break;
end
iter = iter + 1;
if iter == maxiter
    warning('Convergence not reached')
end
end

% % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % The solution to the pde is graphed as a movie
% % going forward in time.
reruns = 3;
framespersec = 5;
numframes = length(time);

Frames = moviein(numframes);

for i = 1 : length(time)
plot(space,squeeze(State(:,:,i))');
axis([x0 xf 0 max(SusInitial)]);
Frames(1) = getframe;
end
```

```
movie(Frames,reruns,framespersec)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Nested functions
function v = rk4(SIRState,SIRxx)

        m1 = Xpde(SIRState,SIRxx);
        m2 = Xpde(SIRState + (dt/2)*m1,SIRxx);
        m3 = Xpde(SIRState + (dt/2)*m2,SIRxx);
        m4 = Xpde(SIRState + dt*m3,SIRxx);

        v = SIRState + (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Xprime = Xpde(SIRState,SIRxx)

Xprime = [
         birth(SIRState) - Death(j,i) * SIRState(1) - Beta(j,i) * SIRState(1)
* SIRState(2) + SusDiff(j,i) * SIRxx(1);
         Beta(j,i) * SIRState(1) * SIRState(2) - (Death(j,i) + InfDeath(j,i)
+ Recovery(j,i)) * SIRState(2) + InfDiff(j,i) * SIRxx(2);
         Recovery(j,i) * SIRState(2) - Death(j,i) * SIRState(3) +
RecDiff(j,i) * SIRxx(3)
          ];
end

function Birth = birth(SIRState)
    Birth = mu * sum(SIRState);
end

function v = rk4_R(RgivenI,Rxx,ifunction,recdiff)

        m1 = Rpde(RgivenI,Rxx,ifunction,recdiff);
        m2 = Rpde(RgivenI + (dt/2)*m1,Rxx,ifunction,recdiff);
        m3 = Rpde(RgivenI + (dt/2)*m2,Rxx,ifunction,recdiff);
        m4 = Rpde(RgivenI + dt*m3,Rxx,ifunction,recdiff);

        v = RgivenI + (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Rprime = Rpde(RgivenI,Rxx,ifunction,recdiff)

Rprime = [ recovery * ifunction - mu * RgivenI + recdiff * Rxx
          ];
end

function v =
rk4_g(functiong,functiongx,functiongxx,functionf,functionfx,functionfxx,funct
ionft,ifunction,rgiveni,susdiff)
```

```
        m1 =
Fpde(functiong,functiongx,functiongxx,functionf,functionfx,functionfxx,functi
onft,ifunction,rgiveni,susdiff);
        m2 = Fpde(functiong +
(dt/2)*m1,functiongx,functiongxx,functionf,functionfx,functionfxx,functionft,
ifunction,rgiveni,susdiff);
        m3 = Fpde(functiong +
(dt/2)*m2,functiongx,functiongxx,functionf,functionfx,functionfxx,functionft,
ifunction,rgiveni,susdiff);
        m4 = Fpde(functiong +
dt*m3,functiongx,functiongxx,functionf,functionfx,functionfxx,functionft,ifun
ction,rgiveni,susdiff);

        v = functiong + (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Fprime =
Fpde(functiong,functiongx,functiongxx,functionf,functionfx,functionfxx,functi
onft,ifunction,rgiveni,susdiff)

Fprime = [(- susdiff * functiong ^2 * functionf) * functiongxx + (susdiff * 2
* functionf * functiong) * functiongx ^ 2 + (- susdiff * 2 * functiong ^ 2 *
functionfx) * functiongx + (mu * ifunction + mu * rgiveni - functionf) *
functiong ^ 4 + (susdiff * functionfxx - functionft ) * functiong ^ 3
         ];
end

function v = rk4_B(StateBeta,SIRxxBeta,IdentifiedBeta)

        m1 = Bpde(StateBeta,SIRxxBeta,IdentifiedBeta);
        m2 = Bpde(StateBeta + (dt/2)*m1,SIRxxBeta,IdentifiedBeta);
        m3 = Bpde(StateBeta + (dt/2)*m2,SIRxxBeta,IdentifiedBeta);
        m4 = Bpde(StateBeta + dt*m3,SIRxxBeta,IdentifiedBeta);

        v = StateBeta + (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Bprime = Bpde(StateBeta,SIRxxBeta,IdentifiedBeta)

Bprime = [
         birth(StateBeta) - Death(j,i) * StateBeta(1) - IdentifiedBeta *
StateBeta(1) * StateBeta(2) + SusDiff(j,i) * SIRxxBeta(1);
         IdentifiedBeta * StateBeta(1) * StateBeta(2) - (Death(j,i) +
InfDeath(j,i) + Recovery(j,i)) * StateBeta(2) + InfDiff(j,i) * SIRxxBeta(2);
         Recovery(j,i) * StateBeta(2) - Death(j,i) * StateBeta(3) +
RecDiff(j,i) * SIRxxBeta(3)
          ];
end

end
```

A.2 TIKHONOV CODE

We want to recover the transmission rate, $\beta(t, x)$ using Tikhonov regularization. This is why, we will use the "RegTools" package (byPer Christian Hansen 16 Apr 1998(Updated 18 Mar 2008)). We used the following:

1. $t \in [0, 1]$.
2. $x \in [0, 1]$.
3. For the population N we used a constant function at time $t = 0$, $N(0, x) = 250$ for all $x \in [0, 1]$.
4. The number of infected people at time $t = 0$ is 1, $I(0, x) = 1$ for all $x \in [0, 1]$.
5. The number of recovered people at time $t = 0$ is 0, $R(0, x) = 0$ for all $x \in [0, 1]$.
6. The birth rate is the same as the death rate is $\mu = 0.001$.
7. The transmission rate is $\beta = 0.03$.
8. Diffusions are $D_S = D_I = D_R = 0.01$.
9. The recovery rate is $r = 0.3$.
10. The extra death rate due to infection, when it is used, is $\gamma = 0.3$.

We will rewrite this system by isolating all the terms that we consider that we know. We will rewrite our system as

$$Ax = b,$$

with A the matrix that has all the coefficients of the unknown x, and b is the column vector that has all the known values. And by applying the Tikhonov regularization function from the package, we will get our results.

The Matlab code used is

```
%In this file we are estimating Beta
%using Tikhonov Regularization

I = [1 1 1 1 1 1; 5 7 6 5 7 6; 25 35 42 24 31 28; 45 50 60 33 52 49; 104 113
93 102 94 92; 193 201 188 175 190 188]; %Random Data
[n, m]= size(I); % Size of the Data matrix
II=ones(n,m);
N = 250*II;
Rp = [0 0 0 0 0 0; 1 1 1 1 1 1; 4 5 3 4 4 4; 12 11 13 10 9 10; 25 30 20 45 30
35; 30 35 30 55 50 52]; % our guess for the number of recovered
BE = 0.003; % Our guess for beta
Sp = N - I - Rp;  % Our guess for the number of Susceptible
BSp=BE*Sp;

It = zeros(n,m);
It(1:n-1,:)=diff(I); % time derivative of the data matrix
d = 0.001*II;
gamma = 0.001*II;
r = 0.3*II;
a=0.01;
D_I = a*II;
D_S = a*II;
D_R = a*II;
Bi = 12*II;

I_xx = zeros(n,m);
dx = 0.2;
I1 = ones(1,m);
S0 = 250*I1 - I(1,:);
R0 = zeros(1,m);
g1=r.*I;
for i = 1 : n - 1
for j = 1 : m

    if j ~= 1 && j ~= m
            I_xx(i,j) = (I(i,j+1) - 2 * I(i,j) + I(i,j-1)) / (dx ^ 2);
    end

end
end

g = It + (d + gamma + r).*I - D_I.*I_xx;

B_S1 = g./I;
B0 = g(1,:)./(I(1,:).*S0);


BS0=B_S1(1,:);

rs = 3*m*(n-1);

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% contruction of the matrix of all known values
```

```
Bi1 = zeros(rs,1);
for i = 1:m;
    for j = 1:n-1;
            Bi1(j+(i-1)*(n-1))=Bi(j+1,i);
            Bi1(j+(i-1)*(n-1)+(n-1)*m)=g(j+1,i);
            Bi1(j+(i-1)*(n-1)+2*(n-1)*m)=g1(j+1,i);
    end
end


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% contruction of the vector of all estimations of the unknowns


x0 = zeros(rs,1);
for i = 1:m;
    for j = 1:n-1;
            x0(j+(i-1)*(n-1))=Sp(j+1,i);
            x0(j+(i-1)*(n-1)+(n-1)*m)=BSp(j+1,i);
            x0(j+(i-1)*(n-1)+2*(n-1)*m)=Rp(j+1,i);
    end
end


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% contruction of the matrix of all coefficients of the unknowns


MSR_xx=zeros(rs,rs);
for i=1:m-2
for j=1:n-2
        MSR_xx(j+i*(n-1),j+i*(n-1))=2*D_S(i+1,j+1)/(dx^2);
        MSR_xx(j+i*(n-1),j+i*(n-1)-(n-1))=-D_S(i+1,j+1)/(dx^2);
        MSR_xx(j+i*(n-1),j+i*(n-1)+(n-1))=-D_S(i+1,j+1)/(dx^2);
        MSR_xx(j+i*(n-1)+2*m*(n-1),j+i*(n-1)+2*m*(n-
1))=2*D_R(i+1,j+1)/(dx^2);
        MSR_xx(j+i*(n-1)+2*m*(n-1),j+i*(n-1)-(n-1)+2*m*(n-1))=-
D_R(i+1,j+1)/(dx^2);
        MSR_xx(j+i*(n-1)+2*m*(n-1),j+i*(n-1)+(n-1)+2*m*(n-1))=-
D_R(i+1,j+1)/(dx^2);
end
end
MSR_t=zeros(rs,rs);
for i=1:m-1
    for j=1:n-2
        MSR_t(j+(i-1)*(n-1),j+(i-1)*(n-1))=-1;
        MSR_t(j+(i-1)*(n-1),j+(i-1)*(n-1)+1)=1;
        MSR_t(j+(i-1)*(n-1)+2*m*(n-1),j+(i-1)*(n-1)+2*m*(n-1))=-1;
        MSR_t(j+(i-1)*(n-1)+2*m*(n-1),j+(i-1)*(n-1)+1+2*m*(n-1))=1;
    end
end
```

```
MdSR = zeros(rs,rs);
for i=1:n
    for j=1:m-1
        MdSR(j+(i-1)*(n-1),j+(i-1)*(n-1))=d(j+1,i);
        MdSR(j+(i-1)*(n-1)+2*m*(n-1),j+(i-1)*(n-1)+2*m*(n-1))=d(j+1,i);
    end
end
MISB = zeros(rs,rs);
for i=1:n
    for j=1:m-1
        MISB(j+(i-1)*(n-1),j+(i-1)*(n-1)+m*(n-1))=I(j+1,i);
        MISB(j+(i-1)*(n-1)+m*(n-1),j+(i-1)*(n-1)+m*(n-1))=I(j+1,i);
    end
end
Z = zeros(rs,rs);
A = MdSR + MSR_t + MSR_xx +MISB;
S=zeros(n,m);
BS=zeros(n,m);
R=zeros(n,m);
S(1,:)=S0;
BS(1,:)=BS0;
R(1,:)=R0;


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve using regTools package


[U,s,V] = csvd(A);
lambda = 0.3;
b = Bi1;
[x_lambda,rho,eta] = tikhonov(U,s,V,b,lambda,x0);


for k=1:m;
    S(2:n,k)=x_lambda(1+(k-1)*(n-1):n-1+(k-1)*(n-1));
    BS(2:n,k)=x_lambda(1+(k-1)*(n-1)+m*(n-1):n-1+(k-1)*(n-1)+m*(n-1));
    R(2:n,k)=x_lambda(1+(k-1)*(n-1)+2*m*(n-1):n-1+(k-1)*(n-1)+2*m*(n-1));
end


beta1=BS./S; % matrix for beta
```

A.3 OPTIMAL CONTROL CODE

We want to recover the transmission rate using the optimal control method. We assume the following:

1. $t \in [0, 1]$.
2. $x \in [0, 1]$.
3. For the population N we used a constant function at time $t = 0$, $N(0, x) = 250$ for all $x \in [0, 1]$.
4. The number of infected people at time $t = 0$ is 1, $I(0, x) = 1$ for all $x \in [0, 1]$.
5. The number of recovered people at time $t = 0$ is 0, $R(0, x) = 0$ for all $x \in [0, 1]$.
6. The birth rate is the same as the death rate is $\mu = 0.001$.
7. The transmission rate is $\beta = 0.03$.
8. Diffusions are $D_S = D_I = D_R = 0.01$.
9. The recovery rate is $r = 0.3$.
10. The extra death rate due to infection, when it is used, is $\gamma = 0.3$.

We used Matlab to do the following:

1. interpolate the known data Z to a function $Z(x, t)$ on all of Q
2. choose initial guess for $\beta(x, t)$ (Here we choose the constant function 0.03)
3. solve state PDE system forward in time. We use a loop because of the diffusion terms which has a second order space derivative. The PDE is solved by fixing a space location and using RK4 to solve forward in time.
4. solve adjoint PDE system backward in time
5. update β using the optimal control characterization.

6. Repeat (3), (4) and (5) until state, adjoint, and β are closed to previous iteration.

This method will give us $\beta(t, x)$ that we found.

The Matlab code used is

```
%In this file the Beta Identification Formula
%using Tikhonov Regularization via Optimal
%Control theory
%is tested using a known beta(x,t).
%This is joint work with Alaa Elkadry.

%Anna Mummert April 2013

function SIR_PDE_Beta_OptCntl_04042013_v1
clear %reinitialize all variables
clc %clear command window
%format longG %displays numbers fully

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SOLVE THE PDE WITH KNOWN BETA

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the model parameter values %%%%%%%%%%%%%%%%%%

% NOTE: if the solution to the pde with a known
% beta is used to generate the known data function,
% then length(time) must be one more than a multiple
% of m and length(space) must be one more than a
% multiple of n.

m = 3; % must be a positive integer
n = 2; % must be a positive integer

t0 = 0;
dt = 0.01;
tf = 1-dt;
time = t0 : dt : tf;

x0 = 0;
dx = 0.1; %cannot be 0.01 if dt = 0.01
xf = 1;
space = x0 : dx : xf;

N = 1000;
PopTemp = N*sin(space* (2*pi))+N;
S = sum(PopTemp);
PopInitial = N / S *PopTemp;

InfInitial = ones(size(space)) ;
SusInitial = PopInitial - InfInitial;
RecInitial = zeros(size(space));

mu = 0.001;
beta = 0.03;
diffusion = 0.01;
recovery = 0.3;
infdeath = 0;
```

```
Beta = beta * ones(length(space),length(time));
Death = mu * ones(length(space),length(time));
SusDiff = diffusion * ones(length(space),length(time));
InfDiff = SusDiff;
RecDiff = SusDiff;
Recovery = recovery * ones(length(space),length(time));
InfDeath = infdeath * ones(length(space),length(time));

MaxBeta = 0.5;
MinBeta = 0.005;
delta = 0.05;
h = 1;

State = zeros(3,length(space),length(time));
State(:,:,1) = [ SusInitial; InfInitial; RecInitial ];

SIRxx = zeros(3,length(space),length(time));

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% an iterated loop to solve the pde.  The loop is
% required because the diffusion terms SIRxx are
% estimated.  The estimation improves after each
% iteration.

% The pde is solved by fixing a space location
% and using RK4 to solve forward in time.

iter = 1;
maxiter = 500;
while iter < maxiter
    oldState = State;

for j = 1 : length(space)
for i = 1 : length(time) - 1

    if j ~= 1 && j ~= length(space)
            SIRxx(:,j,i) = (State(:,j+1,i) - 2 * State(:,j,i) + State(:,j-
1,i)) / (dx ^ 2);
    end % end creation of SIRxx

    Cntl = [Beta(j,i),Beta(j,i)];
    SecDeriv = [SIRxx(:,j,i),SIRxx(:,j,i)];
    State(:,j,i+1) = rk4(State(:,j,i),SecDeriv,Cntl);

end % end time loop
end % end space loop


err = max(max(max(abs(oldState - State))));
if err < 0.01
    break;
end
iter = iter + 1;
if iter == maxiter
```

```
    warning('Convergence not reached')
end
end

if min(min(min(State(:,:,:)))) < 0
display(min(min(min(State(:,:,:)))))
end

dt2 = dt*m;
time2 = t0 : dt2 : tf;

dx2 = dx*n;
space2 = x0 : dx2 : xf;

IData = zeros(length(space2),length(time2));
for i = 1 : length(time2)
for j = 1 : length(space2)
    IData(j,i) = State(2,(j-1)*n+1,(i-1)*m+1);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INTERPOLATE I(x,t) DATA
% Interpolate I(x,t) by hand - linear
IFunction = zeros(length(space),length(time));

for j = 1 : length(space2)
IFunction((j-1)*n+1,:) = interp1(time2,IData(j,:),time);
end
for i = 1 : length(time2)
IFunction(:,(i-1)*m+1) = interp1(space2,IData(:,i),space);
end
for j = 1 : length(space)
for i = 1 : length(time)
    if IFunction(j,i) == 0
        IFunction(j,i) = (IFunction(j-1,i) + IFunction(j+1,i)) / 2;
    end
end
end
% display(IFunction)
if isnan(IFunction(length(space),length(time)))
display('IFunction contains NaN')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Choose Initial Guess for beta(x,t)
% Define adjoint variables

OptBeta =  beta * ones(length(space),length(time));

OptState = zeros(3,length(space),length(time));
OptState(:,:,1) = [ SusInitial; InfInitial; RecInitial ];

OptSIRxx = zeros(3,length(space),length(time));
```

```
Lambda = zeros(3,length(space),length(time));
Lambda(:,:,end) = zeros(3,length(space));

Lambdaxx = zeros(3,length(space),length(time));

test = -1;
count = 0;
endcount = 20;
while (test < 0 && count < endcount)

oldBeta = OptBeta;
oldState = OptState;
oldLambda = Lambda;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve the state PDE forward in time
% using the guessed or updated beta(x,t)

iter = 1;
maxiter = 500;
while iter < maxiter
    oldOptState = OptState;

for j = 1 : length(space)
for i = 1 : length(time) - 1

    if j ~= 1 && j ~= length(space)
            OptSIRxx(:,j,i) = (OptState(:,j+1,i) - 2 * OptState(:,j,i) +
OptState(:,j-1,i)) / (dx ^ 2);
    end % end creation of SIRxx

    Cntl = [OptBeta(j,i),OptBeta(j,i)];
    SecDeriv = [OptSIRxx(:,j,i),OptSIRxx(:,j,i)];
    OptState(:,j,i+1) = rk4(OptState(:,j,i),SecDeriv,Cntl);

end % end time loop
end % end space loop

err = max(max(max(abs(oldOptState - OptState))));
if err < 0.01
    break;
end
iter = iter + 1;
if iter == maxiter
    warning('Convergence not reached')
end
end

if min(min(min(OptState(:,:,:)))) < 0
display(min(min(min(OptState(:,:,:)))))
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve the adjoint PDE backward in time
% using the guessed or updated beta(x,t)

iter = 1;
maxiter = 500;
while iter < maxiter
    oldAdjoint2 = Lambda;

for j = 1 : length(space)
for i = 1 : length(time) - 1
    backtime = (length(time) -1) + 2 - i;
    if j ~= 1 && j ~= length(space)
            Lambdaxx(:,j,backtime) = (Lambda(:,j+1,backtime) - 2 *
Lambda(:,j,backtime) + Lambda(:,j-1,backtime)) / (dx ^ 2);
    end % end creation of Adjointxx

    Cntl = [OptBeta(j,backtime),OptBeta(j,backtime)];
    SecDeriv = [Lambdaxx(:,j,backtime),Lambdaxx(:,j,backtime)];
    NewSt = [OptState(:,j,backtime),OptState(:,j,backtime)];
    Data = [IFunction(j,backtime),IFunction(j,backtime)];
    Lambda(:,j,backtime-1) =
rk4back(Lambda(:,j,backtime),SecDeriv,NewSt,Cntl,Data);

end % end time loop
end % end space loop

err = max(max(max(abs(oldAdjoint2 - Lambda))));
if err < 0.01
    break;
end
iter = iter + 1;
if iter == maxiter
    warning('Convergence not reached')
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Update the control.  Check convergence.

tempControl = (1 / h) * ( Lambda(1,:,:) - Lambda(2,:,:) ) .* OptState(1,:,:)
.* OptState(2,:,:);
Controltemp = squeeze(min(MaxBeta,max(MinBeta,tempControl)));
OptBeta = 0.5*(Controltemp + oldBeta);

temp1 = delta*sum(sum(abs(OptBeta),2),3) - sum(sum(abs(oldBeta -
OptBeta),2),3);
temp2 = delta*sum(sum(abs(OptState),2),3) - sum(sum(abs(oldState -
OptState),2),3);
temp3 = delta*sum(sum(abs(Lambda),2),3) - sum(sum(abs(oldLambda -
Lambda),2),3);

test = min(min(min(temp1),min(temp2)),min(temp3));
```

```
count = count + 1;
end

IdentifiedBeta = OptBeta;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USE IDENTIFIED beta(x,t) IN PDE

StateBeta = zeros(3,length(space),length(time));
StateBeta(:,:,1) = [ SusInitial; InfInitial; RecInitial ];

SIRxxBeta = zeros(3,length(space),length(time));

iter = 1;
maxiter = 500;
while iter < maxiter
    oldStateBeta = StateBeta;

for j = 1 : length(space)
for i = 1 : length(time) - 1

    if j ~= 1 && j ~= length(space)
            SIRxxBeta(:,j,i) = (StateBeta(:,j+1,i) - 2 * StateBeta(:,j,i) +
StateBeta(:,j-1,i)) / (dx ^ 2);
    end % end creation of SIRxx

    Cntl = [IdentifiedBeta(j,i),IdentifiedBeta(j,i+1)];
    SecDeriv = [SIRxxBeta(:,j,i),SIRxxBeta(:,j,i)];
    StateBeta(:,j,i+1) = rk4(StateBeta(:,j,i),SecDeriv,Cntl);

end % end time loop
end % end space loop

err = max(max(max(abs(oldStateBeta - StateBeta))));
if err < 0.01
    break;
end
iter = iter + 1;
if iter == maxiter
    warning('Convergence not reached')
end
end

if min(min(min(StateBeta(:,:,:)))) < 0
display(min(min(min(StateBeta(:,:,:)))))
end


reruns = 3;
framespersec = 5;
numframes = length(time);

Frames = moviein(numframes);
```

```
% %%% Compare the beta functions
if max(max(IdentifiedBeta)) > max(max(Beta))
    yf = max(max(IdentifiedBeta))+MinBeta;
else
    yf = max(max(Beta))+MinBeta;
end

if min(min(IdentifiedBeta)) < min(min(Beta))
    y0 = min(min(IdentifiedBeta));
else
    y0 = min(min(Beta));
end

for i = 1 : length(time)
plot(space,Beta(:,75)',space,IdentifiedBeta(:,75)');
axis([x0 xf 0 yf]);
Frames(50) = getframe;
end
%
movie(Frames,reruns,framespersec)

plot(space,Beta(:,1)',space,IdentifiedBeta(:,1)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Nested functions
function v = rk4(SIRState,SIRxx,CntlBeta)
        SecDerAvg = (SIRxx(:,1) + SIRxx(:,2)) / 2;
        CntlBetaAvg = (CntlBeta(1) + CntlBeta(2)) / 2;


        m1 = Xpde(SIRState,SIRxx(:,1),CntlBeta(1));
        m2 = Xpde(SIRState + (dt/2)*m1,SecDerAvg,CntlBetaAvg);
        m3 = Xpde(SIRState + (dt/2)*m2,SecDerAvg,CntlBetaAvg);
        m4 = Xpde(SIRState + dt*m3,SIRxx(:,2),CntlBeta(2));

        v = SIRState + (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Xprime = Xpde(SIRState,SIRxx,Beta)

Xprime = [
         birth(SIRState) - Death(j,i) * SIRState(1) - Beta * SIRState(1) *
SIRState(2) + SusDiff(j,i) * SIRxx(1);
         Beta * SIRState(1) * SIRState(2) - (Death(j,i) + InfDeath(j,i) +
Recovery(j,i)) * SIRState(2) + InfDiff(j,i) * SIRxx(2);
         Recovery(j,i) * SIRState(2) - Death(j,i) * SIRState(3) +
RecDiff(j,i) * SIRxx(3)
          ];
end

function Birth = birth(SIRState)
    Birth = mu * sum(SIRState);
end
```

```
function v = rk4back(SIRAdjoint,SecDeriv,StateVals,Cntl,Data)
        SecDerAvg = (SecDeriv(:,1) + SecDeriv(:,2)) / 2;
        StValAvg = (StateVals(:,1) + StateVals(:,2)) / 2;
        CntlAvg = (Cntl(1) + Cntl(2)) / 2;
        DataAvg = (Data(1) + Data(2)) / 2;

        m1 =
Adjointpde(SIRAdjoint,SecDeriv(:,2),StateVals(:,2),Cntl(2),Data(2));
        m2 = Adjointpde(SIRAdjoint -
dt/2*m1,SecDerAvg,StValAvg,CntlAvg,DataAvg);
        m3 = Adjointpde(SIRAdjoint -
dt/2*m2,SecDerAvg,StValAvg,CntlAvg,DataAvg);
        m4 = Adjointpde(SIRAdjoint -
dt*m3,SecDeriv(:,1),StateVals(:,1),Cntl(1),Data(1));

        v = SIRAdjoint - (dt/6)*(m1 + 2*m2 + 2*m3 + m4);
end

function Adjointprime = Adjointpde(SIRAdjoint,Adjointxx,SIRState,Beta,Data)

Adjointprime = [
         (Death(j,i) + Beta * SIRState(2) - mu) * SIRAdjoint(1) + ( -Beta *
SIRState(2)) * SIRAdjoint(2) + 0 * SIRAdjoint(3) - SusDiff(j,i) *
Adjointxx(1);
         (SIRState(2) - Data) + (Beta * SIRState(1) - mu) * SIRAdjoint(1) + (
-Beta * SIRState(1) + Death(j,i) + Recovery(j,i) + InfDeath(j,i)) *
SIRAdjoint(2) - Recovery(j,i) * SIRAdjoint(3) - InfDiff(j,i) * Adjointxx(2);
         ( -mu ) * SIRAdjoint(1) + ( 0 ) * SIRAdjoint(2) + ( Death(j,i) ) *
SIRAdjoint(3) - RecDiff(j,i) * Adjointxx(3)
          ];
end

end
```

Appendix B

LETTER FROM INSTITUTIONAL RESEARCH BOARD

The letter of IRB approval follows.

MARSHALL UNIVERSITY®
www.marshall.edu

Office of Research Integrity

September 17, 2012

Alaa Elkadry
435 Township Road 1525
Proctorville, Ohio 45669

Dear Alaa:

This letter is in response to the submitted thesis abstract concerning time-dependent transmission rate in partial differential equation in epidemic models. After assessing the abstract it has been deemed not to be human subject research and therefore exempt from oversight of the Marshall University Institutional Review Board (IRB). The Code of Federal Regulations (45CFR46) has set forth the criteria utilized in making this determination. Since the information in this study will utilize specific computer programs to compute this formula it is not considered human subject research. If there are any changes to the abstract you provided then you would need to resubmit that information to the Office of Research Integrity for review and a determination.

I appreciate your willingness to submit the abstract for determination. Please feel free to contact the Office of Research Integrity if you have any questions regarding future protocols that may require IRB review.

Sincerely,

Bruce F. Day, ThD, CIP
Director
Office of Research Integrity

REFERENCES

[1] H T Banks and K Kunisch, *Estimation techniques for distributed parameter systems*, (1989).

[2] Herbert Egger and Heinz W. Engl, *Tikhonov regularization applied to the inverse problem of option pricing: convergence analysis and rates*, Inverse Problems, 2005, p. 10271045.

[3] William L. Garrison, *Historical transportation development*, Transportation Planning and Engineering in Encyclopedia of Life Support Systems (2002).

[4] K.P. Hadeler, *Parameter identification in epidemic models*, Mathematical Biosciences **229** (2011).

[5] Delaware Heath and Social Services., *Direct and indirect disease transmission*, (2011).

[6] Victor Isakov, *Inverse problems for partial differential equations*, Applied Mathematical Sciences, vol. 127, Springer-Verlag, New York, 1998. MR 1482521 (99b:35211)

[7] Hadeler K, *Parameter estimation in epidemic models: simplied formulas.*

[8] W.O. Kermack and A.G. McKendrick, *A contribution to the mathematical theory of epidemics.*, (1927).

[9] Kenneth F. Kiple, *The Cambridge world history of human disease*, 2012.

[10] Ronald Lagnado and Stanley Osher, *A technique for calibrating derivative security pricing models: Numerical solution of an inverse problem*, Computing in Economics and Finance 1997 101, Society for Computational Economics, http://EconPapers.repec.org/RePEc:sce:scecf7:101.

[11] Suzanne Lenhart and John T Workman, *Optimal control applied to biological models*, Chapman and Hall/CRC, Boca Raton, 2007.

[12] Anna Mummert, *Studying the recovery procedure for the time-dependent transmission rate(s) in epidemic models*, Journal of Mathematical Biology.

[13] N. C. Nguyen, *A note on Tikhonov regularization of linear ill-posed problems*, (2006).

[14] Mark Pollicott, Hao Wang, and Howard (Howie) Weiss, *Extracting the time-dependent transmission rate from infection data via solution of an inverse ODE problem*, Journal of Biological Dynamics **6** (2012), no. 2, 509–523, PMID: 22873603.

[15] Michael Ulbrich, *A generalized Tikhonov regularization for nonlinear inverse ill-posed problems*, 1998.

[16] F. Brauer P. van den Driessche and J. Wu, *Mathematical epidemiology*, (2008).

[17] G. F. Webb, *A reaction-diffusion model for a deterministic diffusive epidemic*, J. Math. Anal. Appl. **84** (1981), no. 1, 150–161, http://dx.doi.org/10.1016/0022-247X(81)90156-6. MR 639529 (83d:35078)

[18] MaryGeorge L. Whitney, *Theoretical and numerical study of Tikhonovs regularization and Morozovs discrepancy principle*, (2009).

Alaa Elkadry

Born April 28, 1988 in Rachaya, Lebanon

Education

- Master of Arts. Marshall University, May 2013. Thesis Advisor: Anna Mummert.
- Bachelor of Science. Lebanene University, September 2009,.

Work

- Teacher Assistant at Marshall University Fall 2011 Spring 2013.
- Tutor at Marshall University Fall 2011 Spring 2013.

Additional Information

- Computing Skills: Knowledge of C++, Python, matlab, R, minitab, Java, html, Latex, Office.
- Languages: Arabic, French and English.

Publications

1. *Transmission rate in partial differential equation in epidemic models.* Master's thesis, Marshall University, May 2013.