ANALYSIS AND PARAMETER ESTIMATION OF VIRUS TRANSPORT THROUGH SUBSURFACE MEDIA

A THESIS

Submitted in partial fulfilment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

in

CIVIL ENGINEERING

by

DWARIKANATH RATHA

DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JULY, 2008

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATES DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **ANALYSIS AND PARAMETER ESTIMATION OF VIRUS TRANSPORT THROUGH SUBSURFACE MEDIA** in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy and submitted in the Department of Civil Engineering of the Indian Institute of Technology Roorkee, Roorkee, is an authentic record of my own work carried out during the period from July 2005 to July 2008 under the supervision of Dr. K.S.Hari Prasad, Associate Professor and Dr. C.S.P. Ojha, Professor, department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee,. The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Dwarika Nath Ratha

(DWARIKANATH RATHA)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

K.S. Hari Prasad

(K.S. HARI PRASAD)
Supervisor

(C.S.P. OJHA)
Supervisor

Date:

The Ph.D Viva-Voce Examination of **Mr. Dwarikanath Ratha**, Research Scholar, has been held on 3/11/2008

K.S. Hari Prasad 3/11/2008

Signature of Supervisor(s)

3/11/08

Signature of External Examiner

ABSTRACT

In recent years there has been a tremendous increase in the contamination of groundwater due to rapid industrial growth and use of fertilizers and pesticides in agriculture. This contaminated water passes through the soil and may produce hazardous chemicals, which are risk to public health. The definition of contaminant is defined as the presence of any objectionable substance in water which make unsafe for drinking. The substance may be physical, chemical, biological or radiological. The biological contaminants are bacteria and virus. It is viruses in drinking water that are an important source of human enteric diseases. Pathogenic microorganisms from sewage sludges, septic tanks and other sources can transport with subsurface water to drinking water wells. Production wells for drinking water must be at an adequate distance from source of contamination. Thus, there is a need to predict the contaminant distribution in ground water once these are released from the source.

Understanding of the movement of contaminants in subsurface is necessary for taking up proper remedial measures. Numerical models are very important tools for studying the movement of contaminants in subsurface. The present study is concerned with the modeling of conservative as well as nonconservative virus transport in subsurface. The model is based on an operator split approach which employs a globally second order accurate explicit finite volume method for the advective transport and an implicit finite difference method for the dispersive transport. The performance of the numerical model in predicting solute/virus movement for both advection dominated and dispersion dominated flow scenario is studied by comparing the model prediction with the corresponding analytical solutions for a wide range of

Peclet numbers and Courant numbers. The comparison is made for various cases of movement of conservative, reactive and virus transport in subsurface. In addition the virus transport numerical model is coupled with Richards equation governing moisture flow through the unsaturated zone. The numerical model simulating moisture flow through unsaturated zone is based on a mass conservative fully implicit finite difference numerical scheme. The application of the flow and transport models on virus movement through unsaturated zone is demonstrated through an example.

The present study is also concerned with the estimation of transport parameters of virus movement in subsurface. The parameter estimation is formulated as a least square minimization problem in which the parameters are estimated by minimizing the deviation between the model predicted and observed virus concentrations. For this purpose, a hybrid finite volume numerical model simulating one dimensional virus transport in subsurface is coupled with Levenberg-Marquadart optimization algorithm. The efficacy and robustness of the optimization procedure is evaluated by estimating the parameter from hypothetically generated virus concentration data in both saturated and unsaturated zones. The present study also investigates the performance of the objective function while estimating transport parameters using inverse procedures in the presence of data errors. In this study the Gaussian noise is added to the hypothetical data generated at discrete times and at discrete distances from the source. A detailed statistical analysis is carried out to study the effect of bias induced by the objective function on the estimated parameters when the data contains the errors. The optimization algorithm is also applied to estimate the transport parameters from the virus concentration data of two column experiments involving MS2 and ΦX174 virus transport in saturated and unsaturated zones.

ACKNOWLEDGEMENTS

At the outset I wish to express my immense sense of gratitude to my supervisors Dr. K.S. Hari Prasad, Associate Professor, and Dr. C.S.P. Ojha, Professor, Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee, for their invaluable guidance, thought provoking discussions and untiring efforts throughout the tenure of this work. Their timely help, constructive criticism, and painstaking efforts made it possible to present the work contained in this thesis in its present form. I express my whole hearted thanks to them for providing me necessary requirements for the work and for spending their valuable time for discussion at any time.

The cooperation and help extended by the Head and faculty members, Department of Civil Engineering, Indian Institute of Technology Roorkee is gratefully acknowledged. I also want to thank my research committee members for providing insightful and constructive comments.

I am thankful to Prof. M. Parida, Department of Civil Engineering, Indian Institute of Technology Roorkee for his cooperation and encouragement during this work. I am also grateful to acknowledge the technical staff of the Hydraulics laboratory, Department of Civil Engineering for constant support during my research work.

I express my deep sense of gratitude and reverence to my parents for their blessings and endeavor to keep my moral high, throughout the period of my work. Thanks to them as they have made it possible for me, to reach where I am today. I am also thankful to my brothers and sister for their immense love, affection, help,

cooperation and encouragement since my schooldays. I thankfully acknowledge the contribution of all my teachers since schooldays, for showing me the right path at different steps of life.

It is a pleasure to acknowledge the support extended by all my fellow research scholars. It has always been very helpful to have technical and general discussions with P.N. Chandramouli, Trilochan Sahu, Niranjan Trivedi, Kranti Borgaonkar, Deepak Sahoo, Jignesh, Kasheo Prasad, Rajesh, Rakesh, Sujit, Susmita, Rajan. The most difficult hurdles seem easy and simple, when friends with their mere presence lead you across. Susanta Panigrahy, Durga Prasad, Bikash Sahoo, Rajeev Kumar, Devi Prasad, Goutam, Nigam, Ranjan and Bana will be remembered for their companionship and frequent encouragement during my difficult days.

Financial assistance provided by the *Ministry of Human Resource and Development, Government of India*, in the form of Teaching Assistantship is thankfully acknowledged.

Finally I thank one and all who have directly and indirectly helped me throughout this work and made this duration happy and memorable.

And above all, I am thankful to the Almighty whose divine grace gave me the required courage, strength and perseverance to overcome various obstacles that stood in my way.

Dwarikanath Ratha

(DWARIKANATH RATHA)

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATIONS

Symbol	Description	Dimension
δC_i	Gradient of the concentration distribution in cell i	$[ML^{-3}]$
σ_0	Standard deviation of the initial concentration field	[L]
ϕ	Porosity of the porous material	[*]
θ	Volumetric moisture content	$[L^3L^{-3}]$
J	Jacobian matrix	-
A	Tridiagonal coefficient matrix	-
B	Vector of known quantities at the time level n	-
b	Parameter vector	-
C	Aqueous phase virus concentration	$[ML^{-3}]$
C(b)	Predicted response for a given parameter vector **b**	$[ML^{-3}]$
$C(h)$	Specific water capacity of a soil	$[L^{-1}]$
$C(\psi)$	Specific moisture capacity	$[L^{-1}]$
C^*	Mass of virus adsorbed on the solid matrix per unit mass of solid	[*]
$\mathbf{C}^*$	Observation vector whose elements represent measured concentrations	$[ML^{-3}]$
C_i	Cell centered concentration	$[ML^{-3}]$
$C_o(t)$	Source concentration	$[ML^{-3}]$
C_u	Courant number	[*]
D	Hydrodynamic dispersion coefficient	$[LT^{-2}]$
D_m	Diffusion coefficient	$[L^2T^{-1}]$
erfc	Complementary error function	-
F	Mass of solute per unit area per unit time	$[ML^{-2}T^{-1}]$
i	Typical interior node while flow in horizontal direction	-
j	Typical interior node while flow in vertical direction	-
k	Overall affinity coefficient	$[L^3M^{-1}]$
k_d	Linear distribution coefficient	$[L^3M^{-1}]$

Symbol	Description	Dimension
k_f	Equilibrium constants	$[L^3M^{-1}]$
K_s	Saturated hydraulic conductivity	$[LT^{-1}]$
K_θ	Hydraulic conductivity	$[LT^{-1}]$
L	Soil domain length	[L]
m	Picard iteration levels	-
m_v	Unsaturated soil parameters	[*]
n	Time level at which solution is known	-
N	Total no of nodes in a solution domain	-
n_v	Unsaturated soil parameters	[*]
$O(\mathbf{b})$	Objective function	-
P_e	Peclet number	[*]
Q	Total amount of surface sites or the maximum sorption capacity	[*]
R	Retardation coefficient	[*]
R_i	Van Albada limiter	[*]
t	Time coordinate	[T]
v	Pore water velocity in the flow direction	$[LT^{-1}]$
v_d	Darcy's velocity	$[LT^{-1}]$
v_m	Mean transport velocity of the microorganisms	$[LT^{-1}]$
v_w	Mean flow velocity of the groundwater	$[LT^{-1}]$
$\mathbf{W}$	Symmetric weighting matrix	-
x	Cartesian coordinate	[L]
x_0	Center of mass of the initial concentration field	[L]
Z	Vertical coordinate taken positive upwards	[L]
α_s	Dynamic dispersivity	[L]
α_v	Unsaturated soil parameters	$[L^{-1}]$
Δt	Temporal increments	[T]
Δx	Horizontal distance between nodes	[L]
Δz	Vertical distance between nodes	[L]
Θ	Effective saturation	[*]
θ_r	Residual water content of the soil	$[L^3L^{-3}]$

Symbol	Description	Dimension
θ_s	Saturated water content of the soil	$[L^3L^{-3}]$
λ	First order inactivation rate coefficient in the aqueous viruses	$[T^{-1}]$
λ^*	First order inactivation rate coefficient in the sorbed viruses	$[T^{-1}]$
ρ	Bulk density of the solid matrix	$[ML^{-3}]$
τ	Duration of the continuous source	[T]
ψ	Pressure head	[L]
Ω	Solution domain	-

Note: [*] denotes dimensionless

CHAPTER 1
INTRODUCTION

1.1 GENERAL

Ground water is an important source of drinking water. Majority of the people depend on the ground water as the principal source of drinking water. Traditionally, ground water has been considered safe for human consumption and hardly requires conventional drinking water treatment. The quality of ground water is better in comparison to surface water because of the natural purification property of the soil. However, due to increase in population, industrial and agricultural activities it is being subjected to contamination. Management of a ground water system means assuring availability of water in terms of both quantity and quality (Dhiman and Keshari, 2003). It is necessary to develop the appropriate ground water quality management plan with proper understanding of physical phenomena of contaminants movement. Pollution of ground water occurs due to mixing of physical, chemical and bacteriological contaminants from different sources. The definition of contaminant is defined as the presence of any objectionable substance in water which makes it unsafe for drinking. The substance may be physical, chemical, biological or radiological in nature. As per Freeze and Cherry (1979), contaminant is defined as "all solutes introduced into the hydrologic environment as a result of man's activities regardless of whether or not the concentrations reach levels that cause significant degradation of water quality." The definition given by Miller (1980) is "Ground water contamination is the degradation of the natural quality of ground water as a result of man's activities." Most of the contaminants are released at or slightly below the land surface

by design, by accident or by neglect. The shallow ground water is more affected by these contaminants as compared to the deep ground water. There are at least four ways by which ground water contamination occurs. a) Infiltration b) Direct migration c) Interaquifer exchange and d) Recharge from surface water.

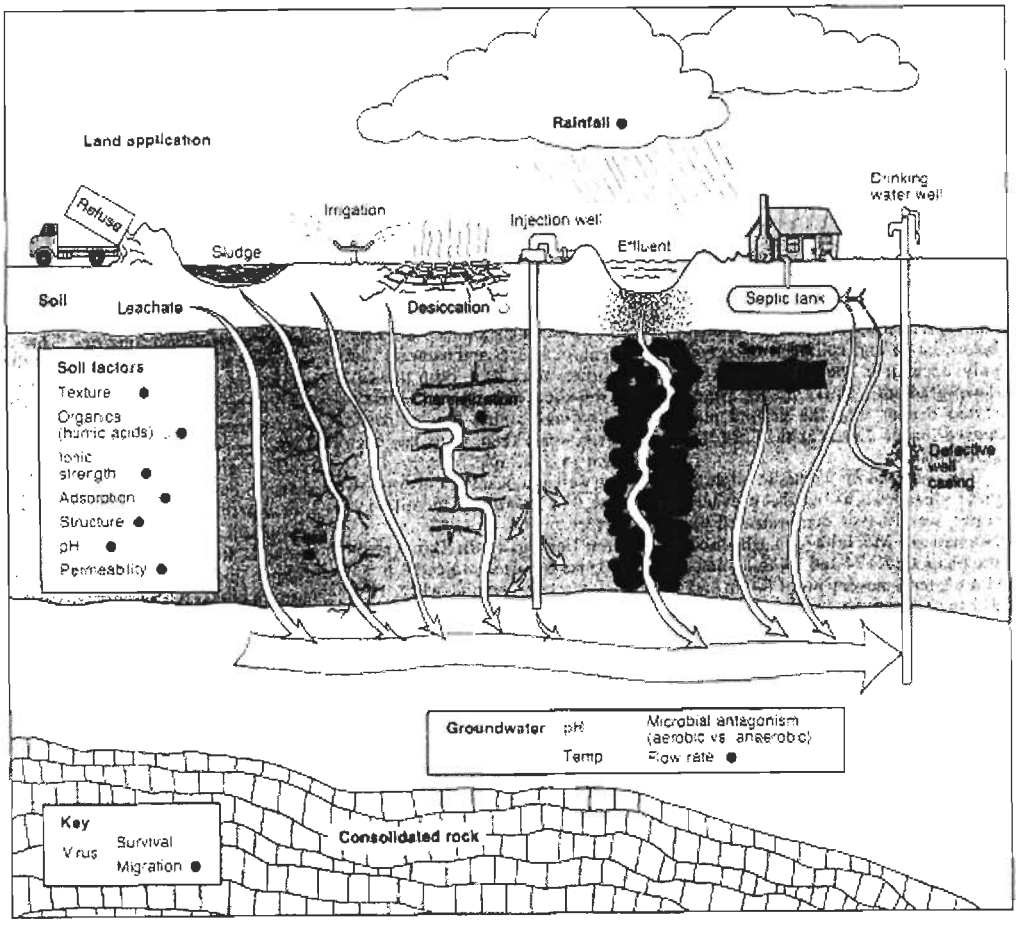

Fig 1.1 Factors affecting the entry, survival, and migration of viruses in groundwater

Fig 1.1 explains the factors affecting the entry survival and migration of viruses in groundwater. The biological contaminants are bacteria and virus. It is viruses in drinking water that are an important source of human enteric diseases (Chu et al., 2003). Pathogenic microorganisms from sewage sludges, septic tanks and other

sources can travel with ground water to drinking water wells. It has been shown that the viruses travel considerable distances through the subsurface depending on their size, their adsorption characteristics and their degree of inactivation. It is considered that soil passage is an important barrier against virus movement. For example, in Germany and in the Netherlands, 60 days traveling of ground water is adequate to inactivate pathogenic microorganisms (Schijven, 2001; Pekdeger and Matthess, 1983). Viruses moved at greater than 300 m/day and survived at least for 7 days (Keswick and Gerba, 1980). Thus, after certain time or certain distance traveled by the water, viruses are removed. The removal of virus is defined as the logarithmic reduction of virus concentration, $\log_{10}(c/c0)$. The removal of viruses from groundwater occurs due to the processes of adsorption and inactivation (Chu et al., 2001). In addition, advection and dispersion affect spreading of viruses and thereby attenuation of virus concentrations. Bacterial breakthrough is slightly faster in columns with lower clay content and that the most rapid rate of bacterial adsorption may occur during the first 60 min of exposure (Banks et al., 2003). If the inactivation coefficient equal to zero then the transport is considered as conservative and if the inactivation coefficient is considered, then the transport is considered as nonconservative.

1.2 PROBLEM IDENTIFICATION

Groundwater may become contaminated with pathogenic microorganisms from artificial recharge with wastewater or surface water, or from septic tanks or leakage of sewage pipes. Therefore, to protect groundwater from contamination, adequate setback distances between these sources of contamination and production wells for drinking water are needed. Surface water may also be contaminated with

pathogenic microorganisms mainly due to discharge of wastewater to produce safe drinking water pathogen need to be removed. To assure production of safe drinking water, adequate travel times and distances are needed. It is difficult to eliminate completely the source activities and subsurface will continue to receive increasing quantity of contaminants. Since the source activities cannot be completely eliminated, it is necessary to protect the ground water from the source of contamination. This goal can be effectively achieved after acquiring the definite knowledge of the transport of biological contaminants in the subsurface environment. The unsaturated zone acts as a conduit for the passage of the water from the ground to water table. The biological contaminants present in the ground can be carried by the water infiltrating into the ground and become a potential threat to the groundwater quality (Jyothish, 1999). In order to prevent or minimize such a water quality hazard, a thorough understanding of the flow processes combined with mechanism of virus transport in the unsaturated zone and groundwater is essential (Powelson et al., 1990; Powelson et al., 1991; Zhuang and Jin., 2003).

Pathogens of major threat to human health are viruses and the pathogenic protozoa. Little is known about the fate of these pathogenic protozoa during soil passage (Hancock et al., 1998). But much more information is available for viruses compared to pathogenic protozoa. It is believed that the processes that determine the removal of viruses during soil passage also apply to protozoa (Scijven, 2001). Therefore, this problem will be confined to the study of virus. The most common types of viruses found in ground water which may infect human body are animal viruses such as: adenovirus, coliphage, coxsackievirus, enterovirus, hepatitis, poliovirus and rotavirus (Gerba and Keswick, 1981; Yates and Yates, 1988, John and Rose, 2005).

Bacteriophages were selected as model virus as they serve as ideal indicators of viral pollution (Chattopadhyay et al., 2002). Two common bacteriophage used as tracer in groundwater are MS2 and PRD1 (Kinoshita et al., 1993). Bacteriophages such as MS2 and PRD1 have properties similar to pathogenic human viruses suggesting that bacteriophages can be used as proxies for virus transport (Corapcioglu et al., 2006). MS2 and PRD1 are considered to be good model viruses because they attach less than most pathogenic viruses and are relatively persistent during transport through subsurface. MS2 and PRD1 have relatively low isoelectric points (Bales et al., 1991) and are therefore expected to attach poorly to most soils (Gerba, 1984).

1.3 OBJECTIVE OF THE PRESENT STUDY

The present study is concerned with the numerical modeling of conservative as well as nonconservative virus transport in ground water and identification of relevant transport parameters. With this in view, the following objectives have been set for the present research.

1. To develop numerical models for virus transport both in saturated and unsaturated zones.
2. To develop an optimization model for estimating virus transport parameters.
3. To address the issue of identifiability of model parameters from the known concentration measurements.
4. To study the effect of data errors on the parameter estimates.
5. To provide an insight into the use of virus transport models with a focus on parameter selection.

1.4 ORGANISATION OF THESIS

A brief description of the layout of the thesis is presented in the following paragraphs.

A comprehensive literature review on modeling of virus transport through saturated zone and unsaturated zone, solution techniques, parameter estimation procedures is presented in Chapter 2.

Chapter 3 deals with the development of a numerical model to solve the virus transport equation through saturated zone. The numerical model solves the one dimensional movement of virus through ground water. The model is based on an operator split approach which employs a globally second order accurate explicit finite volume method for the advective transport and an implicit finite difference method for the dispersive transport. The model is validated with the available analytical solution for advection dispersion equation.

Chapter 4 deals with the optimization technique which is coupled with the virus transport equation in saturated zone to estimate the virus transport parameters. The inverse problem is formulated as a nonlinear optimization problem, i.e parameters are estimated by minimizing a suitable objective function which expresses the discrepancy between observed and predicted system response (Kool and Parker, 1988). Levenberg-Marquardt algorithm is used as the optimization algorithm due to its simplicity and robustness. The performance of inverse procedure is tested by hypothetically generated virus concentrations data. To study the effect of objective function on parameter estimates, Gaussian noise is added to a hypothetically generated data and detailed statistically analysis is carried out. The transport parameters are also estimated from the virus concentrations data from a column

experiment conducted by Bales et al. (1991) and Jin et al. (2000) for MS2 and ΦX174.

Chapter 5 deals with the development of a numerical model for virus transport in unsaturated zone. A fully implicit numerical model is developed to solve Richard's equation and the virus transport equation is coupled with the Richard's equation. The model results in the spatial distribution of nodal pressure head which are used to calculate the velocity at each grid point. These velocities are used in the virus transport equation to determine the concentration of the virus at different locations.

Chapter 6 discusses virus transport and estimation of transport parameters in the unsaturated zone. It includes development of an implicit finite difference model for moisture flow, a finite volume model for virus transport, coupling with the flow and transport models with an optimization algorithm for parameter estimation. To study the effect of objective function on parameter estimates, Gaussian noise is added to a hypothetically generated data and detailed statistically analysis is carried out.

Chapter 7 summarizes the main findings of the study and discusses the scope for future investigations.

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

Soils sustain life on earth. They are important not only from an agronomic standpoint for supporting growth of plants but also from an environmental standpoint for mitigating many of the potentially adverse effects of surface applied contaminants on the quality of ground water resources (Rockhold et al., 2004). Microbial activities directly affect the environmental quality of water, soil and sediments. It should be noted that although sludges are treated by various disinfection methods such as chlorination and heat conditioning prior to disposal, highly resistant virus may remain intact (Berg, 1977).Viruses are colloid particles with size ranging from 0.02 to 0.3 μm (Brock and Madigan, 1991). They vary widely in shape and chemical composition. Their surface charge is established by the ionizable groups comprising the virus surface; and also at natural subsurface conditions viruses are generally negatively charged (Taylor and Bosmann, 1981; Elimelech et al., 1995). Because viruses do not have their own respiratory and biosynthetic functions, they reproduce inside other cells by a process called infection. Therefore, unlike bacteria or protozoa, viruses present in groundwater cannot increase in numbers but only decrease (Sim and Chriskopoulos, 1998).

The transport of virus in ground water and the use of bacteriophage as tracers were first studied by Wimpenny et al. (1972). Virus transport in subsurface has gained much attention during the past few decades. Considering its importance in different areas researchers have made immense efforts towards understanding the complexities

underlying this phenomenon. In the following Section, a comprehensive literature review on i) different mechanisms of virus transport in subsurface ii) the mathematical modeling of virus transport in subsurface and iii) estimation of virus transport parameters is presented.

2.2 VIRUS TRANSPORT MECHANISM

In recent years, studies of transport in groundwater systems have focused on understanding the mobility and degradation characteristics of contaminants (Sureshkumar and Sekhar, 2005). During subsurface transport the virus are subject to variety of hydrological, physical and biochemical process (Mallen et al., 2005; Islam et al., 2001). Subsurface virus migration is controlled by advective-dispersive transport and sorption mechanisms (Kim, 2005; Kim, 2006). In porous media, virus can partition between the aqueous phase and solid matrix and the contaminant transport can be delayed relative to groundwater flow as a result of sorption (Kim and Kim, 2003). Attenuation processes of viruses result mainly from biodegradation and sorption (Hiscock and Grischek, 2002).

There are three mechanisms that govern the transport of virus.

1. Virus transport caused by the flow of ground water is called advection
2. Virus transport caused by the irregular mixing of waters during advection is called dispersion.
3. Chemical mechanisms which occur during advection are called retardation.

2.2.1 Advection

It is the movement of virus caused by ground water flow. Due to advection, viruses travel at an average rate equal to seepage velocity of the fluid. Seepage

velocity is defined as

$$v = \frac{v_d}{\phi} \tag{2.1}$$

where v is seepage velocity or pore velocity of the ground water (LT^{-1}), v_d is Darcy's velocity (LT^{-1}), ϕ is porosity of the porous material. In the above equation it is assumed that all the voids in the media are equally effective in conducting the flow.

2.2.2 Hydrodynamic Dispersion

Hydrodynamic dispersion involves the spreading of the viruses from the path that would be computed to follow according to the advective hydraulics of the flow system (Freeze and Cherry, 1979). Dispersion is due to the mechanical mixing during fluid advection (termed as mechanical dispersion) and because of molecular diffusion.

2.2.2.1 Mechanical dispersion

Mechanical dispersion is most easily viewed as a microscopic process. On the microscopic scale, dispersion is caused by three mechanisms. The first occurs in individual pore channels because molecules travel at different velocities at different points across the channel due to the drag exerted on the fluid by the roughness of the pore surfaces. The second process is caused by the difference in pore sizes along the flow paths followed by the water molecules. Because of differences in surface area and roughness relative to the volume of the water in individual pore channels, different pore channels have different bulk fluid velocities as shown in Fig 2.1. The third dispersive process is related to tortuosity, branching and interfingering of pore channels as shown in Fig 2.2.

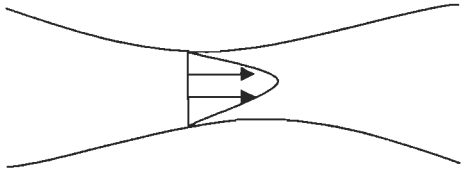

Fig 2.1 Mixing in individual pores

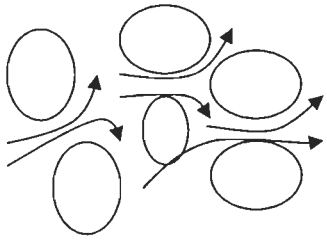

Fig 2.2 Mixing of pore channels

2.2.2.2 Molecular diffusion

Diffusion is the process whereby ionic or molecular constituents move under the influence of their kinetic activity in the direction of their concentration gradient. Diffusion occurs in the absence of any bulk hydraulic movement of the solution. If the solution is flowing, diffusion is a mechanism, along with mechanical dispersion, that causes mixing of ionic or molecular constituents. Diffusion ceases only when concentration gradients become nonexistent. The process of diffusion is often referred to as molecular diffusion.

Fick's first law states that the mass of diffusing substance passing through a given cross section per unit time is proportional to the concentration gradient and is expressed as

$$F = -D_m \frac{dC}{dx} \tag{2.2}$$

where, F is the mass of solute per unit area per unit time [$ML^{-2}T^{-1}$]. D_m is the diffusion coefficient [L^2T^{-1}], C is the concentration of the virus [ML^{-3}], and dC/dx is the

concentration gradient. The negative sign in Eq. (2.2) indicates that the virus moves in the decreasing direction of concentration.

Fick's second law relates the concentrations of a diffusing substance to space and time.

$$\frac{\partial C}{\partial t} = D_m \frac{\partial^2 C}{\partial x^2} \tag{2.3}$$

where x is the space coordinate and t is the time coordinate. The hydrodynamic dispersion is the sum of mechanical dispersion and molecular diffusion and is given as

$$D = \alpha_s v + D_m \tag{2.4}$$

where α_s [L] is the characteristic property of the porous medium known as dynamic dispersivity.

2.2.3 Sorption

During transport, virus undergo some type of reaction, the most common being sorption (Mojid and Vereecken, 2005). There are three types of adsorption, i) physical adsorption ii) chemical adsorption and iii) exchange adsorption (Corapcioglu and Haridas, 1984). Sorption tends to retard or delay the virus movement. Sorption occurs due to chemical and physical processes and implies an exchange of virus mass between mobile fluid and the immobile regions existing in the porous medium (Brusseau and Rao, 1989; Sardin et al., 1991). Adsorption of viruses to soil may be modeled as either reversible or irreversible (Agarwal et al. 2006). In the case of irreversible attachment, there is no detachment. In the case of reversible adsorption, one may have equilibrium and/or kinetic adsorption sites. In general, both kinds of adsorption may occur in a given medium. There are sites where attachment

and detachment are fast relative to flow velocity; allowing equilibrium to occur, i.e. a high mass transfer rate implies fast reactions that are usually approximated by equilibrium mass transfer. Similarly there are some other sites where adsorption is kinetically limited relative to flow velocity, with constant attachment and detachment rate coefficients, i.e. for low mass transfer rates, the reactions are rate limiting and the kinetics of sorption is significant. The transport of virus in heterogeneous porous media is the result of a spatially variable velocity field and spatially variable retardation factor. The bulk velocity (average velocity over travel path) of reactive virus is smaller than that of nonreactive virus (inactivation coefficient is equal to zero) because of the partitioning of the virus onto the solid phase (porous medium). This reduction in velocity of virus transport is called the retardation and is usually described by sorption isotherm as shown in Fig 2.3.

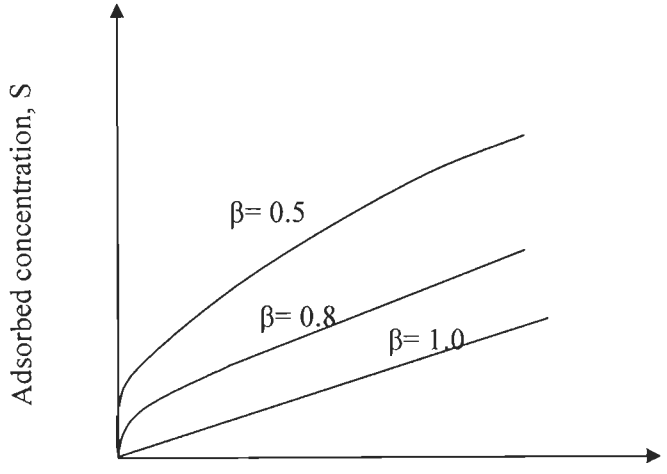

Fig 2.3 Langmuir-Freundlich isotherm with different exponent β

The sorption isotherm can be described by Langumir-Freundlich model (Huang et al., 1998) which is given as

$$C^* = \frac{Q(kC)^\beta}{1+(kC)^\beta} \tag{2.5}$$

where C^* is the concentration of virus in adsorbed phase, Q is the total amount of surface sites or the maximum sorption capacity, k (L^3M^{-1}) is an overall affinity coefficient, $0 < \beta < 1$ is a dimensionless fitting parameter. For $\beta=1$ Eq (2.5) reduces to the Langmuir sorption scenario, given as

$$C^* = \frac{Q(kC)}{1+(kC)} \tag{2.6}$$

In the limiting case $1/k >> C$, Eq (2.5) reduces to Freundlich isotherm (Huang et al., 1998) and is given as

$$C^* = k_f C^\beta \tag{2.7}$$

where

$$k_f = Qk^\beta \tag{2.8}$$

where k_f is equilibrium constants.

Viruses are subject to reversible adsorption and desorption processes which approximately follow the model of the Freundlich's isotherm (Matthess et al., 1988). This model describes the equilibrium between the concentration of suspended (C) and adsorbed (C^*) viruses. In dilute suspensions, if $\beta=1$ the constant k_f equals to distribution coefficient k_d (Matthess et al., 1988). The subsequent adsorption-desorption processes cause a retardation of the virus with respect to the transporting groundwater which may be described by the retardation factor R. R is defined as the ratio of the mean transport velocity of the microorganisms to the mean groundwater flow velocity and can be approximated if the distribution coefficient k_d of the viruses is known and is given as

$$R = \frac{v_m}{v_w} = 1 + \frac{\rho}{\phi} k_d \tag{2.9}$$

where v_w is the mean flow velocity of the groundwater, v_m is the mean transport

velocity of the microorganisms, ρ is the bulk density of aquifer material, ϕ is the effective porosity and k_d is the distribution coefficient.

2.2.4 Inactivation

Viruses lose their ability to infect host cells with time by inactivation. Viruses are inactivated because of disruption of coat proteins and degradation of nucleic acids (Gerba, 1984). The factors that influence inactivation of viruses have already been reviewed by Yates et al. (1987). Virus inactivation is usually regarded as a first order process (Traub et al., 1986; Anders and Chrysikopoulous, 2006). The most important factors that influence virus inactivation rates are temperature, adsorption to particular matter and soil, unsaturated conditions and microbial activity. Effects of other factors are found to be of insignificant importance. Yates et al. (1985) found that pH, ammonia, magnesium hardness, total hardness, nitrate, total dissolved solids and turbidity did not significantly affect inactivation of MS2, poliovirus 1 and echovirus 1. However, inactivation of MS2 increased significantly with calcium hardness. Virus inactivation is generally quantified by their inactivation rate coefficients. During the transport, viruses are inactivated by various mechanisms. Experimental observations suggest that the inactivation rate is smaller for attached phase than for liquid phase viruses (Hurst et al., 1980; Gerba, 1984). Studies of Sobsey et al. (1980) and Yates et al. (1987) indicated that there exists a strong correlation between virus adsorption and inactivation. They showed that the viruses which are adsorbed onto the solid matrix survive longer period because they are protected against disruption of coat protein and degradation of nucleic acid. Thus inactivation rates of liquid phase and attached viruses should not be assumed equal (Sim and Chrysikopoulous, 1995). Table 2.1 shows an inventory of the values of inactivation rate coefficient for viruses in ground water and sewage water from several studies.

Table 2.1: Inactivation rate coefficient of viruses in ground water and waste water [source Schijven, 2001]

^{0}C	Water	MS2	FRNAPH	PRD1	POLIO1	ECHO1	HAV	Reference
4	GRW	0.063						Yates et al. (1985)
7	GRW	0.0058		0.01				Yahya et al. (1993)
7	GRW	0.10		0.10				
10	GRW	0.11		0.025	0.11		0.19	Blanc and Nasser (1996)
10	GRW[a]				0.010			Matthess et al. (1988)
10	GRW-deionized				0.032			
10	GRW				0.013			
10	GRW		0		0		0.10	Nasser et al. (1993)
10	PE[a]		0		0.046		0.17	
10	PE		0.031		0.0077		0.12	
10	SE	0.077		0.054	0.03			Blanc and Nasser (1996)
10	TE	0.091		0.051	0.11	0.17		
12	GRW	0.16			0.18	0.24		Yates et al. (1985)
13	GRW	0.22			0.20	0.25		Yates et al. (1985)
16	GRW(5.4 mg/l O_2)				0.21			Jansons et al.(1989)
16	GRW(0.2 mg/l O_2)				0.069			
17	0.18				0.31	0.28		Yates et al. (1985)
20	GRW		0.0077		0.038		0	Nasser et al. (1993)
20	PE[a]		0.038		0.077		0.15	
20	PE		0.084		0.14		0.15	
20	SE[a]				0.054			Sobsey et al. (1980)
20	SE				0.14			
22	GRW (0.06 mg/l O_2)				0.16			Jansons et al.(1989)
22	GRW				0.10			Bitton et al. (1983)
23	GRW	0.36		0.035	0.17		0.18	Blanc and Nasser (1996)
23	GRW	1.3		0.12				Yahya et al. (1993)
23	GRW	0.58		0.30				
23	GRW	0.73			1.2	0.92		Yates et al. (1985)
23	SE	0.38		0.18	0.23		0.025	Blanc and Nasser (1996)
23	TE	0.28		0.069	0.15			
25	GRW[a]						0.082	Sobsey et al. (1986)
25	GRW						0.33	
25	PE[a]				0.10	0.10	>0.055	
25	PE				0.33	0.33	0.055	
25	SE[a]				0.13	0.082	0.055	
25	SE				0.13	0.13	0.073	
30	GRW		0.031		0.12		0.054	Nasser et al. (1993)
30	PE*		0.038		0.12		0.20	
30	PE		0.015		0.21		0.18	

^{0}C = temperature; GRW = groundwater; PE, SE = primary and secondary effluent; [a]sterilized

2.3 ANALYSIS OF VIRUS TRANSPORT IN SUBSURFACE

Numerous studies have been reported in literature for the analysis of virus transport in subsurface. These studies involve both experimental investigations and mathematical simulations. A brief review of these studies is reported in the following Sections.

2.3.1 Experimental Investigations

Several researchers have conducted the experiments for the analysis of virus transport in porous media. Few of them are presented here. Lance and Gerba (1984) have conducted the experiments on virus movement in soil during both saturated and unsaturated flow. They added the poliovirus to sewage water and applied that water at different rates to a 250 cm long soil column equipped with ceramic samplers at different depths. They found that movement of virus during unsaturated flow of sewage through soil columns is much less than during saturated flow. Viruses did not move below 40 cm level when sewage water was applied at less than the maximum infiltration rate. Virus penetration in columns flooded with sewage was at least 160 cm.

Bales et al. (1991) conducted series of seven column experiments using PRD-1 and MS2 as a bacteriophage to study the attachment of these bacteriophages to silica beads. They found that at pH 5.0-5.5, the attachment was at least partially reversible. However, release of attached phage was slow and breakthrough curves exhibited significant tailing. They found the rate coefficient for attachment and detachments were of the order of 10^{-4} and 10^{-6}-10^{-4} /s respectively.

Jin et al. (2000) conducted column flow experiment to determine the role of unsaturated flow on virus sorption and inactivation during transport through sand

columns. Two bacteriophages namely MS2 and ΦX174 were used in the experiment. The input solution containing bromide tracer and the viruses was applied to the column as a step function and the samples were collected at the effluent end using fraction collector. They concluded that the mechanisms of removal of both the bacteriophages were different. The increased removal of MS2 was due to inactivation whereas the increased removal of ΦX174 was due to sorption. They concluded that this difference was probably due to difference in their isoelectric points.

Scijven (2001) conducted experiments with batch suspensions, recirculating columns and flow through columns involving sandy soils and five bacteriophages: MS2, PRD1, ΦX174, Qβ and PM2. In batch and recirculating column experiments the attachment and detachment rate coefficients were determined. He found MS2 appeared to detach faster in presence of strong advective flow. In the case of flow through column experiment, he found a large proportion of ΦX174 adsorbed to equilibrium sites where as a small proportion of bacteriophages MS2, PRD1, Qβ adsorbed to equilibrium sites.

Chu et al.(2003) conducted experiment to investigate inactivation and sorption of virus during saturated and unsaturated transport in different soils. Their results showed that greater virus removal occurred from unsaturated column than in saturated column. They found that presence of in situ metal oxides was a significant factor responsible for virus sorption and inactivation.

2.3.2 Mathematical Models for Analysis of Virus Transport in Groundwater

The transport of viruses in subsurface is usually studied by solving the advection-dispersion equation governing virus movement involving sorption and inactivation. The one dimensional virus transport in hydraulically homogeneous,

saturated porous media accounting for virus adsorption and inactivation can be expressed as (Sim and Chrysikopoulous, 1996),

$$\frac{\partial C}{\partial t}+\frac{\rho}{\theta}\frac{\partial C^*}{\partial t}=\frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right)-v\frac{\partial C}{\partial x}-\lambda C-\lambda^*\frac{\rho}{\theta}C^* \tag{2.10}$$

where C is the aqueous phase virus concentration, C^* is the mass of virus adsorbed on the solid matrix, D represents the hydrodynamic dispersion coefficient, v is the pore water velocity in the flow direction, ρ is the bulk density of the solid matrix, λ is the first order inactivation rate coefficient in the aqueous viruses, λ^* is the first order inactivation rate coefficient in the sorbed viruses, x is the Cartesian coordinate and t is the time coordinate.

In terms of the virus concentration in the liquid phase, the virus transport equation can be written as (Jin et al., 1997)

$$R\frac{\partial C}{\partial t}=\frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right)-v\frac{\partial C}{\partial x}-\lambda RC \tag{2.11}$$

where R is retardation factor, defined in Eq. (2.9).

The solution of the virus transport equations can be found out both analytically and numerically. There have been numerous studies in modeling of virus transport in groundwater using analytical, numerical and experimental methods. A brief review of various analytical and numerical studies on solute and virus transport in subsurface is provided in the following Sections.

2.3.2.1 Analytical solutions of virus transport in groundwater

Several researchers have given the analytical solutions for virus transport in groundwater for different initial and boundary conditions. Analytical solutions for simple form of the governing equation known as constant parameter advection-

dispersion equation are widely available (Ogata and Banks, 1961; Bear, 1979; Van Genuchten, 1981). The utility of these analytical solutions are 1) they provide an exact solution when problem at hand is aptly described by constant parameter advection-dispersion equation and 2) they provide a means to check the accuracy of numerical solutions that are developed for more complex cases. The analytical solutions for the constant parameter advection-dispersion are available for two types of input loading scenarios. 1) finite amount of mass is instantaneously released at the upstream boundary and 2) solutes are continuously released into system at the upstream boundary. A special case for scenario 2 is continuous source of finite duration. The analytical solution for virus transport given in Eq. (2.11) is given by Van Genuchten and Alves (1982). The analytical solution for virus transport equation without retardation for continuous source of infinite duration is given by O'Loughlin and Bowmer (1975). The analytical solution for virus transport equation without retardation for continuous source of finite duration is given by Runkel (1996). The approximate analytical solutions for continuous source of both finite and infinite duration are given by O'Loughlin and Bowmer (1975) and Rose (1977). Van Genuchten (1981) reviewed the analytical methods available for different initial and boundary conditions of transport involving advection, dispersion and retardation. Runkel et al. (1996) also reviewed and compared the accuracy of the approximate analytical equation with exact analytical equation for the governing transport equation without sorption.

2.3.2.2 Numerical solutions of virus transport in ground water

The development of efficient numerical solution methodologies for the advection-dispersion transport equation has received considerable attention in recent

years (Rao et al., 2005). Numerical instability resulting from the inherent hyperbolic nature of the equation is one of the major numerical problems. The advection term causes several numerical oscillations and instability, especially when advection is dominant in the advection-dispersion transport equation and sharp concentration fronts exist. Eulerian methods such as finite difference and finite elements are commonly used for the solution of mass transport equations (Young et al., 2000). Several studies in the past have suggested that the higher order finite difference schemes lead to numerical oscillations (Hossain, 1999). The oscillation can be eliminated by reducing space steps which considerably increases the computational cost of the solution. First order upwind methods are used for avoiding numerical oscillations but the accuracy of the solution is reduced due to excessive numerical smearing. Extensive research has been carried out in the recent past for improving the accuracy of these numerical schemes. Al-Rabeh (1993) has compared the computational efficiency of various upwinding schemes for the discretisation of the one dimensional advection dominated transport equations. He has compared first order upwind, second order upwind, weighted scheme and the quadratic upstream interpolation for convective kinematics (QUICK) scheme. The result of this study shows the superiority of the central difference scheme for small Peclet numbers. For higher Peclet numbers both central difference and QUICK schemes are observed to show oscillations. It was also concluded that the second order upwind method is accurate and efficient at higher Peclet number cases. It was also shown that for moderate Peclet numbers, QUICK scheme produces a better solution but is computationally more expensive than the second order upwind method. Man and Tsai (2007) developed a numerical scheme to solve advection diffusion equation in

staggered system. They have used Adams-Bashforth predictor-corrector method for time derivative and Von Neumann method for stability analysis.

Another major problem in solving advection-dispersion equation is the mass balance error pertaining to its nonlinear nature when transport involves physical and chemical reactions such as degradation, adsorption and production. Although a good mass balance error does not guarantee an accurate solution, mass conservation is an essential requirement for an accurate numerical algorithm. To reduce the mass balance error, small time steps and iterative procedures are usually required to solve a nonlinear equation or a system of coupled nonlinear equations, which in turn makes the solution very time consuming. In flow and transport modeling, most attention has been paid to overcoming the nonlinearity of the flow problems and eliminating the numerical dispersion and spurious oscillations of the transport problems. Celia et al. (1990) presented an Optimal Test Function (OTF) method to solve the contaminant transport problem involving nonlinear reaction and biodegradation. The operator splitting technique has received much attention in the groundwater modeling literature for nonlinear transport (Yeh and Tripathy, 1989; Wheeler and Dawson, 1987; Rifai and Bedient, 1990). The nonlinear system of advection-dispersion-reaction equations is split into a system of linear partial differential equations involving the advection-dispersion equations and a system of nonlinear ordinary differential equations involving the reaction equations. Cheng et al. (2003) compared the full coupling, operator splitting and predictor-corrector techniques to solve the reactive transport equation. Their investigation has led to conclusion that both operator splitting and predictor-corrector technique can effectively solve reactive transport equation.

Starting with Van Leer (1974), upwind methods based on Godunov's scheme

have been derived that are non-oscillatory and globally second order accurate in space for the solution of hyperbolic equations. These methods are based on exact or approximate solutions to local Riemann problems and on the concept of monotone interpolation of the variable to avoid oscillations. Van Leer (1974, 1977a, 1977b, 1984) developed the Monotone Upwind Scheme for Conservation Laws (MUSCL) method in which differences are first limited and then used for the solution of Local Riemann problems, a procedure known as preprocessing. The method is forced to be monotone and to maintain global second order accuracy. Furthermore it uses cell balances and averages that enhance the non-oscillatory property. Harten (1983) constructed a second order accurate Total Variation Diminishing (TVD) upwind scheme in which an extra diffusion flux is added to minimize the numerical diffusion. The TVD property guarantees that the total variation of the solution will not increase as the solution progresses in time. This property ensures that the integration scheme is monotonicity preserving, thereby preventing the formation of spurious oscillations in the solution.

A triangular finite volume approach based on a second order TVD scheme which is fully explicit was proposed by Putti et al. (1990), for the solution of advection-dispersion equation. It is shown that finite volume method is ideally suited for advective dominated problems and for tracking sharp fronts. However model application is limited to flow fields aligned along the grid directions. Since the advective and dispersive fluxes are calculated explicitly, a serious limitation on the time step for high dispersive situations is imposed. It is observed that for advection dominated cases some numerical smearing occurs which depends on Courant number. Celia et al. (1989) have presented a one dimensional numerical model for the

simulation of reactive transport in porous media using an Optimal Test Function (OTF) method. The test function was automatically selected based on the relative domination of each processes of advection, dispersion and reactions. The applicability of this method for variable coefficient requires interpolation. Kindred and Celia (1989) have used this model for simulating contaminant transport with biodegradation.

Tim and Mostaghimi (1991) developed a numerical model, VIROTRANS for simulating the vertical movement of water and virus through soils treated with wastewater effluent and sewage sludges. They coupled the set of partial differential equations that describe the transient flow of water and suspended virus particle movement through variably saturated media and used Galerkin finite element method to accomplish the solutions to the partial differential equations. They compared the simulated model to an analytical solution and to experimental measurement of soil moisture content and poliovirus transport.

Yates and Ouyang (1992) developed a numerical model VIRTUS for predicting virus fate and transport in unsaturated soils. They compared their model with measured data of virus transport from column experiment. Their model also estimates the number of viruses entering into the groundwater after traveling through the soil from a contamination source. In this model the virus inactivation rate was allowed to vary on the basis of changes in soil temperature.

Park et al., (1992) developed a numerical model VIRALT for simulating the virus transport in groundwater. Chrysikopoulos and Sim (1996) developed a stochastic model for one dimensional virus transport in homogeneous, saturated medium. Their model accounts for first order inactivation of liquid phase and

adsorbed viruses with different inactivation rate constants and time dependent distribution coefficient. They described the virus adsorption process by a local equilibrium expression with stochastic time dependent distribution coefficient.

Clement et al. (1996) developed a numerical transport model that describes subsurface biological processes under radial flow conditions. They used a numerical procedure that incorporates the attractive features of Eulerian-Lagrangian and reaction-operator split methods for solution. They studied the sensitivity of biomass distribution and concluded that the microbial detachment and attachment processes are important transport parameters that control biomass distribution in an aquifer.

Huang et al. (1998) have presented a model for the simulation of reactive transport in groundwater. They have generalized the modified Picard iteration algorithm developed by Celia et al. (1990) for unsaturated flow to solve the nonlinear transport equation. The transport equation was written in a mixed form and the total concentration is expanded in a Taylor series with respect to the solution concentration to linearise the transport equation which is solved by using finite element method. It was shown that the numerical results of the mixed form algorithm resulted in negligible mass balance errors and required less computational time than the conventional iterative scheme. The numerical results were obtained by implementing the proposed modified Picard iteration algorithm into the HYDRUS software code.

Sim and Chrysikopoulous (2000) developed a one dimensional numerical model for virus transport in homogeneous unsaturated porous media. They have considered the virus sorption onto liquid-solid and air liquid interfaces as well as inactivation of viruses suspended in the liquid phase and viruses attached at both interfaces in their model. Also they investigated the effect of moisture content

variation of virus transport in unsaturated porous media.

Verma et al. (2000) developed overlapping control volume method for solute transport. The method is applicable for nonorthogonal grids.

Schijven and Simunek (2002) studied the removal of bacteriophages MS2 and PRD1 by dune recharge and also removal of MS2 by deep well injection. They have used the software HYDRUS-1D and HYDRUS-2D, which simulate the water flow and solute transport in one and two dimensional variably saturated porous media.

Gallo and Manzini (2003) proposed a numerical model that is based on cell centered finite volume method for the system of advection-dispersion equations of contaminants with a mixed hybrid finite element method for the solution of a single phase Darcy's equation.

2.4 FLOW IN UNSATURATED ZONE

The analysis of transport of virus in unsaturated zone is a complex process due to the nonlinearity of flow in this zone (Persson and Berndtsson, 1998). The moisture velocity in the unsaturated zone greatly depends upon the degree of saturation and varies considerably in this zone. In such a situation, it becomes essential to solve the flow equation prior to the solution of virus transport equation.

In unsaturated zone, voids present in the soil are partly filled with water and partly with air. Water is held in the voids due to surface tension forces. The pressure in the unsaturated zone is always less than the atmospheric pressure. The flow and storage characteristics are function of the pressure head. Flow in the unsaturated zone is usually simulated by solving Richards equation given by Richards (1931).

Richards equation can be expressed in several forms with either pressure head or moisture content as dependent variable (Celia et al., 1990). The constitutive

relationships between the moisture content and the pressure head allow conversion of one form of the equation to another. There are three standard forms of Richards equation available.

Pressure head based (ψ- based)

$$C(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left[K(\theta)\left\{\frac{\partial \psi}{\partial z}+1\right\}\right] \tag{2.12}$$

Moisture content based (θ-based)

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[D(\theta)\frac{\partial \theta}{\partial z}\right]+\frac{\partial K(\theta)}{\partial z} \tag{2.13}$$

Mixed form

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left\{K(\theta)\left(\frac{\partial \psi}{\partial z}+1\right)\right\} \tag{2.14}$$

where ψ is the pressure head, θ is the moisture content, z is the vertical coordinate taken positive upwards, t is the time coordinate. $C = \frac{d\theta}{d\psi}$ is the specific moisture capacity of the soil, K is the unsaturated hydraulic conductivity of the soil and $D = K/C$ is the soil moisture diffusivity.

The θ-based formulation results in significantly improved performance (Hills et al., 1989) compared to ψ-based formulation, when modeling infiltration into very dry soils but it can not be used for problems containing saturated regions, since the soil moisture diffusivity becomes infinity in the saturated regions. In contrast ψ-based formulation can be used for both saturated and unsaturated soils. However, while simulating problems involving steep wetting fronts moving into a very dry soil, ψ-based formulation requires very small time steps in order to maintain stability and minimize truncation errors. A mixed form of Richards equation that contains both moisture content and pressure head as unknowns has advantage over the ψ-based

Richards equation because the former is more mass conservative than later (Celia et al., 1990; Clement et al., 1994).

Richards equation is highly nonlinear in nature, since the storage properties K, C and D are functions of dependent variable. The functional relationships between soil hydraulic properties (K, θ, ψ) are needed for analyzing unsaturated water flow in soils (Govindraju et al., 1992). It is common practice to use a K-ψ relationship which is derived from the θ-ψ relationship using some physically based approach such as the distribution of pore sizes (Mualem, 1976).

θ-ψ Relationship:

The water retention characteristics (θ-ψ Relationship) of the soil describe the soil's ability to store and release water. The θ-ψ Relationship is called soil moisture retention curve or soil moisture characteristic curve. The shape of the SMC depends upon the pore size distribution of the soil. Many investigators used empirical and semi empirical relations to describe the characteristics. Among the many empirical functional forms existing in literature for the SMC, the most popular relationships are; Brooks and Corey (1964), Campbell (1974) and Van Genuchten (1980) relationships.

Brooks and Corey's Relationship:

$$\Theta = \begin{cases} \left(\frac{\psi_b}{\psi}\right)^{\lambda} & \text{for } \psi \le \psi_b \\ 1 & \text{for } \psi > \psi_b \end{cases} \tag{2.15}$$

where ψ_b is the bubbling pressure, λ is the pore size index and Θ is the effective saturation defined as

$$\Theta = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{2.16}$$

where θ_s and θ_r are the saturated moisture content and residual moisture content of the soil respectively.

Campbell's Relationship:

$$\frac{\theta}{\theta_s} = \begin{cases} \left(\frac{H_b}{\psi}\right)^{1/b} & \text{for } \psi \le H_b \\ 1 & \text{for } \psi > H_b \end{cases} \tag{2.17}$$

where H_b is the scaling parameter with dimension of length and b is a constant.

Van Genuchten Relationship:

$$\Theta = \begin{cases} \left(\frac{1}{1+\alpha_v|\psi|^{n_v}}\right)^{m_v} & \text{for } \psi \le 0 \\ 1 & \text{for } \psi > 0 \end{cases} \tag{2.18}$$

where α_v and n_v are unsaturated soil parameters. $m_v = 1 - \frac{1}{n_v}$

K-θ Relationship:

The hydraulic conductivity K is a measure of the ability of the soil to transmit water and depends upon both the properties of soil and water. The unsaturated hydraulic conductivity K is a nonlinear function of moisture content θ. Many investigators used the empirical and semi empirical methods for describing the K-θ relationship. Childs and Collis-George (1950), Burdine (1953) and Mualem (1976) proposed the concept of relative hydraulic conductivity for the K-θ relationship. Two approaches are generally used for predicting the hydraulic conductivity in unsaturated soils. The first approach is, the relative hydraulic conductivity K_r is a power function of the effective saturation Θ which is given by

$$K_r = \frac{K}{K_{sat}} = \Theta^{\gamma} \tag{2.19}$$

where K_{sat} is the saturated hydraulic conductivity of the soil. For a wide range of soils γ =3.5, leads to a better agreement with experimental observations (Brooks and Corey, 1964 and Campbell, 1974).

The second approach is the use of saturated moisture content to derive the unsaturated hydraulic conductivity which is given as follows.

Burdine Equation:

$$K_r(\theta) = \Theta^2 \left[\frac{\int_{\theta=0}^{\theta} d\theta/\psi^2}{\int_{\theta=0}^{\theta_{sat}} d\theta/\psi^2} \right] \tag{2.20}$$

Childs-Collis George equation:

$$K_r(\theta) = \Theta^{\zeta} \left[\frac{\sum_{i=1}^{l} \frac{\left[2(1-i)+1\right]}{\psi^2}}{\sum_{i=1}^{s} \frac{\left[2(s-i)+1\right]}{\psi^2}} \right] \tag{2.21}$$

where s represents total number of intervals into which θ domain is divided, i is the number of intervals upto a prescribed value of θ and ζ is the exponent whose value ranges between 0 and 4/3.

Mualem (1976) derived an expression for the relative hydraulic conductivity which is given as

$$K_r(\theta) = \Theta^{1/2} \left[\frac{\int_{\theta=0}^{\theta} d\theta/\psi}{\int_{\theta=0}^{\theta_{sat}} d\theta/\psi} \right] \tag{2.22}$$

Van Genuchten (1980) derived an expression for K_r by combining Eq (2.22) given by Mualem (1976) and the θ-ψ relationship given by Eq (2.18) which is given as

$$K_r = \Theta^{1/2} \left[1 - \left(1 - \Theta^{1/m_v} \right)^{m_v} \right]^2 \tag{2.23}$$

2.4.1 Solution of Richards Equation

There have been numerous studies in modeling unsaturated flows in soils using analytical as well as numerical techniques. Feddes et al. (1988) reviewed the developments in modeling soil moisture movement in the unsaturated zone. Numerous numerical models have been developed based on finite difference and finite element methods for solving Richards equation. Feddes et al. (1978) used Crank-Nicolson finite difference scheme for solving Richards equation. The equation was taken up in suction head form. Narsimhan et al. (1978) adopted finite element method for solving problems in subsurface hydrology using a mixed explicit-implicit scheme. Cooley (1983) used the sub-domain finite element method to solve Richards equation. Huyakorn et al (1986) developed two dimensional Galerkin finite model for solving Richards equation in which the element matrix is evaluated in a simple and efficient manner using influence coefficient technique. Celia et al. (1990) proposed a mixed form of Richards equation to improve the poor mass balance and less accuracy in ψ-based formulation. Kirkland et al. (1992) proposed algorithms for solving θ-based form of Richards equation. But θ-based formulation has limited application since it can not be applied to saturated soils. Kirkland et al. (1992) defined a new variable for the transformed Richards equation which has the characteristics of water content, when soil is unsaturated and of pressure head when soil is at or near saturation. Clement et al. (1994) developed a physically based two dimensional finite difference algorithm based on mixed form of Richards equation proposed by Celia et al. (1990). The finite difference equations are solved by computationally efficient preconditioned conjugate gradient method. Rathfelder and Abriola (1994) developed efficient conservative solutions of the head based form of the Richards equation. They have demonstrated that the proper evaluation of specific moisture capacity term

improves the mass conservation of the numerical schemes. Janz and Stonier (1995) used θ-based Richards equation with a macroscopic sink term to produce soil water content profiles at any time. They used Crank-Nicholson finite difference scheme and solved it by employing implicit central difference approximation. Singh and Murty (1996) developed a numerical model based on MacCormack finite difference scheme, to study the effect of flow depth in the infiltration calculations on the simulation results. Hari Prasad et al. (2001) developed a numerical model to perform the sensitivity analysis of gravity drainage and infiltration processes on unsaturated soil parameters. Sato et al. (2003) investigated the importance of soil texture properties and water content on pore water velocity and associated solute dispersion in unsaturated zone. Dogan and Motz (2005a, b) solved finite difference formulation of mixed form of Richards equation with volumetric source or sink term using modified picard iteration scheme. In their study a new saturated-unsaturated 3-D rainfall driven groundwater flow model (SU3D) has been developed to simulate most of the important elements of the hydrological cycle. A square symmetric positive definite matrix consisting of coefficients of the finite difference formulation is formed. The nonlinear terms in the matrix are linearized at every modified Picard iteration level and then the linear system of the equations is solved using the preconditioned conjugate gradient method, which has advantage over other iterative methods in terms of computer memory requirements and faster convergence.

2.5 INVERSE PROBLEM

Accurate prediction of solute/virus transport in subsurface is essential for adopting proper contaminant remedial measures. Computer simulations based on numerical models have been widely used for these purposes. With greater model

sophistication comes a need for more intensive data requirements, and real improvements in precision will eventually hinge on our ability to accurately determine the required model (transport) parameters (Kool et al., 1987). Hydraulic and transport properties of the unsaturated zone are commonly determined by imposing rather restrictive initial and boundary conditions so that the governing equations can be inverted by analytical or semi-analytical methods. Such procedures allow direct computation of the specific functional form of deterministic model coefficients. However, these direct inversion methods also have a number of limitations which restrict their practicality, in particular when used for calibrating field scale models. Experimental analyses based on direct methods are generally quite time consuming and hence costly owing to the need to meet conditions requisite for the explicit calculation of model coefficients. Another limitation results from the need to impose relatively simple initial and boundary condition. This is especially problematic for field experiments where accurate control of the boundary conditions on a large scale can be difficult and expensive. Parameter estimation using inverse procedures have become an alternative to direct inversion methods (Kool and Parker, 1988). In such a procedure, the parameters are estimated by minimizing the deviations between the observed and model predicted output for prescribed, but arbitrary initial and boundary conditions. Contrary to the direct inversion methods, the optimization approach does not put any inherent constraint on the form or complexity of the model, on the stipulation of the initial and boundary conditions, on the constitutive relationships, or on the treatment of inhomogeneities via deterministic or stochastic representations. Thus, a major advantage is that experimental conditions can be selected on the basis of convenience and expeditiousness, rather than by a need to simplify the mathematics of the direct inversion process.

2.5.1 Posedness, Identifiability, Uniqueness and Stability

Three important factors that need attention while estimating parameters using inverse procedure are i) identifiability ii) uniqueness and iii) stability (Russo et al, 1991). Consider a functional relationship between the response $\boldsymbol{R}$ and the set of parameters $\boldsymbol{p}$, i.e. $\boldsymbol{R=F(p)}$. The inverse relationship i.e. $\boldsymbol{p=I(R)}$ determines the parameters which is known as inverse problem. This problem is properly posed if and only if i) a solution exists; ii) the solution is unique for any given $\boldsymbol{R}$; and iii) the solution is stable (Russo et al., 1991). The illposedness may sometimes be due to nonuniqueness, sometimes due to nonidentifiability or sometimes due to stability (Carrera and Neumann, 1986). If the inverse problem fails to satisfy one or more of these requirements, it is then referred to as being ill posed. Uniqueness refers to the inverse relationships $\boldsymbol{I}$. When $\boldsymbol{I}$ represents the minimization of an estimation criterion (such as the deviation between observed and predicted concentration), the inverse solution is nonunique whenever the criterion to be minimized is nonconvex, i.e. it has local minima or global minimum at more than one point in the parameter space. In other words, if a given response $\boldsymbol{R}$ leads to more than one set of parameter values $\boldsymbol{p}$, the inverse solution is nonunique. If more than one parameter set $\boldsymbol{p}$ leads to a given response $\boldsymbol{R}$, the parameters are unidentifiable. In contrast Stability means that small errors in the response data must not result in large changes in the estimated parameters. Instability may arise from a lack or poor degree of identifiability and it is generally associated with an estimation criterion that is flat near minimum.

2.5.2 Classification of Parameter Identification Methods

Newman (1973) classified the inverse problem of parameter estimation into two different approaches. 1) direct 2) indirect. The direct approach treats the model

parameters as dependent variables in a formal inverse boundary value problem. The indirect approach is based upon an output error criterion where an existing estimate of the parameters is iteratively improved until the model response is sufficiently close to that of the measured output (Yeh, 1986). Kubrusly (1977) classified the distributed parameter estimation procedures into three categories. i) direct method which consists of those methods that use optimization techniques directly to the distributed model ii) reduction to a lumped parameter system, which consists of those methods that reduce the distributed parameter system to a continuous or discrete-time lumped parameter system which is described by ordinary differential equation and iii) reduction to an algebraic equation which consists of those methods that reduce the partial differential equation to an algebraic equation.

2.5.3 General Formulation of the Estimation Problem

Many parameter estimation problems can be formulated as a weighted least-squares minimization problem.

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^T \mathbf{W}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right] \tag{2.24}$$

where the objective function, $O(\mathbf{b})$, is a function of the model parameters $\mathbf{b}$, $\mathbf{b} = \{b_1, b_2, ---b_m\}^T$; $\mathbf{C}^* = \{C_1^*, ---C_n^*\}^T$ is the observation vector whose elements represent measured concentrations; $\mathbf{C}(\mathbf{b}) = \{C_1(b), ---C_n(b)\}^T$ represents the predicted response for a given parameter vector $\mathbf{b}$, and $\mathbf{W}$ is symmetric weighting matrices. The coefficient 1/2 in above equation is purely for notational convenience. The objective is to find the parameter vector $\mathbf{b}$ that minimizes the Eq (2.24) or in other words, results in a best fit between the model and available data. The weighting

matrices **W** contain information about measurement accuracy, as well as possible correlations between measurement errors and between parameters. In the absence of any additional information besides the observations $\mathbf{C}^*$, the simplest and recommended approach is to set **W** equal to the identity matrix **W**=1. In this case the Eq (2.24) reduces to the well known ordinary least squares (OLS) problem.

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^T \left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right] = \frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^2 \tag{2.25}$$

The OLS formulation has probably been the most popular one for parameter estimation problems. Its attraction is due to its simplicity and because it requires a minimum amount of information. When observation errors are normally distributed, are uncorrelated and have a constant variance, the OLS estimates possess optimal statistical properties. When these conditions are not met, the OLS method will no longer yield optimal parameter estimates in terms of precision and minimum variance. More serious difficulties arise due to violation of the constant variance and uncorrelated errors assumptions. These situations often occur in practical problems. For instance, error variances are commonly found to increase with the magnitude of the property being measured.

2.5.4 Studies on Estimation of Parameters using Inverse Procedure

Parameter estimation techniques have been widely used in subsurface flow and transport (Yeh, 1986; Kool et al., 1987; Ghidaoui and Prasad, 2000; Barth and Hill, 2005a, b). Parameter uncertainty issues are particularly relevant to the transport of biodegradable contaminants, as the coupling of hydrodynamic, chemical and microbiological processes results in significant complexity, with numerous sources of

variability and attendant uncertainty in associated parameters (Brusseau et al., 2006). However, very few studies have been reported on estimation of transport parameters in case of virus transport in ground water. Schijven (2001) estimated the virus transport parameters by calibrating HYDRUS 1D and HYDRUS 2D model, with measured data using iterative procedure.

Barth and Hill (2005a) examined the use of observed value weighting, breakthrough curve temporal moment observation and transport time step size in sensitivity analysis of virus transport parameters. The results suggest that i) sensitivities using observed value weighting are more susceptible to numerical solution variability ii) temporal moments of the breakthrough curve are a more robust measure of sensitivity than individual conservative transport observations, and iii) the transport simulation time step size is more important than the inactivation rate in solution and about as important as at least two other parameters, reflecting the ease with which results can be influenced by numerical issues. For the numerical simulation of virus transport, they have used flow program MODFLOW96 and MT3DMS. MODFLOW's PCG2 solver has been used to solve for heads and flows.

Barth and Hill (2005b) evaluated the importance of seven types of parameters to virus transport namely hydraulic conductivity, porosity, dispersivity, sorption rate and distribution coefficients and in-solution and adsorbed inactivation. The importance of four type of observations such as hydraulic heads, flow, temporal moment of conservative transport concentrations and virus concentrations are evaluated to estimate the virus transport parameters. They concluded that these observations are not sufficient to estimate the parameters uniquely.

The present study focuses on

1. developing numerical models for virus transport in soil during saturated and unsaturated flow using finite volume method.
2. developing an optimization model for estimating virus transport parameters.
3. addressing the issue of identifiability of model parameters from the known concentration measurements.
4. effect of data errors on the parameter estimates and also bias induced by the objective function on the estimated parameters when the data contains the errors.
5. providing insight into the use of virus transport models with a focus on parameter selection.

CHAPTER 3

ANALYSIS OF VIRUS TRANSPORT IN SATURATED ZONE- MODEL DEVELOPMENT

3.1 INTRODUCTION

The present Chapter discusses the development of a numerical model for the analysis of virus transport in groundwater. The differential equation describing virus transport in groundwater is solved using finite volume method. The finite volume method is based on monotone upwind schemes for conservation laws (MUSCL) by Van Leer (1977a) which is globally high-order accurate and non-oscillatory. The performance of the numerical model is evaluated for both advective and dispersive dominated transport by comparing the model results with available analytical solutions. The development of the numerical model is described in detail in the following Sections.

3.2 GOVERNING EQUATION

The one dimensional virus transport in hydraulically homogeneous, saturated porous media accounting for virus adsorption and inactivation can be expressed as (Sim and Chrysikopoulous, 1996),

$$\frac{\partial C}{\partial t}+\frac{\rho}{\theta}\frac{\partial C^*}{\partial t}=\frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right)-v\frac{\partial C}{\partial x}-\lambda C-\lambda^*\frac{\rho}{\theta}C^* \qquad (3.1)$$

where C is the aqueous phase virus concentration, C^* is the mass of virus adsorbed on the solid matrix, D represents the hydrodynamic dispersion coefficient, v is the pore water velocity in the flow direction, ρ is the bulk density of the solid matrix, λ is the

first order inactivation rate coefficient in the aqueous viruses, λ^* is the first order inactivation rate coefficient in the sorbed viruses, x is the Cartesian coordinate and t is the time coordinate.

The left hand side of the Eq. (3.1) consists of the virus accumulation terms and last two terms on the right hand side represent the inactivation of liquid phase and adsorbed viruses respectively.

Eq. (3.1) consists of advection, dispersion, sorption and inactivation. In the absence of sorption, Eq (3.1) reduces to (Runkel and Bencala, 1995)

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) - v\frac{\partial C}{\partial x} - \lambda C \tag{3.2}$$

If there is no sorption and inactivation, then Eq (3.1) reduces to

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) - v\frac{\partial C}{\partial x} \tag{3.3}$$

3.3 INITIAL AND BOUNDARY CONDITIONS

Eq. (3.1) needs initial and boundary conditions to obtain unique solution for a given problem.

Initial condition:

Initially, i.e. at $t = 0$, the concentration of virus is usually assumed to be zero, i.e.

$$t = 0, \; C(x) = 0, \qquad 0 \leq x \leq \infty \tag{3.4}$$

Boundary conditions:

At the source ($x = 0$), two different types of boundary conditions are usually employed. In the first type, Dirichlet type boundary condition wherein the virus concentration at the inlet is set to the virus concentration of the incoming flux C_0. i.e.

$$t \geq 0, \; C(t) = C_0, \quad x = 0 \tag{3.5}$$

In the second type, flux type boundary condition is applied which is represented as,

$$t \geq 0, \quad -D\frac{\partial C}{\partial x} + vC = vC_o(t), \; x = 0 \tag{3.6}$$

where C_o (t) denotes the source concentration.

For away from the source ($x \rightarrow \infty$) the concentration flux is set to zero. i.e.

$$t \geq 0, \quad \frac{\partial C}{\partial x} = 0, \qquad x \rightarrow \infty \tag{3.7}$$

3.4 NUMERICAL SCHEME

In the present work, the governing transport equation Eq. (3.3) is solved using an operator-split approach for advection and dispersion. In such a situation, the governing transport equation Eq. (3.3) can be written as (Putti etal., 1990):

Advective transport:

$$\frac{\partial C}{\partial t} = -v\frac{\partial C}{\partial x} = -\frac{\partial F}{\partial x} \tag{3.8}$$

where F is the advective flux

Dispersive transport:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) \tag{3.9}$$

Since an operator-split approach is used in the present study for the advective and dispersive transport parts, there is a necessity to specify two separate inlet boundary conditions for both parts. One of the ways, this can be implemented is by specifying that the total incoming flux as the advective flux while specifying the dispersive flux as zero. For the case where the flow is leaving the domain, the flux for the dispersive part is ignored, while it is assumed that the advective flux is kept constant across the boundary.

In the present study, the finite volume method which is globally second order accurate is used for advective transport and central finite difference method for dispersive transport.

3.4.1 Advective Transport

The finite volume method used for solving the advective transport is based on monotone upwind schemes for conservation laws (MUSCL) by Van Leer (1977a) which is globally high-order accurate and non-oscillatory (Putti et al., 1990). The advective transport is given as

$$\frac{\partial C}{\partial t}+\frac{\partial F}{\partial x}=0 \tag{3.10}$$

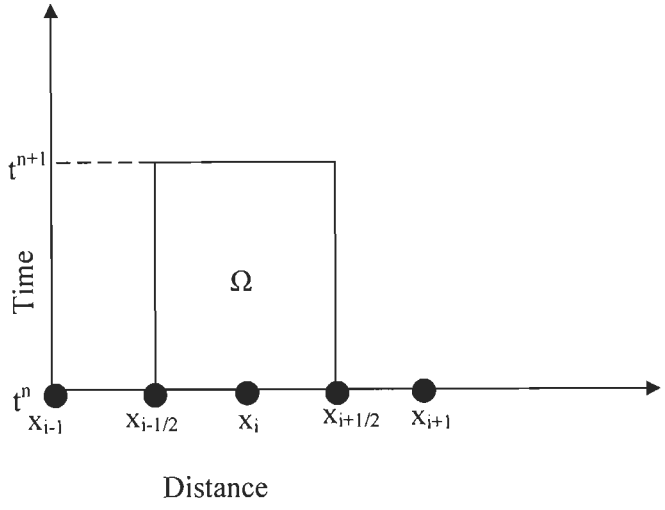

Fig 3.1 Space-time diagram

In the finite volume method, Eq. (3.10) is integrated over the solution domain Ω in space and time as

$$\iint_{\Omega}\left(\frac{\partial C}{\partial t}+\frac{\partial F}{\partial x}\right)dxdt=0 \tag{3.11}$$

Taking the limits of the domain Ω as depicted in Fig. 3.1, Eq. (3.11) can be written as

$$\int_{x_{i-1/2}}^{x_{i+1/2}} C dx + \int_{t^n}^{t^{n+1}} F dt = 0 \tag{3.12}$$

Integration of Eq. (3.12) leads to

$$\frac{\left(\overline{C}_i^{n+1} - \overline{C}_i^n\right)}{\Delta t} + \frac{\left(\overline{F}_{i+1/2}^n - \overline{F}_{i-1/2}^n\right)}{\Delta x} = 0 \tag{3.13}$$

where,

$$\begin{aligned} \overline{C}_i^n &= \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} C_i^n dx \\ \overline{F}_{i+1/2}^n &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F_{i+1/2}^n dt \end{aligned} \tag{3.14}$$

The Eq. (3.10) can be written in a discrete form by assuming the cell centered concentration C_i as cell averaged concentration as,

$$\frac{\left(C_i^{n+1} - C_i^n\right)}{\Delta t} + \frac{\left(\overline{F}_{i+1/2}^n - \overline{F}_{i-1/2}^n\right)}{\Delta x} = 0 \tag{3.15}$$

where C_i is the cell centered concentration and the flux $\overline{F}_{i+1/2}^n$ is the time averaged flux.

Various methods have been used for the calculation of the flux at each cell boundary. In this work, the advective flux is calculated using a second order upwind method similar to the one adopted by Putti et al. (1990).

Assuming a linear distribution in the cell, the mass concentration value at the cell interface is reconstructed using a MUSCL approach as (Van Leer, 1977a):

$$C_{i+1/2}^L = C_i + \frac{1}{2}\delta C_i \tag{3.16a}$$

$$C^{R}_{i+1/2} = C_{i+1} - \frac{1}{2}\delta C_{i+1} \tag{3.16b}$$

where δC_i is the gradient of the concentration distribution in cell i while L and R represent the left and right faces of the cell interface.

The value of gradient δC_i is calculated using limiters to ensure that no overshoot or undershoot occurs. The limiters give maximum allowable gradient in cell i without causing numerical oscillations. The limiter adds a certain amount of dissipation to the scheme. The accuracy of the advective transport is influenced by the choice of suitable limiter. The minmod limiter is used by Putti et.al., (1990) and Cox and Nishikawa (1991). In this work, comparison is made in the solutions of the finite volume scheme when Superbee, Van Albada and Minmod limiters are employed.

The gradient of the concentration distribution in cell i for monotonicity, prescribed by the Superbee and Minmod limiters is,

$$(\delta C_i)_{mon} = ave(\Delta_- C_i, \Delta_+ C_i) \tag{3.17}$$

with

$$\begin{aligned} \Delta_- C_i &= C_i - C_{i-1} \\ \Delta_+ C_i &= C_{i+1} - C_i \end{aligned} \tag{3.18}$$

where $ave(a,b)$ in Eq (3.17) for Minmod and Superbee limiters are as follows,

Minmod limiter:

$$ave(a,b) = \begin{cases} sign(a)\min(|a|,|b|) & \text{if ab } > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

Superbee limiter:

$$ave(a,b) = \begin{cases} sign(a)\min(2|a|, \max(|a|,|b|), 2|b|) & \text{if ab } > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

The gradient of the concentration distribution in cell i, for applying Van Albada limiter,

$$\delta C_i = R_i\left[\frac{1}{2}(1-k)(\Delta_- C_i)+\frac{1}{2}(1+k)(\Delta_+ C_i)\right] \tag{3.21}$$

where R_i is the Van Albada limiter,

$$R_i = \frac{\left[2(\Delta_+ C_i)(\Delta_- C_i)+\varepsilon\right]}{\left[(\Delta_+ C_i)^2+(\Delta_- C_i)^2+\varepsilon\right]} \tag{3.22}$$

where ε is very small number, used to avoid occurrence of division by zero in Eq. (3.22). In Eq. (3.21), when k=1/3, gives a third order accurate scheme.

The fluxes at the interface are evaluated based on Eqs. (3.16) as,

$$\overline{F}^n_{i+1/2} = v_i C^L_{i+1/2} \qquad \text{when } v_i > 0 \tag{3.23a}$$

$$\overline{F}^n_{i+1/2} = v_{i+1} C^R_{i+1/2} \qquad \text{otherwise} \tag{3.23b}$$

Integration over time:

Explicit and implicit methods are commonly used for the time integration of the discretised Eq. (3.15). A choice of $\overline{F}^n$ as given in Eq. (3.23) in Eq. (3.15), results in an Euler scheme which is first order accurate in time and is unconditionally unstable (Tai et al. 1997). To alleviate this difficulty, Hancock's Scheme (Van Albada et al., 1982) is usually employed which is a two step second-order accurate explicit scheme (Putti et al., 1990). The two half time steps in this method can be represented as predictor and corrector steps as follows,

Predictor step:

$$C_i^{n+1/2} = C_i^n - \frac{\Delta t}{2\Delta x} v_i \delta C_i \tag{3.24}$$

Corrector Step:

$$C_i^{n+1} = C_i^n - \frac{\Delta t}{\Delta x}\left(\overline{F}^{n+1/2}_{i+1/2} - \overline{F}^{n+1/2}_{i-1/2}\right) \tag{3.25}$$

The C_i obtained from the predictor step is used for calculation the fluxes. The same gradient δC_i is used in both predictor and corrector steps. For the above scheme, the time step is limited by the courant number. For stability of this scheme, Courant number, $C_u = \frac{v\Delta t}{\Delta x}$, should be less than or equal to 1.

3.4.2 Dispersive Transport

The dispersive transport is performed on the concentrations, resulting from the advective transport in each time step.

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) \tag{3.26}$$

The dispersive part is solved by a conventional, fully implicit, finite-difference scheme, which is unconditionally stable, for the final concentrations.

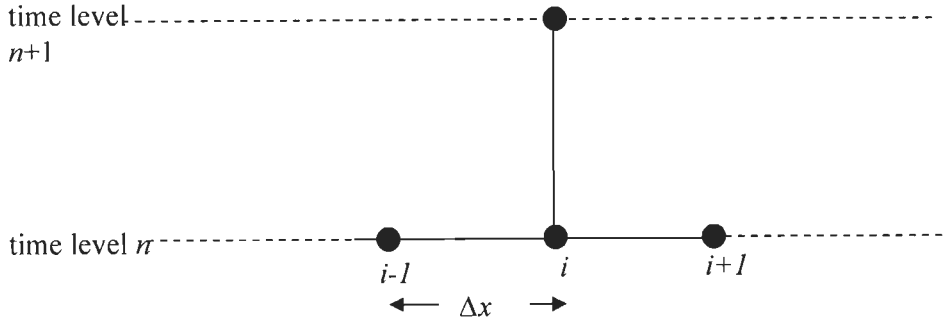

Fig 3.2 Definition sketch of finite difference discretization

3.4.2.1 Discretization in space and time

For a typical interior node i, surrounded by the adjoining nodes i-1 and i+1, the finite difference approximation of Eq. (3.26) can be written as

$$-\left(\frac{D}{\Delta x^2}\right)C_{i-1}^{n+1} + \left(\frac{2D}{\Delta x^2} + \frac{1}{\Delta t}\right)C_i^{n+1} - \left(\frac{D}{\Delta x^2}\right)C_{i+1}^{n+1} = \frac{C_i^n}{\Delta t} \tag{3.27}$$

where Δx and Δt are the spatial and temporal increments, n denotes the time level at which solution is known and n+1 denotes the time level where the solution is sought. Eq. (3.27) can be written in matrix form as,

$$[\mathbf{A}][\mathbf{C}] = [\mathbf{B}] \tag{3.28}$$

where **A** is a tridiagonal coefficient matrix, **C** is the vector of unknown concentration C_i at time level n+1 and **B** is the vector of known quantities at the time level n. The tridiagonal systems of Eqs (3.28) is solved using Thomas algorithm (Remson et al., 1971)

The present formulation has the advantage of using an implicit numerical scheme for the dispersive transport, while using an explicit numerical scheme for advection transport there by either advection dominated or dispersion dominated systems can be accurately handled. However, in this approach the restriction on the time step is due to Courant number.

A code is written using FORTRAN 90 for the implementation of the numerical model which is provided in APPENDIX-I.

3.5 NUMERICAL RESULTS

The performance of the numerical model in predicting solute/virus movement for both advection dominated and dispersion dominated flow scenario is studied by comparing the model prediction with the corresponding analytical solutions for a wide range of Peclet numbers $\left(P_e = \frac{v\Delta x}{D}\right)$ and Courant numbers $\left(C_u = \frac{v\Delta t}{\Delta x}\right)$. The comparison is made for various cases of movement of conservative, reactive and virus transport in subsurface.

3.5.1 Continuous Conservative Source of Infinite Duration

The problem consists of studying the movement of a conservative solute. A continuous source of conservative solute with a concentration of 100 units (C_0 = 100) applied at the inlet. A steady state flow velocity of 0.5 m/day is assumed the movement of such a solute is governed by Eq. (3.3). The solute concentration prior to the injection is assumed to be zero in the entire domain. The analytical solution for Eq. (3.3) subjected to initial and boundary conditions Eqs.(3.4, 3.5 & 3.7) is given by Ogata and Banks (1961) as

$$\frac{C}{C_0} = \frac{1}{2}\left[erfc\left(\frac{x - \bar{v}t}{2\sqrt{Dt}} \right) + \exp\left(\frac{\bar{v}x}{D} \right) erfc\left(\frac{x + \bar{v}t}{2\sqrt{Dt}} \right) \right] \tag{3.29}$$

In Eq. (3.29) erfc denotes the complementary error function.

The present model is applied to predict the solute movement. The spacing Δx between the consecutive grid points is taken as 1.0m and the time step is taken such that the Courant number is less than unity. The dispersion coefficient is varied to obtain required Peclet number.

Figs. 3.3a to 3.3c show the comparison of numerical and analytical solutions for advection dominated transport (P_e=200) at time equal to 150 days. It is seen from Fig 3.3a that numerical solution matches exactly with the analytical solution indicating that numerical model performs very well for the case of advection dominated transport. Fig 3.3b shows the effect of Courant number on the numerical solution. It is seen from Fig 3.3b that Courant number does not have much effect on the accuracy of the numerical solution. In Fig 3.3c, the choice of limiters on the numerical solution is shown. It is clear from Fig. 3.3c that the numerical solution with Superbee limiter is least dissipative while the Minmod limiter is most dissipative. Superbee limiter is recommended for accurately tracking the sharp advective fronts.

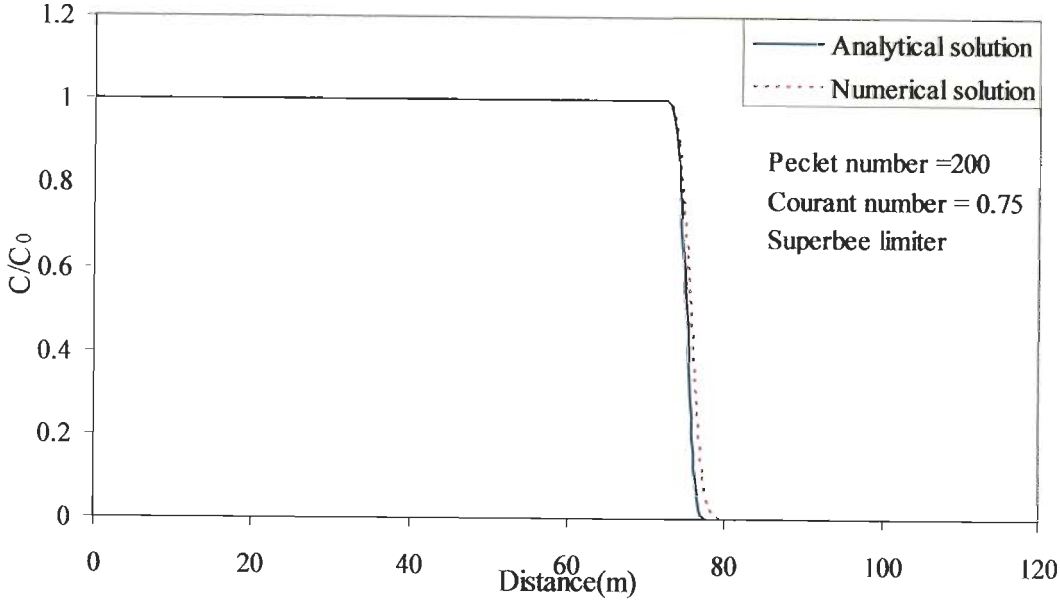

Fig 3.3(a) Comparison of analytical and numerical solution for advection dominated transport

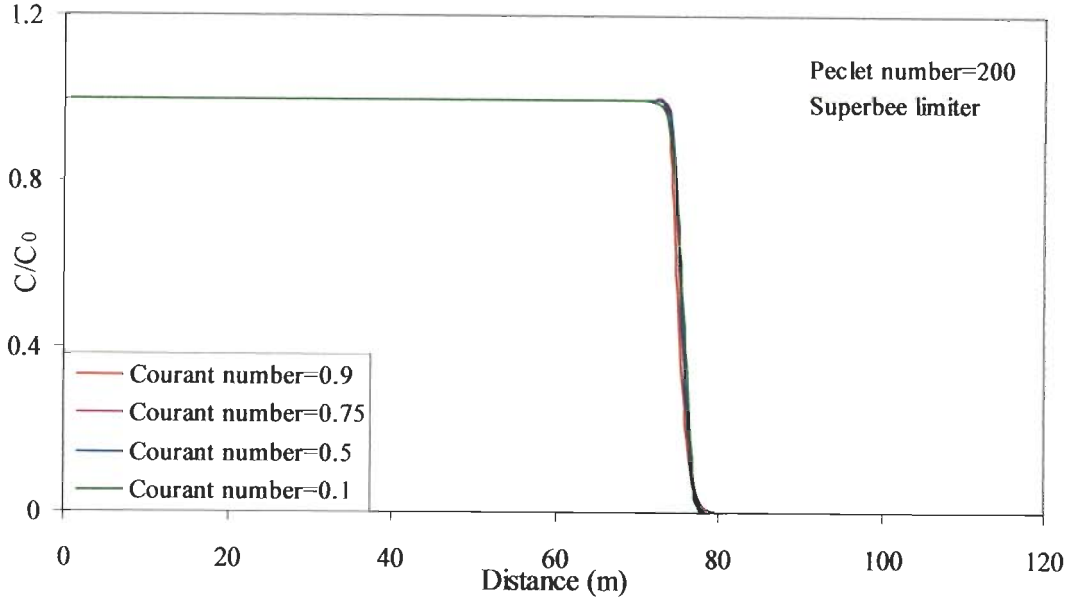

Fig 3.3(b) Effect of Courant number on the numerical solution for advection dominated transport

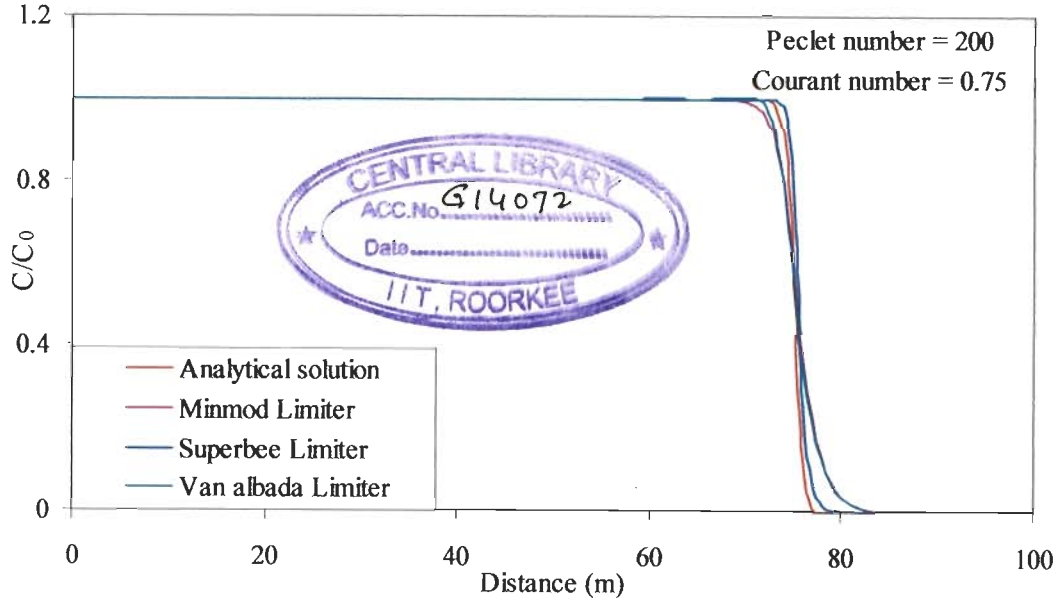

Fig 3.3(c) Effect of limiter on the numerical solution for advection dominated transport

Fig 3.4 shows the comparison of numerical and analytical solution for dispersion dominated transport ($P_e = 2$) at time equals to 150 days. Fig 3.4a indicates that numerical model also performs very well at low Peclet numbers with the numerical solution very closely matching with the analytical solution. Fig 3.4b shows the effect of Courant number on the numerical solution at low Peclet numbers. It is also seen from Fig 3.4b that Courant number does not affect on the accuracy of the numerical solution. In Fig 3.4c, it can be seen that there is no significant effect of limiters in the dispersion dominated transport. For dispersion dominated case all the limiters perform equally well while tracking the concentration profile.

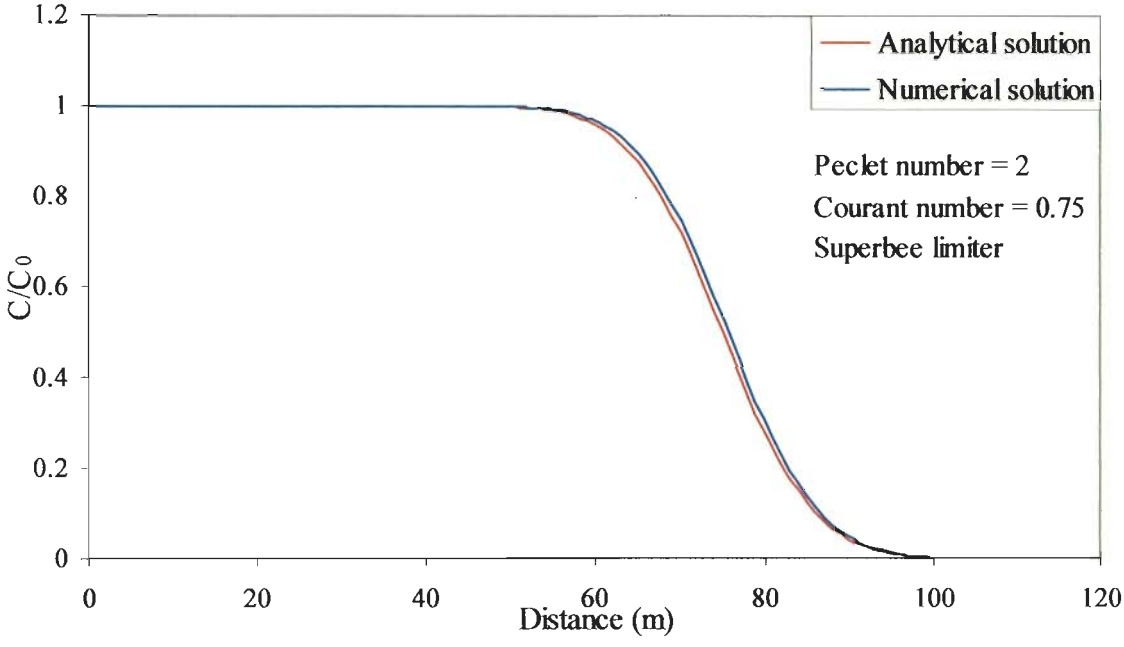

Fig 3.4(a) Comparison of analytical and numerical solution for dispersion dominated transport

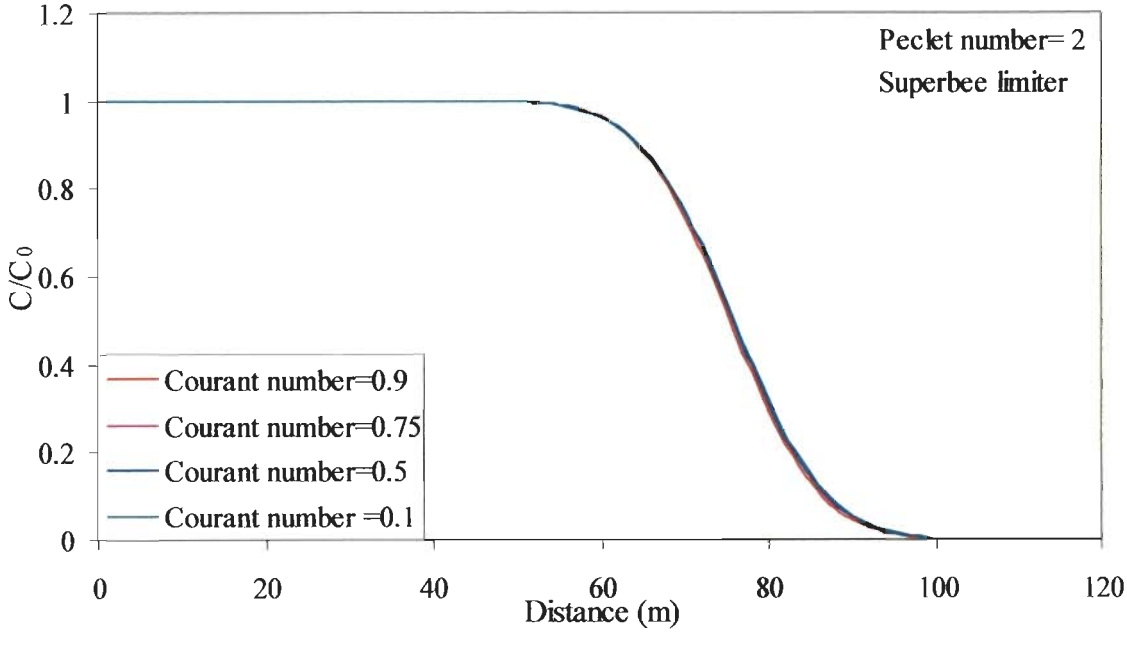

Fig 3.4(b) Effect of Courant number on the numerical solution for dispersion dominated transport

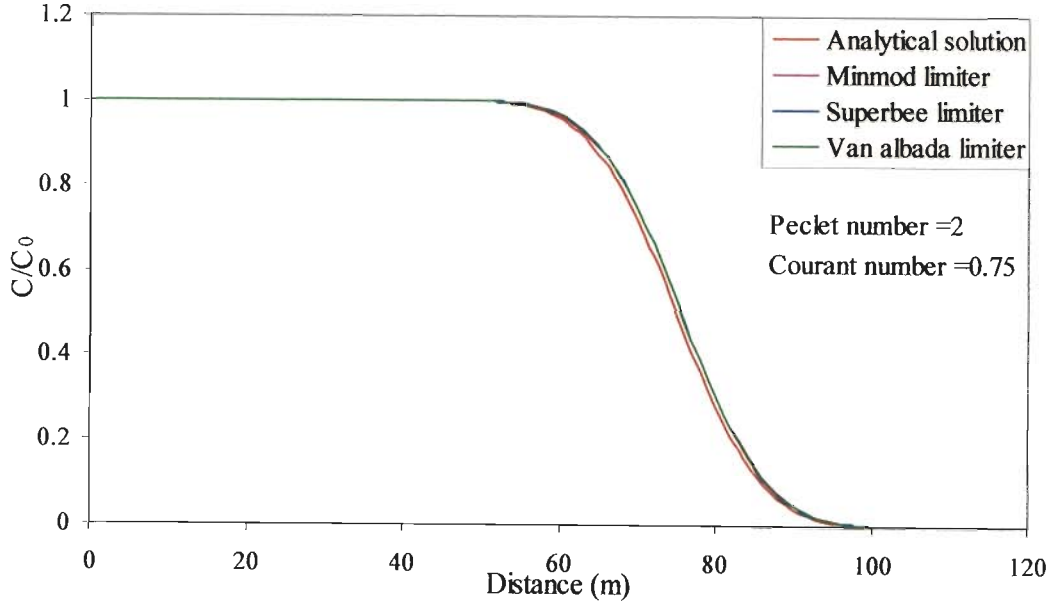

Fig 3.4(c) Effect of limiter on the numerical solution for dispersion dominated transport

3.5.2 Instantaneous Gaussian Conservative Solute Pulse

In this case, the model performance in simulating transport of a solute which has an initial distribution represented by a Gaussian hill is studied in a uniform steady flow field. For the governing equation (3.3), the initial and boundary conditions are as follows,

$$C(x,0) = C_0(x) = \exp\left(-\frac{(x-x_0)^2}{2\sigma_0^2}\right) \tag{3.30}$$

$$C(x,t) \to 0 \qquad x \to \infty \tag{3.31}$$

where $C(x,t)$ is the concentration at a distance x and time t, x_0 is the center of mass of the initial concentration field and σ_0 is the standard deviation of the initial concentration field. The analytical solution for the above case is given as (Yeh, 1990),

$$C(x,t) = \frac{\sigma}{\sigma_0} \exp\left(-\frac{\left(x-\bar{x}\right)^2}{2\sigma^2} \right) \tag{3.33}$$

where $\sigma^2 = \sigma_0^2 + 2Dt$ and $\bar{x} = x_0 + \int_0^t v(t)dt$

The numerical model is used to simulate the transport for different Peclet and Courant numbers and different limiters. The flow velocity is taken as 0.5 m/day. The domain is discretized into 100 grids so that spacing between the grids Δx is equal to 1.0m. Center of mass of the initial concentration field $x_0 = 45$m. Standard deviation of the initial concentration field $\sigma_0 = 5$m, and Courant number equal to 0.75 is used in this simulation.

Fig 3.5 shows the comparison between numerical solution and analytical solution for different Peclet numbers wherein the concentration breakthrough curves are shown with respect to distance at the end of simulation period of 150 days. It is seen from Fig 3.5 that the numerical solution is in good agreement with the exact solution for an advection dominated case (P_e = 200) with little peak clipping. It is also observed that the present numerical scheme matches very well with analytical solution for lower Peclet numbers (P_e = 2 and 20), which is expected for a second order accurate scheme. A comparison of the numerical solution with analytical solution for various Courant numbers is studied in Fig. 3.6. It is observed from Fig. 3.6 that the numerical accuracy slightly decreases with a decrease in Courant number. The effect of various limiters on the numerical solution for a high Peclet number case is presented in Fig. 3.7. It is clear from Fig. 3.7 that the Suprebee limiter captures the peak better than the other two limiters.

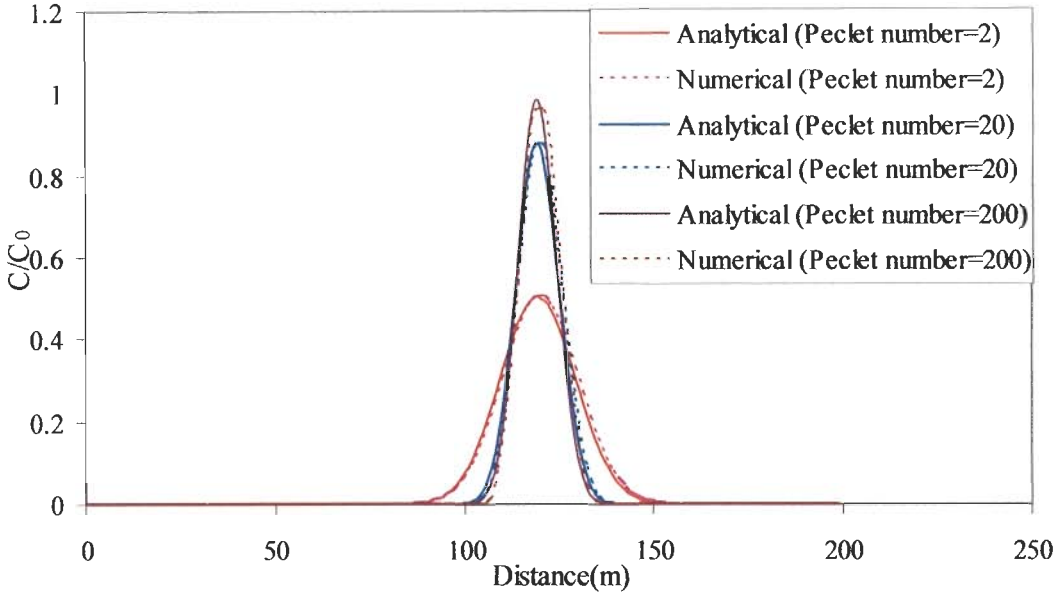

Fig 3.5 Comparison of numerical and analytical solutions for the transport of instantaneous Gaussian solute pulse

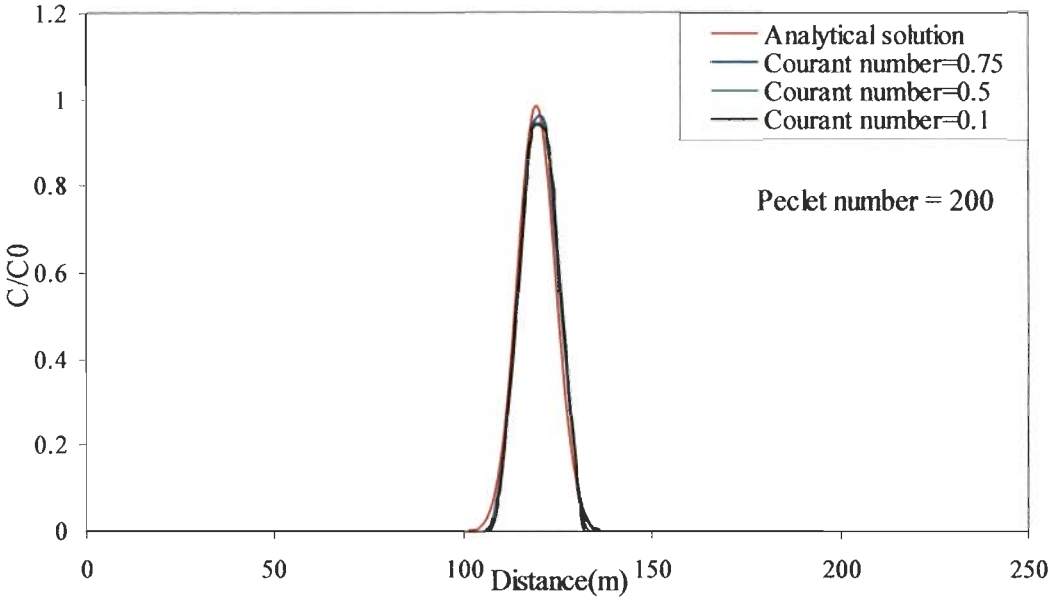

Fig 3.6 Effect of time step in the numerical solution for one dimensional transport of an instantaneous Gaussian solute pulse

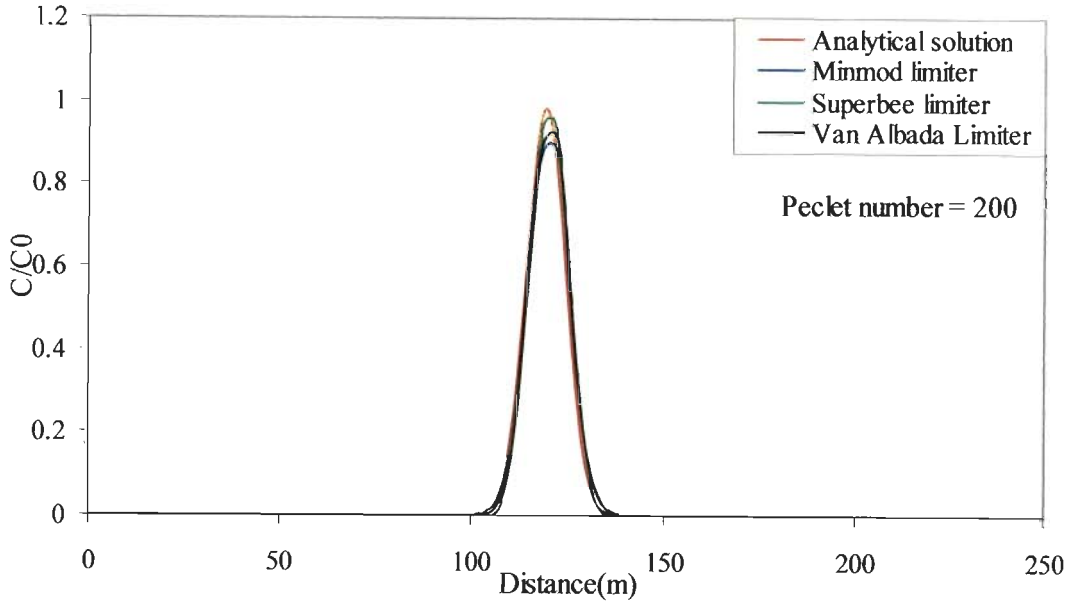

Fig 3.7 Effect of limiter in the numerical solution for advection dominated one dimensional transport of an instantaneous Gaussian solute pulse

3.5.3 Continuous Source of Finite Duration with Biodegradation

Analytical and numerical solutions are compared for various cases to assess the performance of numerical scheme for the contaminants with only biodegradation described in Eq. (3.2). The effect of Courant number on the accuracy is also studied. To compare the numerical solution with the analytical solutions, a continuous source of 2 hour duration is imposed such that the concentration at the upstream boundary is 100 units. The flow velocity and dispersion coefficient are assumed as 0.1m/s and 5.0 m^2/s respectively. The problem is solved using the finite volume method with Courant number of 0.75 and employing Superbee limiter. Let τ represent the duration of the continuous source and for this problem the initial and boundary conditions is given as follows.

$$
\begin{aligned}
&C(x,0) = 0 \qquad \text{for } x \geq 0 \\
&C(0,t) = C_0 \qquad \text{for } \tau \geq t \geq 0 \\
&C(0,t) = 0 \qquad \text{for } t > \tau \\
&C(\infty,t) = 0 \qquad \text{for } t \geq 0
\end{aligned}
\tag{3.34}
$$

The exact analytical solution for the governing equation 3.2 subject to initial and boundary conditions (Eq. 3.34) for nonconservative solute ($\lambda \neq 0$) is presented by O'loughlin and Bowmer (1975)

$$
C(x,t) = \frac{C_0}{2}\left\{\begin{aligned}
&\exp\left[\frac{vx}{2D}(1-\Gamma)\right]\left[erfc\left(\frac{x - vt\Gamma}{2\sqrt{dt}}\right) - erfc\left(\frac{x - v(t-\tau)\Gamma}{2\sqrt{d(t-\tau)}}\right)\right] \\
&+\exp\left[\frac{vx}{2D}(1+\Gamma)\right]\left[erfc\left(\frac{x + vt\Gamma}{2\sqrt{Dt}}\right) - erfc\left(\frac{x + v(t-\tau)\Gamma}{2\sqrt{D(t-\tau)}}\right)\right]
\end{aligned}\right\}
\tag{3.35}
$$

where

$$\Gamma = \sqrt{1+2H} \tag{3.36}$$

$$H = \frac{2\lambda D}{v^2} \tag{3.37}$$

Fig 3.8a and Fig 3.8b show the comparison of concentration profiles predicted by numerical and analytical solution as a function of time at 100m and 2000m from the source respectively for the case of conservative solute (λ =0). It can be seen from Fig 3.8a and Fig 3.8b that the numerical predictions are in close agreement with the exact solution for both the distances.

Fig 3.9a and Fig 3.9b show the variation of model predicted and analytically predicted concentration as a function of time at distances 100m and 2000m respectively for a nonconservative solute with λ is equal to 0.36 /hr. In this case also the model predictions are in close agreement with the analytical predictions. From Figs 3.8 and 3.9 it can be observed that the deviation between the numerical and analytical predictions decreases with distances.

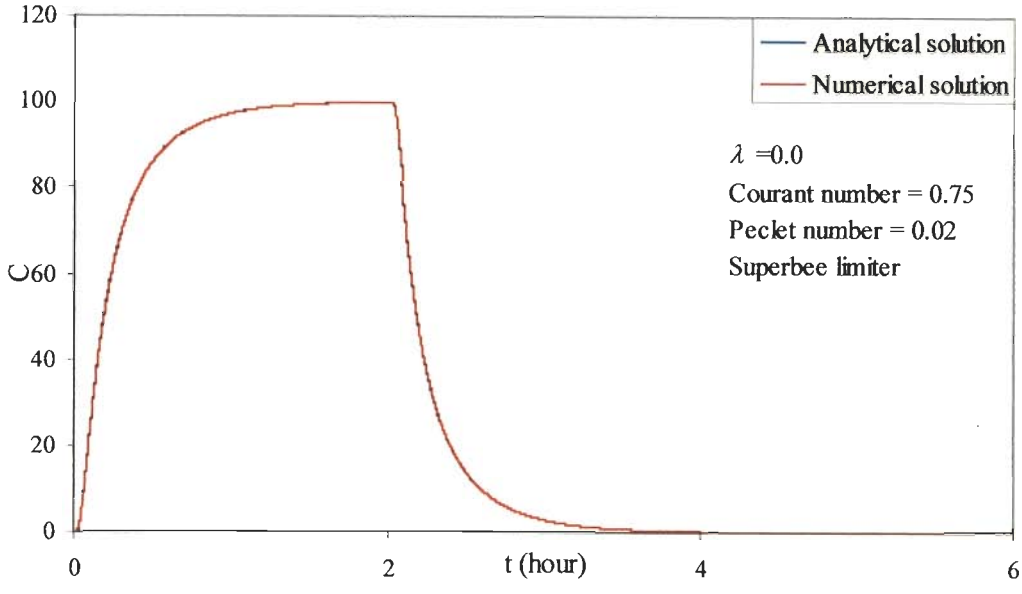

Fig 3.8(a) Comparison of analytical and numerical solution for conservative solute transport at 100m

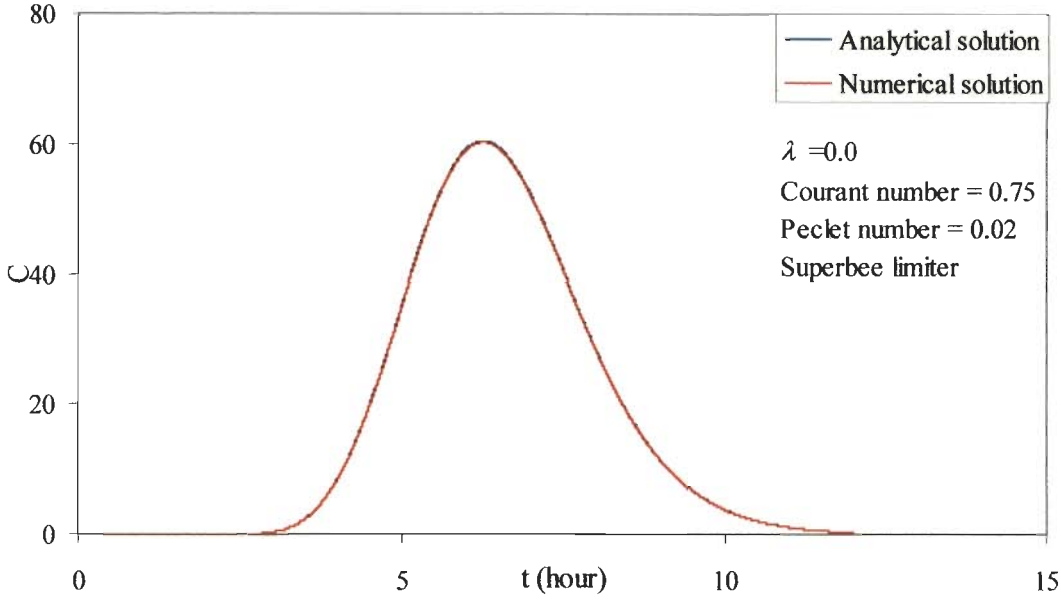

Fig 3.8(b) Comparison of analytical and numerical solution for conservative solute transport at 2000m

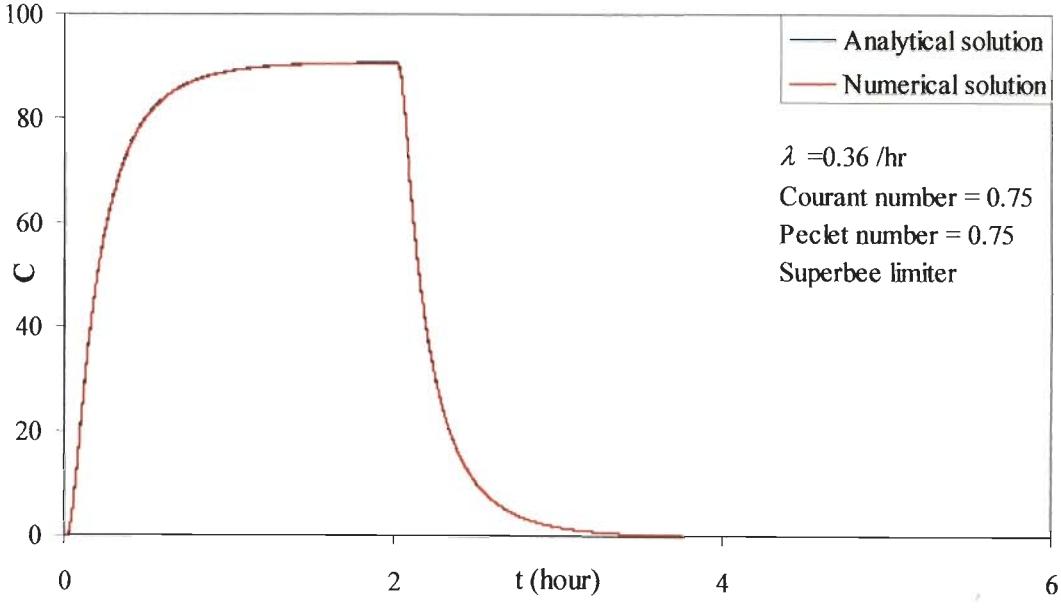

Fig 3.9(a) Comparison of analytical and numerical solution for nonconservative solute transport at 100m

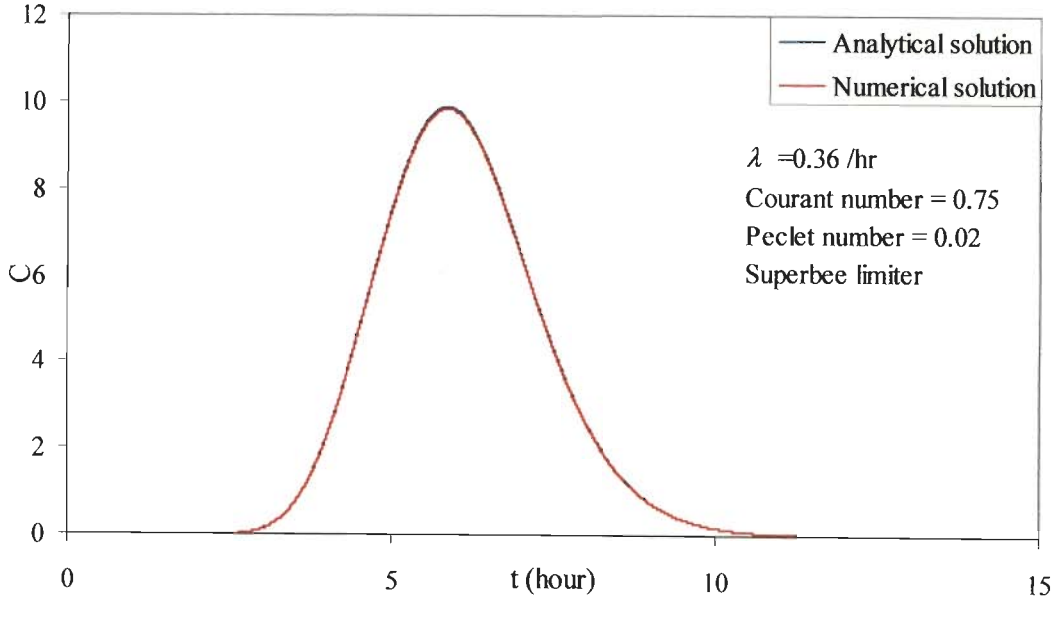

Fig 3.9(b) Comparison of analytical and numerical solution for nonconservative solute transport at 2000m

Table 3.1 shows the maximum error between the numerical and analytical solutions as a percentage of peak analytical concentration at 100 and 2000m for different values of λ. From Table 3.1, it can be seen that at a given distance the error percentage increases as the decay rate λ increases. In addition, for a given λ, the error percentage decreases as the distance increases.

Table 3.1. Maximum Error as percentage of peak concentration

Decay rate (hr)	Distance from the source	
	x = 100m	x = 2000m
$\lambda = 0.0$	0.4532	0.2903
$\lambda = 0.36$	0.4962	0.414
$\lambda = 0.72$	0.6552	0.5396

Fig 3.10 shows the effect of Courant number on the numerical solution for the nonconservative solute transport. It is seen that the Courant number has no significant effect on the accuracy of the numerical solution for the nonconservative solute transport.

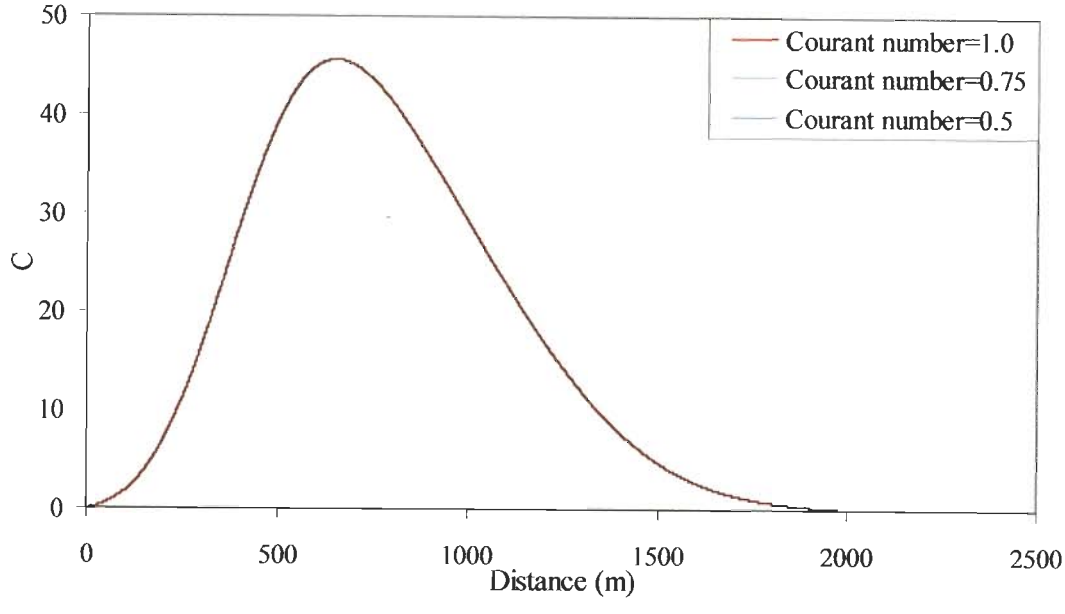

Fig 3.10 Effect of Courant number on the numerical solution for nonconservative solute transport

3.5.4 Virus Transport

The one dimensional virus transport in hydraulically homogeneous, saturated porous media accounting for virus adsorption and inactivation is given in Eq. (3.1). Virus adsorption in homogeneous porous media is commonly described by an equilibrium adsorption relationship assuming an instantaneous equilibrium between viruses in the liquid phase and onto the solid matrix (Chrysikopoulos and Sim, 1996). A general expression for the adsorption isotherm is given by

$$C^* = f(C) \tag{3.38}$$

where *f(C)* is an arbitrary function which is commonly given by either Langmuir or Freundlich model (Schijven, 2001). Maraqa (2007) investigated the effects of assumption of linear sorption on retardation of nonlinearly sorbed solutes and concluded that it is appropriate to estimate the retardation coefficient of a nonlinearly sorbed solute using a linearized isotherm if all soil particles experience sorption with liquid concentration equal to the induced concentration. For linear sorption the equilibrium solid phase concentration C^* will be linearly proportional to the equilibrium liquid phase concentration C.

$$C^* = k_d C \tag{3.39}$$

where k_d is linear distribution coefficient. Substituting Eq (3.39) in Eq. (3.1), Eq. (3.1) reduces to

$$R\frac{\partial C}{\partial t} = D\left(\frac{\partial^2 C}{\partial x^2}\right) - v\frac{\partial C}{\partial x} - \lambda C - \lambda^* \frac{\rho k_d}{\theta} C \tag{3.40}$$

where R is the retardation coefficient which is expressed as

$$R = 1 + \frac{\rho k_d}{\theta} \tag{3.41}$$

In terms of the virus concentration in the liquid phase, the virus transport equation can be written as (Jin et al., 1997)

$$R\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) - v\frac{\partial C}{\partial x} - \lambda RC \tag{3.42}$$

The analytical solution of the Eq. (3.42) subjected to boundary conditions given in Eqs. (3.4), (3.5) and (3.7) is given as (Van Genuchten and Alves, 1982)

$$\frac{C}{C_0}=\frac{1}{2}\exp\left(\frac{xv}{2D}\right)\left[\begin{array}{l}\exp\left(-\frac{x}{2D}\left(\sqrt{v^2+4\lambda D}\right)\right)erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)t}{2\sqrt{DRt}}\right)+\\ \exp\left(\frac{x}{2D}\left(\sqrt{v^2+4\lambda D}\right)\right)erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)t}{2\sqrt{DRt}}\right)\end{array}\right] \tag{3.43a}$$

The analytical solution of the Eq. (3.42) subjected to boundary conditions given in Eq. (3.34) is given as

$$\frac{C}{C_0}=\frac{1}{2}\exp\left(\frac{xv}{2D}\right)\left[\begin{array}{l}\exp\left(-\frac{x}{2D}\left(\sqrt{v^2+4\lambda D}\right)\right)\left[\begin{array}{l}erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)t}{2\sqrt{DRt}}\right)\\ -erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)(t-\tau)}{2\sqrt{DR(t-\tau)}}\right)\end{array}\right]+\\ \exp\left(\frac{x}{2D}\left(\sqrt{v^2+4\lambda D}\right)\right)\left[\begin{array}{l}erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)t}{2\sqrt{DRt}}\right)\\ -erfc\left(\frac{Rx-\left(\sqrt{v^2+4\lambda D}\right)(t-\tau)}{2\sqrt{DR(t-\tau)}}\right)\end{array}\right]\end{array}\right] \tag{3.43b}$$

To test the accuracy of the numerical scheme, the numerical solutions are compared with analytical solutions (Van Genuchten and Alves, 1982) for a wide range of Peclet numbers. These comparisons are made for continuous source of both infinite and finite durations.

3.5.4.1 Virus injection of infinite duration

In this problem, a continuous source of virus is imposed such that the concentration at the upstream boundary is 100 concentration units (C_0 = 100). The

pore water velocity, bulk density of soil and Distribution coefficient are assumed as 34 cm/day, 1.11 gm/cm^3 and 0.02 ml/gm. The domain is discretized into 100 grids so that spacing between the grids Δx is equal to 1.0cm. The governing equation 3.1 is solved numerically subject to initial and boundary conditions given in Eqs (3.4, 3.5 & 3.7). The simulation is carried out for both advection dominated (Peclet numbers equal to 100) and dispersion dominated (Peclet number equal to 1) situations.

Fig. 3.11 to Fig. 3.13 shows the comparison of numerical and analytical solutions for a continuous source of infinite duration. Fig. 3.11 shows the comparison of virus concentration after 2 days for advection dominated virus transport (P_e =100) considering inactivation coefficient $\lambda = \lambda^* = 0$. It is seen from Fig. 3.11 that the model predicted concentrations are in an excellent agreement with those predicted analytically. In contrast, Fig. 3.12 shows the comparison of virus concentration after 2 days for dispersion dominated transport (P_e =1) and inactivation coefficient $\lambda = \lambda^* = 0$. It is also seen from Fig. 3.12 that both numerical and analytical solutions are in very good agreement with each other. In addition, Fig. 3.13 shows the comparison of numerical and analytical solutions involving virus inactivation with inactivation coefficient $\lambda = \lambda^* = 0.58$ /day. In this case also, the finite volume model predicted virus concentration matches very well with the analytically obtained concentrations.

3.5.4.2 Virus injection of finite duration

This case is similar to virus injection of infinite duration except the virus is injected only for 2 days. The simulation is carried out for 6 days. Fig 3.14 shows the comparison of breakthrough curves obtained by both numerical and analytical solutions at 20m from the source. It is seen from Fig. 3.14 that the numerical model slightly underpredicts the peak as compared to the analytical solution.

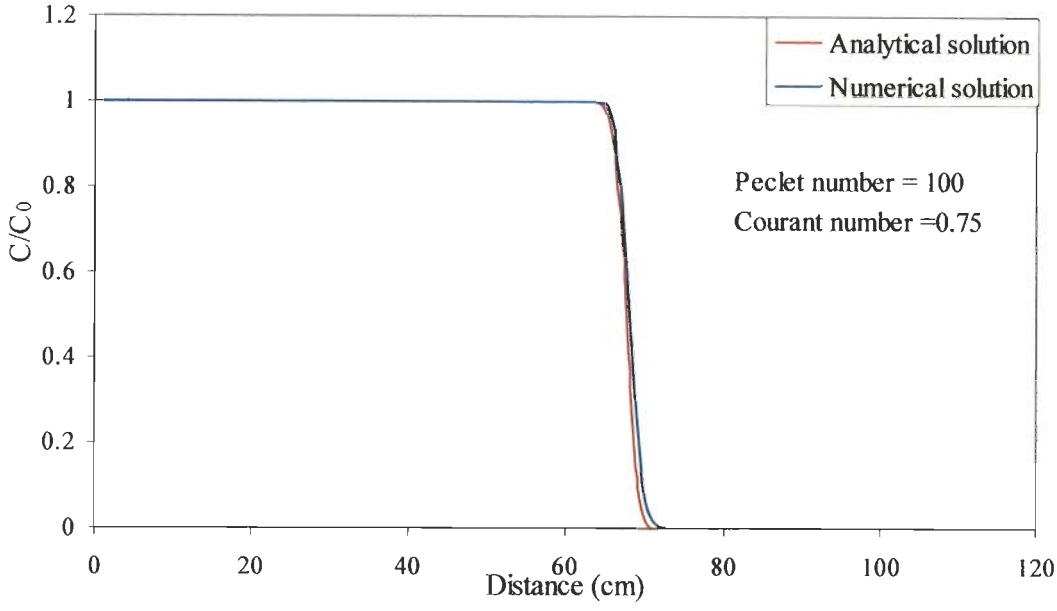

Fig 3.11 Comparison of analytical and numerical solution of advection dominated virus transport considering inactivation coefficient $\lambda = \lambda^* = 0.0$

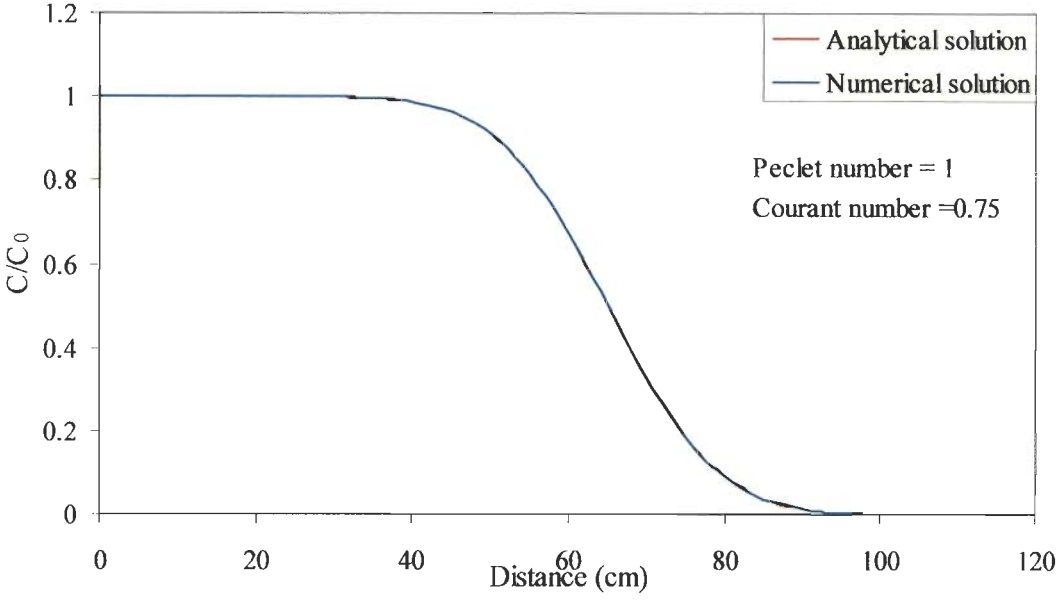

Fig 3.12 Comparison of analytical and numerical solution of dispersion dominated virus transport considering inactivation coefficient $\lambda = \lambda^* = 0.0$

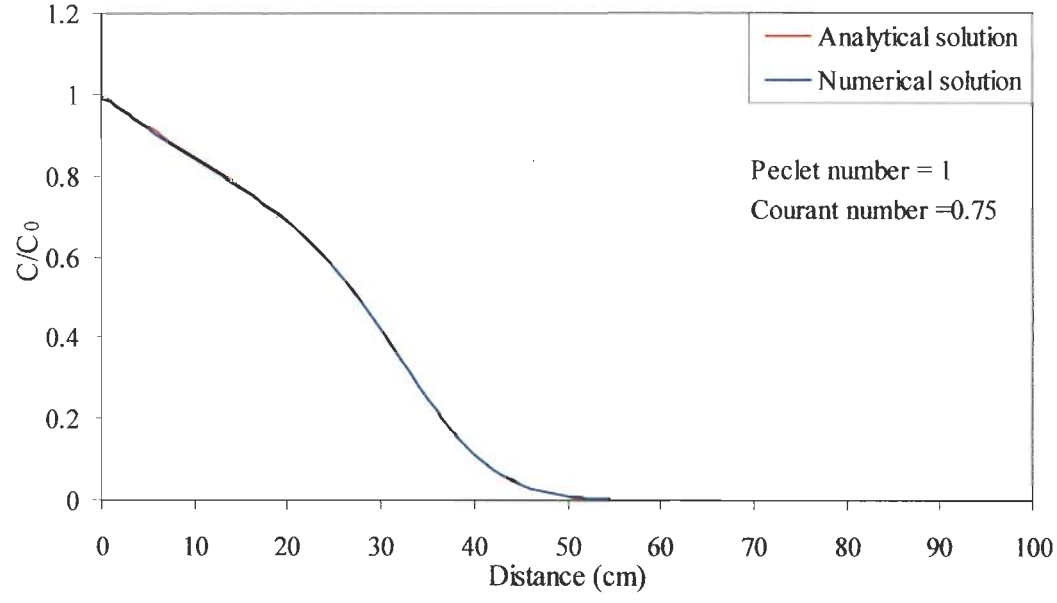

Fig 3.13 Comparison of analytical and numerical solution of virus transport considering inactivation coefficient $\lambda = \lambda^* = 0.58$/day

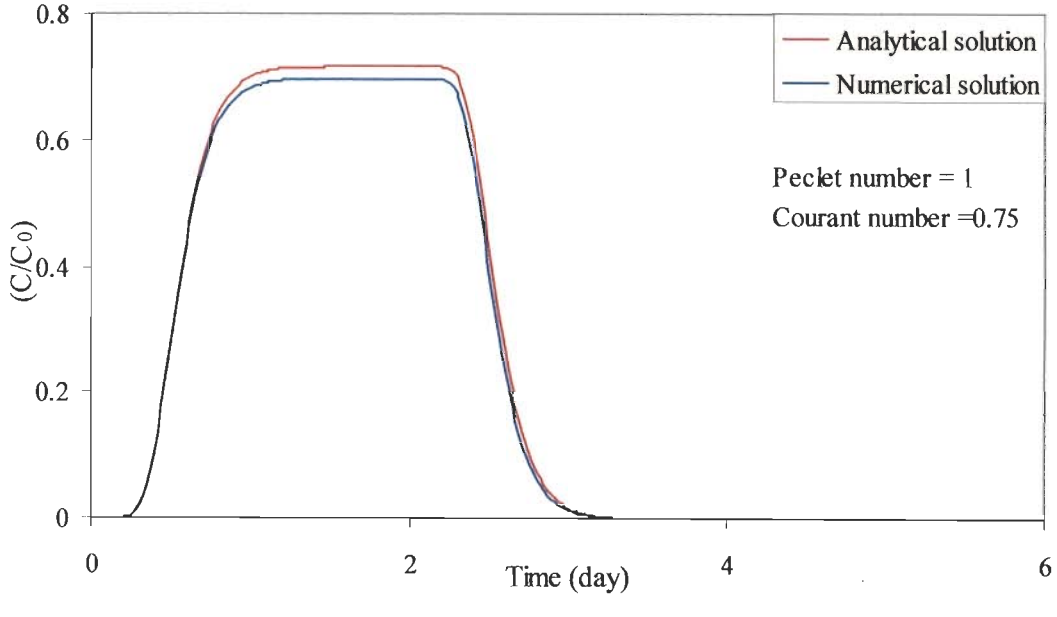

Fig 3.14 Comparison of analytical and numerical solution of virus transport at 20 cm subject to a continuous load of finite duration

3.6 CONCLUDING REMARKS

In the present study a hybrid finite volume numerical model has been developed using operator split approach for solving conservative, nonconservative solute and virus transport equation in ground water. This approach uses a globally second order accurate explicit finite volume method for the advective transport and an implicit central difference method for the dispersive transport. It is observed that the numerical model is capable of simulating transport of conservative, nonconservative solute and virus under advection dominated and dispersion dominated situations. The effect of Courant numbers on numerical solution is found to be insignificant for all cases considered. The accuracy of the model is also tested for different types of limiters. It is observed that the Suprebee limiter is least dissipative, while Minmod limiter is most dissipative among the three limiters.

CHAPTER 4

ESTIMATION OF TRANSPORT PARAMETERS IN SATURATED ZONE

4.1 INTRODUCTION

In Chapter 3, a numerical model is developed for simulating conservative and nonconservative virus transport in groundwater. The model predicts the transport of virus as a function of space and time for a given set of transport parameters. However, the accurate prediction depends upon the reliability of the transport parameters used in the analysis. Virus transport parameters are usually estimated by conducting laboratory column experiments and the parameters are obtained by direct numerical inversion of the governing equation (Kool et al., 1987). These methods are time consuming and require rather restrictive initial and boundary conditions. In addition, the parameters so obtained may not be applicable at field scale level. An alternative to these methods are the estimation of transport parameters using inverse procedure. In such an approach, the transport parameters are obtained by minimizing the deviation between the model predicted and experimentally observed virus concentrations in an experiments. The advantage of such an approach is that the experiments can be selected on the basis of convenience and expeditiousness, rather than by a need to simplify the mathematics of the direct inversion process. The main disadvantage of inverse procedure is that the inverse problem is often ill posed (Carrera and Neumann, 1986). The illposedness may be due to non-identifiability, nonuniqueness or instability.

In the present Chapter, a parameter estimation algorithm is developed to estimate virus transport using inverse procedure. For this purpose, the numerical model developed in Chapter 3 is coupled with Levenberg-Marquadart optimization algorithm. The identifiability of the transport parameters is discussed in detail by estimating the transport parameters from synthetically generated virus concentration data. The effect of errors in the data on the estimated parameters is also studied and a detailed statistical analysis is carried out to study the bias induced by the objective function at different noise levels.

4.2 GENERAL FORMULATION OF THE ESTIMATION PROBLEM

The inverse problem is formulated as a nonlinear optimization problem i.e. the transport parameters are estimated by minimizing the deviation between observed and model predicted response as

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^T \mathbf{W}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right] \tag{4.1}$$

where the objective function, $O(\mathbf{b})$, is a function of the model parameters $\mathbf{b}$, $\mathbf{b} = \{b_1, b_2, - - - b_m\}^T$; $\mathbf{C}^* = \{C_1^*, - - - C_n^*\}^T$ is the observation vector whose elements represent measured concentrations; $\mathbf{C}(\mathbf{b}) = \{C_1(b), - - - C_n(b)\}^T$ represents the predicted response for a given parameter vector $\mathbf{b}$ and $\mathbf{W}$ is the symmetric weighting matrix. In the present study, the parameter vector $\mathbf{b}$ comprises inactivation coefficient in liquid phase λ, inactivation coefficient in sorption phase λ^*, dispersion coefficient D, and distribution coefficient k_d i.e. $\mathbf{b} = \{\lambda, \lambda^*, D, k_d\}^T$. The objective is to find the optimum parameter vector $\mathbf{b}$ that minimizes the objective function 4.1. When the observation errors are assumed to be independent and normally distributed the

weighting matrices **W** becomes an identity matrix and Eq. (4.1) reduces to ordinary least squares (OLS) problem.

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^T \left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right] = \frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^2 \tag{4.2}$$

where N is the number of observations.

The OLS formulation has probably been the most popular one for parameter estimation. Its attraction is due to its simplicity and because it requires a minimum amount of information. When observation errors are normally distributed, are uncorrelated and have a constant variance, the OLS estimates possess optimal statistical properties (Kool et al., 1987). When these conditions are not met, the OLS method will no longer yield optimal parameter estimates in terms of precision and minimum variance.

4.3 SOLUTION ALGORITHM

In the present section, Levenberg-Marquadart algorithm is used to optimize the objective function (4.2) and is explained in detail. Let **e** represents the vector of residuals defined as

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} C_1^* - C_1(\mathbf{b}) \\ C_2^* - C_2(\mathbf{b}) \\ \vdots \\ C_n^* - C_n(\mathbf{b}) \end{bmatrix} \tag{4.3}$$

Then Eq. (4.2) can be written as

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\mathbf{e}^T\mathbf{e} = \frac{1}{2}\sum_{i=1}^{N}\mathbf{e}_i^2 \tag{4.4}$$

The objective function (4.4) is a nonlinear function of the parameter vector **b** and hence the minimization has to be carried out iteratively. At each iteration *i*, the

parameter correction vector $\Delta\mathbf{b}^i$ is determined such that

$$O\left(\mathbf{b}^i + \Delta\mathbf{b}^i\right) \leq O\left(\mathbf{b}^i\right) \tag{4.5}$$

Newton's method can be derived by writing 3-term Taylor series expansion for $O(\mathbf{b})$ around $\mathbf{b}^i$.

$$O\left(\mathbf{b}^i + \Delta\mathbf{b}\right) = O\left(\mathbf{b}^i\right) + \nabla O\left(\mathbf{b}^i\right)^T \Delta\mathbf{b} + \frac{1}{2}\Delta\mathbf{b}^T \nabla^2 O\left(\mathbf{b}^i\right)\Delta\mathbf{b} + small \tag{4.6}$$

One wishes to select $\Delta\mathbf{b}^i$ such that $O\left(\mathbf{b}^i + \Delta\mathbf{b}^i\right)$ is approximately minimized. Newton's method (Kool et al., 1987) for obtaining the parameter correction leads to the system of equation

$$\left(\mathbf{J}^T\mathbf{J} + \mathbf{S}\right)\Delta\mathbf{b}^i = -\mathbf{J}^T\mathbf{e} \tag{4.7}$$

where $\mathbf{J}$ is a Jacobian matrix whose columns contain the partial derivatives of the residuals $\mathbf{e}$ with respect to the elements of the parameter vector $\mathbf{b}$ and $\mathbf{S}$ is a matrix containing second derivative of $\mathbf{e}$ with respect to element of $\mathbf{b}$. The implementation of Newton's method is computationally expensive, because of computation of second derivatives and hence is not generally adopted in practice (Kool et al., 1987).

In Gauss-Newton method, $\mathbf{S}$ is neglected in Eq. (4.7) and $\Delta\mathbf{b}^i$ is obtained by solving the system of equations.

$$\mathbf{J}^T\mathbf{J}\Delta\mathbf{b}^i = -\mathbf{J}^T\mathbf{e} \tag{4.8}$$

Jacobian has dimension $N \times P$, where N is the number of observations and P is the number of unknown parameters to be estimated. The element of $\mathbf{J}$ are obtained by forward finite difference approximation as

$$J_{i,j} = \frac{\partial e_i}{\partial b_j} = -\frac{\partial \hat{C}_i}{\partial b_j} \approx \frac{\hat{C}(x_i, t_i; b_j + \delta b_j) - \hat{C}(x_i, t_i; b_j)}{\delta b_j} \tag{4.9}$$

where $\delta b_j = 0.01 \times b_j$ with j denotes the index for the parameter.

$$\left(\mathbf{J}^T\mathbf{J} + \beta\mathbf{D}^T\mathbf{D}\right)\Delta\mathbf{b} = -\mathbf{J}^T\mathbf{e} \tag{4.10}$$

In Eq. (4.10) β is a positive scalar and $\mathbf{D} = diag(d_1, d_2, ...d_p)$ is a scaling matrix that takes into account differences in the magnitude of the sensitivities for the different parameters. Following Kool and Parker (1988) the elements of $\mathbf{D}$ are updated in each iteration i, as

$$\begin{aligned} d_j^i &= \left\|\mathbf{J}_j^i\right\| \\ d_j^i &= \max\left\{d_j^{i-1}, \left\|\mathbf{J}_j^i\right\|\right\} \quad i > 0 \end{aligned} \tag{4.11}$$

where $\mathbf{J}_j$ denotes the jth column of $\mathbf{J}$ and the vertical bars denote the Euclidian norm. The parameter β controls both the step direction and step length. Following Osborne (1976) and More (1977) scheme for updating β is coupled with the solution of Eq. (4.10) as discussed below. The system of Eq. (4.10) can be represented as normal equations for the linear least squares problem as

$$\min_{\Delta\mathbf{b}} e^T e = \left\| \begin{pmatrix} \mathbf{J} \\ \beta^{1/2}\mathbf{D} \end{pmatrix} \Delta\mathbf{b} - \begin{pmatrix} -\mathbf{e} \\ 0 \end{pmatrix} \right\|^2 \tag{4.12}$$

The minimum of Eq. (4.12) is found in two steps. In the first step Eq. (4.12) is multiplied by an orthogonal Householder matrix $\mathbf{Q}^T$ so that

$$\mathbf{Q}^T\mathbf{J} = \mathbf{R}_1 \tag{4.13}$$

where $\mathbf{R}_1$ is $P \times P$ upper triangular. The right hand side of Eq. (4.12) becomes

$$\left\| \begin{pmatrix} \mathbf{R}_1 \\ 0 \\ \beta^{1/2}\mathbf{D} \end{pmatrix} \Delta\mathbf{b} + \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \\ 0 \end{pmatrix} \right\|^2 \tag{4.14}$$

Then the subdiagonal elements $\beta^{1/2}$ $\mathbf{D}$ in Eq. (4.14) are zeroed out by applying a series of orthogonal Givens Rotations which yield

$$\left\| \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} \Delta\mathbf{b} + \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \right\|^2 \tag{4.15}$$

where $\mathbf{R}$ is $P \times P$ upper triangular and the subvectors $\mathbf{f}$ and $\mathbf{g}$ have lengths P and N respectively. The minimum of Eq. (4.12) can be obtained if parameter corrections $\Delta\mathbf{b}$ are chosen such that

$$\mathbf{R}\Delta\mathbf{b} = -\mathbf{f} \tag{4.16}$$

The accepted value of β is one that reduces the value of the objective function. After obtaining the acceptable value, β is updated in each iteration by computing γ as

$$\gamma = \frac{\mathbf{e}^T\mathbf{e} - \mathbf{e}_+^T\mathbf{e}_+}{\mathbf{f}^T\mathbf{f}} \tag{4.17}$$

where, $\mathbf{e} \equiv \mathbf{e}(\mathbf{b})$ *and* $\mathbf{e}_+ \equiv \mathbf{e}(\mathbf{b} + \Delta\mathbf{b})$

If $\gamma \geq 0.75$, β is decreased by multiplying it by a factor $\omega_2 \leq 1$. If $\gamma \leq 0.25$, β is increased by multiplying it by a factor $\omega_1 \geq 1$. The optimization is terminated when either of the following termination criteria is satisfied.

$$\mathbf{f}^T\mathbf{f} \leq \left(1 + \mathbf{e}^T\mathbf{e}\right)^{1/2} \times \varepsilon_1 \tag{4.18}$$

$$\frac{|\Delta b_i|}{b_i + 10^{-20}} \leq \varepsilon_2 \tag{4.19}$$

where ε_1 and ε_2 are the specified tolerance limits. The following values are used for optimization parameters in the study; β_{initial} =1.0. ω_1=1.75, ω_2=0.25, ε_1 = 0.000002 and ε_2 = 0.0001.

As the objective function does not exhibit a convex nature uniformly over the entire parametric space, it is found necessary to impose some constraints on the corrections to ensure convergence to the global minimum, from the initial guess in any region of the parametric space. The constraints proposed by Cobb et al. (1982) on

the relative changes of the parameter are imposed in the computations which is given by

$$-0.2 \leq \frac{\Delta b_j^i}{b_j^i} \leq 0.5 \tag{4.20}$$

Cobb et al., (1982) observed that with these constraints the convergence to true values can be obtained even when initial assumptions differs from true values by orders of magnitude. These constraints are incorporated in the Levenberg-Marquardt algorithm in the present study.

A computer code is written in FORTRAN 90 to implement the inverse procedure and is presented in APPENDIX-II.

4.4 IDENTIFICATION OF TRANSPORT PARAMETERS

In the present Section, the parameter estimation algorithm developed in Section 4.3 is used to estimate the virus transport parameters λ, D, λ^* *and* k_d. Initially it is checked whether the optimization yields unique estimates of the transport parameters from hypothetical generated virus concentration data and the identifiability of the parameters are discussed. Later, the effect of noise (errors) in the measurements on the parameter estimates is studied.

Hypothetical virus concentration data:

Hypothetical data of virus concentration as a function of time are generated by solving Eq. (3.1) subject to initial and boundary conditions given by Eqs. (3.4), (3.5) and (3.7). The initial concentration of virus in flow domain is considered to be zero. A virus concentration of 1 unit is applied continuously at the source. The virus concentration far away from the source is assumed to be zero. A steady groundwater velocity (v) of 34 cm/day is considered. The transport parameters used in

the simulation are λ = 0.58 /day, D = 34 cm^2/day, λ^* = 0.46/day, k_d = 0.02 ml/gm, ρ = 1.11 gm/cm^3 and θ = 0.4. Virus concentration at discrete times (0.5, 1.0, 1.5, 2.0, and 2.5 days) and at a discrete distance from the source (1, 3, 5, 7, 9, 11, 13, 15, 17, 20, 22 cm) is generated by solving Eq (3.1) subjected to initial and boundary conditions given in Eq. (3.4), Eq. (3.5) and Eq. (3.7). Table 4.1 presents the hypothetically generated virus concentration data. These data are used as observed concentration data in the parameter estimation.

Table 4.1: Hypothetical virus concentrations data for parameter estimation

Time (Days) / Distance in cm	0.5	1.0	1.5	2.0	2.5
1	0.96962	0.98128	0.98193	0.98193	0.98193
3	0.88107	0.94213	0.94773	0.94785	0.94785
5	0.74232	0.89680	0.91364	0.91496	0.91496
7	0.55359	0.84059	0.87860	0.88315	0.88320
9	0.33772	0.76859	0.84116	0.85201	0.85255
11	0.15664	0.67732	0.79944	0.82074	0.82295
13	0.05572	0.56651	0.75127	0.78865	0.79428
15	0.01554	0.44019	0.69455	0.75487	0.76614
17	0.00348	0.30658	0.62775	0.71831	0.73792
20	0.00026	0.13602	0.50802	0.65573	0.69438
22	0.00004	0.06578	0.41728	0.60726	0.66369

The robustness of the optimization procedure is studied by changing the number of transport parameters to be estimated from 1 to 4. In addition, the efficacy of the optimization procedure is analyzed by starting the initial guesses of individual parameters considerably far away from their true values. The parameter estimation is discussed in detail in the following Section.

4.4.1 Case 1: Estimation of One Unknown Parameter

Case 1 considers the estimation of one unknown transport parameters while treating the other three parameters as constant to their respective values used for the generation of hypothetical data. Further two sub cases (case A and case B) are considered. In case A, the initial guess parameter is over estimated by one order from its true value while in case B it is under estimated by one order from its true value. Table 4.2 presents the initial guess values and the optimal estimated values of the transport parameter λ, λ^*, D, and k_d. It is clear from Table 4.2 that the optimization algorithm converges to the true values in both sub cases (case A and Case B). Further, it can be seen from Table 4.2 that starting the initial guess as over estimated value results in less number of iterations to converge to the optimal solution.

4.4.2 Case 2: Estimation of Two Unknown Parameters

In case 2, two among the four transport parameters are considered as unknown and are estimated while keeping other two parameters a constant to their respective values used for the generation of the hypothetical data. Such an estimation results in six combinations of two unknown parameters; (λ, D), (λ, k_d), (λ, λ^*), (D, k_d), (λ^*, D) *and* (λ^*, k_d). For each of these combinations, four subcases are considered. In case A, the initial guess of the parameters are over estimated by one order from their true values. In case B, the initial guess of the parameters are underestimated by one order. In case C, the initial guess of the first parameter is over estimated by one order while the initial guess of the second parameter is under estimated by one order. In contrast in case D, the initial guess of the first parameter is under estimated by one order while initial guess of second parameter is over estimated by one order. During optimization, it is observed that for the particular combination in which the inactivation coefficients λ and λ^* are considered as unknown parameters, the optimization resulted in non

unique solutions. The identifiability of these two parameters is discussed later in this Section. Table 4.3 presents the initial guess values and the final estimated values of the parameter for the other five combinations of parameters considering all the four sub cases (case A to case D). It is seen from Table 4.3 that the optimization algorithm yields true values of the parameters in all the four sub cases. Table 4.3 also suggests that the convergence to the optimal solution is most rapid when initial guesses are over estimated. The convergence of the optimization to the true values for the five combinations discussed is further evidenced by the convexity of the objective function in their respective parametric spaces. Fig 4.1 to 4.5 show the contours of the objective function in the parametric spaces (λ, D), (λ, k_d), (D, k_d), (λ^*, D) and (λ^*, k_d) respectively. It can be seen from Figs 4.1 to 4.5 that the objective function ϕ is strictly convex in nature with one global minimum at their true values in all these five parametric spaces. This results in the optimization algorithm converging to the true values from any point in the parametric space.

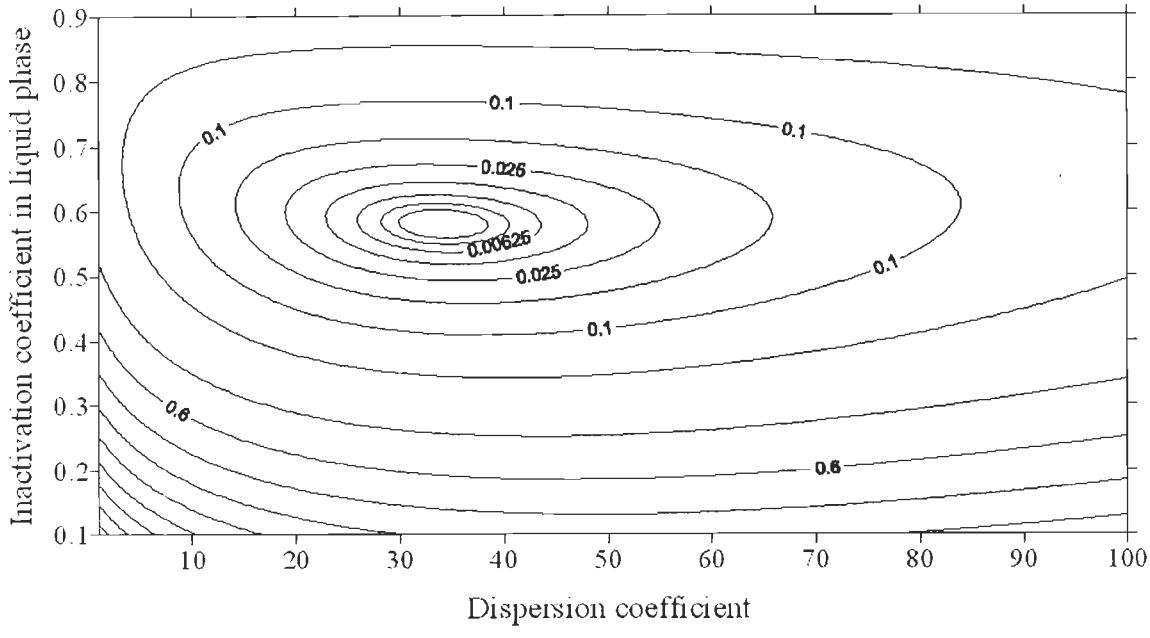

Fig 4.1 Contour showing objective function as a function of dispersion coefficient D (cm^2/day) and inactivation coefficient in liquid phase λ (/day).

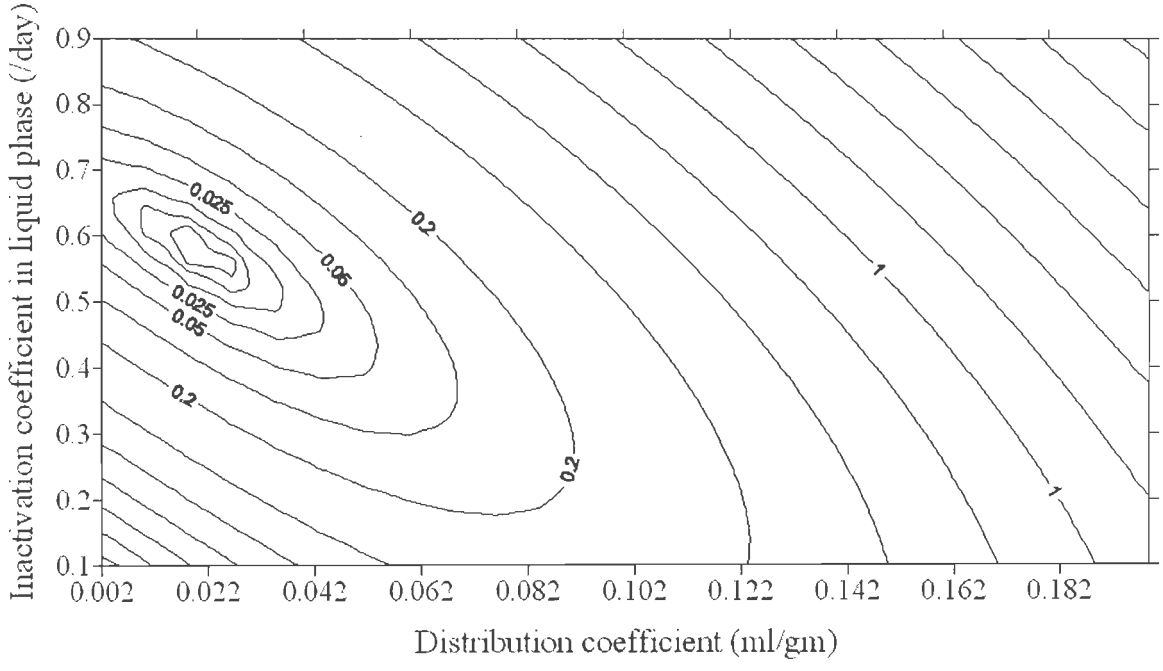

Fig 4.2 Contour showing objective function as a function of distribution coefficient k_d (ml/gm) and inactivation coefficient in liquid phase λ (/day).

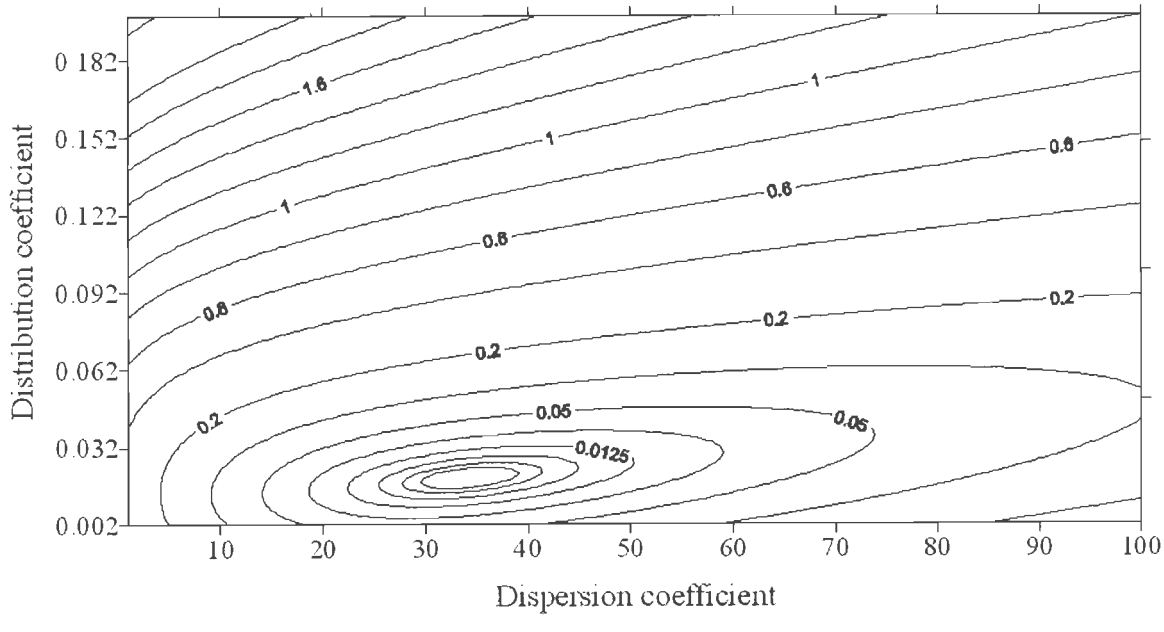

Fig 4.3 Contour showing objective function as a function of dispersion coefficient D (cm^2/day) and distribution coefficient k_d (ml/gm).

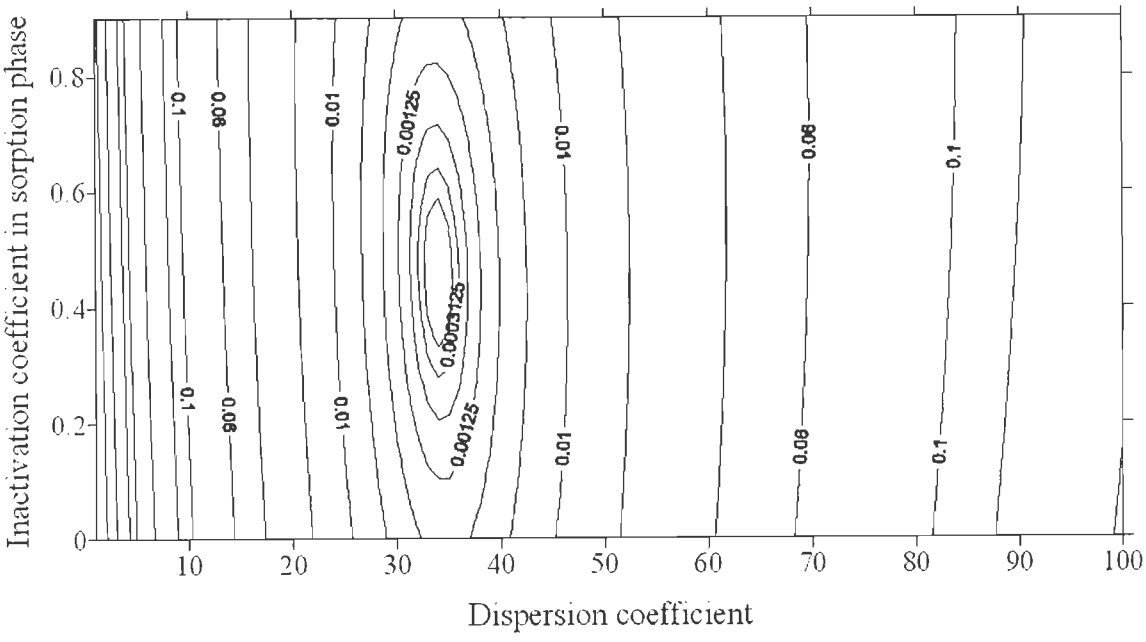

Fig 4.4 Contour showing objective function as a function of dispersion coefficient D (cm^2/day) and inactivation coefficient in sorption phase λ^* (/day).

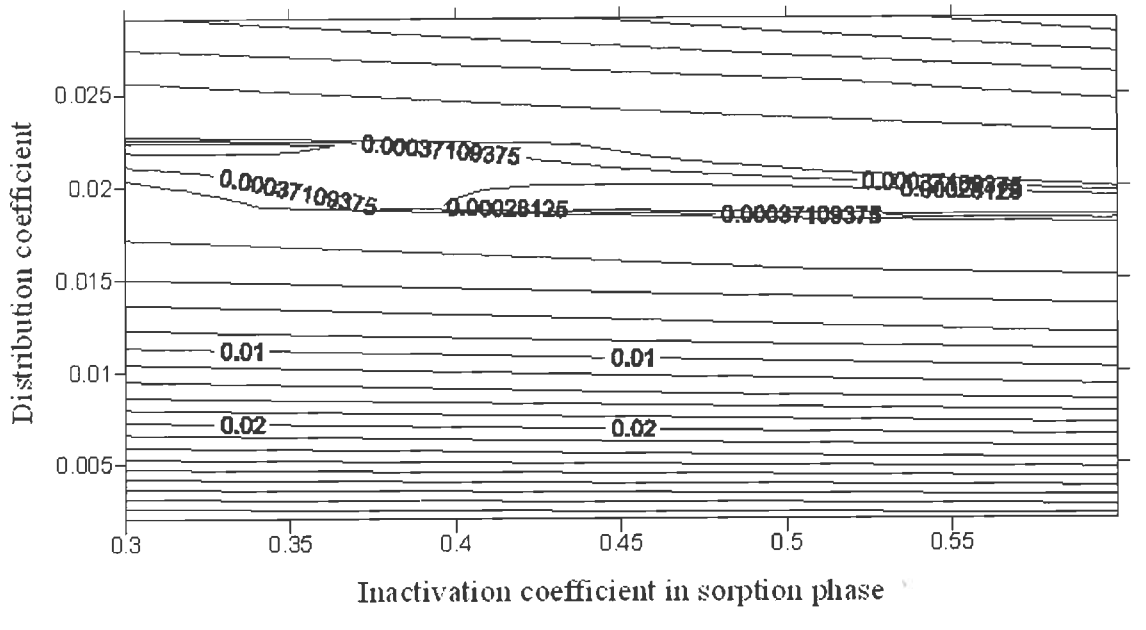

Fig 4.5 Contour showing objective function as a function of distribution coefficient k_d (ml/gm) and inactivation coefficient in sorption phase λ^* (/day).

4.4.3 Identifiability of Inactivation Coefficients λ and λ^*

The movement of viruses in ground water is greatly influenced by inactivation (Bales et al., 1991) and it has been observed that the inactivation process is different in aqueous and sorbed phases and inactivation coefficients vary significantly for these two phases (Gerba, 1984). An attempt has been made to test whether the inverse procedure uniquely estimates the inactivation coefficients from the measured virus concentration data. For this purpose optimization is carried out by treating λ and λ^* as unknown parameters. Table 4.4 presents the initial guess values and the optimal estimates of parameters λ and λ^* for four sub cases of over estimated, under estimated and mixed. It is evident from Table 4.4 that the optimization results in nonunique estimates of λ and λ^*. Further it can be observed that in all the sub cases, the estimated values of the parameter λ are close to the true value. However, the estimates of parameter λ^* deviates significantly from the true value. Convexity of the objective function ϕ on the λ - λ^* parametric space is studied and is shown in the form of contours in Fig. 4.6.

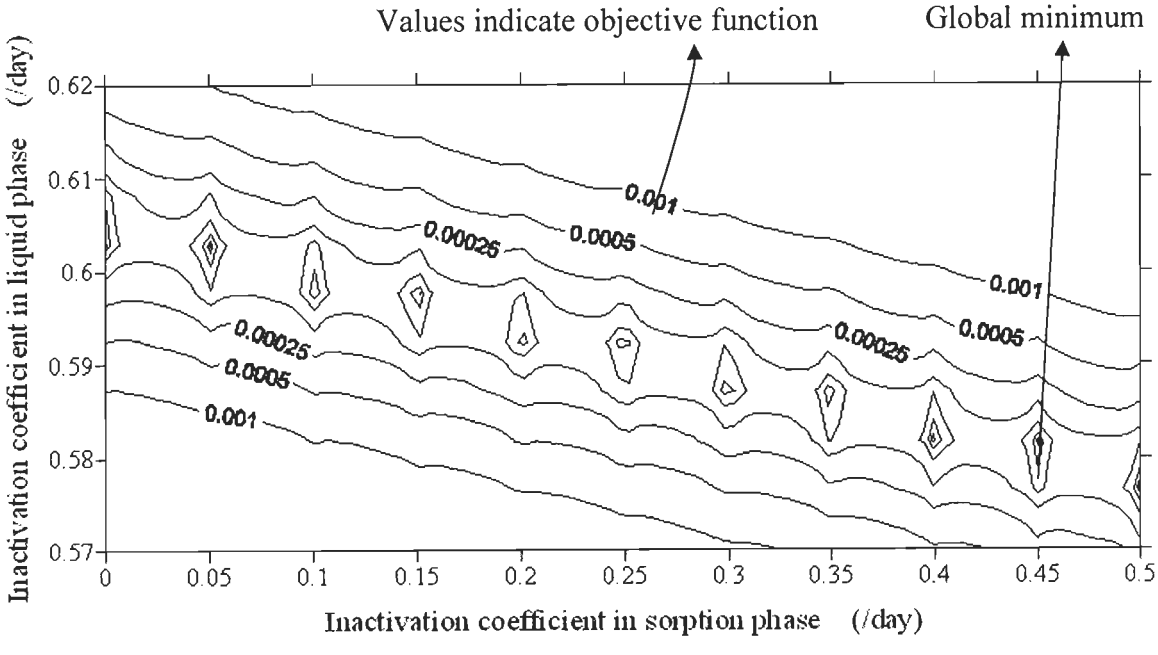

Fig 4.6 Contour showing objective function as a function of inactivation coefficient in sorption phase λ^* (/day) and inactivation coefficient in liquid phase λ (/day).

From Fig. 4.6, it is clear that the objective function has many local minima in λ- λ^* parametric space and hence an optimal estimate depends upon the initial guess values from which the optimization is started. It can also be inferred from Fig. 4.6 that the local minima are concentrated over a small range around the true value of the parameter λ (= 0.58/day) resulting in parameter estimates close to the true value. However, for the case of λ^*, the local minima is spread over a wide range around its true value (= 0.46/day). As a result, the optimal parameter estimates deviate significantly from the true value.

4.4.4 Case 3: Estimation of Three Unknown Parameters

Case 3 considers the estimation of three unknown transport parameters simultaneously while keeping the remaining one parameter as constant to its respective value used for generation of hypothetical data. Such an estimation results in four combination of three unknown parameters; (λ, D, k_d), (λ, λ^*, k_d), (λ, λ^*, D) and (λ^*, D, k_d). For each of three combinations, three subcases are considered. In case A, initial guess parameters are over estimated by one order from their true values, while for case B, the initial guess parameters are underestimated by one order and case C considers the case where initial guess of some parameters are over estimated while the initial guess of rest of the other parameters are underestimated. In case 3 also, it is observed that the simultaneous presence of inactivation coefficient λ and λ^* in any combination of unknown parameters, the algorithm converges to non unique solution. However, if one of the parameter (λ or λ^*) is considered as a known parameter, the optimization resulted in the unique estimation of the other three unknown parameters as shown in Table 4.5. From Table 4.5, it can be seen that if the initial guess values are over estimated, the optimization required less number of iterations for convergence as compared to the cases of under estimation and mixed.

Table 4.2: Parameter estimates for the hypothetical data – Case 1

Parameter	True values	Case A (over estimated)			Case B (Under estimated)		
		Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations
λ (/day)	0.58	5.8	0.58	6	0.058	0.5797	8
D (cm^2/day)	34	340.0	34.0	9	3.4	34.0	10
λ^* (/day)	0.46	4.6	0.4609	4	0.046	0.4586	4
K_d (ml/gm)	0.02	0.2	0.02	4	0.002	0.02	5

Table 4.3: Parameter estimates for the hypothetical data – Case 2

Parameters	True values	Case A (over estimated)			Case B (Under estimated)			Case C (Mixed) Type I			Case D (Mixed) Type II		
		Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations
λ (/day)	0.58	5.8	0.58	16	0.058	0.58	34	0.058	0.58	21	5.8	0.5799	40
D (cm^2/day)	34	340.0	34.0		3.4	33.95		340.0	34.0		3.4	33.985	
λ (/day)	0.58	5.8	0.58	23	0.058	0.5797	30	0.058	0.58	23	5.8	0.5797	35
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.2	0.02		0.002	0.0199	
D (cm^2/day)	34.0	340.0	34.0	21	3.4	33.93	45	340.0	33.99	38	3.4	34.0039	25
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.002	0.0199		0.2	0.02002	
λ^* (/day)	0.46	4.6	0.461	18	0.046	0.468	20	4.6	0.4614	20	0.046	0.4615	35
D (cm^2/day)	34.0	340.0	34.0		3.4	33.987		3.4	34.1		340.0	33.15	
λ^* (/day)	0.46	4.6	0.4674	27	0.046	0.461	32	4.6	0.463	29	0.046	0.461	28
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.002	0.02		0.2	0.02102	

Table 4.4: Non uniqueness of decay parameters λ and λ^*

Parameters	True values	Case A (over estimated)		Case B (under estimated)		Case C (Mixed) Type I		Case D (Mixed) Type II	
		Initial guess	Final estimated values	Initial guess	Final estimated values	Initial guess	Final estimated values	Initial guess	Final estimated values
λ(/day)	0.58	5.8	0.5191	0.058	0.5998	5.8	0.6034	0.058	0.3477
λ^*(/day)	0.46	4.6	1.553	0.046	0.1055	0.046	0.0409	4.6	4.6421

Table 4.5: Parameter estimates for the hypothetical data –Case 3

Parameters	True values	Case A (over estimated)			Case B (under estimated)			Case C (mixed)		
		Initial guess	Final estimated values	No of iterations	Initial guess	Final estimated values	No of iterations	Initial guess	Final estimated values	No of iterations
λ (/day)	0.58	5.8	0.5797	28	0.058	0.5797	48	0.058	0.58	42
D (cm^2/day)	34.0	340.0	34.041		3.40	33.9		340.0	33.943	
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.002	0.0199	
λ^* (/day)	0.46	4.6	0.459	27	0.046	0.4535	49	0.046	0.4585	39
D (cm^2/day)	34.0	340.0	33.96		3.40	33.9		340.0	33.94	
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.002	0.0199	

4.5 DATA ERROR AND BIAS

In Section 4.4, the performance of the optimization algorithm is studied by estimating parameters from hypothetically generated error free virus concentration data. As stated by Kool et al. (1987), the reliability of the parameters estimates obtained by inverse procedure greatly depend upon the quality of the experimental data. In addition, Khatibi et al. (1997) have demonstrated that the nature of the objective function may induce bias in the parameter estimates obtained from inverse procedure. The present Section discusses the effect of data errors and bias induced by the objective function on the transport parameter estimates. The least square minimization techniques based on the following assumptions i) the error vector has zero mean and constant variance ii) the errors are mutually uncorrelated and iii) the error distribution is normal in a statistical sense (Diskin and Simon, 1977). In the absence of errors, the identification procedure normally identifies a unique set of values for the parameters. In their presence, the individual values of the identified parameters depend on the individual samples and therefore the identified parameters do not appear to be unique. However, the treatment of the problem in a statistical sense using the sampling theory clearly renders the uniqueness of the means, provided that the objective function does not introduce any undue bias. Bias is induced in the parameter estimates due to following reasons (Williams, 1978). i) the statistical distribution of the sample is different from that of the population ii) the error measurements which creates "inconsistent" data and iii) the functional form of the estimator is such that the average overall samples is not equal to the true value. The present study is aimed at evaluating the bias induced by the objective function in the presence of errors. For this purpose, Gaussian noise is added to hypothetically generated virus concentration data (Refer Section 4.4) through specifying a mean μ

and a standard deviation σ as follows:

$$C_{o,i,j} = C_{t,i,j}\varepsilon \tag{4.21}$$

with

$$\varepsilon = N(\mu, \sigma) \tag{4.22}$$

where $C_{o,i,j}$ is the observed virus concentration and $C_{t,i,j}$ is the hypothetically generated virus concentration data and ε is the Gaussian error. i and j are the number of space and time observations respectively. The μ of every sample is assigned a value equal to 1 to ensure unbiased perturbation through Eq (4.21), where the errors are randomly distributed above and below $C_{t,i,j}$. For each value of σ, 10 samples are generated by changing the seeding of the random number generator. In the present study, σ is changed from 0.025 to 0.15 in increments of 0.025.

As discussed in Section 4.4, when the data contains no errors, the objective function converges to the true values of the parameters as shown in Table 4.2, 4.3 and 4.5 for one two and three unknown parameters respectively. This indicates that when data is free from errors, the objective function does not induce any bias in the estimated parameters. When the data contains errors, the objective function does not converge to true values. Percentage errors in estimating the transport parameters λ, λ^*, D and k_d are presented in Tables 4.6 to 4.9 respectively at two noise levels 0.05 and 0.1 for 10 different samples. From Tables 4.6 to 4.9, it is concluded that i) the noisy data introduces errors in the estimated parameters. ii) the amount of induced error increases with the noise level and iii) for any given noise level, the individual values of identified parameters deviate from the true value. In addition it is observed that the error percentage in case of λ^* is very high as compared to other parameters. As the individual values of the identified parameters generally deviate from their true values, the error contained in these values reveal very little in their behavior and hence statistical analysis is necessary in terms of means and confidence intervals. In the present study, 95% confidence interval is used for carrying out the statistical analysis.

Table 4.6: Effect of data error and objective function on estimated parameter λ

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of λ	Error percentage in λ	Value of λ	Error percentage in λ
1	0.5733	1.15	0.5667	2.2931
2	0.5711	1.534	0.5624	3.034
3	0.5693	1.844	0.5586	3.6896
4	0.5672	2.206	0.5547	4.362
5	0.6232	-7.448	0.6672	-15.0345
6	0.5799	0.0172	0.5799	0.0172
7	0.573	0.7	0.566	2.4137
8	0.56	3.448	0.5405	6.8103
9	0.5483	5.465	0.5174	10.7931
10	0.5617	3.155	0.5436	6.2758

Table 4.7: Effect of data error and objective function on estimated parameter λ^*

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of λ^*	Error percentage in λ^*	Value of λ^*	Error percentage in λ^*
1	0.3282	28.6521	0.2564	44.26
2	0.3005	34.6739	0.132	71.3
3	0.2795	39.2391	0.09194	80.013
4	0.2538	44.826	0.0706	84.652
5	1.2402	-169.6	2.0286	-341.0
6	0.4509	1.9782	0.4654	-1.1739
7	0.3307	28.1087	0.2551	44.5434
8	0.0961	79.1087	0.01737	96.2239
9	0.0188	95.913	0.04242	90.7782
10	0.1285	72.0652	0.2528	45.0434

Table 4.8: Effect of data error and objective function on estimated parameter D

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of D	Error percentage in D	Value of D	Error percentage in D
1	34.672	-1.9764	35.349	-3.9676
2	34.965	-2.8382	35.976	-5.8117
3	33.806	0.5705	33.603	1.1676
4	34.766	-2.2529	35.477	-4.3441
5	33.191	2.3794	32.676	3.8941
6	33.306	2.0411	32.601	4.1147
7	35.068	-3.1411	36.225	-6.5441
8	35.077	-3.1676	36.284	-6.7176
9	35.122	-3.3	36.288	-6.7294
10	33.393	1.7852	32.694	3.8411

Table 4.9: Effect of data error and objective function on estimated parameter k_d.

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of k_d	Error percentage in k_d	Value of k_d	Error percentage in k_d
1	0.0198	1.0	0.0198	1.0
2	0.01944	2.8	0.01884	5.8
3	0.0179	10.5	0.01599	20.05
4	0.01833	8.35	0.01659	17.05
5	0.02512	-25.6	0.0308	-54.0
6	0.02257	-12.85	0.0251	-25.5
7	0.01885	5.75	0.01767	11.65
8	0.01794	10.3	0.01601	19.95
9	0.01588	20.6	0.01374	31.3
10	0.02157	-7.85	0.02349	-17.45

Figs 4.7 to 4.10 show the mean values and the 95% confidence intervals of estimated parameters λ, λ^*, D and k_d respectively. The following conclusions can be drawn from Figs 4.7 to 4.10. i) the deviation of the mean from the true value increases with an increase in noise level in case of parameters λ and D. However, the mean remains significantly unchanged and is closer to the true value in case of λ^* and k_d. ii) the true value is contained within the 95% confidence interval of all the identified parameters at all noise levels indicating that the objective function does not induce bias on the parameter estimates, when they are estimated individuals.

Fig 4.11a and 4.11b show the variation of means of identified λ and D with noise level σ by treating λ and D is unknown parameters. Similarly Fig 4.12a and 4.12b, Fig 4.13a and 4.13b, Fig 4.14a and 4.14b, and Fig 4.15a and 4.15b show the variation of means of identified λ and k_d, λ^* and D, λ^* and k_d, and D and k_d with noise level σ respectively by treating them as unknown parameters. From Figs 4.11 to 4.15, it can be seen that in the case of estimation of two unknown parameters also, the objective function does not induce any undue bias in estimated parameters. This is evident from the fact that the true value falls within the 95% confidence interval of the identified parameters at all noise levels.

Fig 4.16a, 4.16b, 4.16c and Fig 4.17a, 4.17b, Fig 4.17c show the variation of means of identified λ, D, k_d, and λ^*, D, k_d with noise level σ while estimating three unknown parameters. From Figs 4.16 and 4.17, it is seen that the objective function induces bias in estimated parameters as the true value does not fall within the 95% confidence interval of the identified parameters.

It is to be noted that in some cases, the lower confidence limit for λ^* (Fig. 4.8, Fig. 4.13a, Fig. 4.14a and Fig. 4.17a) show negative value which have no physical relevance. However, they are retained in these figures for showing the upper and lower confidence intervals.

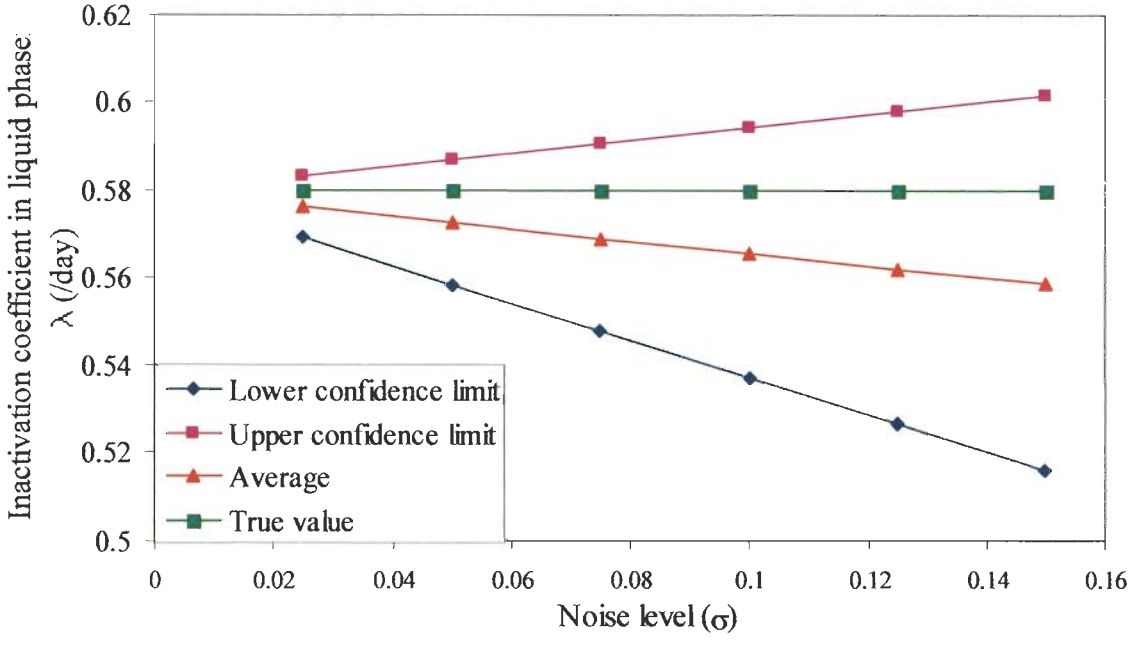

Fig 4.7 Variation of means of identified λ with noise level σ

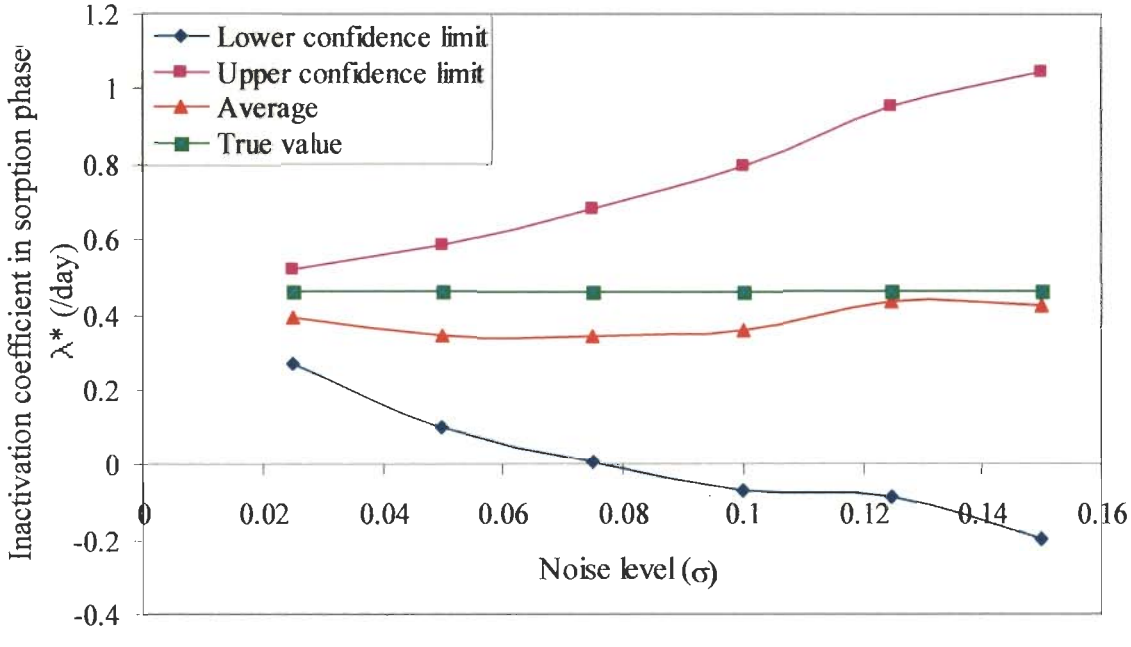

Fig 4.8 Variation of means of identified λ^* with noise level σ

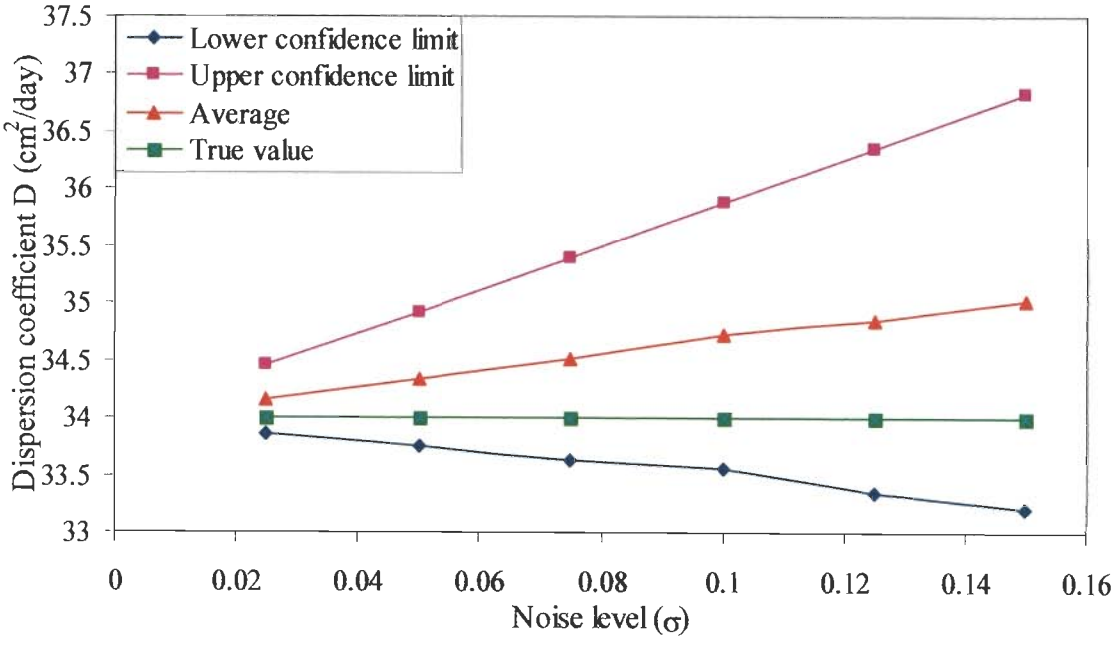

Fig 4.9 Variation of means of identified D with noise level σ

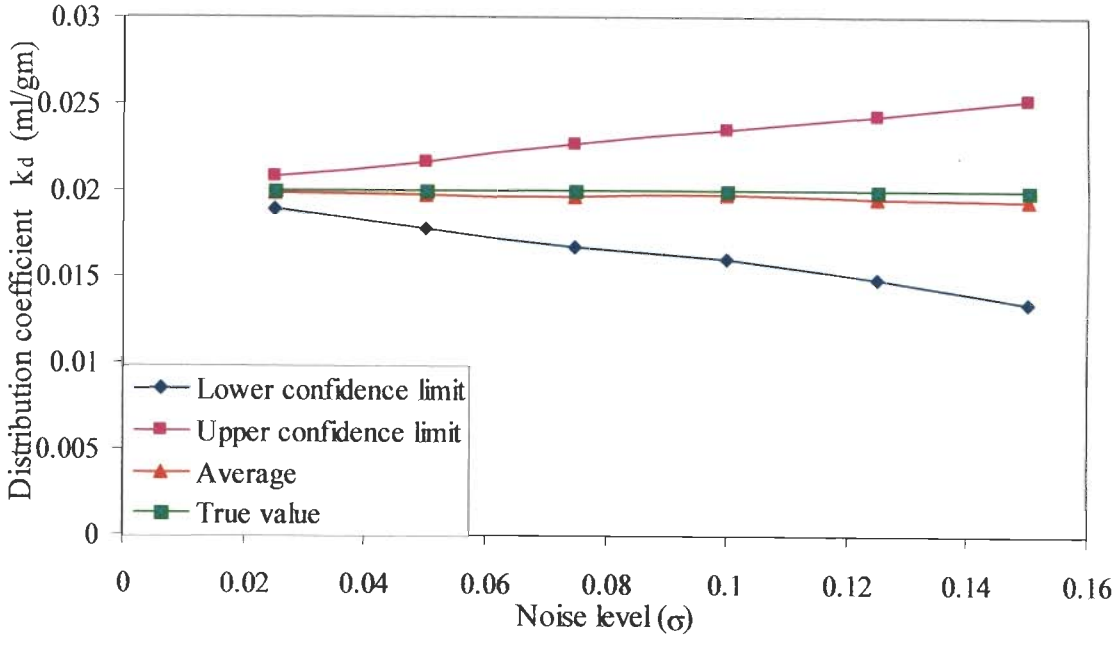

Fig 4.10 Variation of means of identified k_d with noise level σ

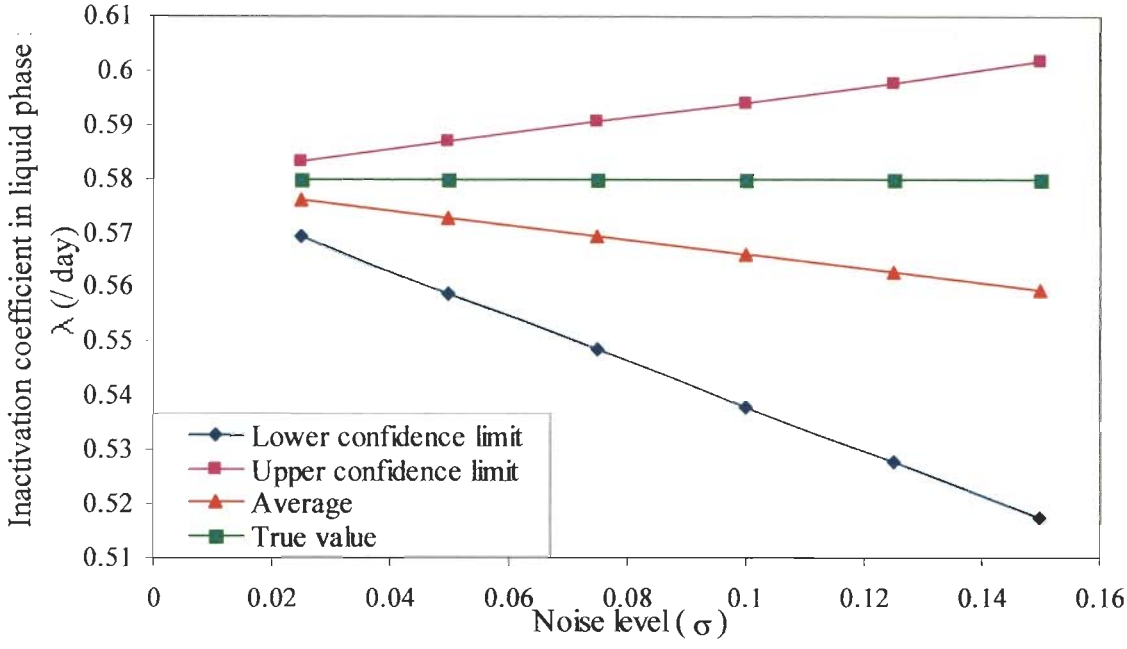

Fig 4.11a Variation of means of identified λ with noise level σ while λ and D are unknown parameter

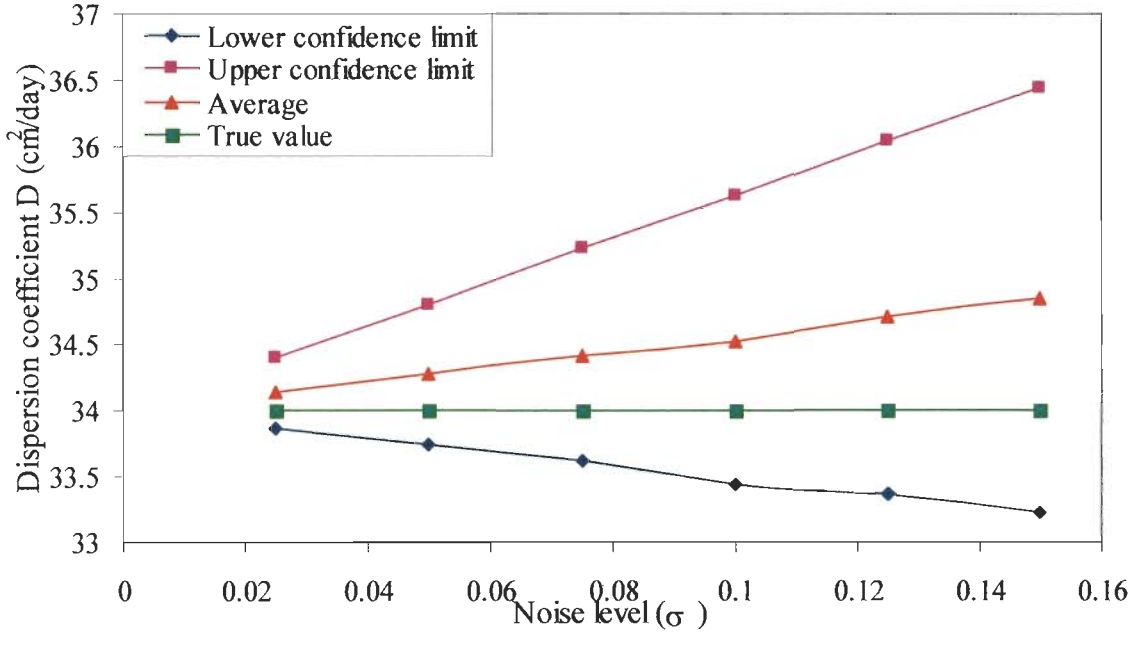

Fig 4.11b Variation of means of identified D with noise level σ while λ and D are unknown parameter

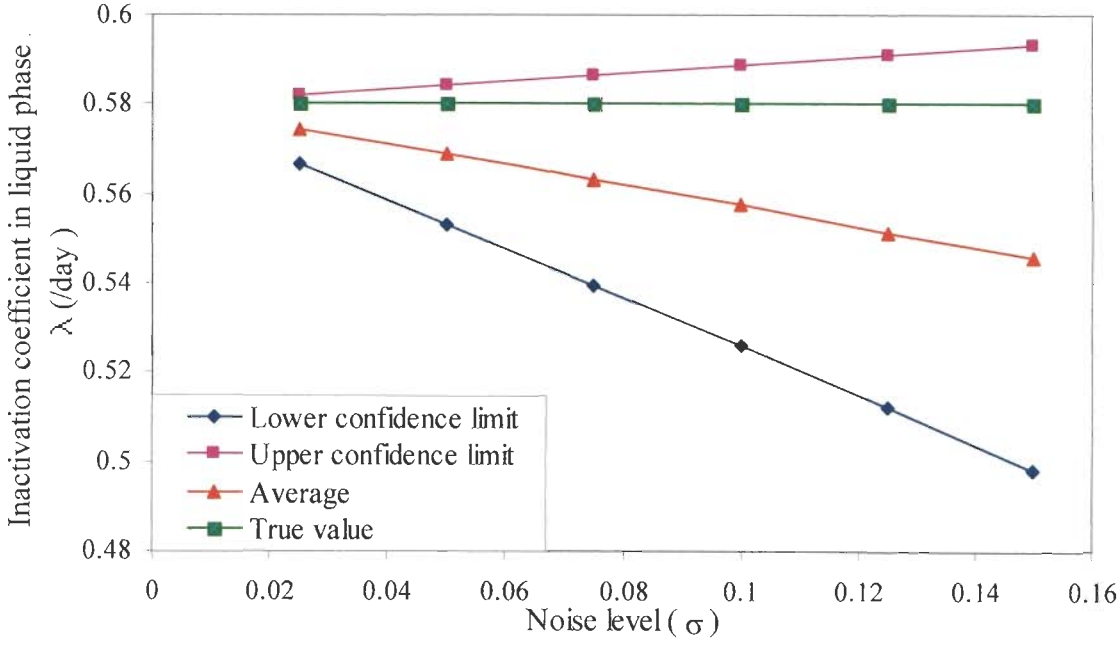

Fig 4.12a Variation of means of identified λ with noise level σ while λ and k_d are unknown parameter

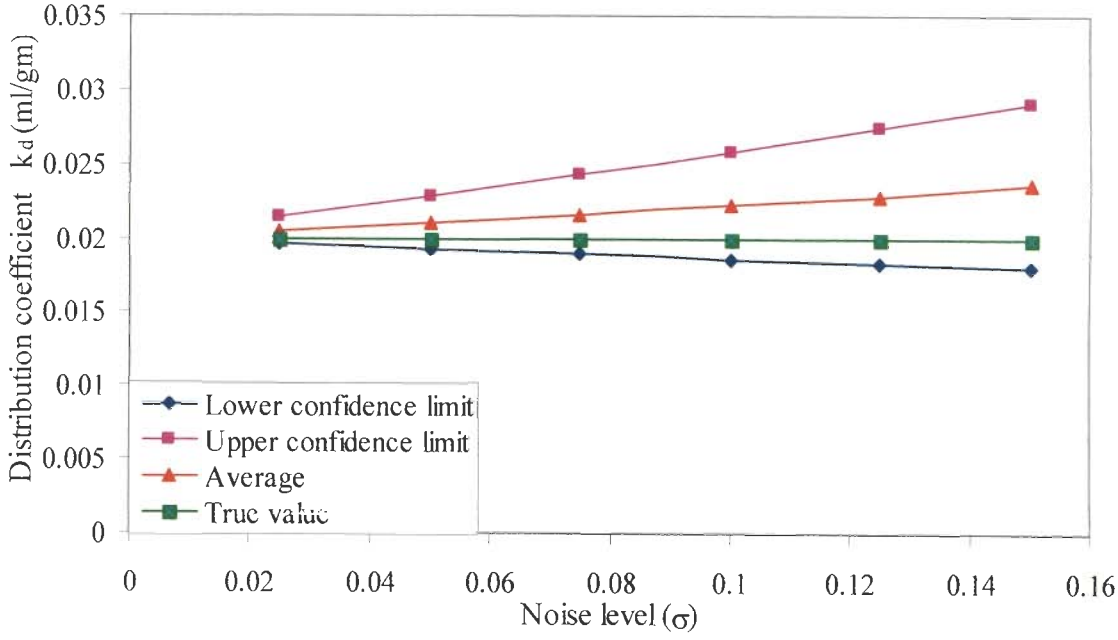

Fig 4.12b Variation of means of identified k_d with noise level σ while λ and k_d are unknown parameter

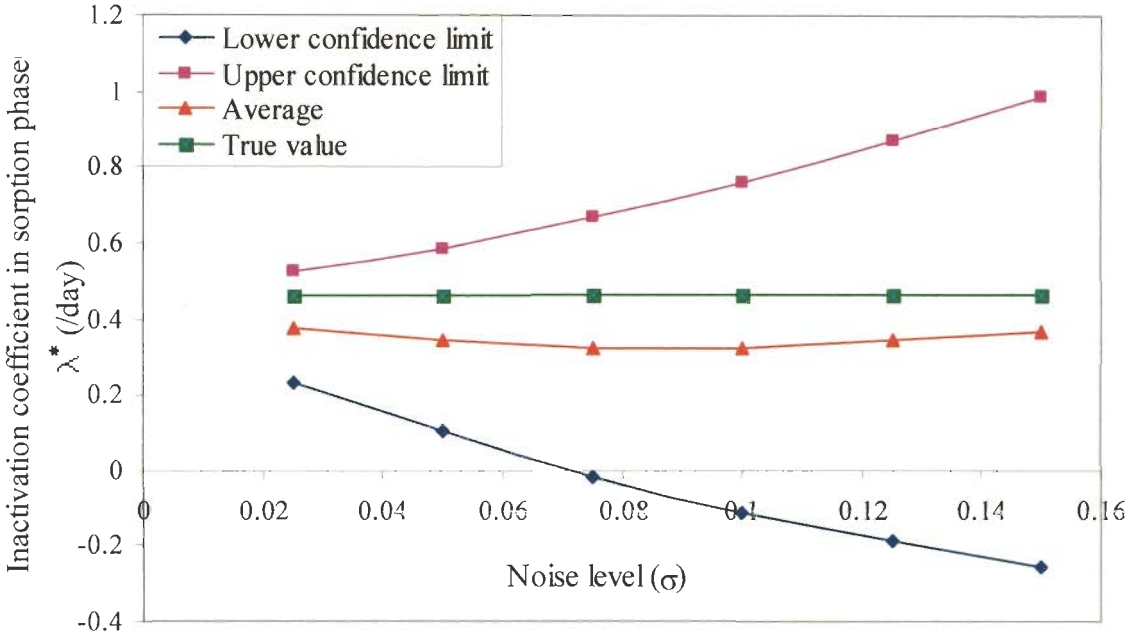

Fig 4.13a Variation of means of identified λ^* with noise level σ while λ^* and D are unknown parameter

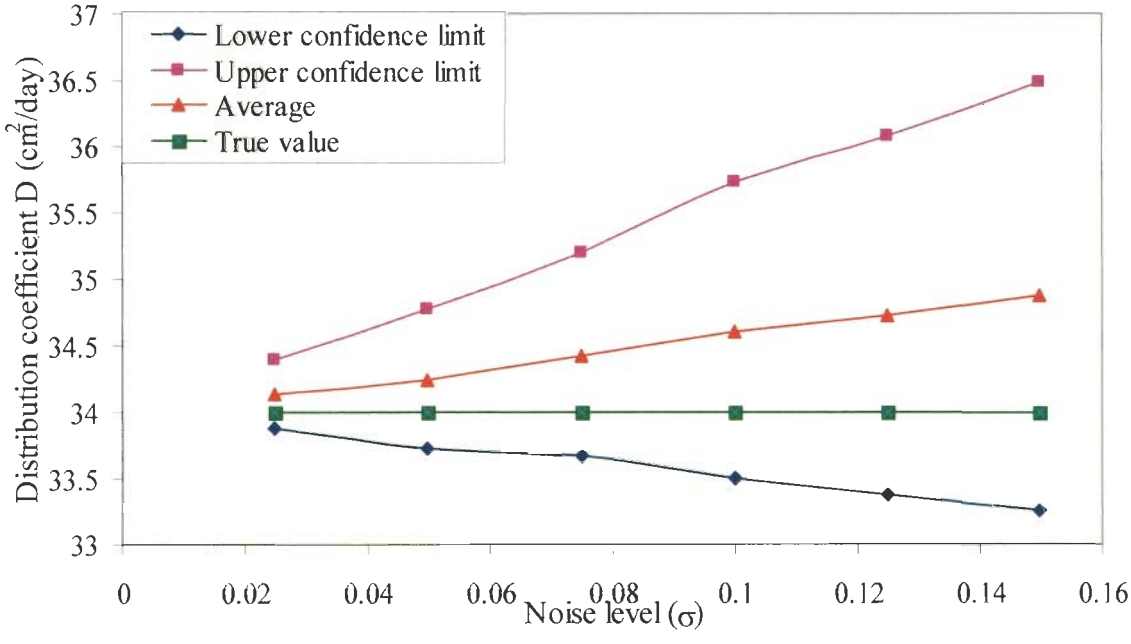

Fig 4.13b Variation of means of identified D with noise level σ while λ^* and D are unknown parameter

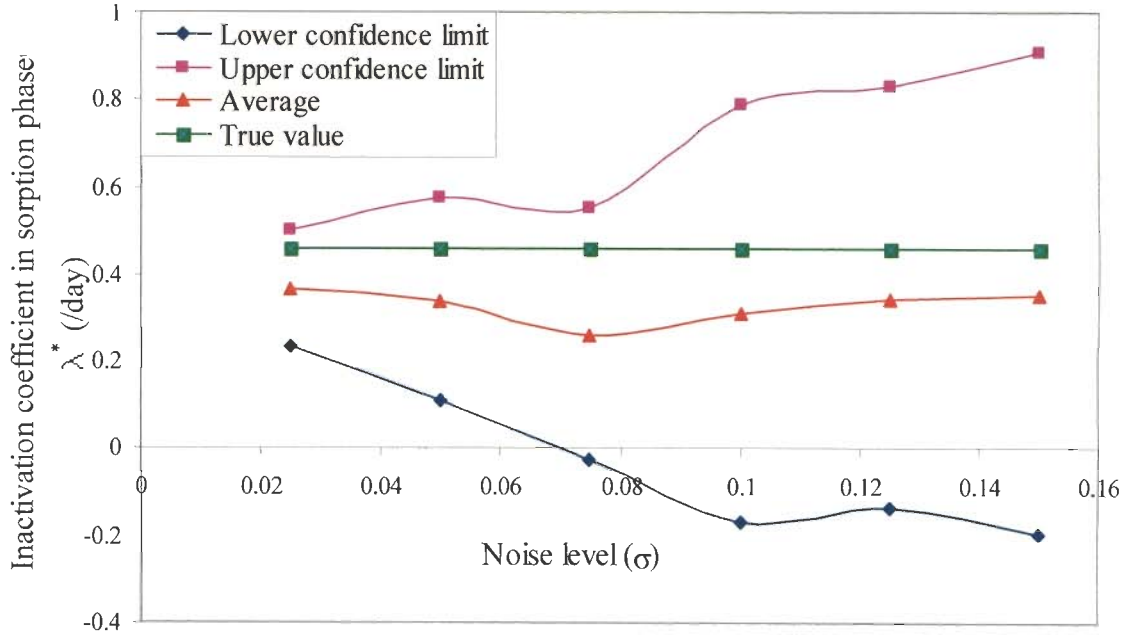

Fig 4.14a Variation of means of identified λ^* with noise level σ while λ^* and k_d are unknown parameter

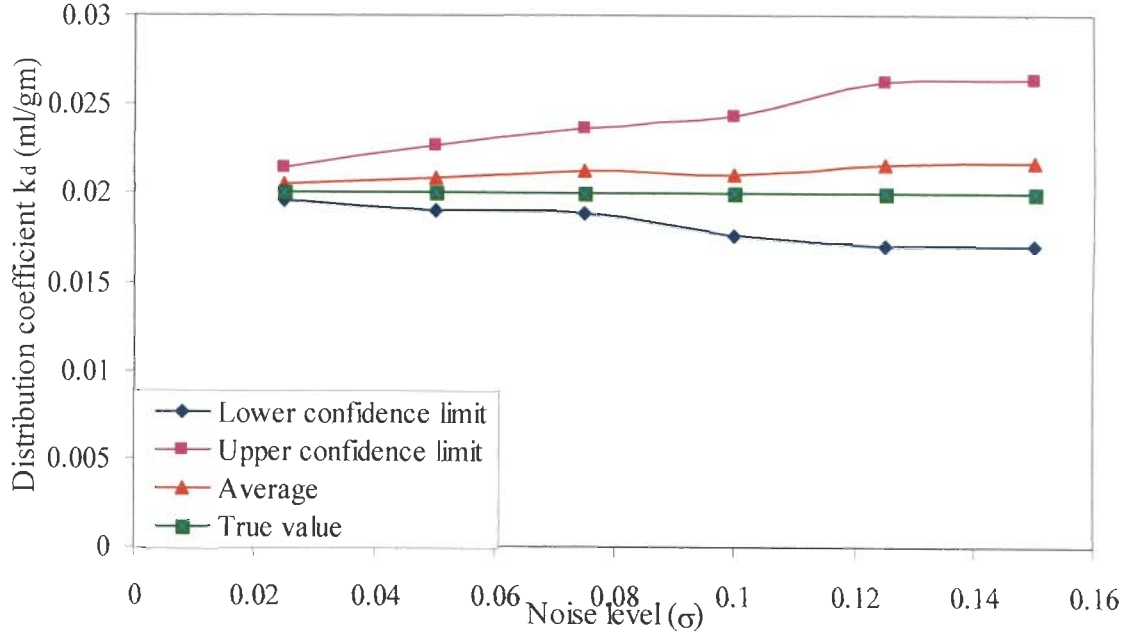

Fig 4.14b Variation of means of identified k_d with noise level σ while λ^* and k_d are unknown parameter

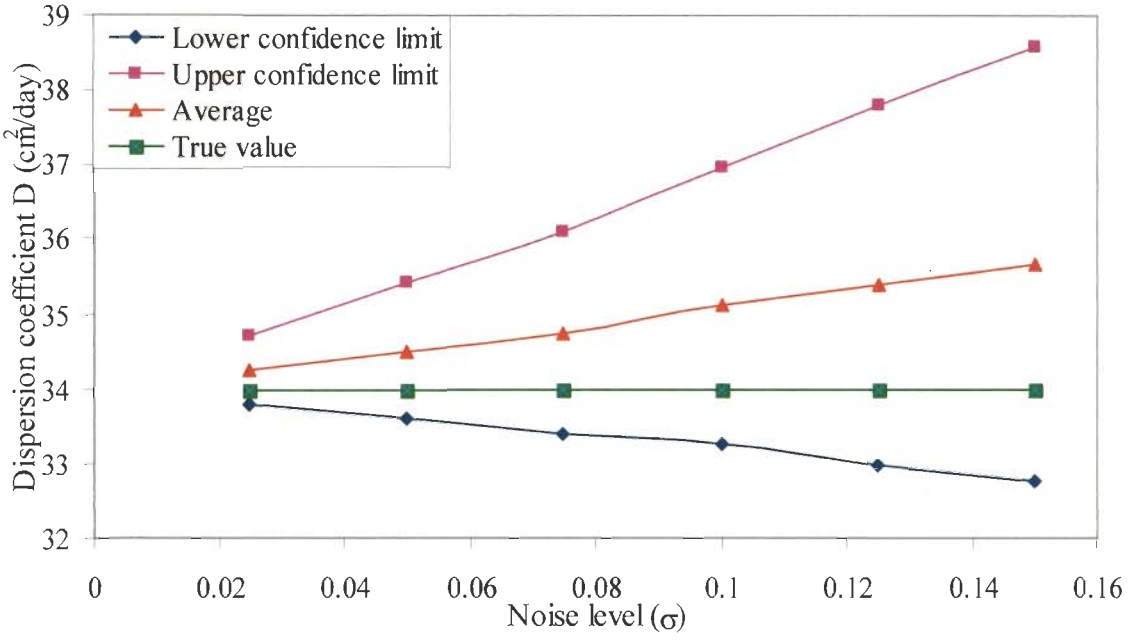

Fig 4.15a Variation of means of identified D with noise level σ while D and k_d are unknown parameter

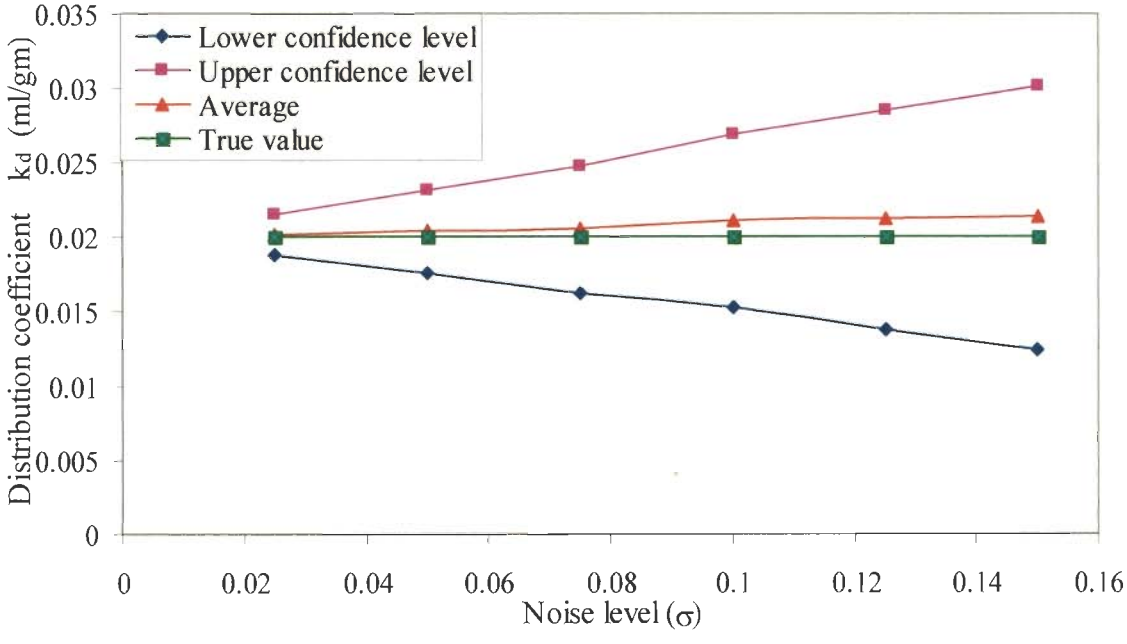

Fig 4.15b Variation of means of identified k_d with noise level σ while D and k_d are unknown parameter

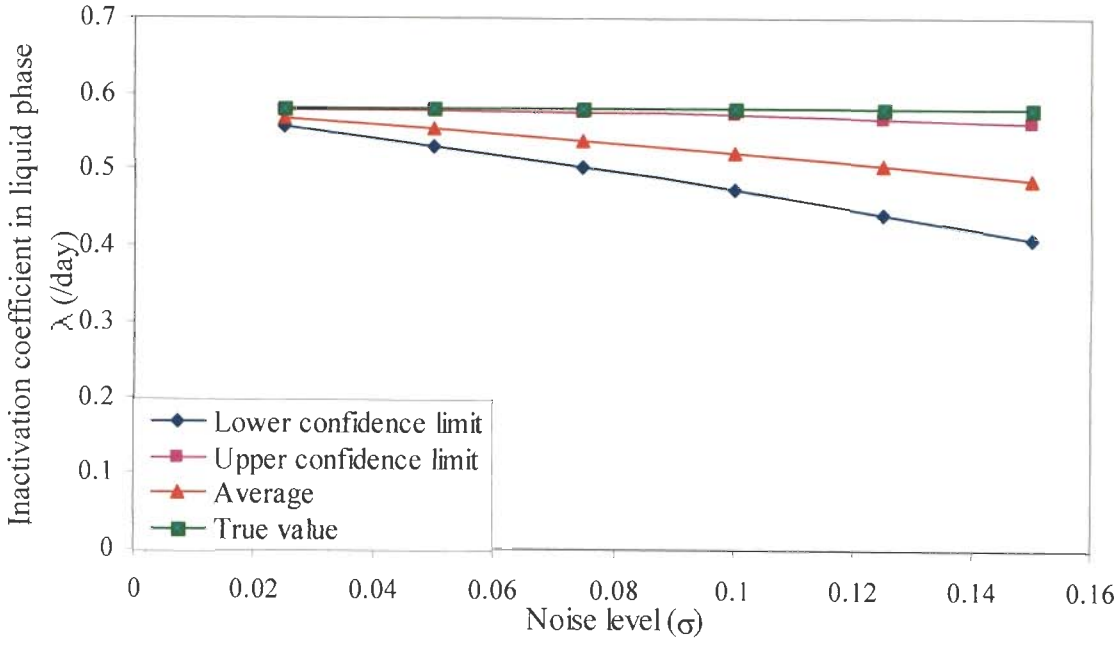

Fig 4.16a Variation of means of identified λ with noise level σ while λ, D and k_d are unknown parameter

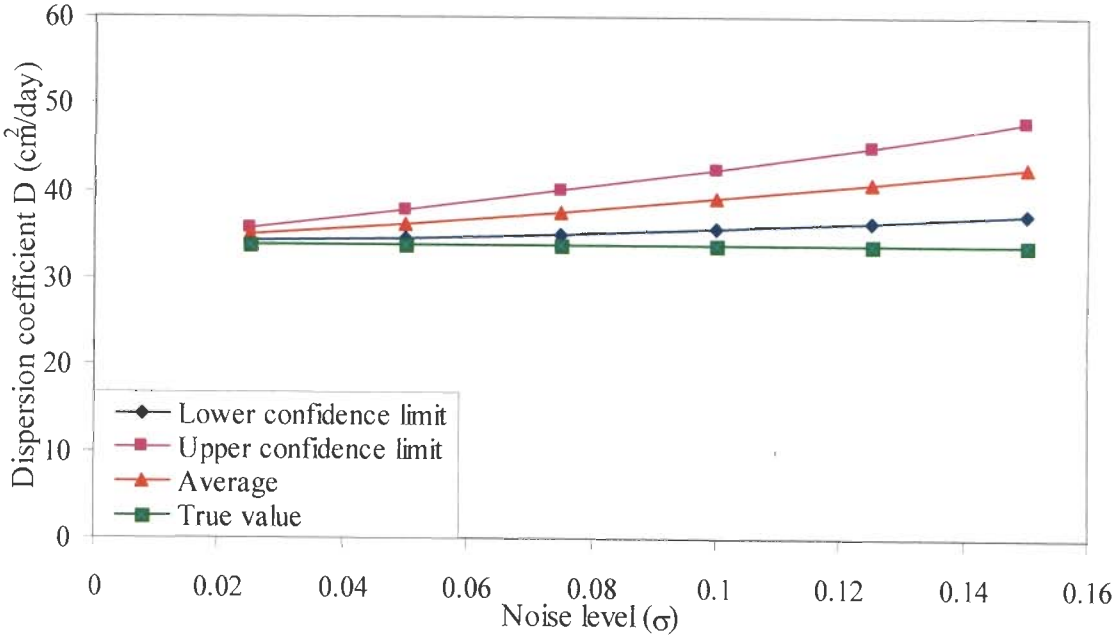

Fig 4.16b Variation of means of identified D with noise level σ while λ, D and k_d are unknown parameter

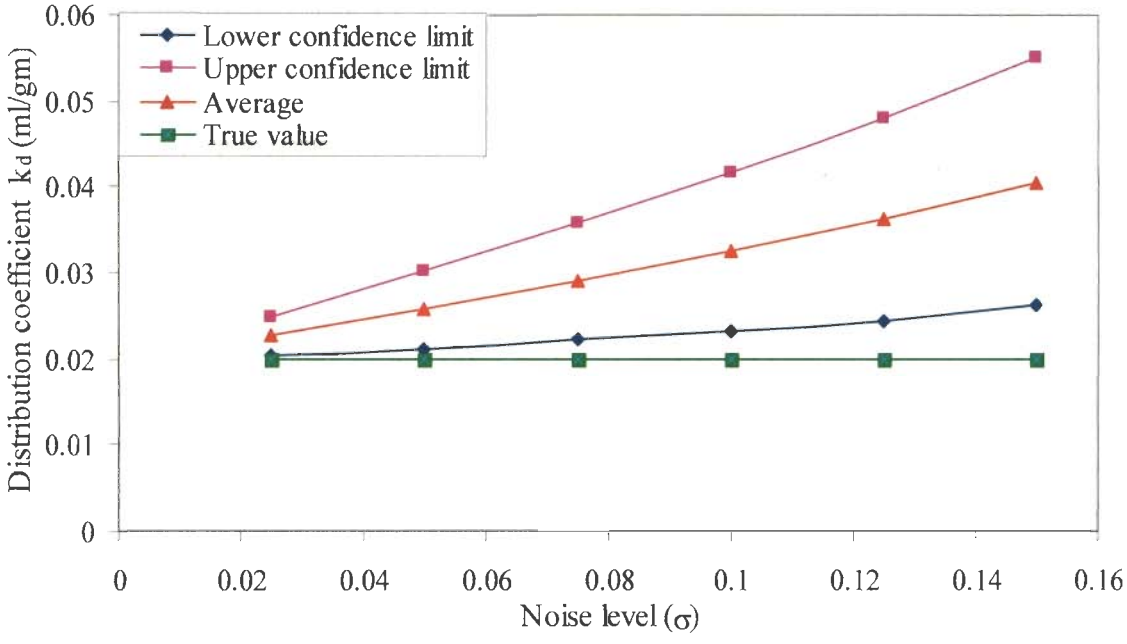

Fig 4.16c Variation of means of identified k_d with noise level σ while λ, D and k_d are unknown parameter

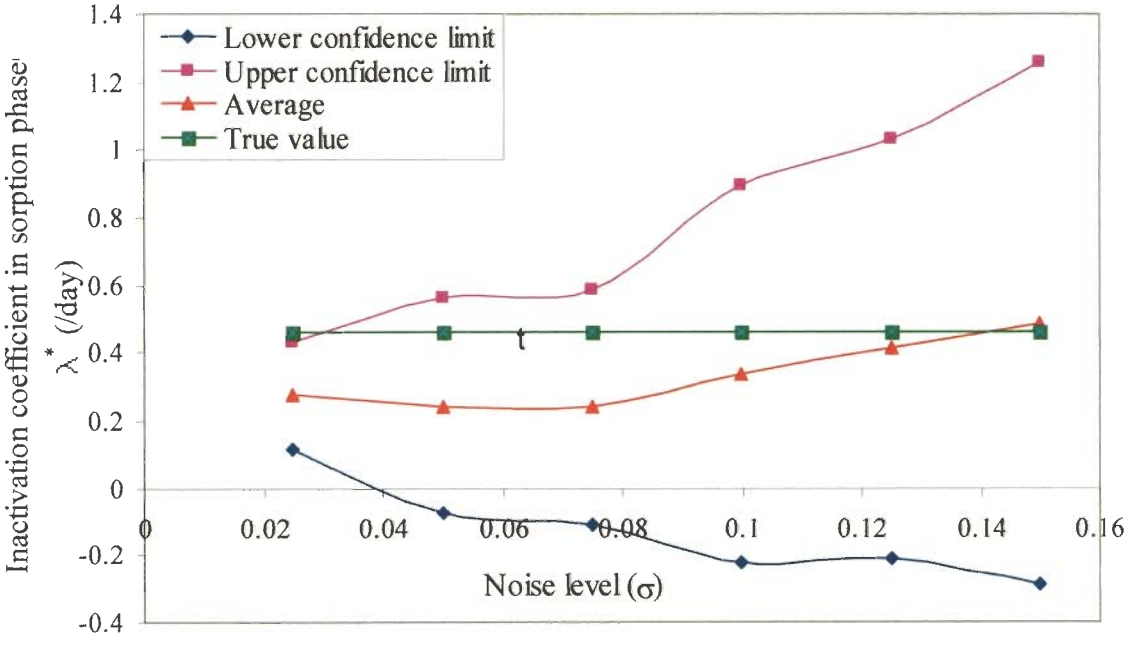

Fig 4.17a Variation of means of identified λ^* with noise level σ while λ^*, D and k_d are unknown parameter

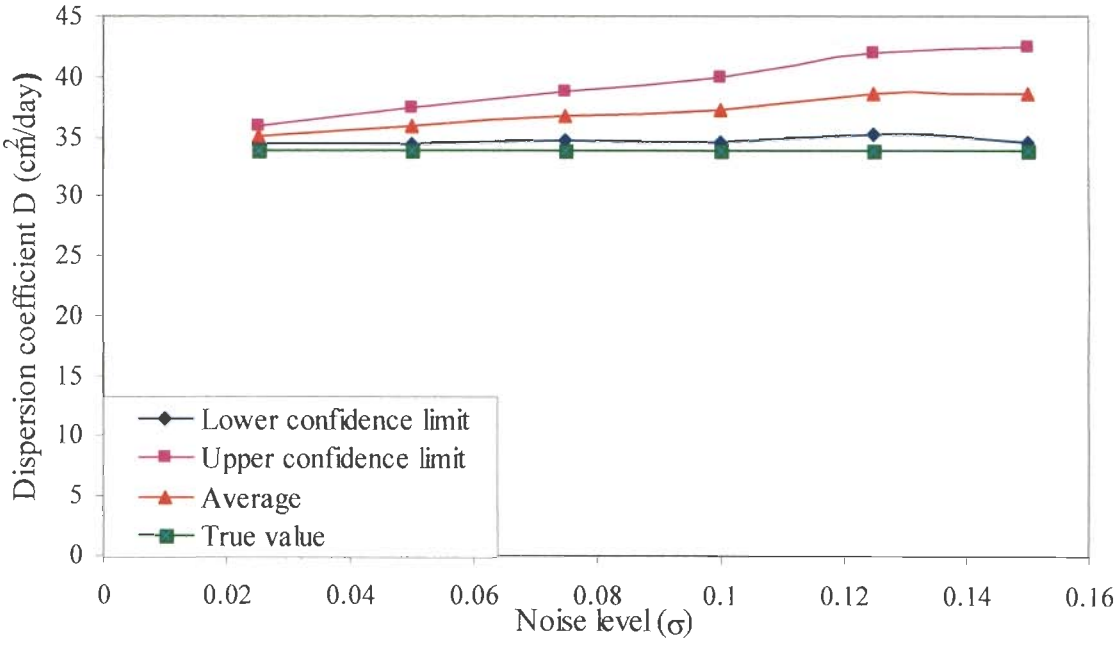

Fig 4.17b Variation of means of identified D with noise level σ while λ^*, D and k_d are unknown parameter

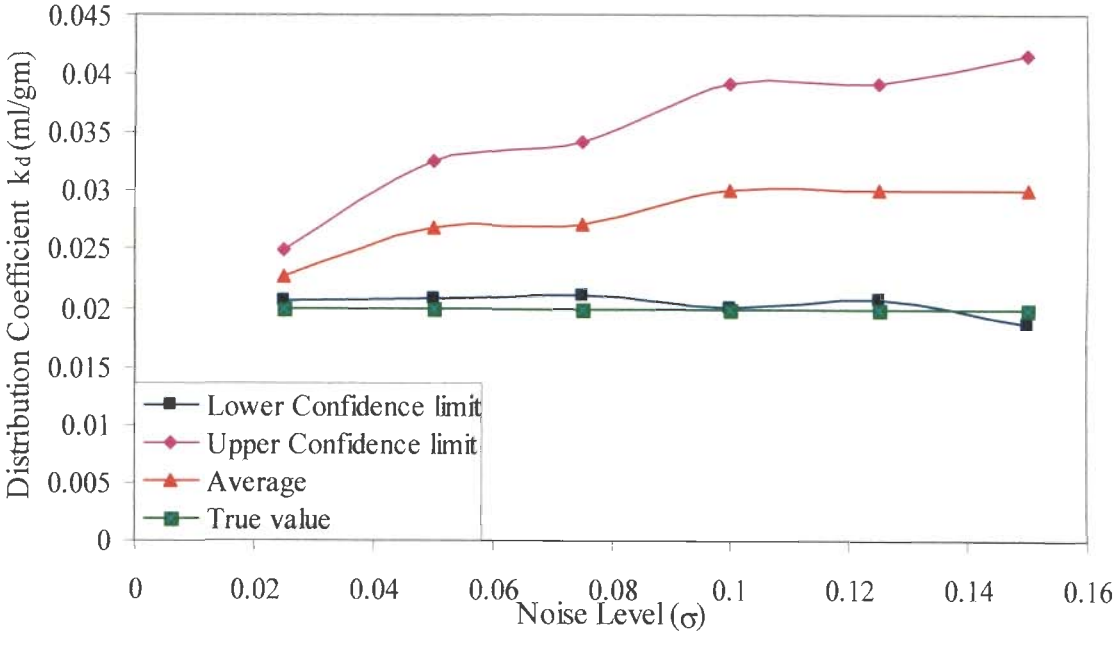

Fig 4.17c Variation of means of identified k_d with noise level σ while λ^*, D and k_d are unknown parameter

4.6 ESTIMATION OF TRANSPORT PARAMETERS FROM COLUMN EXPERIMENT

In this Section, the optimization algorithm is also applied to estimate the transport parameters from the virus concentration data of two column experiments.

4.6.1 Column Experiment 1

Bales et al., (1991) conducted three column experiments with MS2 virus at pH 5. Experiments were done at 4^0 C using a 15 cm × 0.9 cm i.d. precision-bore glass chromatography column packed with 45-90 μm glass beads. Flow rates monitored continuously remained fairly steady. The average moisture content during the experiment was about 0.35. The mass density of glass beads was 1.6 gm/cm^3 and the average velocity was about 13.32 cm/hr. The virus concentrations at the outlet for a period of 4.7 hr are given as the observed data for the estimating parameters. Since the experiments were conducted at 4^0 C, the inactivation coefficients λ and λ^* are taken as 0.0 (Sim and Chrysikopoulous, 1996). The remaining two parameters D and k_d are estimated using the inverse procedure as explained in Section 4.3. Table 4.10 presents the initial guess value and the optimal parameter estimates of D and k_d. It can be seen that, in all the cases (over estimated, under estimated and mixed) the optimization resulted in more or less unique values for the estimated parameters. The RMS error in all the cases is almost the same. Further it can be seen that starting the parameter estimation with over estimated values resulted in least number of iterations. Fig. 4.18 compares the observed and model predicted virus concentration with optimal parameter values (D=41.472 cm^2/hr and k_d = 0.0286 ml/gm). Fig 4.18 suggests that the model predictions with optimal parameter estimates match reasonably well with the observed virus concentrations.

Table 4.10: Estimation of transport parameters from Column Experiment 1 (Bales et al., 1991)

Parameters	Case A (over estimated)				Case B (under estimated)				Case C (mixed)			
	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations
D (cm^2/day)	400.0	41.472	0.0847	22	4.0	41.457	0.0847	150	4.0	40.911	0.0853	24
k_d (ml/gm)	0.40	0.0286			0.004	0.0286			0.40	0.02858		

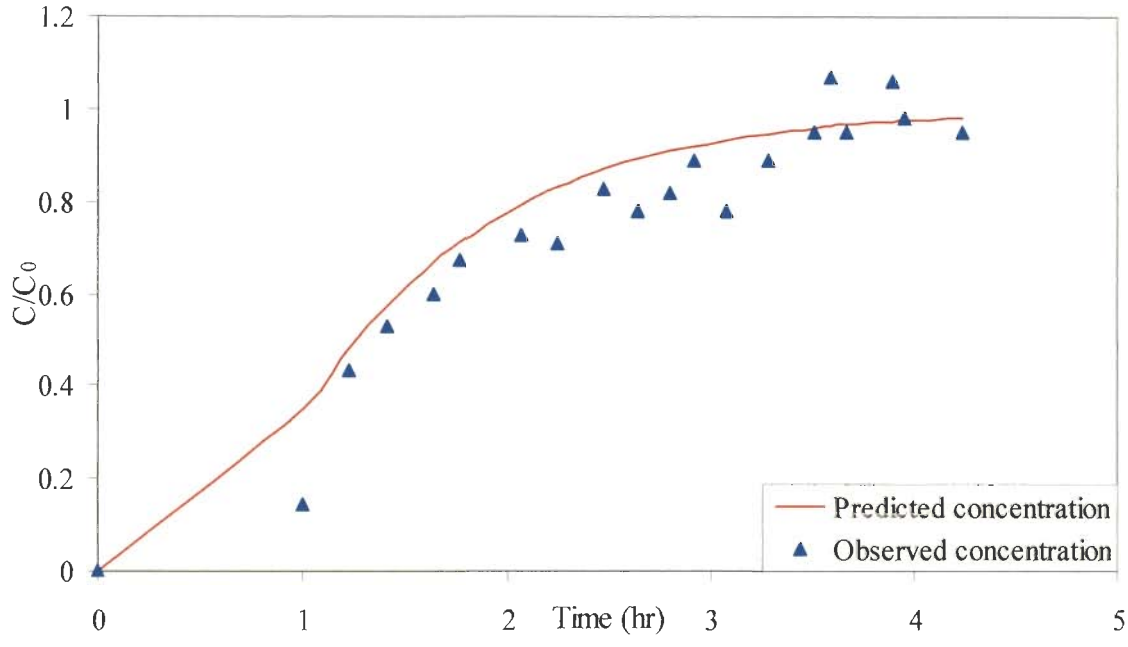

Fig 4.18 Comparison of normalized virus MS2 breakthrough concentration from Column experiment 1 (Bales et al., 1991)

4.6.2 Column Experiment 2

Jin et al. (2000) conducted ΦX174 virus transport experiment for saturated conditions. The column used for saturated experiments consisted of a top and bottom plate and was sealed by an o-ring on each end. There were 17 syringe needles evenly distributed on the bottom solution-filling column to ensure a uniform supply of the input solution. A stainless steel porous plate (3.175 mm thick) with a pore size of 0.5 μm and a bubbling pressure of 10.1-13.5 kpa was placed at the bottom of the column. Column outlet was connected to a vacuum chamber with a fraction collector inside. By adjusting the vacuum pressure and flow rate of the input solution, steady state and essentially uniform water content was reached. Two small tensiometers were installed at depths of 3.3 and 6.6 cm to verify that the column indeed had a uniform water distribution.

The column was made of acrylate and was 7.6 cm in diameter and 10 cm long. The experiments were conducted in a cold room at 4^0 C to minimize the inactivation

due to high temperature. Viruses were added to the sand column as a constant input at an approximate concentration of 5×10^4 pfu/ml. Input solution containing bromide tracer, and ΦX174 was applied with a peristaltic pump. Outflow samples were collected in 15 ml polypropylene centrifuge tubes with a fraction collector. The bromide tracer was used to test the performance of the column and to obtain transport parameters. The average moisture content during the experiment was about 0.363. The mass density of sand column was 1.72 gm/cm^3 and the average velocity was about 1.38 cm/hr. The inactivation coefficient in both liquid and sorption phase is taken zero. The remaining two parameters D and k_d are estimated using the inverse procedure as explained in Section 4.3.

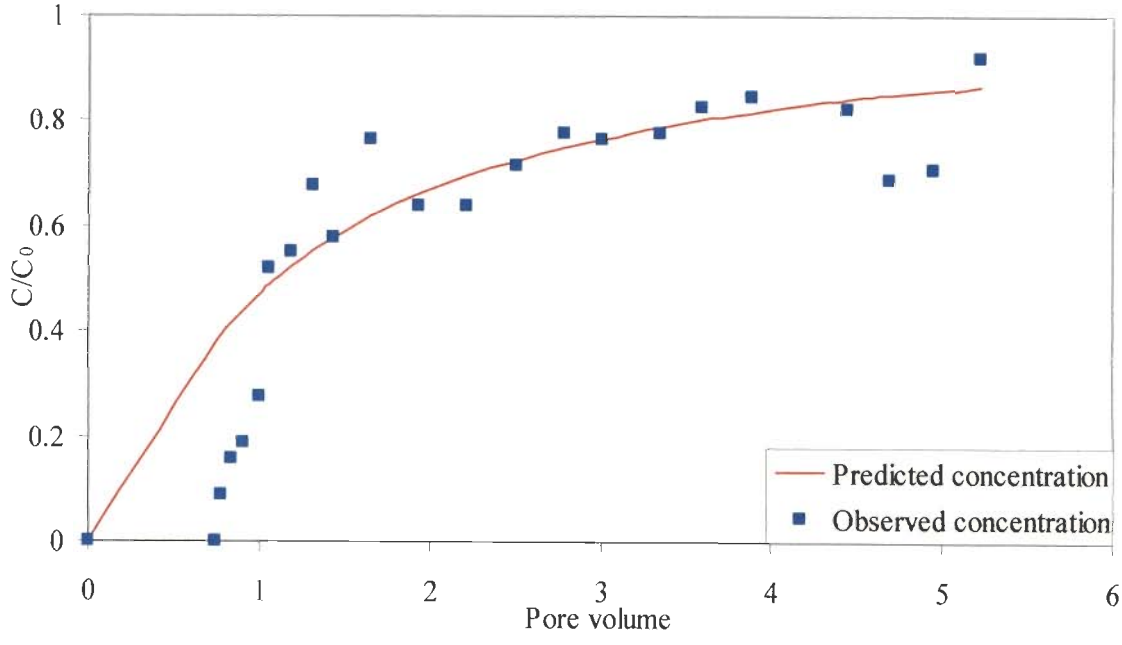

Fig 4.19 Comparison of normalized virus ΦX174 breakthrough concentration from Column experiment 2 (Jin et al., 2000)

Table 4.11 presents the initial guess values and the optimal parameter estimates of D and k_d by considering different initial guess values. The optimal parameters estimated in each case are used to predict the final virus concentrations.

Table 4.11: Estimation of transport parameters of ΦX174 from Column Experiment 2 (Jin et al., 2000)

Parameters	Case A (over estimated)				Case B (under estimated)				Case C (mixed)			
	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations
D (cm^2/hr)	100	31.882	0.1744	33	1.0	22.357	0.1476	18	100.0	29.8835	0.1727	20
k_d (ml/gm)	0.4	0.005879			0.004	0.00277			0.004	0.00235		
D (cm^2/hr)	200	32.765	0.1748	35	2.0	29.8341	0.1723	14	200.0	28.0837	0.1671	18
k_d (ml/gm)	0.02	0.009796			0.0002	0.00022			0.0002	0.00022		

The RMS error in all the cases is also found out for each case and is shown in Table 4.11. The parameters corresponding to minimum RMS error is finally used to predict the virus concentrations. Table 4.11 shows that for the minimum RMS error is 0.1476 for which the value of the parameters are D=22.357 cm^2/hr and k_d = 0.00277 ml/gm respectively. Fig. 4.19 compares the observed and model predicted ΦX174 virus concentration with optimal parameter values (D=22.357 cm^2/hr and k_d = 0.00277 ml/gm). Fig. 4.19 suggests that the model predictions with optimal parameter estimates match reasonably well with the observed virus concentrations.

4.7 CONCLUDING REMARKS

In the present Chapter, the parameter estimation is formulated as a least square minimization problem in which the parameters are estimated by minimizing the deviation between the model predicted and observed virus concentrations. Levenberg-Marquadart algorithm is employed for the nonlinear optimization. The efficacy and robustness of the optimization procedure is evaluated by estimating the parameter from hypothetically generated virus concentration data. It is found that with the virus concentration data, it is not possible to estimate the four transport parameters D, λ, λ^*, and k_d uniquely. If the number of parameters to be estimated is less than or equal to three, the inverse procedure uniquely estimates the unknown parameters. Further it is observed that in the cases of estimation of three or two parameters, if the parameters to be estimated involve λ and λ^*, the optimization does not yield unique estimates. The analysis of the convexity of the objective function in λ-λ^* parametric space shows the presence of local minima which result in the nonunique estimation of the parameters λ and λ^*. It is concluded that apriori estimation of one of the inactivation coefficient is necessary for unique estimation of other unknown parameters. Optimization results on

the hypothetical data indicate that starting the initial guess from over estimated values results in least number of iterations for the algorithm to reach the optimal solution.

The present Chapter also investigates the performance of the objective function in the presence of noisy data during estimation of transport parameters. To study the effect of objective function on parameter estimation, Gaussian noise is added to hypothetically generated data and detailed statistical analysis is carried out. It is found that the objective function does not induce any bias into the estimated parameters when unknown parameters are less than three but when the unknown parameters are equal to three, then the objective function induces bias into the estimated parameters. The parameter estimation is also applied to estimate the transport parameter from a column experiment involving virus transport. It is found that the model predictions with optimal parameter estimates match reasonably well with experimental data.

CHAPTER 5

ANALYSIS OF VIRUS TRANSPORT THROUGH UNSATURATED ZONE- MODEL DEVELOPMENT

5.1 INTRODUCTION

In Chapter 3 and 4, the numerical modeling for virus transport in saturated zone and estimation of virus transport parameters are discussed in detail. Numerous studies (Yates and Ouyang, 1992; Sim and Chrysikopoulous, 2000) have indicated that nature of virus movement in unsaturated zone is significantly different from that in saturated zone. These studies have shown that the virus removal in unsaturated zone is much higher as compared to saturated zone. Modeling of virus transport in unsaturated zone is much more difficult than modeling in the saturated zone since the velocities and moisture content in unsaturated zone depends on the pressure head unlike saturated zone. The nonlinearity of the governing flow equation (Richards equation) needs mass conservative schemes for accurate prediction of velocities and moisture contents. The objective of the present Chapter is to develop moisture flow and transport numerical model for predicting virus movement in the unsaturated zone. A mass conservative fully implicit finite difference scheme (Celia et al., 1990) for predicting moisture movement in unsaturated zone is coupled with hybrid finite volume numerical model (developed in Chapter 3) predicting virus movement.

5.2 GOVERNING EQUATION

The mass conservation equations for the simultaneous transport of water and suspended virus particles through variably saturated media under transient flow condition can be written as (Tim and Mostaghimi, 1991)

Flow equation (Richards equation):

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[K(\theta)\left\{\frac{\partial \psi}{\partial z} + 1\right\}\right] \tag{5.1}$$

Virus transport equation:

$$R\frac{\partial C}{\partial t} = \frac{\partial}{\partial z}\left(D\frac{\partial C}{\partial z}\right) - v\frac{\partial C}{\partial z} - \lambda RC \tag{5.2}$$

In Eq. (5.1) and (5.2), R is the retardation coefficient, ψ is pressure head, θ is volumetric moisture content, and $K(\theta)$ is hydraulic conductivity. Here the vertical coordinate is taken positive upwards.

Eqs. (5.1) and (5.2) are coupled equations since the velocity v appearing in Eq. (5.2) has to be obtained by solving the flow equation (5.1). The solution proceeds in two steps. First, the Eq. (5.1) is solved first to obtain the pressure head in the solution domain as a function of space and time. From the computed pressure head distribution, the velocity of moisture v is computed using Darcy's law. Knowing v, Eq. (5.2) is solved for the virus concentration. The solution of Eq. (5.2) has been discussed in detail in Chapter 3. The solution of Eq. (5.1) is discussed in detail in the following Section.

5.3 NUMERICAL SOLUTION OF RICHARDS EQUATION

Eq. (5.2) is nonlinear in nature as the hydraulic conductivity (K) and moisture content (θ) on the pressure head ψ and needs the constitutive relationships for solution. The constitutive relationships proposed by Van Genuchten (1980) are used in the present study.

5.3.1 Constitutive Relationships

The relationship given by Van Genuchten (1980) are used for θ–ψ and K-θ relationships which are given as

θ–ψ Relationship:

$$\Theta = \left\{ \frac{1}{1+\left(\alpha_v |\psi|\right)^{n_v}} \right\}^{m_v} \tag{5.3}$$

where α_v and n_v are unsaturated soil parameters with

$$m_v = 1 - \frac{1}{n_v} \tag{5.4}$$

Θ is the effective saturation defined as

$$\Theta = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{5.5}$$

where θ_s is saturated water content and θ_r is residual water content of the soil.

K-θ Relationship:

$$K(\Theta) = K_s \left\{ 1 - \left\{ 1 - \Theta^{\left(1/m_v\right)} \right\}^{m_v} \right\}^2 \Theta^{\left(1/2\right)} \tag{5.6}$$

where K_s is saturated hydraulic conductivity.

5.3.2 Initial and Boundary Conditions

Initial condition:

Usually, the pressure head or moisture content distribution at the beginning of the simulation is used as initial condition, i.e.

$$t = 0,\ \psi = \psi_0 \qquad 0 \le z \le L \tag{5.7}$$

or

$$t = 0,\ \theta = \theta_0 \qquad 0 \le z \le L \tag{5.8}$$

where ψ_0 and θ_0 are the specified pressure head and moisture contents at the beginning of the simulation.

Lower boundary condition:

Depending on the presence of water table, the following two boundary conditions are used as lower boundary condition.

In the presence of water table very near to the ground surface, atmospheric pressure head ($\psi = 0$) is applied at the water table. i.e.

$$t > 0, \psi = 0 \qquad z = 0 \tag{5.9}$$

where L is the depth of the water table.

For the case of the deeper groundwater table, a gravity drainage $\left(\frac{\partial \psi}{\partial z} = 0\right)$ boundary condition is applied at certain depth below the ground surface. i.e.

$$t > 0, \quad \left(\frac{\partial \psi}{\partial z} = 0\right), \qquad z = 0 \tag{5.10}$$

Upper boundary condition:

The process occurring at the ground surface such as infiltration or evaporation is assigned as upper boundary condition. For infiltration under ponding conditions, Dirichlet type boundary condition is assigned at the ground surface i.e.

$$t > 0 \quad \psi = \psi_{top} \qquad z = L \tag{5.11}$$

For infiltration/evaporation with constant flux, Neuman type condition is assigned at the ground surface. i.e.

$$t > 0, \quad -K\left(\frac{\partial \psi}{\partial z} + 1\right) = q_{top}, \qquad z = L \tag{5.12}$$

where q_{top} is the infiltration/evaporation rate.

5.3.3 Discretization in Space and Time

For solving the differential equation (5.2), a finite difference grid is superposed over the solution domain. The solution domain L is divided into a number of grids of equal length Δz as shown in Fig 5.1. The spatial index, j, in z direction increases from bottom to top. $j = 1$ coincides with the bottom boundary and $j = N$ coincides with ground surface and n is the temporal index. The time domain is

discretized by finite number of discrete times of size Δt. n denotes the discrete time level at which the solution is known, n+1 denotes the discrete time level at which solution is unknown. The previous and current Picard iteration levels are denoted as m and m+1 respectively.

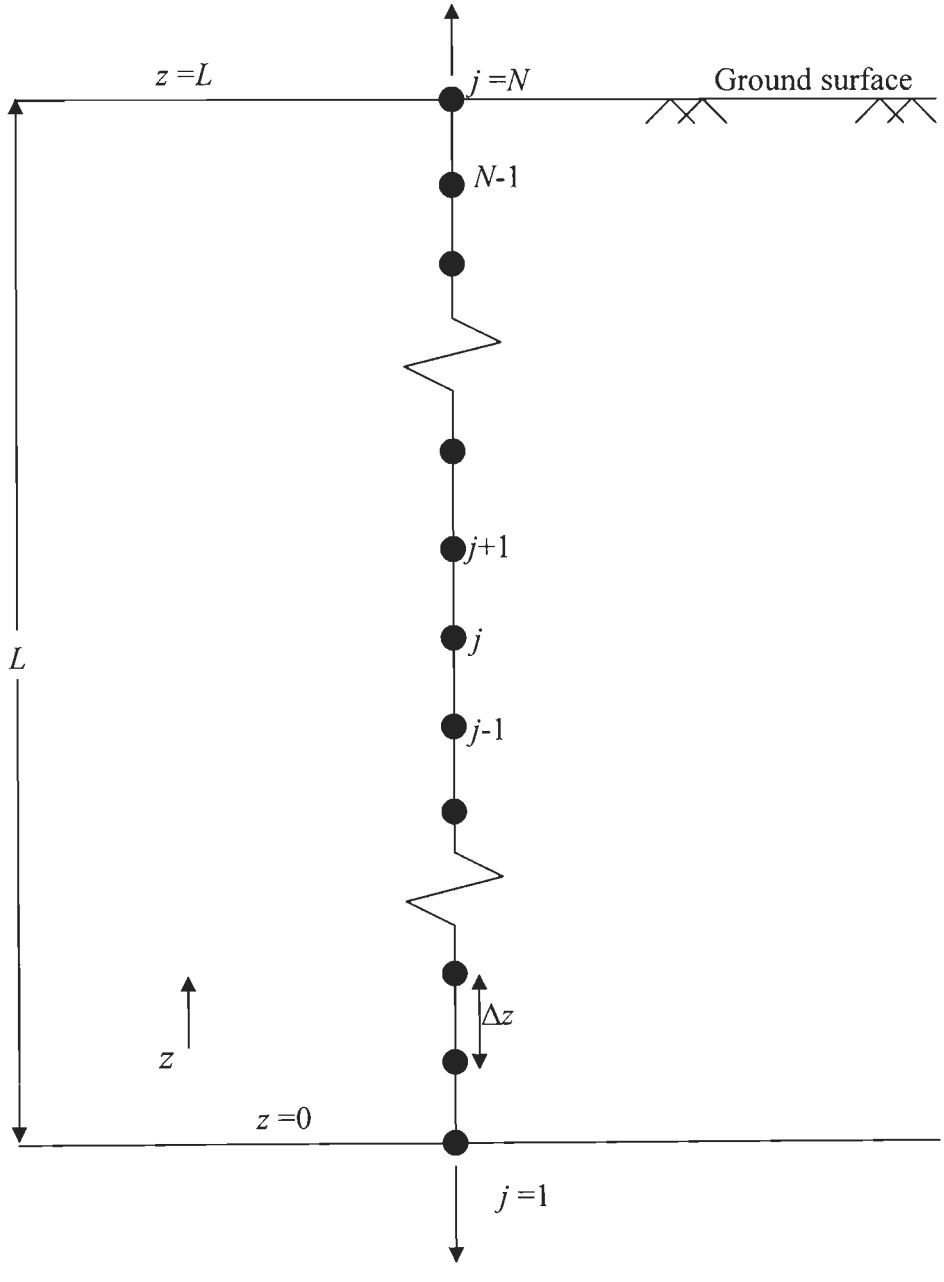

Fig 5.1 Finite difference discretization of solution domain

5.3.3.1 Spatial approximation

For a typical interior node j, a fully implicit finite difference approximation of the term on right side of the Eq. (5.2), based on Picard's scheme for the non linear terms can be written as (Clement et al., 1994)

$$\frac{\partial}{\partial z}\left\{K(\theta)\left(\frac{\partial \psi}{\partial z}+1\right)\right\} = \frac{1}{\Delta z}\left[\left(\frac{K_j^{n+1,m}+K_{j+1}^{n+1,m}}{2}\right)\left(\frac{\psi_{j+1}^{n+1,m+1}-\psi_j^{n+1,m+1}}{\Delta z}\right)\right.$$
$$\left.-\left(\frac{K_j^{n+1,m}+K_{j-1}^{n+1,m}}{2}\right)\left(\frac{\psi_j^{n+1,m+1}-\psi_{j-1}^{n+1,m+1}}{\Delta z}\right)\right]$$
$$+\frac{1}{\Delta z}\left[\left(\frac{K_j^{n+1,m}+K_{j+1}^{n+1,m}}{2}\right)-\left(\frac{K_j^{n+1,m}+K_{j-1}^{n+1,m}}{2}\right)\right] \qquad (5.13)$$

where n denotes the discrete time level at which the solution is known. $\Delta t = t^{n+1} - t^n$ is the time step. $K(\theta)$ is a nonlinear function of θ; it is linearized using a Picard iteration scheme. The current and previous Picard iteration levels are denoted as $m+1$ and m respectively. The hydraulic conductivity is arithmetically averaged between nodes. Use of arithmetic mean is justified by the finding of Kirkland et al. (1992) that solution of the Richards equation is relatively insensitive to the interblock-averaging scheme used for hydraulic conductivity. Kirkland et al. (1992) also found that use of a Crank-Nicholson scheme on the mixed form of Richards equation fails to reduce truncation error and subject to potential inabilities. So here fully implicit formulation is used.

5.3.3.2 Temporal approximation

A backward Euler approximation, coupled with a Picard iteration scheme, is used to discretize the left hand side of Eq. (5.1), containing the time derivative of water content as

$$\frac{\partial \theta}{\partial t} \approx \left[\frac{\theta_j^{n+1,m+1} - \theta_j^n}{\Delta t} \right] \tag{5.14}$$

where m denotes the Picard's iteration and n denotes the time level.

Using a fully implicit (backward Euler) time approximation and representing the water content, $\theta^{n+1,\,m+1}$, by the first order approximation

$$\theta_j^{n+1,m+1} \approx \theta_j^{n+1,m} + \left(\frac{d\theta}{d\psi} \right)_j^{n+1,m} \left[\psi_j^{n+1,m+1} - \psi_j^{n+1,m} \right] \tag{5.15}$$

The specific water capacity of a soil is defined as follows

$$C(\psi) = \frac{d\theta}{d\psi} \tag{5.16}$$

The time derivative of moisture content of Eq. (5.1) is approximated as follows.

$$\frac{\partial \theta}{\partial t} \approx \left[\frac{\theta_j^{n+1,m} - \theta_j^n}{\Delta t} \right] + C_j^{n+1,m} \left[\frac{\psi_j^{n+1,m+1} - \psi_j^{n+1,m}}{\Delta t} \right] \tag{5.17}$$

The first term on the right side of Eq. (5.17) is an explicit estimate for time derivative of water content, based on the m^{th} Picard level estimates of pressure head. In the second term of the right side of Eq. (5.17), the numerator of the bracketed fraction is an estimate of the error in the pressure head at node j between two successive Picard iterations. Its value diminishes as the Picard iteration process converges. As a result, as the Picard process proceeds, the contribution of the specific water capacity is diminished.

Interior Node:

The finite difference expressions for the spatial and temporal derivatives given in Eqs. (5.13) and (5.17) are rearranged by collecting all the unknowns on the left side and all the known on the right, in agreement with Eq (5.1).

$$\left(\frac{K_j^{n+1,m}+K_{j-1}^{n+1,m}}{2\Delta z^2}\right)\psi_{j-1}^{n+1,m+1}-\left(\frac{K_{j-1}^{n+1,m}+2K_j^{n+1,m}+K_{j+1}^{n+1,m}}{2\Delta z^2}+\frac{c_j^{n+1,m}}{\Delta t}\right)\psi_j^{n+1,m+1}$$

$$+\left(\frac{K_j^{n+1,m}+K_{j+1}^{n+1,m}}{2\Delta z^2}\right)\psi_{j+1}^{n+1,m+1}=\left(\frac{\theta_j^{n+1,m}-\theta_j^{n,m}}{\Delta t}\right)-\left(\frac{K_{j+1}^{n+1,m}-K_{j-1}^{n+1,m}}{2\Delta z}\right)-\left(\frac{c_j^{n+1,m}\psi_j^{n+1,m}}{\Delta t}\right) \tag{5.18}$$

Using the above implicit finite difference approximation, the pressure heads at the $n+1^{th}$ time level and $m+1^{th}$ Picard level are obtained from solution of the following system of simultaneous linear algebraic equations.

$$a\psi_{j-1}^{n+1,m+1}+b\psi_j^{n+1,m+1}+c\psi_{j+1}^{n+1,m+1}=d+e-f\psi_j^{n+1,m} \tag{5.19}$$

where coefficients a,b,c,d,e,f are defined as

$$a=\left(\frac{K_j^{n+1,m}+K_{j-1}^{n+1,m}}{2\Delta z^2}\right);\quad c=\left(\frac{K_j^{n+1,m}+K_{j+1}^{n+1,m}}{2\Delta z^2}\right);\quad d=\left[\frac{\theta_j^{n+1,m}-\theta_j^{n,m}}{\Delta t}\right]$$

$$e=\frac{K_{j-1}^{n+1,m}-K_{j+1}^{n+1,m}}{2\Delta z};\qquad f=\frac{c_j^{n+1,m}}{\Delta t};\qquad b=-[a+c+f] \tag{5.20}$$

Eq. (5.19) applies to all interior nodes; at boundary nodes this equation is modified to reflect the appropriate boundary conditions.

Top Node:

Drichilet type boundary condition at top node is assigned as

$$\psi_N^{j+1,m}=\psi_{top} \tag{5.21}$$

where, ψ_{top} is a prescribed head at the ground surface.

Coefficients present in the left side of the Eq. (5.19) for the top node can be written as:

$$a_N=0 \tag{5.22a}$$

$$b_N=1 \tag{5.22b}$$

$$c_N=0 \tag{5.22c}$$

Coefficient present in the right side of the Eq. (5.19) for the top node can be written as ψ_{top}.

For flux type boundary condition, the coefficients of Eq. (5.19) for the top node are:

$$a_N = \frac{K_N^{n+1,m} + K_{N-1}^{n+1,m}}{2\Delta z^2} + \frac{K_N^{n+1,m} + K_{N+1}^{n+1,m}}{2\Delta z^2} \tag{5.23a}$$

$$b_N = \frac{K_{N-1}^{n+1,m} + 2K_N^{n+1,m} + K_{N+1}^{n+1,m}}{2\Delta z^2} + \frac{C_N^{n+1,m}}{\Delta t} \tag{5.23b}$$

$$c_N = 0 \tag{5.23c}$$

$$d_N + e_N - f_N \psi_N^{n+1,m} = \left(\frac{\theta_N^{n+1,m} - \theta_N^{n,m}}{\Delta t}\right) - \left(\frac{K_{N+1}^{n+1,m} - K_{N-1}^{n+1,m}}{2\Delta z}\right) - \left(\frac{C_N^{n+1,m}\psi_N^{n+1,m}}{\Delta t}\right) - \left(\frac{-2q_{top}}{K_N^{n+1,m} + K_{N+1}^{n+1,m}} - 1\right)\left(\frac{K_N^{n+1,m} + K_{N+1}^{n+1,m}}{\Delta z}\right) \tag{5.23d}$$

In Eq. (5.23) hydraulic conductivity at imaginary node N+1 is assumed to be same as that of N^{th} node.

Bottom Node:

Dirichlet boundary condition for the bottom node is assigned as $\psi_1^{j+1,m} = \psi_{bottom}$ where ψ_{bottom} is the prescribed head at the bottom boundary. For Dirichlet boundary condition coefficients present in the left side of the Eq. (5.19) for the bottom node are:

$$a_1 = 0 \tag{5.24a}$$

$$b_1 = 1 \tag{5.24b}$$

$$c_1 = 0 \tag{5.24c}$$

Coefficients present in the right side of the Eq. (5.19) for the bottom node can be written as ψ_{bottom}.

If gravity drainage is considered at the bottom boundary, then it is a Neumann/

flux type boundary condition. To formulate the finite difference approximation, procedure similar to the one at the top node is followed for the bottom node too. For gravity drainage a unit hydraulic gradient is assumed and hence q_{bottom}, which is considered as the moisture flowing out of the boundary, is kept equal to incident hydraulic conductivity at the bottom node. The coefficients of Eqn. (5.19) for the bottom node are:

$$a_1 = 0 \tag{5.25a}$$

$$b_1 = \frac{K_0^{n+1,m} + 2K_1^{n+1,m} + K_2^{n+1,m}}{2\Delta z^2} + \frac{C_1^{n+1,m}}{\Delta t} \tag{5.25b}$$

$$c_1 = \frac{K_0^{n+1,m} + K_1^{n+1,m}}{2\Delta z^2} + \frac{K_1^{n+1,m} + K_2^{n+1,m}}{2\Delta z^2} \tag{5.25c}$$

$$d_1 + e_1 + f_1\psi_1^{n+1,m} = \left(\frac{\theta_1^{n+1,m} - \theta_1^{n,m}}{\Delta t}\right) - \left(\frac{K_2^{n+1,m} - K_0^{n+1,m}}{2\Delta z}\right) - \left(\frac{C_1^{n+1,m}\psi_1^{n+1,m}}{\Delta t}\right) + \left(\frac{-2q_{bottom}}{K_0^{n+1,m} + K_1^{n+1,m}} - 1\right)\left(\frac{K_0^{n+1,m} + K_1^{n+1,m}}{\Delta z}\right) \tag{5.25d}$$

In Eq. (5.25) hydraulic conductivity at imaginary node $N = 0$ is assumed to be same that of 1st node. The term q_{bottom} in the approximation is equivalent to the corresponding hydraulic conductivity at each iteration and time level.

The resulting system linear algebraic equations, for the unknown pressure-head values, is written in a matrix notation

$$\mathbf{A}\boldsymbol{\psi} = \mathbf{B} \tag{5.26}$$

where **A** is coefficient matrix consisting of coefficients of the finite difference equation (5.19), $\boldsymbol{\psi}$ is vector of unknown pressure heads and **B** is known right hand side vector. The coefficient matrix **A** is tridiagonal in nature. The set of linear equations thus formed for all the nodes in the solution domain, results into a tridiagonal matrix, which is solved by *Thomas algorithm* (Remson et al., 1971).

Convergence Criteria:

For the first time step of a transient simulation, the initial estimate used is same as the initial conditions supplied as input to the numerical program. For all subsequent time steps the initial estimates can be extrapolated from the pressure heads at previous time steps (Cooley, 1971) and Huyakorn et al., 1984).

Convergence in the computer implementations of the Picard scheme is monitored by computing the maximum error norm $\left\| \psi_j^{n+1,m+1} - \psi_j^{n+1,m} \right\|_\infty$. Convergence is achieved when norm falls below some specified tolerance level. This represents an absolute convergence criterion. Input tolerance level affects the accuracy of the numerical solution, within limits imposed by spatial and temporal truncation error (Paniconi et al., 1991). In the present work the error at each node, for each iteration level is obtained by

$$\varepsilon_j = \left\| \psi_j^{n+1,m+1} - \psi_j^{n+1,m} \right\| \qquad 1 \le j \le N \tag{5.27}$$

The solution at each iteration level converges when maximum change falls below the pre-stipulated value of the convergence factor ε_2, which is taken as 0.001 m in the present work.

$$\text{Max } (\varepsilon_j) < \varepsilon_2$$

5.3.4 Model Validation

The unsaturated flow numerical model is validated by considering different problems of unsaturated flow with diverse boundary conditions, available in literature. The model results are compared with the reported results and are discussed below.

5.3.4.1 Infiltration into a very dry soil with Dirichlet type boundary condition at top

Problem of infiltration into a very dry soil solved by Celia et al. (1990) is taken here for model validation. This problem considers infiltration into a

homogeneous soil column, which is initially dry. The soil parameters are α_v = 0.0335 cm^{-1}, θ_s = 0.368, θ_r = 0.102, n_v=2, m_v=0.5, K_s = 0.00922 cm/s. Length of the soil sample is 100 cm. The initial and boundary conditions are;

$$\psi(z,0) = -1000 \text{ cm}, \quad 0 \leq z \leq 100 \text{ cm}$$

$$\psi(0,t) = \psi_{bottom} = -1000 \text{ cm}$$

$$\psi(40, t) = \psi_{top} = -75 \text{ cm}$$

Celia et al. (1990) obtained finite element as well as finite difference solution using coarse and dense grid. For the dense grid consideration, the over all soil domain length L is divided into 101 grids such that the distance between two grids will be 1.0 cm. The problem is simulated using the present model with Δt = 20 sec. Fig. 5.2 compares the model predicted pressure head after one day of simulation with Celia et al. (1990) dense grid simulation.

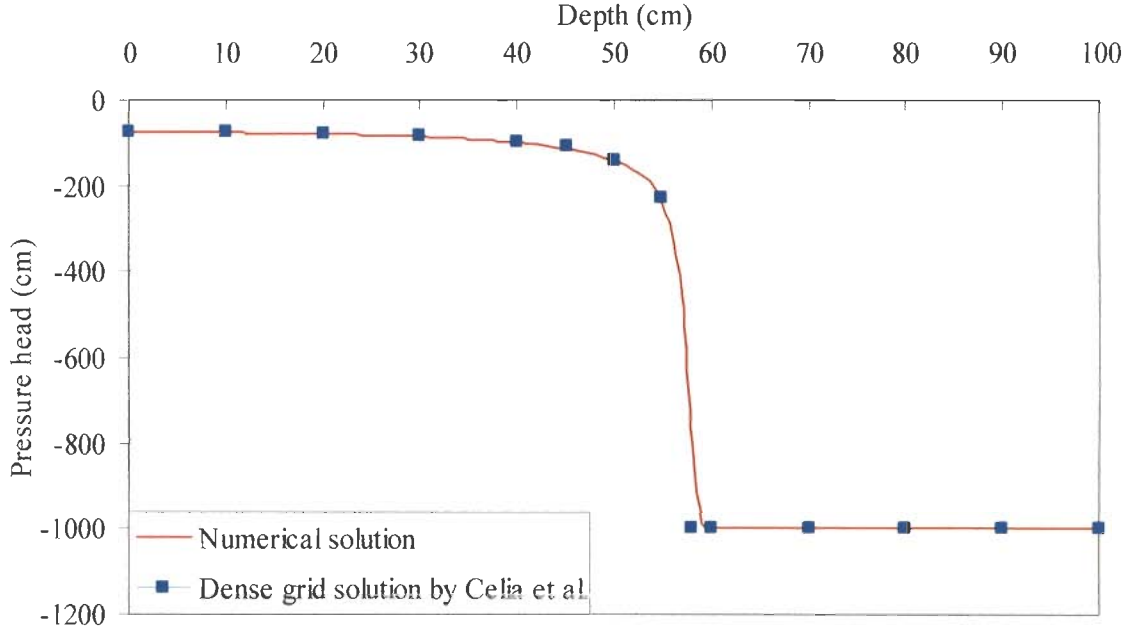

Fig 5.2 Model validation for infiltration into a very dry soil – Dirichlet boundary condition at top

It is clear from Fig. 5.2 that the model predictions are in excellent agreement with the reported predictions.

5.3.4.2 Gravity drainage from an initially saturated soil

Dane and Hruska (1983) simulated gravity drainage from a hypothetical soil with the following set of soil parameters. α_v = 0.02912 cm^{-1}, n_v = 3.57168, K_s = 0.00305 cm/sec, θ_s = 0.365 and θ_r = 0.069.

The problem involves allowing a soil column of length (L) = 1.4m, which is at an initial moisture content of 0.3 throughout, to drain due to gravity at the lower boundary. The Darcy's flux at the top is zero.

The initial and boundary conditions for the problem are as follows

$t = 0$: $\theta = 0.30$ $\quad 0 \le z \le 1.4$m

$t > 0$: $q_{bottom} = -K$ $\quad z = 0$

$q_{top} = 0$ $\quad z = 1.4$ m

The problem is simulated using the present model with Δz=2 cm. Fig 5.3 shows a comparison between the moisture contents obtained after 12 hours of simulation by Dane and Hruska and the present model.

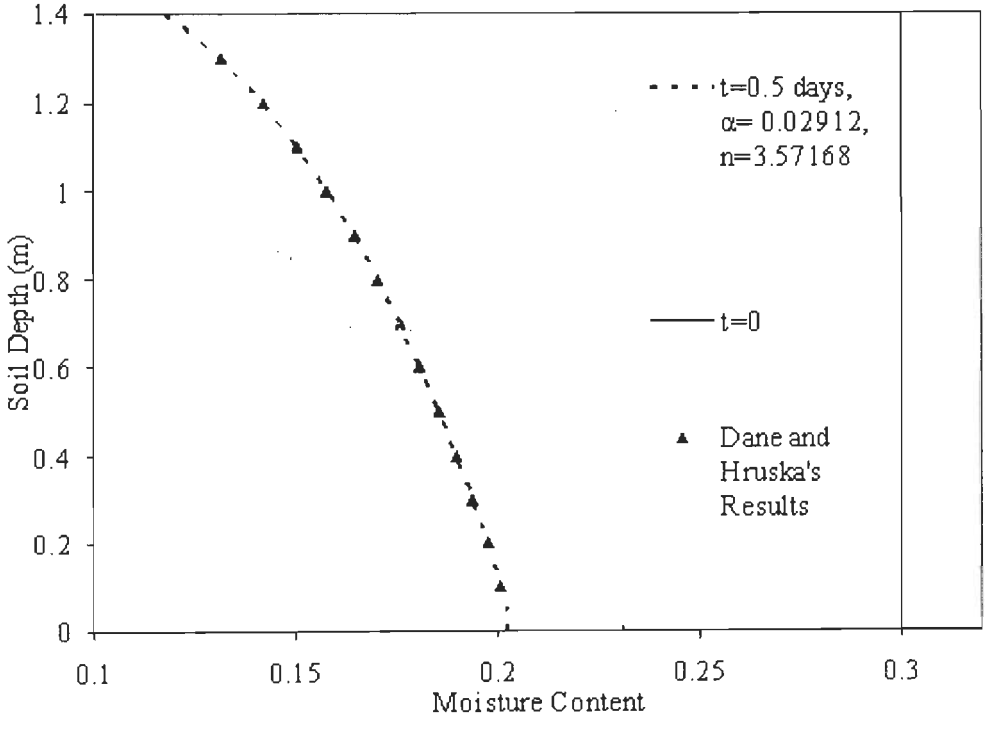

Fig 5.3 Model validation for gravity drainage from an initially saturated soil

It can be seen from Fig. 5.3 that results are in excellent agreement with those of Dane and Hruska.

5.3.4.3 Infiltration into a very dry soil with Neuman type boundary condition at top

Problem 3 considers infiltration into an initially dry, producing sharp moisture fronts and a four order of magnitude change in relative hydraulic conductivity across the wetting front (Paniconi et al., 1991). The parameter values for sharp front infiltration are, K_s = 1.11 × 10^{-5} cm/sec, θ_s = 0.38, θ_r = 0.15, n_v = 4.0 and L =1.25m. The initial and boundary conditions are

$t = 0$: ψ = -3.0 m	$0 \leq z \leq 1.25$ m
$t > 0$: ψ = -3.0m	$z = 0$
$t > 0$: q = 0.0008 m^3/hr	z=1.25m

The problem is simulated using the present model with Δz = 0.004167 m and Δt = 0.01hr. Figs 5.4 and 5.5 show a comparison between pressure head and moisture content obtained after 120 hours of simulation by Paniconi et al. (1991) and the present model.

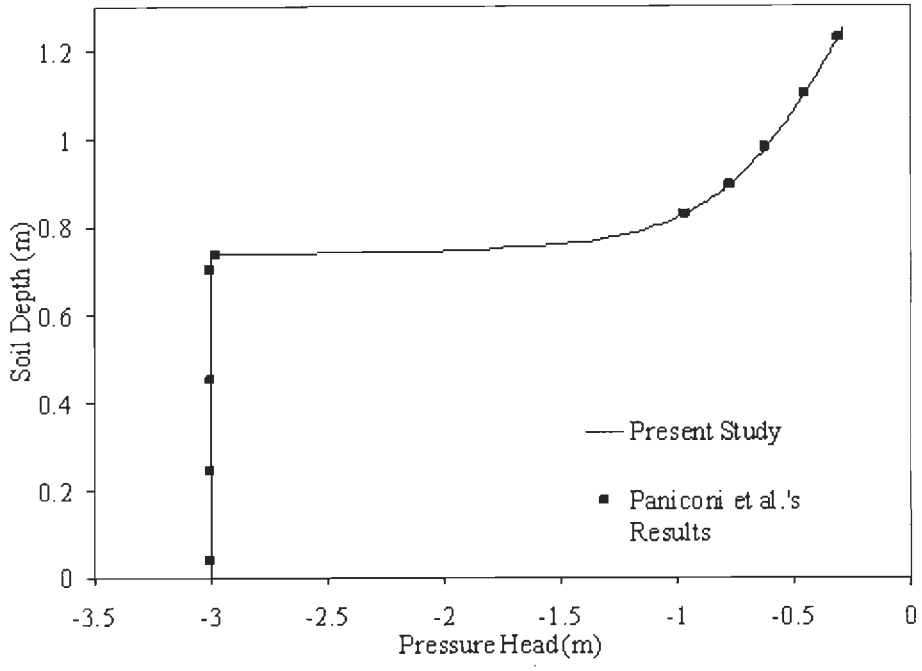

Fig. 5.4 Model validation for infiltration into a dry soil- Neuman boundary condition: Comparison of Pressure heads

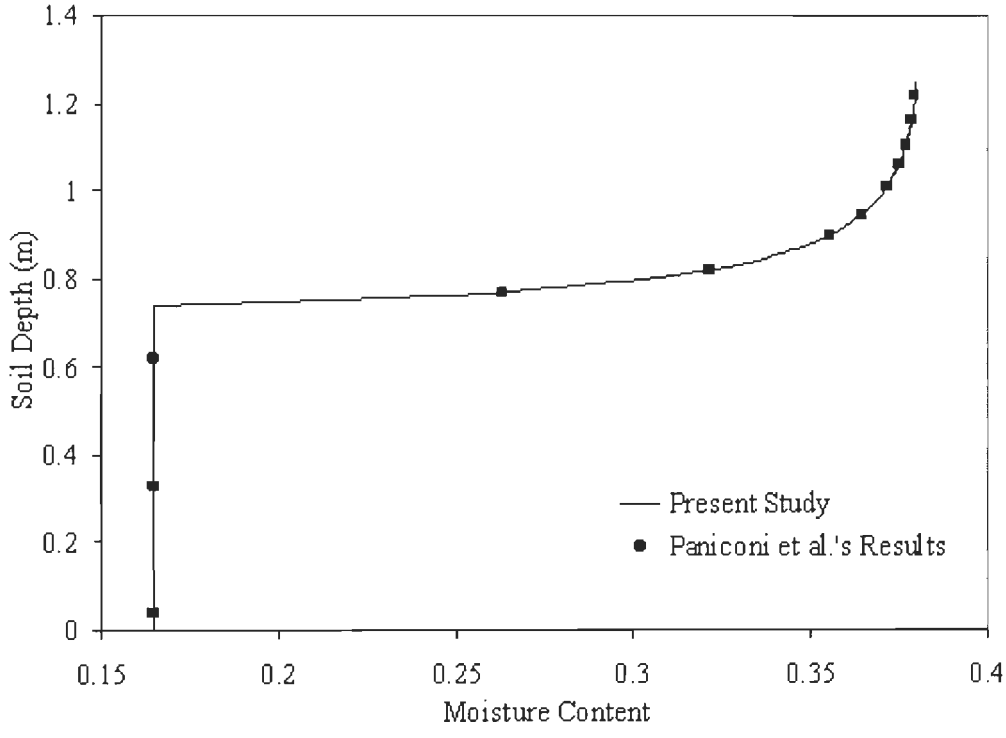

Fig. 5.5 Model validation for infiltration into a dry soil- Neuman boundary condition: Comparison of moisture content

It is evident from Fig. 5.4 and Fig. 5.5 that the results of the present model are in excellent agreement with those of Paniconi et al. (1991).

5.4 COUPLING OF MOISTURE FLOW AND VIRUS TRANSPORT MODELS

As discussed earlier, the moisture flow and virus transport models (Eq. 5.1 and 5.2) are coupled partial differential equations as velocity v, appearing in Eq. (5.2) has to be obtained by solving Eq. (5.1).

Numerical solution of Eq. (5.1) discussed in Section 5.4, provides the nodal pressure heads in the solution domain at successive time steps. Let ψ_j^n represent the

pressure head at j^{th} node at time level n. From nodal pressure heads, the corresponding seepage velocity are computed as

$$v_j^n = -\frac{K_j^n}{\theta_j^n}\left(\frac{\psi_{j+1}^n - \psi_{j-1}^n}{2\Delta z} + 1\right) \tag{5.28}$$

where θ_j^n is the moisture content and K_j^n is the unsaturated hydraulic conductivity at node j at time level n. These model seepage velocities are used in Eq. (5.2) and Eq. (5.2) is solved using hybrid finite volume model as discussed in Chapter 3. A computer code is written in FORTRAN 90 for the implementation of the numerical model which is provided in APPENDIX-III. Since the performance of the virus transport model has been discussed in detail in Chapter 3, in the present Chapter the application of the model for the analysis of virus transport in unsaturated zone is presented.

5.5 MODEL APPLICATION

Analytical solutions for virus transport through unsaturated zone are available only for steady state flow and constant moisture content throughout the solution domain (Ogata and banks, 1961; Van Genuchten and Alves, 1982). A review of literature suggests that analytical solution for virus transport in case of unsteady unsaturated flow with variable moisture content in the solution domain is not available. Hence, in the present Section the application of the developed numerical model is demonstrated through an example.

5.5.1 Analysis of Moisture Flow

The unsaturated medium is initially assumed to be very dry at a pressure head

of -1000 cm in a soil length of 500 cm. A pressure head of -75 cm is applied at the ground surface. Due to the change in pressure head, the moisture starts flowing into the soil. The soil parameters are α_v = 0.0335 cm^{-1}, θ_s = 0.368, θ_r = 0.102, n_v = 2, m_v = 0.5 and K_s = 0.00922 cm/s. The initial and boundary conditions are;

Initial condition:

Initially at t = 0, the head is assumed to be -1000 cm at each node, i.e.

$$\psi(z,0) = -1000 \text{ cm}, \quad 0 \leq z \leq 100 \text{ cm} \tag{5.29}$$

Boundary condition:

For $t > 0$, the head at bottom of the solution domain is assumed to be -1000 cm where as 75 cm is assumed at the top.

$$\psi\ (0,t) = \psi_{bottom} = -1000 \text{ cm} \tag{5.30}$$

$$\psi\ (100, t) = \psi_{top} = -75 \text{ cm} \tag{5.31}$$

The numerical model discussed in Section 5.4 is used to obtain the nodal pressure heads and corresponding seepage velocities by solving Eq. (5.2) subjected to initial and boundary conditions given in Eq. (5.29) to (5.31). Fig. 5.7 shows the variation of seepage velocity at each node after 1, 2, 4, 8, 10, 15 days interval. It is seen from Fig. 5.6 that as the time period increases the nodal seepage velocity becomes constant at higher depth. Fig. 5.7 shows the variation of moisture content with depth at each node after 1, 2, 4, 8, 10, 15 days interval. It is seen from Fig. 5.7 that the moisture has flown upto 450 cm from ground surface after 15 days.

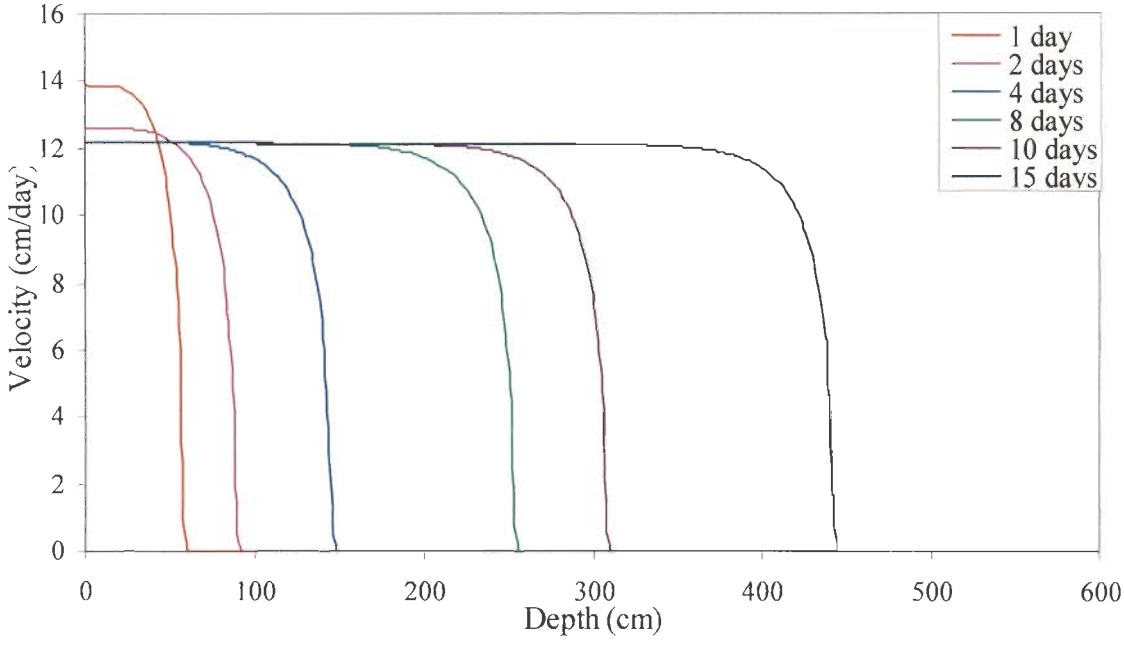

Fig. 5.6 Variation of seepage velocity with depth

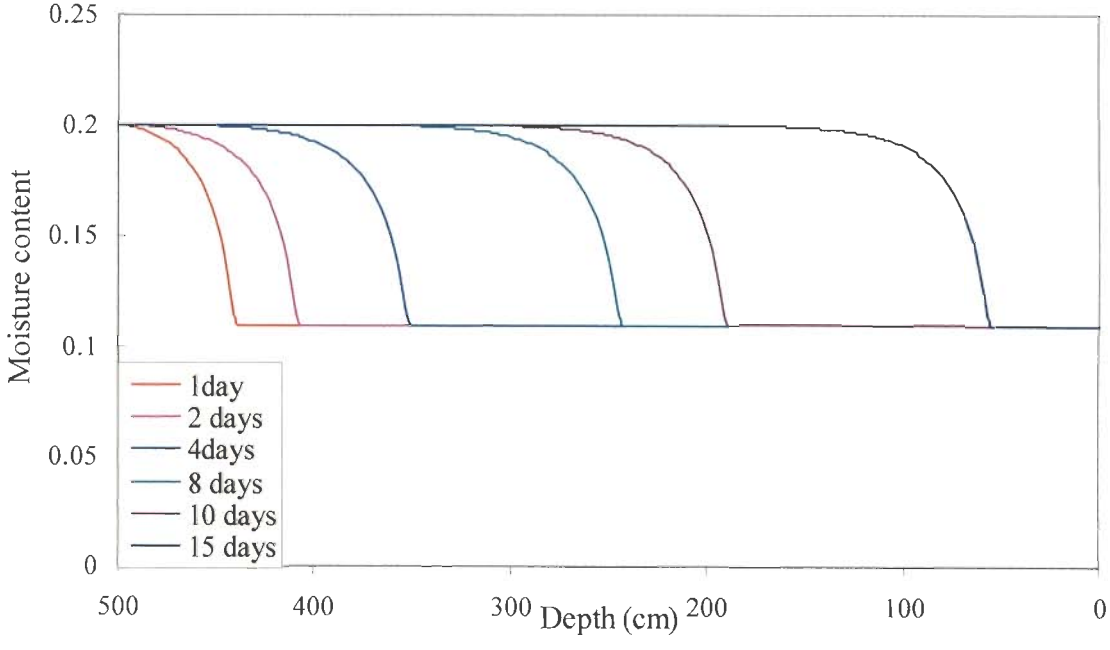

Fig. 5.7 Variation of moisture content with depth

5.5.2 Analysis of Virus Transport due to Injection of Infinite Duration

Initially, the virus concentration in the unsaturated soil domain is assumed to be zero. A continuous source of virus is imposed such that the concentration at the ground surface is 1 concentration units (C_0 = 1). The bulk density of soil is assumed as 1.11 gm/cm^3. The inactivation coefficient of liquid and sorption phase is assumed as 0.58 /day. The distribution coefficient is assumed as 0.02 ml/gm. The initial and boundary conditions are;

Initial condition:

Initially, i.e. at t=0, the concentration of virus is usually assumed to be zero, i.e.

$$t = 0,\ C(x) = 0, \qquad 0 \le x \le \infty \tag{5.32}$$

Boundary conditions:

At the source (x=0), a constant concentration boundary condition is used. i.e.

$$t \ge 0,\ C(t) = C_0, \qquad x = 0 \tag{5.33}$$

where C_o (t) denotes the source concentration.

For away from the source ($x \rightarrow \infty$) the concentration flux is set to zero. i.e.

$$t \ge 0, \quad \frac{\partial C}{\partial x} = 0, \qquad x \rightarrow \infty \tag{5.34}$$

The virus transport model is used to obtain the virus concentrations for Peclet numbers 1 and 100 by solving Eqs. (5.2) subjected to initial and boundary conditions (Eqs. 5.32 to 5.34). Fig. 5.8 and Fig. 5.9 show the model predicted virus concentrations at each node after 1, 2, 4, 8, 10, 15 days interval for Peclet numbers 100 and1 respectively. It can be seen from Fig. 5.8 and 5.9 that for advection dominated transport (P_e = 100), the systems reaches steady state in 8 days as compared to 10 days in dispersion dominated transport.

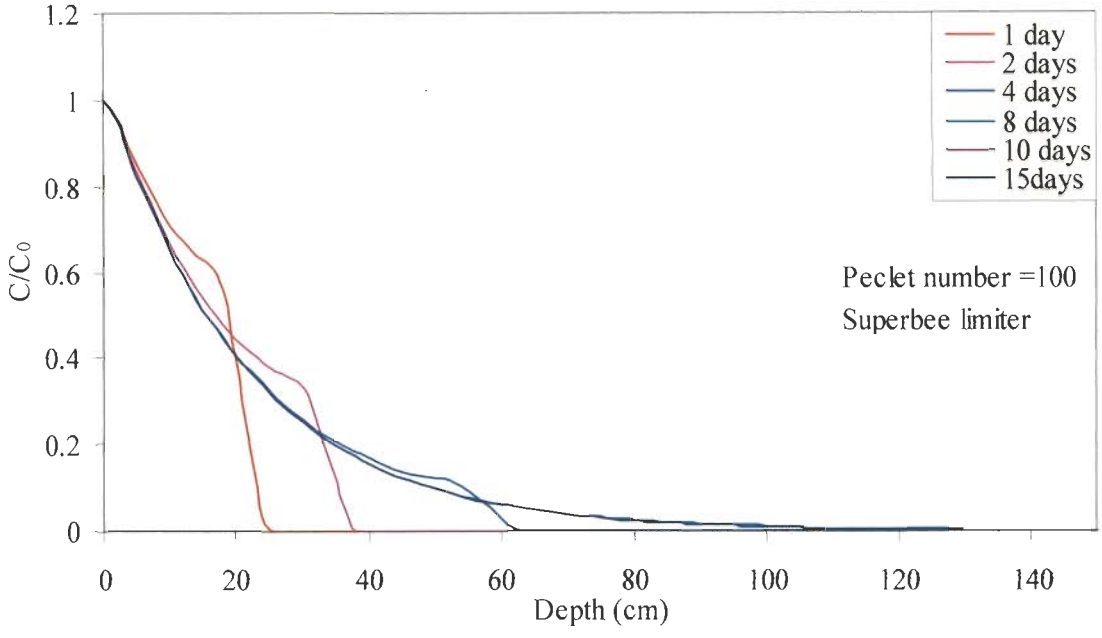

Fig 5.8 Variation of concentration of virus in unsaturated soil after 10 days for Peclet number 100

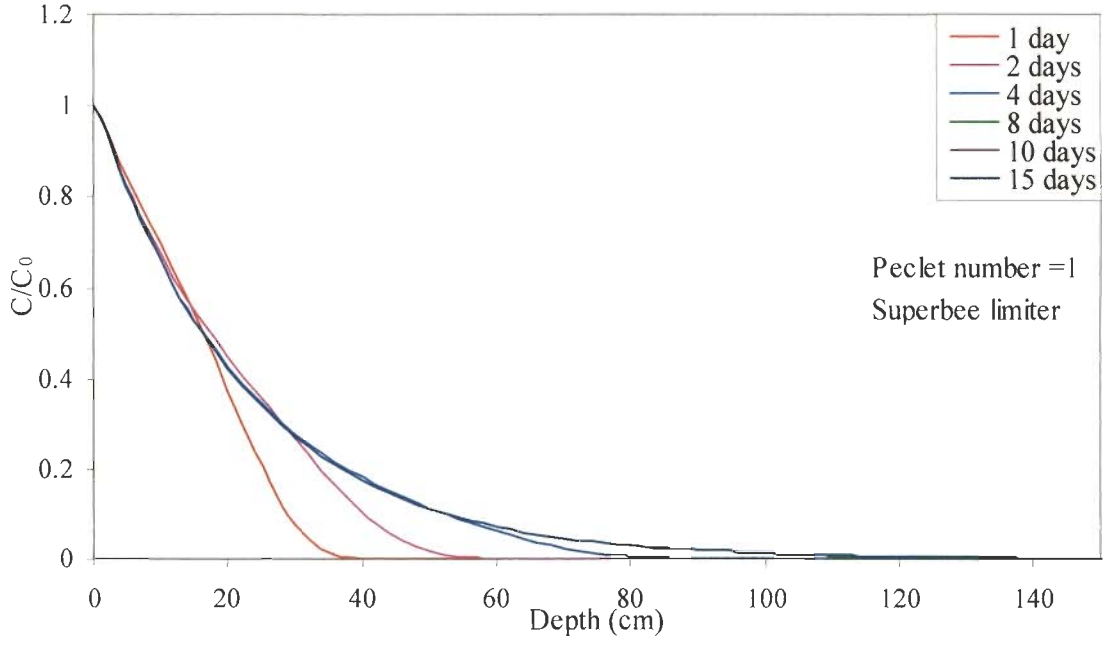

Fig 5.9 Variation of concentration of virus in unsaturated soil for Peclet number 1.0

5.6 CONCLUDING REMARKS

In the present Chapter, a numerical model is developed for analysis of virus transport in unsaturated zone. The model couples a mass conservative fully implicit finite difference model simulating moisture flow in the unsaturated zone with the hybrid finite volume model (Discussed in detail in Chapter 3) for virus transport. The applicability of the model is demonstrated with an example. It is observed that virus transport reaches steady state earlier in advection dominated transport as compared to dispersion dominated transport.

CHAPTER 6

ESTIMATION OF VIRUS TRANSPORT PARAMETERS IN UNSATURATED ZONE

6.1 INTRODUCTION

In Chapter 5, a numerical model is developed for the analysis of virus transport in unsaturated zone. In the present Chapter, the numerical model is coupled with Levenberg-Marquadart algorithm to estimate the virus transport parameters. Identification of flow and transport parameters in the unsaturated zone is much more complicated as compared to the saturated zone due to the significant variation of the seepage velocity and the moisture content which affect the virus movement considerably. The present Chapter deals with the estimation of only the transport parameters and does not involve the estimation of flow parameters. An optimization model is developed using Levenberg-Marquadart algorithm and is applied to estimate the transport parameters from hypothetical virus concentration data. Further the effect of data errors and bias is also discussed. The algorithm is also used to estimate the virus transport parameters from a column experiment.

6.2 GENERAL FORMULATION OF THE ESTIMATION PROBLEM

As discussed in Chapter 4, the inverse problem is formulated as a nonlinear optimization problem i.e. the transport parameters are estimated by minimizing the deviation between observed and model predicted response as

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right]^T \mathbf{W}\left[\mathbf{C}^* - \mathbf{C}(\mathbf{b})\right] \tag{6.1}$$

where the objective function, $O(\mathbf{b})$, is a function of the model parameters **b**, $\mathbf{b} = \{b_1, b_2, ---b_m\}^T$; $\mathbf{C}^* = \{C_1^*, ---C_n^*\}^T$ is the observation vector whose elements represent measured concentrations; $\mathbf{C(b)} = \{C_1(b), ---C_n(b)\}^T$ represents the predicted response for a given parameter vector **b** and **W** is the symmetric weighting matrix. In the present study, the parameter vector **b** comprises inactivation coefficient in liquid phase λ, inactivation coefficient in sorption phase λ^* and distribution coefficient k_d i.e. $\mathbf{b} = \{\lambda, \lambda^*, k_d\}^T$. The objective is to find the optimum parameter vector **b** that minimizes the objective function 6.1. When the observation errors are assumed to be independent and normally distributed the weighting matrices **W** becomes an identity matrix and Eq. (6.1) reduces to ordinary least squares (OLS) problem.

$$\min_{\mathbf{b}} O(\mathbf{b}) = \frac{1}{2}\left[\mathbf{C}^* - \mathbf{C(b)}\right]^T \left[\mathbf{C}^* - \mathbf{C(b)}\right] = \frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{C}^* - \mathbf{C(b)}\right]^2 \qquad (6.2)$$

where N is the number of observations.

The OLS formulation has probably been the most popular one for parameter estimation problems. Its attraction is due to its simplicity and because it requires a minimum amount of information. When observation errors are normally distributed, are uncorrelated and have a constant variance, the OLS estimates possess optimal statistical properties. When these conditions are not met, the OLS method will no longer yield optimal parameter estimates in terms of precision and minimum variance.

6.3 SOLUTION ALGORITHM

The solution algorithm involves the estimation of transport parameters by coupling the numerical model with Levenberg-Marquadart algorithm. The details of

the procedure have been already discussed in Section 4.3. A computer code is written in FORTRAN 90 to implement the algorithm and is presented in APPENDIX-IV.

6.4 IDENTIFICATION OF TRANSPORT PARAMETERS

In the present Section, the parameter estimation algorithm developed is used to estimate the virus transport parameters λ, λ^* and k_d. Initially it is checked whether the optimization yields unique estimates of the transport parameters from hypothetical generated virus concentration data and the identifiability of the parameters are discussed. Later, the effect of noise (errors) in the measurements on the parameter estimates is studied.

Hypothetical virus concentration data:

Hypothetical data of virus concentration as a function of time are generated by solving Eqs. (5.1) and (5.2) subject to initial and boundary conditions given by Eqs. (5.7 to 5.9 and 5.29 to 5.31). The soil is initially assumed to be very dry at a pressure head of -1000 cm and the virus concentration in the soil is assumed to be zero. A pressure head of -75 cm and a virus concentration of 1 unit is applied at the ground surface. The pressure head at far below the ground surface is assumed to be -1000 cm and the virus concentrations is assumed to be zero. The soil parameters used in the simulation are α_v = 0.0335 /cm, θ_s = 0.368, θ_r = 0.102, n_v = 2, m_v = 0.5, K_s = 0.00922 cm/sec and the transport parameters are assumed as λ =0.58 /day, λ^* = 0.46 /day, k_d = 0.02 ml/gm, ρ = 1.11 gm/cm^3. Virus concentration at discrete times (0.5, 1.0, 2.0, 4.0 and 5.0 days) and at a discrete distances from the source (1.0, 2.0, 4.0, 8.0, 10.0, 12.0, 15.0, 20.0 and 22.0 cm) is generated by solving Eq (5.1) and (5.2) subjected to initial and boundary conditions given in Eqs. (5.29 to 5.34). Table 6.1 presents the hypothetically generated virus concentration data. These data are used as observed concentration data in the parameter estimation.

Table 6.1: Hypothetical virus concentrations data for parameter estimation

Time (Days) / Distance (cm)	0.5	1.0	2.0	4.0	5.0
1.0	0.97379	0.96879	0.96556	0.96442	0.96433
2.0	0.94094	0.92906	0.92142	0.91874	0.91855
4.0	0.87664	0.84834	0.83069	0.82455	0.82408
8.0	0.75499	0.71616	0.6757	0.66179	0.66083
10.0	0.59968	0.66853	0.61076	0.59302	0.59181
12.0	0.27042	0.62264	0.55436	0.53155	0.52997
15.0	0.02234	0.51631	0.4816	0.4514	0.44928
20.0	0.00006	0.06739	0.39465	0.34455	0.3414
22.0	0.0	0.01417	0.36527	0.30961	0.30603

The robustness of the optimization procedure is studied by changing the number of transport parameters to be estimated from 1 to 3. In addition, the efficacy of the optimization procedure is analyzed by starting the initial guesses of individual parameters considerably far away from their true values. The parameter estimation is discussed in detail in the following Sections.

6.4.1 Case 1: Estimation of One Unknown Parameter

Case 1 considers the estimation of one unknown transport parameters while treating the other two parameters as constant to their respective values used for the generation of hypothetical data. Further two sub cases (case A and case B) are considered. In case A, the initial guess parameter is over estimated by one order from its true value while in case B it is under estimated by one order from its true value. Table 6.2 presents the initial guess values and the optimal estimated values of the transport parameter λ, λ^*, and k_d. It is clear from Table 6.2 that the optimization algorithm converges to the true values in both sub cases (case A and case B). Further it can be seen from Table 6.2 that starting the initial guess as under estimated value results in less number of iterations to converge to the optimal solution.

Table 6.2: Parameter estimates for the hypothetical data- Case 1

Parameter	True values	Case A (over estimated)			Case B (Under estimated)		
		Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations
λ (/day)	0.58	5.8	0.5804	15	0.058	0.5798	9
λ^* (/day)	0.46	4.6	0.4629	13	0.046	0.4574	9
k_d (ml/gm)	0.02	0.2	0.02013	13	0.002	0.0198	9

Table 6.3: Parameter estimates for the hypothetical data – Case 2

Parameters	True values	Case A (over estimated)			Case B (Under estimated)			Case C (Mixed)			Case D (Mixed)		
		Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations	Initial guess	Final estimate value	No of iterations
λ (/day)	0.58	5.8	0.58	12	0.058	0.5813	7	5.8	0.5792	7	0.058	0.574	18
k_d (ml/gm)	0.02	0.2	0.0203		0.002	0.0199		0.002	0.0199		0.2	0.021	
λ^* (/day)	0.46	4.6	0.4574	32	0.046	0.461	27	4.6	0.4573	29	0.046	0.452	32
k_d (ml/gm)	0.02	0.2	0.02		0.002	0.0199		0.002	0.02		0.2	0.0208	

6.4.2 Case 2: Estimation of Two Unknown Parameters

In case 2, two among the three transport parameters are considered as unknown and are estimated while keeping other parameter as constant to its value used for the generation of the hypothetical data. Such an estimation results in three combinations of two unknown parameters; (λ, k_d), (λ, λ^*) and (λ^*, k_d). For each of these combinations, four subcases are considered. In case A, the initial guess of the parameters are over estimated by one order from their true values. In case B, the initial guess of the parameters are underestimated by one order. In case C, the initial guess of the first parameter is over estimated by one order while the initial guess of the second parameter is under estimated by one order. In contrast in case D, the initial guess of the first parameter is under estimated by one order while initial guess of second parameter is over estimated by one order. During optimization, it is observed that for the particular combination in which the inactivation coefficients λ and λ^* are considered as unknown parameters, the optimization resulted in non unique solutions. Table 6.3 presents the initial guess values and the optimal parameter estimates for case 2. It can be seen from Table 6.3 that the optimization results in the estimation of true values in all the cases. Further, starting the initial guess as under estimated values resulted in the least number of iterations needed for convergence to the optimal solution.

6.5 DATA ERROR AND BIAS

In Section 6.4, the performance of the optimization algorithm is studied by estimating parameters from hypothetically generated error free virus concentration data. As stated by Kool et al. (1987), the reliability of the parameters estimates

obtained by inverse procedure greatly depends upon the quality of the experimental data. The present Section discusses the effect of data errors and bias induced by the objective function on the transport parameter estimates. In the absence of errors, the identification procedure normally identifies a unique set of values for the parameters. In their presence, the individual values of the identified parameters depend on the individual samples and therefore the identified parameters do not appear to be unique. However the treatment of the problem in a statistical sense using the sampling theory clearly render the uniqueness of the means, provided that the objective function does not introduce any undue bias. The present study is aimed at evaluating the bias induced by the objective function in the presence of errors. For this purpose Gaussian noise is added to hypothetically generated virus concentration data (Refer Section 6.4) through specifying a mean μ and a standard deviation σ as follows:

$$C_{o,i,j} = C_{t,i,j}\varepsilon \tag{6.3}$$

with

$$\varepsilon = N(\mu, \sigma) \tag{6.4}$$

where $C_{o,i,j}$ is the observed virus concentration and $C_{t,i,j}$ is the hypothetically generated virus concentration data and ε is the Gaussian error. i and j are the number of space and time observations respectively. The μ of every sample is assigned a value equal to 1 to ensure unbiased perturbation through Eq (6.3), where the errors are randomly distributed above and below $C_{t,i,j}$. For each value of σ, 10 samples are generated by changing the seeding of the random number generator. In the present study, σ is changed from 0.025 to 0.15 in increments of 0.025.

As discussed in Section 6.4, when the data contains no errors, the objective function converges to the true values of the parameters as shown in Table 6.2 and 6.3

for one and two unknown parameters respectively. This indicates that when data is free from errors, the objective function does not induce any bias in the estimated parameters. When the data contains errors, the objective function does not converge to true values. Percentage errors in estimating the transport parameters λ, λ^* and k_d are presented in Tables 6.4 to 6.6 respectively at two noise levels 0.05 and 0.1 for 10 different samples.

Table 6.4: Effect of data error and objective function on estimated parameter λ

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of λ	Error percentage in λ	Value of λ	Error percentage in λ
1	0.49401	14.82586	0.40859	29.55
2	0.39255	32.31897	0.21825	62.37
3	0.55605	4.12931	0.53198	8.2793
4	0.54788	5.537937	0.51632	10.979
5	0.96919	-67.1017	1.38399	-138.61
6	0.56928	1.8482	0.55824	3.7517
7	0.43024	25.8206	0.27323	52.891
8	0.13669	76.4327	0.00188	99.675
9	0.021474	96.2795	0.15838	72.693
10	0.78158	-34.7552	0.99207	-71.046

Table 6.5: Effect of data error and objective function on estimated parameter λ^*

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of λ^*	Error percentage in λ^*	Value of λ^*	Error percentage in λ^*
1	0.042427	90.7767	0.021722	95.277
2	0.03394	92.6217	0.20231	56.0195
3	0.2704	41.2173	0.25631	44.2804
4	0.49392	-7.3739	0.49392	-7.3739
5	4.0933	-789.84	7.86999	-1610.87
6	0.39513	14.1021	0.39513	14.1021
7	0.003503	99.2384	0.003473	99.245
8	0.20231	56.0195	0.39513	14.1021
9	0.77175	-67.7717	0.02715	94.0978
10	2.4565	-434.022	4.4955	-877.283

Table 6.6: Effect of data error and objective function on estimated parameter k_d.

Sample No	Noise level σ =0.05		Noise level σ =0.1	
	Value of k_d	Error percentage in k_d	Value of k_d	Error percentage in k_d
1	0.017269	13.655	0.014163	29.185
2	0.02169	-8.45	0.023047	-15.235
3	0.019139	4.305	0.018245	8.775
4	0.015301	23.495	0.009232	53.84
5	0.022316	-11.58	0.024326	-21.63
6	0.006186	69.07	0.000198	99.01
7	0.028831	-44.155	0.03862	-93.1
8	0.0255	-27.5	0.003036	84.82
9	0.01484	25.8	0.008399	58.005
10	0.04	100	0.061985	-209.925

From Tables 6.4 to 6.6, it is concluded that i) the noisy data introduces errors in the estimated parameters. ii) the amount of induced error increases with the noise level and iii) for any given noise level, the individual values of identified parameters deviate from the true value. In addition it is observed that the error percentage in case of λ^* is very high as compared to other parameters. As the individual values of the identified parameters generally deviate from their true values, the error contained in these values reveal very little in their behavior and hence statistical analysis is necessary in terms of means and confidence intervals. In the present study, 95% confidence interval is used for carrying out the statistical analysis.

Figs 6.1 to 6.3 show the mean values and the 95% confidence intervals of estimated parameters λ, λ^* and k_d respectively.

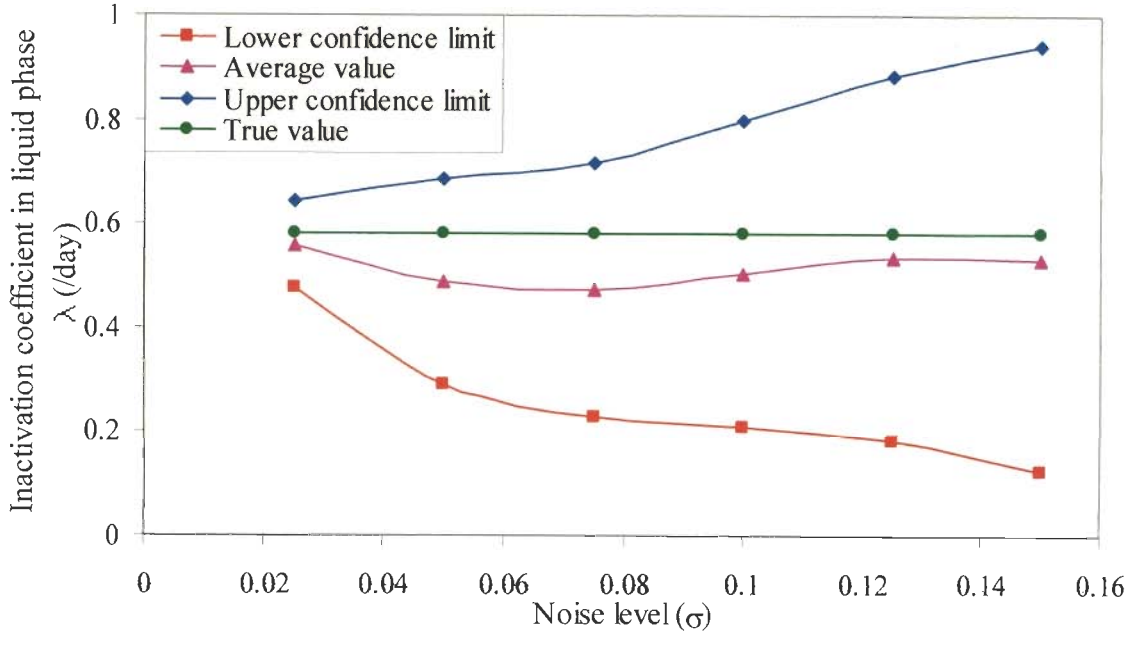

Fig 6.1 Variation of means of identified λ with noise level σ

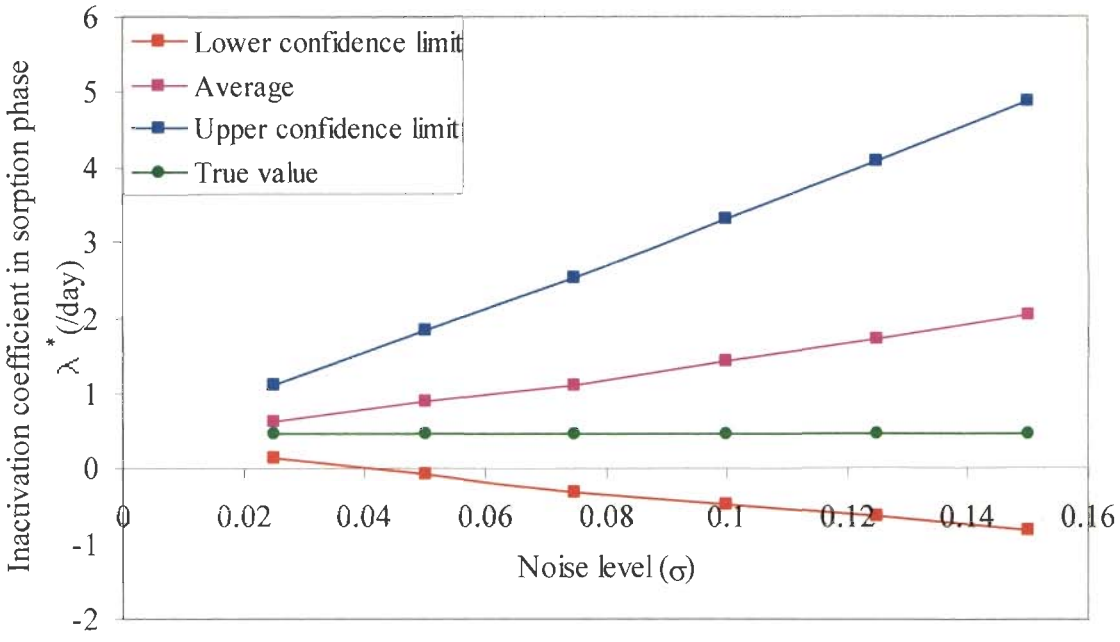

Fig 6.2 Variation of means of identified λ^* with noise level σ

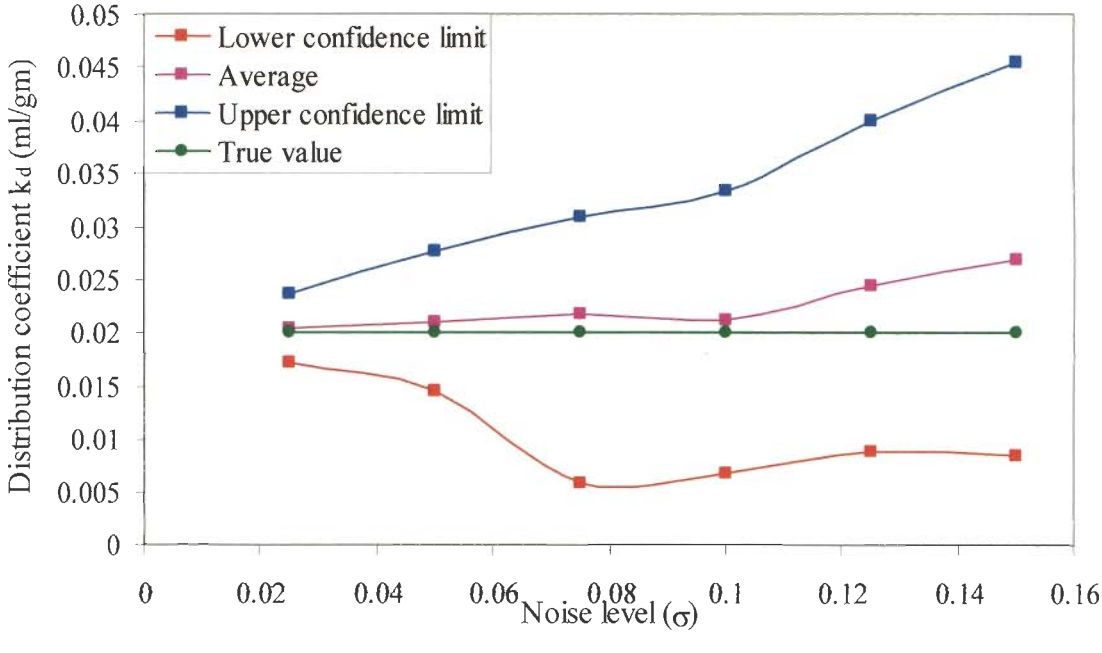

Fig 6.3 Variation of means of identified k_d with noise level σ

The following conclusions can be drawn from Figs 6.1 to 6.3. i) the deviation of the mean from the true value increases with an increase in noise level in case of parameters λ^*. However, the mean remains significantly unchanged and is closer to the true value in case of λ and k_d. ii) the true value is contained within the 95% confidence interval of all the identified parameters at all noise levels indicating that the objective function does not induce bias on the parameter estimates, when they are estimated individuals.

6.6 ESTIMATION OF TRANSPORT PARAMETERS FROM COLUMN EXPERIMENT

The optimization algorithm is also applied to estimate the transport parameters from the virus concentration data of a column experiment involving virus movement in unsaturated zone. Jin et al. (2000) conducted virus transport experiment for unsaturated conditions also. The apparatus used for unsaturated experiments was more complicated, with an additional solution-filling column positioned on top of the transport column. Column outlet was connected to a vacuum chamber with a fraction collector inside. By adjusting the vacuum pressure and flow rate of the input solution, steady state and essentially uniform water content was reached.

The column was made of acrylate and was 7.6 cm in diameter and 10 cm long. The experiments were conducted in a cold room at 4^0 C to minimize the inactivation due to high temperature. Viruses were added to the sand column as a constant input at an approximate concentration of 5×10^4 pfu/ml. Input solution containing bromide tracer and ΦX174 was applied with a peristaltic pump. Outflow samples were collected in 15 ml polypropylene centrifuge tubes with a fraction collector. The average moisture content in unsaturated condition during the experiment was about 0.077. The mass density of sand column was 1.75 gm/cm^3 and the average velocity

was about 1.42 cm/hr. The inactivation coefficient during unsaturated condition was 3.31 /hr. The remaining two parameters D and k_d are estimated using the inverse procedure as explained in Section 4.3.

Table 6.7 presents the initial guess value and the optimal parameter estimates of D and k_d by considering different initial guess value for ΦX174. The parameters estimated in each case are used to predict the final virus concentration. The RMS error in all the cases is also found out for each case and is shown in Table 6.7. The parameter estimates corresponding to minimum RMS error is finally considered as optimal estimates and are used to predict the virus concentration. Table 6.7 shows that for the case of ΦX174 the minimum RMS error is 0.0275 for which the value of the parameters are D=270.742 cm^2/hr and k_d = 0.1997 ml/gm respectively. Fig. 6.4 compares the observed and model predicted ΦX174 virus concentration with optimal parameter values (D=270.742 cm^2/hr and k_d = 0.1997 ml/gm). Fig 6.4 suggests that the model predictions with optimal parameter estimates match reasonably well with the observed virus concentrations.

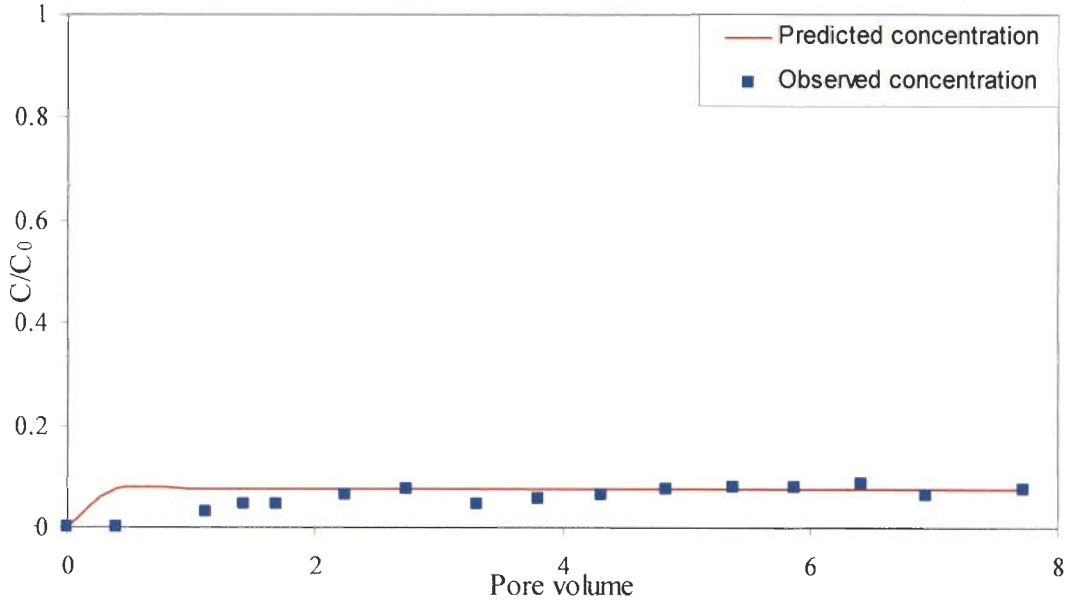

Fig 6.4 Comparison of normalized virus ΦX174 breakthrough concentration from column experiment (Jin et al., 2000)

Table 6.7: Estimation of parameters of ΦX174 from a column experiment for unsaturated condition conducted by Jin et al. (2000)

Parameters	Case A (over estimated)				Case B (under estimated)				Case C (mixed)			
	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations	Initial guess	Final estimated values	RMS error	No of iterations
D (cm^2/hr)	400	270.742	0.0275	10	40	47.144	0.0281	9	400.0	47.5356	0.028	18
k_d (ml/gm)	0.2	0.1997			0.002	0.00197			0.002	0.00235		

6.7 CONCLUDING REMARKS

In the present Chapter, the parameter estimation is formulated as a least square minimization problem in which the parameters are estimated by minimizing the deviation between the model predicted and observed virus concentrations. Levenberg-Marquadart algorithm is employed for the nonlinear optimization. The efficacy and robustness of the optimization procedure is evaluated by estimating the parameter from hypothetically generated virus concentration data. It is found that with the virus concentration data, the three transport parameters λ, λ^*, and k_d can be estimated uniquely if the number of parameters to be estimated is equal to one. If number of unknown parameters is more than two then it is impossible to estimate the transport parameters uniquely using the inverse procedure.

The present Chapter also investigates the performance of the objective function in the presence of noisy data during estimation of transport parameters. To study the effect of objective function on parameter estimation, Gaussian noise is added to synthetically generated data and detailed statistical analysis is carried out. It is found that the objective function does not induce any bias into the estimated parameters. The inverse procedure is also used to estimate the transport parameters from a column experiment involving virus movement in unsaturated zone.

CHAPTER 7
CONCLUSIONS

7.1 GENERAL

In the present study a hybrid finite volume numerical model is developed using operator split approach for solving advection-dispersion equation of solute movement in ground water. This approach uses a globally second order accurate explicit finite volume method for the advective transport and an implicit central difference method for the dispersive transport. The numerical model is used to analyse the virus transport equation. The accuracy of the model is tested for a wide range of Peclet and Courant numbers. The accuracy of the model is also tested for different types of limiters such as superbee, mimmod and van albada. The accuracy of the numerical model in predicting virus movement for both advection and dispersion dominated from continuous source of infinite and finite duration is tested. The comparison of model prediction with analytical solutions indicates that the numerical model accurately predicts virus movement in all the cases.

The present study is also concerned with the estimation of transport parameters of virus movement in ground water. The parameter estimation is formulated as a least square minimization problem in which the parameters are estimated by minimizing the deviation between the model predicted and observed virus concentrations. For this purpose, the hybrid finite volume numerical model simulating one dimensional virus transport in groundwater is coupled with Levenberg-Marquadart optimization algorithm. The efficacy and robustness of the

optimization procedure is evaluated by estimating the parameter from hypothetically generated virus concentration data in saturated zone. The present study also investigates the performance of the objective function while estimating transport parameters using inverse procedures in the presence of data errors. Gaussian noise is added to the hypothetical data generated at discrete times and at discrete distances from the source. A detailed statistical analysis is carried out to study the effect of bias induced by the objective function on the estimated parameters when the data contains the errors. The optimization algorithm is also applied to estimate the transport parameters from the virus concentration data of two column experiments involving MS2 and ΦX174 virus transport.

A numerical model has also been developed to analyze the virus transport in unsaturated zone. The model couples a mass conservative fully implicit finite difference model simulating moisture flow in the unsaturated zone with the hybrid finite volume model for virus transport. The applicability of the model to analyse virus transport in unsaturated zone is demonstrated with an example. Further the virus transport parameters are estimated from hypothetically generated virus concentration data in unsaturated zone. The performance of the objective function while estimating transport parameters in unsaturated zone using inverse procedures in the presence of data errors is also studied. A detailed statistical analysis is carried out to study the effect of bias induced by the objective function on the estimated parameters in unsaturated zone when the data contains the errors. Finally, the optimization algorithm is also applied to estimate the transport parameters from the virus concentration data of a column experiments involving ΦX174 virus transport. The following conclusions are drawn from the study.

7.2 CONCLUSIONS

1. The numerical model is capable of simulating virus transport under advection dominated and dispersion dominated situations.

2. The present model simulates the transport of virus very well for a wide range of Peclet numbers. The effect of Courant numbers on virus transport is found to be insignificant.

3. Among the limiters, the Suprebee limiter is least dissipative, while Minmod limiter is most dissipative among the three limiters tested in the study. It is also found that the numerical model predictions are quite accurate for the virus injected for infinite as well as finite duration.

4. It is found that with the virus concentration data in saturated zone, it is not possible to estimate the four transport parameters D, λ, λ^*, and k_d uniquely while estimating the transport parameters. If the number of parameters to be estimated is less than or equal to three, the inverse procedure uniquely estimates the unknown parameters. In the cases of estimation of three or two parameters, if the parameters to be estimated involve the combination of λ and λ^*, the optimization does not yield unique estimates. The analysis of the convexity of the objective function in λ-λ^* parametric space shows the presence of local minima which result in the nonunique estimation of the parameters λ and λ^*. So apriori estimation of the inactivation coefficient in sorption phase is necessary for unique estimation of other unknown parameters. In all

the cases, starting the initial guess parameter as overestimated results in least number of iteration to converge to the optimal solution.

5. In case of estimating one and two unknown virus transport parameter in saturated zone, it is concluded that in the absence of noise, parameters estimated coincide with the true values but the noisy data induces errors in the estimated parameters. The amount of induced errors increases with an increase in the noise level. The true value of the parameter lies with in 95% confidence interval of the identified parameters. The deviation of the mean from the true value increases with an increase in noise level in case of parameters λ and D. However, the mean remains significantly unchanged and is closer to the true value in case of λ^* and k_d. The objective function does not induce any bias in the estimation of one unknown parameters

6. In case of estimating the three unknown virus transport parameter in saturated zone, the study shows that in the absence of noise, parameters estimated coincide with the true values but the noisy data induces errors in the estimated parameters. The true value of the parameter does not lie with in 95% confidence interval of the identified parameters. The error in the average value of the estimated parameters increases with the increase in the noise level. The objective function induces bias that increase with increase in noise levels.

7. The model predicted virus concentrations with optimal parameter estimates match reasonably well with experimental data of both MS2 and ΦX174 virus transport in saturated zone.

8. The numerical model is also capable of simulating the transport of virus in unsaturated zone. It is found that with the virus concentration data in unsaturated zone, it is not possible to simultaneously estimate all the three transport parameters λ, λ^*, and k_d uniquely. If the number of parameters to be estimated is less than or equal to two, the inverse procedure uniquely estimates the unknown parameters. The results also show that, in the cases of estimation two parameters, if the parameters to be estimated involve the combination of λ and λ^*, the optimization does not yield unique estimates. Further, it is observed that starting the initial guess parameters as underestimated leads to less number of iterations to converge to optimal solution.

9. The statistical analysis in terms of means and confidence intervals while unsaturated virus concentration data containing errors shows that the deviation of the mean from the true value increases with an increase in noise level in case of parameters λ. However, the mean remains significantly unchanged and is closer to the true value in case of λ^* and k_d. The true value is contained within the 95% confidence interval of all the identified parameters at all noise levels indicating that the objective function does not induce bias on the parameter estimates, when they are estimated individuals.

10. It is also found that the model prediction with optimal parameter estimates matches reasonably well with experimental data of ΦX174 virus transport in unsaturated zone.

7.3 SCOPE FOR FUTURE WORK

The present work is mainly focused on numerical modeling of virus transport and estimating the transport parameters. There are certain issues, which are worth mentioning for future investigations.

1. In the present work, equilibrium adsorption has been considered for numerical modeling of the virus transport. The model can be developed by considering non equilibrium adsorption or kinetic sorption for virus transport.
2. Laboratory and field experiment can be conducted for better understanding of virus transport in saturated and unsaturated zones.

BIBLIOGRAPHY

1. Agarwal, G.S., Bhuptawat, H.K. and Choudhary, S. (2006) Biosorption of aqueous chromium by tamarindus indica seeds, Bioresource technology, 97(7), 949-956.

2. Al-Rabeh, A. (1993) On the computational efficiency of certain upwinding schemes. Computational Methods Applied Mechanical Engineering, 109, 131-141.

3. Anders, R and Chrysikopoulos, C.V. (2006) Evaluation of the factors controlling the time dependent inactivation rate coefficients of bacteriophage MS2 and PRD1.Environment Science Technology, 40(10), 3237-3242.

4. Bales, R.C., Hinkle, S.R., Kroeger, T.W., Stocking, K. and Gerba, C.P. (1991) Bacteriophage adsorption during transport through porous media: Chemical perturbation and reversibility. Environmental Science Technology, 25, 2088-2095.

5. Banks, M.K., W. Yu. and Govindaraju, R.S. (2003) Bacterial transport in saturated soil columns. Journal of Environmental Science and Health, A38(12), 2749-2758.

6. Barth, R.G. and Hill, C.M. (2005a). Numerical methods for improving sensitivity analysis and parameter estimation of virus transport simulated using sorptive-reactive process. Journal of Contaminant hydrology, 76, 251-277.

7. Barth, R.G. and Hill, C.M. (2005b). Parameter and observation importance in modeling virus transport in saturated porous media- investigation in a homogeneous system. Journal of Contaminant Hydrology, 80, 107-129.

8. Bear, J. (1979) Hydraulics of ground water, McGraw-Hill.

9. Berg, G. (1977) Viruses in the environment: Criteria for the risk, in: B.P.Sagik and C.A. Sorber (eds.), Risk assessment and health effects of municipal wastewater and sludges, proceedings, Univ. Texas, San Antonio, 216-229.

10. Bitton, G., Farrah, S.R., Ruskin, R.H., Butner, J. and Chou, Y.J. (1983). Survival of pathogenic and indicator organisms in ground water. Ground Water, 21, 405-410.

11. Blanc, R. and Nasser, A. (1996), Effect of effluent quality and temperature on the persistence of viruses in soil, Water Science Technology, 33, 237-242.

12. Brock, T.D. and Madigan, M.T. (1991) Biology of microorganisms, 6th edn, Prentice-Hall, Englewood Cliffs, 874.

13. Brooks, R.H. and Corey, A.T. (1964) Hydraulic properties of porous media, Hydrology paper No 3, Civil Engineering Department, Colorado State University, Fort Collins, Colorado.

14. Brusseau, M.L., and Rao, P.S.C. (1989) Sorption nonideality during organic contaminant transport in porous media. CRC Crit. Rev. Environ. Control., 19, 33-99.

15. Brusseau, M.L., Sandrin, S.K., Li, L., Yolcubal, I, Jordan, F.L. and Maier, R.M. (2006) Biodégradation during contaminant transport in porous media :8. The influence of microbial system variability on transport behavior and parameter determination. Water Resources Research, 42.

16. Burdine, N.T. (1953) Relative permeability calculations from size distribution data. Trans., ASME, 198, 71-78.

17. Campbell, G.S. (1974) A simple method for determining unsaturated conductivity from moisture retention data. Soil Science, 117, 311-314.

18. Carrera, J. and Neumann, S.P. (1986), Estimation of aquifer parameters under transient and steady state conditions, uniqueness, stability and solution algorithms, Water Resources Research, 22, 211-227.

19. Celia, M.A., Bouloutas, E.T. and Zarba, R.L. (1990) A general mass conservative numerical solution for the unsaturated flow equation. Water Resources Research, 26, 1483-1496.

20. Celia, M.A., Kindred, S.J. and Herrera, I. (1989) Contaminant transport and biodegradation: 1. a numerical model for reactive transport in porous media. Water Resources Research, 25(6), 1141-1148.

21. Chattopadhyay, D., Chattopadhyay, S., Lyon, W.G. and Wilson, J.T. (2002). Effect of surfactants on the survival and sorption of viruses. Environment Science Technology, 36(19), 4017-4024.

22. Cheng, Hwai-Ping., Li, Ming-Hsu., Cheng, Jing-Ru. (2003). An assessment of using the predictor-corrector technique to solve reactive transport equations. International Journal For Numerical Methods in Engineering, 56, 739-766.

23. Childs, E.C. and Collis-George, N. (1950) The permeability of porous materials. Proc., Roy. Soc. Ser.A., 201, 392-405.

24. Chrysikopoulos, C.V. and Sim, Y. (1996) One dimensional virus transport in homogeneous porous media with time-dependent distribution coefficient, Journal of Hydrology, 185, 199-219.

25. Chu, Y., Jin, Y., Baumann, T. and Yates, V.M. (2003). Effect of soil properties on saturated and unsaturated virus transport through columns. J.Environmental Quality, 32, 2017-2025.

26. Chu, Y., Jin, Y., Flury, M. and Yates, V.M. (2001). Mechanisms of virus removal during transport in unsaturated porous media. Water Resources Research, 37(2), 253-263.

27. Clement, T.P., Hooker, B.S. and Skeen, R.S. (1996) Numerical modeling of biologically reactive transport near nutrient injection well. Journal of Environmental Engineering, ASCE, 122(9), 833-839.

28. Clement, T.P., Wise, W.R. and Molz, F.J. (1994). A physically based, two-dimensional, finite-difference algorithm for modeling variably saturated flow. Journal of Hydrology, 164, 71-90.

29. Cobb, P.M., McElwee, C.D. and Butt, M.A. (1982) Analysis of leaky aquifer pumping test data: An automated numerical solution using sensitivity analysis. Ground Water, 20, 325-333.

30. Cooley, R.L. (1971) A finite difference method for unsteady flow in a variably saturated porous media: application to a single pumping well. Water Resources Research, 19(5), 1271-1285.

31. Cooley, R.L. (1983) Some new procedures for numerical solution of variably saturated flow problems. Water Resources Research, 19(5), 1271-1285.

32. Corapcioglu, M.Y. and Haridas, A. (1984). Transport and fate of microorganisms in porous media: A theoretical investigation. Journal of hydrology, 72, 149-169.

33. Corapcioglu, M.Y., Vogel, J.R., Munster, C.L., Pillai, S.D., Dowd, S. and Wang, S. (2006) Virus transport experiment in a sandy aquifer. Water, Air and Soil Pollution, 169, 47-65.

34. Cox, R.A., Nishikawa, T., (1991). A new total variation diminishing scheme for the solution of advective-dominant solute transport. Water Resources Research, 27(10), 2645-2654.

35. Dane, J.H. and Hruska, S. (1983) In-situ determination of soil hydraulic properties during drainage. Soil Science Soc. Am. J., 47, 619-624.

36. Dhiman, S.D. and Keshari, A.K. (2003) Quantifying uncertainties using Fuzzy logic for groundwater driven contaminant exposure assessment, Proceedings of the Probabilistic Approaches and Groundwater Modeling symposium held in Philadelphia, Pennsylvania. 236-247.

37. Diskin, M.H. and Simon, E. (1977) A procedure for the selection of objective functions for hydraulic simulation models. Journal of Hydrology, 34, 129-149.

38. Dogan, A. and Motz, L.H. (2005). Saturated-Unsaturated 3D groundwater model. I: Development. Journal of Hydrologic Engineering, 10(6), 492-504.

39. Dogan, A. and Motz, L.H. (2005). Saturated-Unsaturated 3D groundwater model. II: Verification and Application. Journal of Hydrologic Engineering, 10(6), 505-515.

40. Elimelech, M., Gregory, J., Jia, X. and Williams, R.A. (1995) Particle deposition and aggregation: Measurement, Modeling and Simulation, Butterworth-Heinemann, Oxford, Great Britain.

41. Feddes, R.A., Kabat, P., Bakel, P.J.T., Bronswijk, J.J.B. and Halbertsma, J. (1988) Modeling soil water dynamics in the unsaturated zone-state of the art. Journal of Hydrology, 100, 69-111.

42. Feddes, R.A., Kotwalik, P.J. and Zaradny, H. (1978) Simulation of field water use and crop yield. Center for Agricultural Publishing and Documentation, Wageningen, The Netherlands.

43. Freeze, R.A. and Chery, J.A.,(1979). Ground water. Prentice Hall,Inc, Englewood Cliffs, New Jersey.

44. Gallo, C. and Manzini, G. (2003) Finite volume/ Mixed finite element analysis of pollutant transport and bioremediation in heterogeneous saturated aquifers, International Journal for Numerical Methods in Fluids, 42, 1-21.

45. Gerba, C.P. (1984) Applied and theoretical aspects of virus adsorption to surfaces. Advances Applied microbiology, 30, 133-168.

46. Gerba, C.P. and Keswick, B.H. (1981) Survival and transport of enteric bacteria and viruses in groundwater, Stud. Environmental Science, 17, 511-515.

47. Ghidaoui, M.S. and Hariprasad, K.S.(2000) A priori identifiability of unsaturated soil parameters. Journal of Irrigation and drainage Engineering, ASCE, 126(3), 163-171.

48. Govindraju, R.S., Kavvas, M.L., Rolston, D.E. and Biggar, J. (1992) Error analysis of simplified unsaturated flow models under large uncertainty in hydraulic properties. Water Resources Research, 28 (11), 2913-2924.

49. Hancock, C.M., Rose, J.B. and Callahan, M. (1998) Crypto and Giardia in US groundwater, J. AWWA., 90, 58-61.

50. Hari Prasad, K.S., Mohan Kumar, M.S., Sekhar, M. and Chandrasekhar, B. (2005) A simple numerical model for assessment of groundwater recharge. Journal of IWRS, 25(1), 49-60.

51. Harten, A. (1983) High resolution schemes for hyperbolic laws. Journal of Computational Physics, 49, 357-393.

52. Hills, R.G., Porro, I., Hudson, D.B. and Wierenga, J.P. (1989) Modeling one dimensional infiltration into very dry soils: model development and evaluation. Water Resources Research, 25(6), 1259-1269.

53. Hiscock, K.M. and Grischek, T. (2002) Attenuation of groundwater pollution by bank filtration. Journal of Hydrology, 266, 139-144.

54. Hossain, A., (1999). Modeling advective-dispersive transport with reaction: An accurate explicit finite difference model. Applied Mathematics and Computation, 102, 101-108.

55. Huang, K., Mohanty, B.P., Leij, F.K. and Van Genuchten, M.Th. (1998) Solution of the nonlinear transport equation using modified Picard iteration. Advances in Water resources, 21, 237-249.

56. Hurst, C.J., Gerba, C.P. and Cech, I., (1980) Effects of environmental variables and soil characteristics on virus survival in soil. Applied Environmental Microbiology, 40, 1067-1079.

57. Hurst, C.J., Gerba, C.P., Lance, J.C. and Rice, R.C. (1980) Survival of enteroviruses in rapid-infiltration basins during the land application of waste water, Applied Environmental Microbiology, 40, 192-200.

58. Huyakorn, P.S., Springer, E.P., Guvansen, V. and Woodsworth, T.D. (1986) A three dimensional finite element model for simulating water flow in variably saturated porous media. Water Resources Research, 22(13), 1790-1808.

59. Huyakorn, P.S., Thomas, S.D. and Thompson, B.M. (1984) Techniques for making finite elements competitive in modeling flow in a variably saturated media. Water Resources Research, 20, 1099-1115.

60. Islam, J. Singhal, N. and O'Sullivan, M. (2001) Modeling biogeochemical processes in leachate-contaminated soils: A review. Transport in Porous Media, 43, 407-440.

61. Jansons, J., Edmonds, L.W., Speight, B. and Bucens, M.R. (1989) Survival of viruses in ground water. Water Research, 23, 301-306.

62. Janz, T.C. and Stonier, R.J. (1995) Modeling water flow in cropped soils: Water uptake by plant roots. Environment International, 21, 705-709.

63. Jin Y., Pratt, E. and Yates, M.V. (2000) Effect of Mineral colloids on virus transport through sand columns. Journal of Environmental Quality, 29(2), 532-539.

64. Jin, Y., Yates, M.V., Thompson, S.S. and Jury, W.A. (1997) Sorption of viruses during flow through saturated sand column. Environment Science Technology, 31(2), 548-555.

65. Jin. Y., Chu. Y. and Li, Y. (2000). Virus removal and transport in saturated and unsaturated sand columns. Journal of Contaminant Hydrology, 43, 111-128.

66. John, D.E. and Rose, J.B. (2005) Review of factors affecting microbial survival in ground water. Environment Science Technology, 39(10), 7345-7356.

67. Jyothish, S. (1999). A hybrid finite volume model for reactive solute transport in ground water. A PhD thesis, Department of Civil Engineering, Indian Institute of Science, Bangalore.

68. Keswick, B.H. and Gerba, C.P. (1980) Viruses in ground water, Environmental science and technology, 14(11), 1290-1297.

69. Khatibi, R.H., Williams, J.J.R. and Wormleaton, P.R. (1997) Identification problem of open channel friction parameters, Journal of Hydraulic Engineering, ASCE, 123 (12), 1078-1088.

70. Kim, S.B. and Kim, D.J., (2003). Application of generalized contaminant retardation factor to a multi-phase system. Hydrological processes, 17, 3059-3068.

71. Kim, S.B., (2005). Contaminant transport and biodegradation in saturated porous media: model development and simulation. Hydrologic processes, 19, 4069-4079.

72. Kim, S.B., (2006). Numerical analysis of bacterial transport in saturated porous media. Hydrological Processes, 20, 1177-1186.

73. Kindred, S.J. and Celia, M.A. (1989) Contaminant transport and biodegradation: 2. Conceptual model and test simulations. Water Resources Research, 25(6), 1149-1159.

74. Kinoshita, T., Bales, R.C., Yahya, M.T. and Gerba, C.P. (1993) Effect of pH on bacteriophage transport through sandy soils. Journal of Contaminant Hydrology, 14, 1197-1202.

75. Kirkland, M.R., Hills, R.G. and Wierenga, P.J. (1992) Algorithms for solving equations for variably saturated soils. Water Resources Research, 28(8), 2049-2058.

76. Kool, J.B. and Parker, J.C. (1988) Analysis of the inverse problem for transient unsaturated flow. Water Resources Research, 24(6), 817-830.

77. Kool, J.B., Parker, J.C. and Van Genuchten, M.Th. (1987). Parameter estimation for unsaturated flow and transport models- A review. Journal of Hydrology, 91, 255-293.

78. Kubrusly, C.S.(1977). Distributed parameter system identification, a survey. Intr. J. Contr., 26(4), 509-535.

79. Lance, J.C. and Gerba, C.P. (1984). Virus movement in soil during saturated and unsaturated flow. Applied Environmental Microbiology, 47(2), 335-337.

80. Malleen, G., Meloszewski, P., Flynn, R. , Rossi, P., Engel, M. and Seiler, P.K. (2005). Determination of bacterial and viral transport parameters in a gravel aquifer assuming linear kinetic sorption and desorption. Journal of Hydrology, 306, 21-36.

81. Man, C. and Tsai, C.W., (2007). A higher-order predictor-corrector scheme for two-dimensional advection-diffusion equation. International Journal for Numerical Methods in Engineering, in press.

82. Maraqa, M.A. (2007) Retardation of nonlinearly sorbed solutes in porous media. Journal of Environmental Engineering, ASCE, 133 (6), 587-594.

83. Matthess, G., Pekdeger, A. and Schroeter, J. (1988) Persistance and transport of bacteria and viruses in ground water: a conceptual evaluation. Journal of Contaminant Hydrology. 2, 171-188.

84. Miller, D.W., (1980). Waste Disposal effects on ground water. Premier Presss. Berkeley, California

85. Mojid, M.A. and Vereecken, H., (2005). On the physical meaning of retardation factor and velocity of a nonlinearly sorbing solute. Journal of Hydrology, 302, 127-136.

86. More, J.J. (1977) The Levenberg-Marquardt algorithm: implementation and theory. In: G.A. Watson (Editor), Lecture notes in Mathematics 630, Springer, New York, N.Y., 105-116.

87. Mualem,Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. Water Resources Research, 12(3), 513-522.

88. Narsimhan, T.N., Neumann, S.P. and Witherspoon, P.A. (1978) Finite element method for subsurface hydrology using a mixed explicit-implicit scheme. Water Resources Research, 14(5), 863-877.

89. Nasser, A., Tchorch, Y., and Fattal, B. (1993). Comparative survival of E.coli, F+bacteriophages, HAV and poliovirus1 in waste water and ground water, Water Science Technology, 27, 401-407.

90. Newman, S.P. (1973) Calibration of distributed parameter groundwater flow models viewed as a multiple objective decision process under uncertainty, Water Resources Research, 9(4), 1006-1021

91. O'Loughlin, E.M., and Bowmer, K.H.(1975). Dilution and decay of aquatic herbicides in flowing channels. Journal of Hydrology, 26, 217-235.

92. Ogata, A., and Banks, R.B.(1961). A solution of the differential equation of longitudinal dispersion in porous media. Prof. paper No. 411-a, U.S. Geological Survey, Washington, D.C.

93. Osborne, M.R.(1976) Nonlinear least squares- The Levenberg algorithm revisited, J. Aust. Math. Soc., Ser.B, 19, 343-357.

94. Paniconi, C., Aldama, A.A. and Wood, E.F. (1991) Numerical evaluation of iterative and numerical methods for the solution of non-linear Richards equation. Water Resources Research, 27, 1147-1163

95. Park, N., Blanford, T.N. and Huyakorn, P.S. (1992) VIRALT: A modular semi analytical and numerical model for simulating viral transport in groundwater, Int. Groundwater Model. Cent., Colo. Sch. of Mines, Golden, 1992.

96. Pekdeger, A. and Matthess, G. (1983). Factors of bacteria and virus transport in ground water. Environmental Geology, 5(2), 49-52.

97. Persson, M. and Berndtsson, R. (1998) Estimating transport parameters in an undisturbed soil column using time domain reflectometry and transfer function theory. Journal of Hydrology, 205, 232-247.

98. Powelson, D.P., Simpson, J.R. and Gerba, C.P. (1990) Virus transport and survival in saturated and unsaturated flow through soil columns. Journal of Environmental Quality, 19, 396-401.

99. Powelson, D.P., Simpson, J.R. and Gerba, C.P. (1991). Effect of organic matter on virus transport in unsaturated flow. Applied and Environmental Microbiology, 57(8), 2192-2196.

100. Putti, M., Yeh, G.W. and Mulder, A.W. (1990). A triangular finite volume approach with high-resolution upwind terms for the solution of ground water transport equations. Water Resources Research, 26(12), 2865-2880.

101. Rao, P., Miguel, A., and Medina, Jr., (2005). A multiple domain algorithm for modeling one-dimensional transient contaminant transport flows. Applied Mathematics and Computation, 167, 1-15.

102. Rathfelder, K. and Abriola, L.M. (1994) Mass conservative numerical solutions of the head based Richards equation. Water Resources Research, 30(9), 2579-2586.

103. Remson, I., Hornberger, G.M. and Molz, F.Z. (1971) Numerical methods in subsurface hydrology. Willey-International, New York, 389.

104. Richards, L.A. (1931) Capillary conduction of liquids through porous medium. Physics, 1, 318-333.

105. Rifai, H.S. and Bedient, P.B. (1990) Comparison of biodegradation kinetics with an instantaneous reaction model for groundwater. Water Resources Research, 26(4), 637-645.

106. Rockhold, M.L., Yarwood, R.R. and Selker, J.S. (2004) Coupled microbial and transport process in soils. Vadose Zone Journal, 3, 368-383.

107. Rose, D.A. (1977). Dilution and decay of aquatic herbicides in flowing channels- comments. Journal of Hydrology, 32, 399-400.

108. Runkel, L.R. (1996). Solution of the advection-dispersion equation: Continuous load of finite duration. Journal of Environmental Engineering, ASCE, 122(9), 830-832.

109. Runkel, R.L. and Bencala, K.E. (1995) Transport of reacting solutes in rivers and streams, Environment Hydrology, V.P. Singh, ed., Kluwer, Dordrecht, The Netherlands.

110. Russo, D, Bresler, E, Shani, U, and Parker, J.C. (1991) Analysis of infiltration events in relation to determining soil hydraulic properties by inverse problem methodology, Water Resources Research, 27(6), 1361-1373.

111. Sardin, M., Schweich, D., Leij, F.J. and Van Genuchten, M.Th. (1991) Modeling the nonequilibrium transport of linearly interacting solutes in porous media : a review. Water Resources Research, 27, 2287-2307.

112. Sato, T., Tanahashi, H. and Loaiciga, H.A. (2003). Solute dispersion in variably saturated sand. Water Resources Research, 39(6), 1155.

113. Schijven, J.F. and Simunek, J. (2002). Kinetic modeling of virus transport at field scale. J. of Contaminant Hydrology, 55, 113-135.

114. Schijven, J.F., (2001). Virus removal from ground water by soil passage modeling, field, and laboratory experiments. PhD thesis.

115. Sim, Y. and Chrysikopoulos, C.V. (1996). One dimensional virus transport in porous media with time-dependent inactivation rate coefficients. Water Resources Research, 32(8), 2607-2611.

116. Sim, Y. and Chrysikopoulos, C.V. (2000). Virus transport in unsaturated porous media, Water Resources Research, 36(1), 173-179.

117. Sim, Y., and C.V. Chrysikopoulos (1995) Analytical models for one dimensional virus transport in saturated porous media. Water Resources Research, 31(5), 1429-1437.

118. Sim, Y., and C.V. Chrysikopoulos (1998) Three dimensional analytical models for virus transport in saturated porous media. Transport in Porous Media, 30, 87-112.

119. Singh, V. and Murty, B. (1996). A complete hydrodynamic border strip irrigation model. Journal of Irrigation and Drainage Engineering, 122(4), 189-197.

120. Sobsey, M.D., Dean, C.H., Knuckles, M.E. and Wagner, R.A. (1980) Interaction and survival of enteric viruses in soil materials. Applied Environmental Microbiology, 40, 92-101.

121. Sobsey, M.D., Shields, P.A., Hauchmann, F.H., Hazard, R.L. and Caton, L.W. (1986). Survival and transport of hepatitis A virus in soils, groundwater, and waste water. Water Science Technology, 18, 97-106.

122. Suresh Kumar, G. and Sekhar, M. (2005) Spatial moment analysis for transport of nonreactive solutes in fracture matrix system. Journal of Hydrologic Engg., ASCE, 10(3), 192-199.

123. Tai, C.H., Chiang, D.C., and Su, Y.P.(1997). Explicit time marching methods for the time dependent Euler computations. Journal of Computational Physics, 130, 191-202.

124. Taylor, D.H. and Bossman, H.B. (1981) The electrokinetic properties of reovirus type 3: Electrophoretic mobility and zeta potential in dilute electrolytes, Journal of Colloid Interface Science, 83, 153-162.

125. Tim, S.U. and Mostaghimi, S. (1991). Model for predicting virus movement through soils, Ground Water. 29(2), 251-259.

126. Traub, F, Spillmann, S.K. and Wyler, R. (1986). Method for determining virus inactivation during sludge treatment processes. Applied Environmental Microbiology, 52(3), 498-503.

127. Van Albada, G.D., and Van Leer, B., and Robers jr., W.W., (1982). A comparative study of computational methods in cosmic gas dynamics, Astronomy and Astrophysics, 108, 76-86.

128. Van Genuchten, M.Th. (1980) A closed form equation for predicting the hydraulic conductivity of unsaturated soil. Soil Science Society of American Journal, 44, 892-898.

129. Van Genuchten, M.Th. (1981). Analytical solutions for chemical transport with simultaneous adsorption, zero order production and first order decay. Journal of Hydrology, 49, 213-233.

130. Van Genuchten, M.Th. and Alves, W.J. (1982) Analytical solution of the one dimensional convective-dispersive solute transport equation. USDA/ARS Technical Bulletin. 1661.

131. Van Leer, B., (1974). Towards the ultimate conservative difference scheme. II: Monotonicity and conservation combined in a second order scheme. Journal of Computational Physics, 14: 361-375.

132. Van Leer, B., (1977a). Towards the ultimate conservative difference scheme. III: Upstream centered finite difference schemes for ideal compressive flow. Journal of Computational Physics, 23: 263-275.

133. Van Leer, B., (1977b). Towards the ultimate conservative difference scheme. IV: A new approach to numerical convection. Journal of Computational Physics, 23: 276-299.

134. Van Leer, B., (1984). On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher, and Roe. SIAM journal of Scientific and Statistical Computing, 5(1), 1-20.

135. Verma, A.K., Murty, B.S. and Eswaran, V. (2000). Overlapping control volume method for solute transport. Journal of Hydrologic Engineering, ASCE, 5(3), 308-316.

136. Wheeler, M.F. and Dawson, C.N. (1987) An operator splitting method for advection-diffusion-reaction problems. Technical Report 87-9, Department of Mathematical Science, Rice University, Houston, Texas.

137. Williams, W.H. (1978) How bad can good data really be? The American Statistician, 32(2), 61-65.

138. Wimpenny, J.W.T., Cotton, N. and Statham, M. (1972) Microbes as tracers of water movement. Water research, 6, 731-739.

139. Yahya, M.T., Galsomies, L., Gerba, C.P. and Bales, R.C. (1993). Survival of bacteriophages MS2 and PRD1 in ground water., Water Science Technology, 27, 409-412.

140. Yates, M.V. and Ouyang, Y. (1992). Virtus, A model of virus transport in unsaturated soils, Applied and Environmental Microbiology, 58(5), 1609-1616.

141. Yates, M.V. and Yates, S.R. (1988) Modeling microbial fate in the subsurface environment, Crit. Rev. Environment Control, 17(4), 307-344.

142. Yates, M.V., Gerba, C.P. and Kelley, L.M. (1985). Virus persistence in ground water, Applied and Environmental Microbiology, 49(4), 778-781.

143. Yates, M.V., Yates, S.R., Wagner, J. and Gerba, C.P. (1987). Modeling virus survival and transport in the subsurface. Journal of Contaminant Hydrology, 1, 329-345.

144. Yeh, G.T. (1990) A Langrangian-Eulerian method with zoomable hidden fine-mess approach to solving advection-dispersion equations. Water Resources Research, 26(6), 1133-1144.

145. Yeh, G.T. and Tripathi, V.S. (1989) A critical evaluation of recent developments of hydrogeochemical transport model of reactive multichemical components. Water Resources Research, 25(1), 93-108.

146. Yeh, W. (1986) Review of parameter identification procedures in groundwater hydrology; The inverse problem. Water Resources Research, 22(2), 95-108.

147. Young, D.L., Wang, Y.F. and Eldho, T.I. (2000) Solution of advection-diffusion equation using the Eulerian-Langrangian boundary element method, Engineering Analysis with Boundary Elements. 24(6), 449-457.

148. Zhuang, J. and Jin, Y. (2003). Virus retention and transport as influenced by different forms of soil organic matter. Journal of Environmental Quality, 32, 816-823.

LIST OF PUBLICATIONS

Ratha, D.N., Hari Prasad, K.S. and Ojha, C.S.P. (2007) A finite volume model for the solution of the advection-dispersion equation., ISH Journal of Hydraulic Engineering, Vol 13, No 2, pp. 122-135.

Ratha, D.N., Hari prasad, K.S. and Ojha, C.S.P. Analysis of virus transport in ground water and identification of transport parameters, Communicated to Practice Periodical of Hazardous, Toxic and Radioactive Waste Management, ASCE.

APPENDIX –I

NUMERICAL MODEL FOR SIMULATING VIRUS TRANSPORT IN SATURATED ZONE

```
      ! l   =Total Length
      ! Nnode=no of nodes
      ! v   =velocity
      ! Cou  =Courant number
      ! Pe   =Peclet Number
      ! delx =Grid length
      ! d    =Dispersion coefficient

      parameter(Nnode=101)
      real x(Nnode),cold(Nnode),cpre(Nnode),cadv(Nnode),delc(Nnode),ai(Nnode),bi(Nnode)
      real ae(Nnode),be(Nnode),ce(Nnode),re(Nnode),alpha(Nnode),bita(Nnode),cte(Nnode)
      real y(Nnode),c(Nnode),ri(Nnode),cnew(Nnode)
      real s1(Nnode),s2(Nnode),s3(Nnode),s4(Nnode),cana(Nnode),y1(Nnode),y2(Nnode)
      real l,Cou,Pe,delx,delt,tmax,v,time,d,fl1,fl2,erfc,lamda,k,eps,max,t0,row,kd,theta

      open(5,file="dn.in")
      open(1,file="ana.out")
      open(2,file="ana1.out")
      open(3,file="ana2.xls")
      open(6,file="dn1.out")
      open(7,file="dn2.out")
      open(8,file="dn3.xls")
      open(9,file="dn4.xls")
      open(10,file="dn5.xls")
      open(11,file="final1.xls")
      open(12,file="break.xls")
      read (5,*) l,v,tmax,Cou,Pe,k,eps,c0,coef1,coef2,t0,row,kd,theta,limitercode
      delx=l/(Nnode-1)
      delt=Cou*delx/v
      d=v*delx/Pe
      print*,d
      x(1)=0.0
  do j=2,Nnode
      x(j)=x(j-1)+delx
      enddo
!**********************ANALYTICAL SOLUTION BEGINS*****************************
!     time=delt
!     do while(time.le.tmax)
!      cana(1)=1.0
!            do j=2,(Nnode-1)
!            s1(j)=x(j)-v*time
!            s2(j)=x(j)+v*time
!            s3(j)=x(j)*v/d
!            s4(j)=2*sqrt(d*time)
!            enddo
!            write(1,*)(s1(j),s2(j),s3(j),s4(j),j=2,Nnode-1)
!            do j=2,(Nnode-1)
!            y1(j)=(s1(j)/s4(j))
```

```
!                   y2(j)=(s2(j)/s4(j))
!                   enddo
!                   write(2,*)(y1(j),y2(j),j=2,Nnode-1)
!
!                   do j=2,Nnode-1
!                           if(s3(j).ge.500) then
!                           cana(j)=0.5*c0*(erfc(y1(j)))
!                           else
!                           cana(j)=0.5*c0*((erfc(y1(j)))+exp(s3(j))*erfc(y2(j)))
!                           endif
!                   enddo
!                   write(3,100)time,cana(100)
!                   100 format(1x,f15.5,f9.5)
!                   time=time+delt
!           enddo

!************************ANALYTICAL SOLUTION ENDS**************************


!************************NUMERICAL SOLUTION BEGINS**************************


        r=1+(row*kd/theta)

        do  j=2,(Nnode-1)
        cold(j)=0.0
    enddo
        cold(1)=c0
        time=delt
        do while (time .le. tmax)
                if(time .le. t0)then
                cold(1)=c0
                else
                cold(1)=0
                endif
        delc(1)=0.0
                do j=2,(Nnode-1)
                ai(j)=cold(j)-cold(j-1)
                bi(j)=cold(j+1)-cold(j)
                        if (ai(j) .ge. 0.0)then
                        sign=1
                        else
                        sign=-1
                        endif
                        if(limitercode .eq.1)then
                                if ((ai(j)*bi(j)) .gt.0.0)then
                                        if (abs (ai(j)).gt. abs(bi(j))) then
                                        delc(j)=sign*abs(bi(j))
                                        else
                                        delc(j)=sign*abs(ai(j))
                                        endif
                                else
                                delc(j)=0.0
                                endif
                        else if(limitercode .eq. 2)then
                                if ((ai(j)*bi(j)) .gt.0.0)then
                                        if (abs (ai(j)).ge. abs(bi(j))) then
```

```
                                    max=abs(ai(j))
                                    else
                                    max=abs(bi(j))
                                    endif
                                    if(2*abs(ai(j)) .le. 2*abs(bi(j)))then
                                            if(2*abs(ai(j)) .le. max)then
                                            delc(j)=sign*2*abs(ai(j))
                                            else
                                            delc(j)=sign*max
                                            endif
                                    else
                                            if(2*abs(bi(j)) .le. max)then
                                            delc(j)=sign*2*abs(bi(j))
                                            else
                                            delc(j)=sign*max
                                            endif
                                    endif
                            else
                            delc(j)=0
                            endif
                    else if(limitercode .eq. 3)then
                    ri(j)=((2*ai(j)*bi(j))+eps)/(ai(j)**2+bi(j)**2+eps)
                    delc(j)=ri(j)*((0.5*(1-k)*ai(j))+(0.5*(1+k)*bi(j)))
                    endif
            cpre(1)=cold(1)
            cpre(j)=cold(j)-0.5*(delt/delx)*v*delc(j)
            fl1=cpre(j)+0.5*delc(j)
            fl2=cpre(j-1)+0.5*delc(j-1)
            cadv(j)=cold(j)-(1/r)*(delt/delx)*v*(fl1-fl2)
            enddo
            cadv(1)=cold(1)
            write(8,10)(cadv(j),j=1,Nnode)
10          format(1x,f9.6)

lamda=(d*delt/(r*delx**2))

            do j=1,Nnode-1
                    if (j.eq.1)  then
                    ae(j)=0
                    be(j)=1
                    ce(j)=0
                    re(j)=cadv(1)
                    else
                    ae(j)=    -lamda
                    be(j)=     1+(2*lamda)
                    ce(j)=    -lamda
                    re(j)=cadv(j)
                    endif
            enddo
            ae(Nnode)=        0
            be(Nnode)=         1
            ce(Nnode)=0.0
            re(Nnode)=cadv(Nnode)

            alpha(1)=be(1)
            bita(1)=ce(1)/alpha(1)
            y(1)=re(1)/alpha(1)
```

```
                do j=2,(Nnode)
                alpha(j)=be(j)-(ae(j)*bita(j-1))
                bita(j)=ce(j)/alpha(j)
                y(j)=((re(j)-ae(j)*y(j-1))/alpha(j))
                enddo

                cte(Nnode)=y(Nnode)
                do j=(Nnode-1),1,-1
                cte(j)=y(j)-(bita(j)*cte(j+1))
                enddo
                do j=1,(Nnode)
                c(j)=cte(j)
                enddo

                write(9,20)(x(j),c(j),j=1,Nnode)
        20  format(1x,f4.0,f9.6)
                coef=(coef1+(coef2*row*kd/theta))
                do j=1,Nnode
                cnew(j)=c(j)-(coef*c(j)*delt/r)
                enddo
                do j=1,Nnode
                cold(j)=cnew(j)
                enddo
                write(12,50)time,cnew(21)
                50 format(1x,f15.5,f10.5)
                time=time+delt
        enddo
        write(11,200)(x(j),cnew(j),j=1,Nnode)
        200 format (1x,f10.5,f20.15)
!200 format(E15.8)
        stop
        end
!**********************NUMERICAL SOLUTION ENDS*******************************

        function erfc(y1)
        real z,t
        z=abs (y1)
        t=1.0/(1.0+0.5*z)
        erfc=t*exp(-z*z-1.26551223+t*(1.00002368+t*(0.37409196+t*(0.09678418+t*(-
0.18628806+t*(0.27886907+t*(-1.13520398+t*(1.48851587+t*(-0.82215223+t*0.17087277)))))))))
        if(y1.lt.0.0) then
        erfc=2.0-erfc
        endif
        return
        end
```

APPENDIX- II

NUMERICAL MODEL FOR ESTIMATING VIRUS TRANSPORT PARAMETERS IN SATURATED ZONE

```
parameter(nz=101)
parameter(nm=7)
real cobs(nz,nm),z(nz),b(nm),paracode(nm),cpred(nz,nm),cori(nz,nm),timeobs(nm)
real ccal(nz,nm),nelt(nz),zobs(nz),p(nm),plast(nm),err(nz),wj(nz,nm),d(nm),dold(nm)
real qt1(nz,nz),wjnew(nz,nz),errnew(nz),qt2(nz,nz),wjfin(nz,nz),errfin(nz),delp(nm),ratio(nm)
real depth,row,kd,theta,r,v,coef1,coef2,dis
integer maxpar,maxunpar
open(5,file="dn.in")
open(6,file="cnew.xls")
open(11,file="cobs.xls")
open(12,file="cpred.xls")
open(13,file="cori.xls")
open(14,file="err.xls")
open(15,file="jac.xls")
open(16,file="householder.xls")
open(17,file="wjnew.xls")
open(18,file="given.xls")
open(19,file="wjfinal.xls")
open(20,file="error.xls")
open(21,file="check.out")

read(5,*) maxpar,maxunpar,depth,row,delt,timemax,maxtobs,maxzobs,almb,v
read(5,*) (paracode(i),i=1,maxpar)
read(5,*) (b(i),i=1,maxpar)
read(5,*) (timeobs(i),i=1,maxtobs)
read(5,*) (zobs(i),i=1,maxzobs)
read(5,*) (p(i),i=1,maxpar)
read(5,*) omega1,omega2,tol1,tol2
delz=depth/(nz-1)
z(1)=0.0
do j=2,nz
        z(j)=z(1)+(j-1)*delz
enddo
i=1
k=1
z1=0
itotal=maxzobs*maxtobs
do while(i .le. maxzobs)
        if (zobs(i) .gt. z1)then
        k=k+1
        z1=z1+delz
        else
        nelt(i)=k-1
        diff=zobs(i)-z1
        i=i+1
        endif
enddo
```

```
        do j=1,maxpar
                if(j .eq.1)then
                coef1=b(1)
                elseif(j .eq.2)then
                theta=b(2)
                elseif(j .eq.3)then
                coef2=b(3)
                elseif(j .eq.4)then
                dis=b(4)
                elseif(j .eq.5)then
                kd=b(5)
                endif
        enddo
        r=1+((row*kd)/theta)
call
transport(nz,nm,delz,r,delt,timemax,coef1,coef2,v,dis,kd,row,theta,timeobs,maxtobs,ccal,maxzobs,nelt,
zobs,cobs,objf)
                        do j=1,maxzobs
                                do k=1,maxtobs
                                cobs(j,k)=ccal(j,k)
                                enddo
                        enddo
                        write(11,100)((cobs(j,k),k=1,maxtobs),j=1,maxzobs)
                        100 format(1x,5f10.5)
iout=1
ioutconv=0
do while (ioutconv .eq. 0)
do ipar=1,(maxunpar+1)
        if (ipar .eq. 1)then
                do j=1,maxpar
                plast(j)=p(j)
                enddo
        icountpar=1
        icpar=0
        else
        ikode=0
                do while (ikode .eq. 0)
                        if (paracode(icountpar) .eq. 1)then
                        ikode=1
                        icpar=icountpar
                        icountpar=icountpar+1
                        else
                        icountpar=icountpar+1
                        endif
                enddo
        endif
        do j=1,maxpar
                if (j .eq. icpar)then
                p(j)=plast(j)+0.01*plast(j)
                else
                p(j)=plast(j)
                endif
        enddo
        do j=1,maxpar
                if(j .eq.1)then
                coef1=p(1)
                else if(j .eq.2)then
                theta=p(2)
```

```
                else if(j .eq.3)then
                coef2=p(3)
                else if(j .eq.4)then
                dis=p(4)
                else if(j .eq.5)then
                kd=p(5)
                endif
        enddo

        r=1+((row*kd)/theta)

        call
transport(nz,nm,delz,r,delt,timemax,coef1,coef2,v,dis,kd,row,theta,timeobs,maxtobs,ccal,maxzobs,nelt,
zobs,cobs,objf)
        write(21,*)objf
                        if (ipar .eq. 1)then
                        objfpre=objf
                        do j=1,maxzobs
                                do k=1,maxtobs
                                cori(j,k)=ccal(j,k)
                                enddo
                        enddo
                        write(13,200)((cori(j,k),k=1,maxtobs),j=1,maxzobs)
                        200 format(1x,5f10.5)
                        else
                        do j=1,maxzobs
                                do k=1,maxtobs
                                cpred(j,k)=ccal(j,k)
                                enddo
                        enddo
                        write(12,300)((cpred(j,k),k=1,maxtobs),j=1,maxzobs)
                        300 format(1x,5f10.5)
                        endif
        if (ipar .eq. 1)then
                do j=1,maxtobs
                ierr=(j-1)*maxzobs
                        do i=1,maxzobs
                        err(ierr+i)=cobs(i,j)-cori(i,j)
                        enddo
                enddo
        write (14,40) (err(i),i=1,itotal)
        40 format(1x,f10.5)
        else
                do j=1,maxtobs
                ierr=(j-1)*maxzobs
                        do i=1,maxzobs
                        wj(ierr+i,ipar-1)=(cpred(i,j)-cori(i,j))/(p(icpar)-plast(icpar))
                        wj(ierr+i,ipar-1)=-wj(ierr+i,ipar-1)
                        enddo
                enddo
        endif
enddo
write (15,50) ((wj(i,j),j=1,maxunpar),i=1,itotal)
50 format(1x,3f10.3)
do i=1,maxunpar
sum=0
        do j=1,itotal
        sum=sum+wj(j,i)*wj(j,i)
```

```
        enddo
d(i)=abs(sqrt(sum))
enddo
if (iout .eq. 1)then
        do i=1,maxunpar
        dold(i)=d(i)
        enddo
else
        do i=1,maxunpar
                if (dold(i) .gt. d(i))then
                d(i)=dold(i)
                dold(i)=d(i)
                else
                dold(i)=d(i)
                endif
        enddo
endif
call householder(wj,qt1,nz,nm,maxunpar,itotal)
write(16,60)((qt1(i,j),j=1,itotal),i=1,itotal)
60 format(1x, 55f15.8)
mtot=itotal+maxunpar
almbcode=1.0
do while(almbcode .eq. 1)
write(21,*)almb
do i=1,mtot
        do j=1,maxunpar
        wjnew(i,j)=0
        enddo
enddo
do i=1,mtot
errnew(i)=0
enddo
do i=1,mtot
        do j=1,maxunpar
                if (i .le. itotal)then
                        do k=1,itotal
                        wjnew(i,j)=wjnew(i,j)+qt1(i,k)*wj(k,j)
                        errnew(i)=errnew(i)+qt1(i,k)*err(k)
                        enddo
                else
                lll=i-itotal
                        if(j .eq.lll)then
                        wjnew(i,j)=almb*d(j)
                        else
                        wjnew(i,j)=0
                        endif
                errnew(i)=0
                endif
        enddo
enddo
write(17,70)((wjnew(i,j),j=1,maxunpar),i=1,mtot)
70      format(1x,f15.5)
write(17,80)(errnew(i),i=1,mtot)
80      format(1x,f10.5)

call givrot(itotal,maxunpar,nz,nm,wjnew,qt2)
write(18,90)((qt2(i,j),j=1,mtot),i=1,mtot)
90      format(1x,56f10.5)
```

```
do i=1,mtot
        do j=1,maxunpar
        wjfin(i,j)=0
        enddo
        enddo
do i=1,mtot
errfin(i)=0
enddo
do i=1,mtot
        do j=1,maxunpar
                do k=1,mtot
                wjfin(i,j)=wjfin(i,j)+qt2(i,k)*wjnew(k,j)
                errfin(i)=errfin(i)+qt2(i,k)*errnew(k)
                enddo
        enddo
enddo
write(19,150)((wjnew(i,j),j=1,maxunpar),i=1,mtot)
150 format(1x,3f15.5)
write(19,250)(errnew(i),i=1,mtot)
250 format(1x,f10.5)
ffin=0
do j=1,maxunpar
ffin=ffin+errfin(j)*errfin(j)
enddo
gfin=0
do j=maxunpar+1,mtot
gfin=gfin+errfin(j)*errfin(j)
enddo
call gauss(nz,nm,maxunpar,wjfin,errfin,delp)
print*,(delp(j),j=1,maxunpar)
icount=1
do i=1,maxpar
        if(paracode(i) .eq. 1)then
        con=delp(icount)/plast(i)
                if((con .ge. -0.2) .and. (con .le. 0.5))then
                p(i)=plast(i)+delp(icount)
                elseif (delp(icount) .lt. (-0.2))then
                delp(icount)=plast(i)*(-0.2)
                p(i)=plast(i)+delp(icount)
                elseif (delp(icount) .gt. (0.5))then
                delp(icount)=plast(i)*(0.5)
                p(i)=plast(i)+delp(icount)
                endif
        ratio(icount)=abs(delp(icount))/(plast(i)+10.0**(-20))
        icount=icount+1
        endif
enddo

do j=1,maxpar
                if(j .eq.1)then
                coef1=p(1)
                elseif(j .eq.2)then
                theta=p(2)
                elseif(j .eq.3)then
                coef2=p(3)
                elseif(j .eq.4)then
                dis=p(4)
                elseif(j .eq.5)then
```

```
                kd=p(5)
                endif
        enddo
        r=1+((row*kd)/theta)

call
transport(nz,nm,delz,r,delt,timemax,coef1,coef2,v,dis,kd,row,theta,timeobs,maxtobs,ccal,maxzobs,nelt,
zobs,cobs,objf)
write(21,*)iout,objf,objfpre

if (objf .gt. objfpre)then
almb=almb*omega1
almbcode=1
else
almbcode=0
endif
enddo
ratio1=gfin/ffin
sump1=sqrt(1.0+objfpre)
if(ratio1 .ge. 0.75)then
almb=almb*omega2
else if(ratio1 .le. 0.25)then
almb=almb*omega1
endif
if(ffin .le. (sump1*tol1))then
ioutconv=1
endif
great=ratio(1)
if(maxunpar .ge.2)then
        do i=1,maxunpar
                if (ratio(i) .gt.great)then
                great=ratio(i)
                endif
        enddo
endif
if (great .le. tol2)then
ioutconv=1
endif
write(20,*) iout,objf
iout=iout+1
enddo
print*,p
 stop
 end

!*********************CONCENTRATION CALCULATION STARTS**********************

subroutine
transport(nz,nm,delz,r,delt,timemax,coef1,coef2,v,dis,kd,row,theta,timeobs,maxtobs,ccal,maxzobs,nelt,
zobs,cobs,objf)

        real ae(nz),be(nz),ce(nz),re(nz),cdis(nz),cold(nz)
   real alpha(nz),bita(nz),y(nz),cte(nz)
        real cpre(nz),cadv(nz),cobs(nz,nm),delc(nz),ai(nz),bi(nz),fl1(nz),fl2(nz)
        real cnew(nz),ccal(nz,nm),timeobs(nm),nelt(nz),zobs(nz)
        real v,dis,lamda,kd
        do  j=2,(nz)
        cold(j)=0.0
```

```
        enddo
        cold(1)=1.0
  time=0.0
         ll2=1

do while(time .le.timemax)
        cold(1)=1.0
        delc(1)=0.0
                do j=2,(nz-1)
                ai(j)=cold(j)-cold(j-1)
                bi(j)=cold(j+1)-cold(j)
                        if (ai(j) .ge. 0.0)then
                        sign=1
                        else
                        sign=-1
                        endif
                        if ((ai(j)*bi(j)) .gt.0.0)then
                                if (abs (ai(j)).gt. abs(bi(j))) then
                                delc(j)=sign*abs(bi(j))
                                else
                                delc(j)=sign*abs(ai(j))
                                endif
                        else
                        delc(j)=0.0
                        endif
                enddo
                cpre(1)=cold(1)
                do j=2,nz-1
                cpre(j)=cold(j)-0.5*(delt/delz)*(v)*delc(j)
                enddo
                do j=1,nz-1
                fl1(j)=cpre(j)+0.5*delc(j)
                fl2(j)=cpre(j)-0.5*delc(j)
                cadv(j)=cold(j)-(1/r)*(delt/delz)*(v)*(fl1(j)-fl2(j))
                enddo
                cadv(1)=cold(1)
                lamda=(dis*delt)/(delz**2*r)
                do j=1,nz-1
                        if (j.eq.1)  then
                        ae(j)=0
                        be(j)=1
                        ce(j)=0
                        re(j)=cadv(1)
                        else
                        ae(j)=    -lamda
                        be(j)=     1+(2*lamda)
                        ce(j)=    -lamda
                        re(j)=cadv(j)
                        endif
                enddo
                ae(nz)=  0
                be(nz)=   1
                ce(nz)=0.0
                re(nz)=cadv(nz)
                alpha(1)=be(1)
                do j=1,nz-1
                bita(j)=ce(j)/alpha(j)
                alpha(j+1)=be(j+1)-(ae(j+1)*bita(j))
```

```
                enddo
                y(1)=re(1)/alpha(1)
                do j=2,(nz)
                y(j)=((re(j)-ae(j)*y(j-1))/alpha(j))
                enddo
                cte(nz)=y(nz)
                do j=(nz-1),1,-1
                cte(j)=y(j)-(bita(j)*cte(j+1))
                enddo
                do j=1,(nz)
                cdis(j)=cte(j)
                enddo
                do j=1,nz
                cnew(j)=cdis(j)-((coef1+(coef2*kd*row/theta))*cdis(j)*delt/r)
                enddo
                cnew(1)=1.0
                if (((time-delt) .le. timeobs(ll2)) .and. (timeobs(ll2) .lt. time))then
                do j=1,maxzobs
                cdiff1=cnew(nelt(j))-cold(nelt(j))
                tdiff1=timeobs(ll2)+delt-time
                c1=cold(nelt(j))+((cdiff2/delt)*tdiff1)
                cdiff2=cnew(nelt(j)+1)-cold(nelt(j)+1)
                tdiff2=timeobs(ll2)+delt-time
                c2=cold(nelt(j)+1)+((cdiff2/delt)*tdiff2)
                if (nelt(j) .eq. 1)then
                cdiff=c2-c1
                zdiff=zobs(j)
                ccal(j,ll2)=c1+(cdiff*zdiff)/delz
                else
                cdiff=c2-c1
                zdiff=zobs(j)-((nelt(j)-1)*delz)
                ccal(j,ll2)=c1+(cdiff*zdiff)/delz
                endif
                enddo
                ll2=ll2+1
                endif
                do j=1,nz
          cold(j)=cnew(j)
                enddo
                time=time+delt
        enddo
        sum=0
do i=1,maxzobs
        do j=1,maxtobs
        sum=sum+(cobs(i,j)-ccal(i,j))**2
        objf=sum
        enddo
enddo
return
end

!************************************************************************
!program for QR decomposition using House holder transformation for an unsymmetric matrice
subroutine householder(wj,qt1,nz,nm,maxunpar,itotal)
real wj(nz,nm),h(nz,nz),q(nz,nz),r2(nz,nm),w(nz),temp1(nz,nm),temp2(nz,nz)
real prod(nz,nm),qqt(nz,nz),qt1(nz,nz),r1(nz,nm)

nr=itotal
```

```
nc=maxunpar
do ii=1,nr
        do jj=1,nc
        r2(ii,jj)=0
        enddo
enddo
do ii=1,nr
        do jj=1,nc
        prod(ii,jj)=0
        enddo
enddo
do ii=1,nr
        do jj=1,nr
        qqt(ii,jj)=0
        enddo
enddo
do i=1,nc
sum=0
        do k=i,nr
                if(i .eq. 1)then
                sum=sum+wj(k,i)*wj(k,i)
                else
                sum=sum+r2(k,i)*r2(k,i)
                endif
        enddo
s=sum**(0.5)
        if(i .eq. 1)then
        den=(2.0*s*(s+abs(wj(i,i))))**(0.5)
        else
        den=(2*s*(s+abs(r2(i,i))))**(0.5)
        endif
        do k=1,nr
                if(i .eq. 1)then
                        if(k .lt. i)then
                        w(k)=0
                        else if(k .eq. i)then
                                if (wj(k,i) .gt. 0)then
                                w(k)=(wj(k,i)+s)/den
                                else if(wj(k,i) .eq. 0)then
                                w(k)=(wj(k,i))/den
                                else
                                w(k)=(wj(k,i)-s)/den
                                endif
                        else
                        w(k)=wj(k,i)/den
                        endif
                else
                        if (k .lt. i)then
                        w(k)=0
                        else if(k .eq. i)then
                                if(r2(k,i) .gt. 0)then
                                w(k)=(r2(k,i)+s)/den
                                else if(r2(k,i) .eq. 0)then
                                w(k)=r2(k,i)/den
                                else
                                w(k)=(r2(k,i)-s)/den
                                endif
                        else
```

```
                    w(k)=r2(k,i)/den
                    endif
            endif
    enddo

    !calculation of house holder matrix
    do j=1,nr
            do k=1,nr
                    if(k .eq. j)then
                    h(j,k)=1.0-2.0*w(j)*w(k)
                    else
                    h(j,k)=-2.0*w(j)*w(k)
                    endif
            enddo
    enddo

!   calculation of r matrix
    do ii=1,nr
            do jj=1,nc
            temp1(ii,jj)=0
            enddo
    enddo
    do j=1,nr
            do k=1,nc
                    do l=1,nr
                            if(i .eq. 1)then
                            temp1(j,k)=temp1(j,k)+h(j,l)*wj(l,k)
                            else
                            temp1(j,k)=temp1(j,k)+h(j,l)*r2(l,k)
                            endif
                    enddo
            enddo
    enddo
    do j=1,nr
            do k=1,nc
            r2(j,k)=temp1(j,k)
            enddo
    enddo
!   calculation of q matrix
    do ii=1,nr
            do jj=1,nr
            temp2(ii,jj)=0
            enddo
    enddo
    if (i .eq. 1)then
            do j=1,nr
                    do k=1,nr
                    temp2(j,k)=h(j,k)
                    enddo
            enddo
    else
            do j=1,nr
                    do k=1,nr
                            do l=1,nr
                            temp2(j,k)=temp2(j,k)+q(j,l)*h(l,k)
                            enddo
                    enddo
            enddo
```

```
        endif
        do j=1,nr
                do k=1,nr
                q(j,k)=temp2(j,k)
                enddo
        enddo
enddo
do i=1,nr
        do j=1,nr
        qt1(j,i)=q(i,j)
        enddo
enddo
do ii=1,nr
        do jj=1,nc
                do kk=1,nr
                r1(ii,jj)=r1(ii,jj)+qt1(ii,kk)*wj(kk,jj)
                enddo
        enddo
enddo
do ii=1,nr
        do jj=1,nc
                do kk=1,nr
                prod(ii,jj)=prod(ii,jj)+q(ii,kk)*r2(kk,jj)
                enddo
        enddo
enddo
do ii=1,nr
        do jj=1,nr
                do kk=1,nr
                qqt(ii,jj)=qqt(ii,jj)+q(ii,kk)*qt1(kk,jj)
                enddo
        enddo
enddo
return
end
!*******************************************************************************

 !program for Given's rotation
  subroutine givrot(itotal,maxunpar,nz,nm,wjnew,qt2)
real g1(nz,nz),wjnew(nz,nm),r3(nz,nm),temp3(nz,nm),temp4(nz,nz)
real q1(nz,nz),prod1(nz,nz),qt2(nz,nz),qqt1(nz,nz)

nr=itotal+maxunpar
nc=maxunpar
nin=itotal
do ii=1,nr
        do jj=1,nr
        prod1(ii,jj)=0
        qqt1(ii,jj)=0
        enddo
enddo
do ii=1,nr
        do jj=1,nc
        r3(ii,jj)=0
        enddo
enddo
do i=1,nc
        if (i .eq. 1)then
```

```
den=sqrt(wjnew(i,i)*wjnew(i,i)+wjnew(i+nin,i)*wjnew(i+nin,i))
cos=wjnew(i,i)/den
sin=wjnew(i+nin,i)/den
else
den=sqrt(r3(i,i)*r3(i,i)+r3(i+nin,i)*r3(i+nin,i))
cos=r3(i,i)/den
sin=r3(i+nin,i)/den
endif
do j=1,nr
        do k=1,nr
                if(j .eq. i)then
                        if(j .eq. k)then
                        g1(j,k)=cos
                        else if(k .eq. (j+nin))then
                        g1(j,k)=sin
                        else
                        g1(j,k)=0
                        endif
                else if(j .eq. (i+nin))then
                        if(k .eq. (j-nin))then
                        g1(j,k)=-sin
                        else if(j .eq. k)then
                        g1(j,k)=cos
                        else
                        g1(j,k)=0
                        endif
                else
                        if(j .eq. k)then
                        g1(j,k)=1
                        else
                        g1(j,k)=0
                        endif
                endif
        enddo
enddo
do j=1,nr
        do k=1,nc
        temp3(j,k)=0
        enddo
enddo
do j=1,nr
        do k=1,nc
                do l=1,nr
                        if(i .eq. 1)then
                        temp3(j,k)=temp3(j,k)+g1(j,l)*wjnew(l,k)
                        else
                        temp3(j,k)=temp3(j,k)+g1(j,l)*r3(l,k)
                        endif
                enddo
        enddo
enddo
do j=1,nr
        do k=1,nc
        r3(j,k)=temp3(j,k)
        enddo
enddo
write(6,700)((r3(j,k),k=1,nc),j=1,nr)
700 format (1x,3f15.5)
```

```
!       write (20,*) 'R2 matrix'
!       do j=1,nr
!       write(20,*)(r2(j,k),k=1,nc)
!       enddo
        do j=1,nr
                do k=1,nr
                temp4(j,k)=0
                enddo
        enddo
        if(i .eq. 1)then
                do j=1,nr
                        do k=1,nr
                        temp4(j,k)=g1(j,k)
                        enddo
                enddo
        else
                do j=1,nr
                        do k=1,nr
                                do l=1,nr
                                temp4(j,k)=temp4(j,k)+qt2(j,l)*g1(l,k)
                                enddo
                        enddo
                enddo
        endif

        do j=1,nr
                do k=1,nr
                qt2(j,k)=temp4(j,k)
                enddo
        enddo
!       write (20,*) 'QT given matrix'
!       do j=1,nr
!       write(20,*)(qt2(j,k),k=1,nr)
!       enddo
enddo
do ii=1,nr
        do jj=1,nr
                q1(ii,jj)=qt2(jj,ii)
        enddo
enddo
!write (20,*) 'Q matrix'
!do j=1,nr
!write(20,*)(q(j,k),k=1,nr)
!enddo
do j=1,nr
        do k=1,nc
                do l=1,nr
                prod1(j,k)=prod1(j,k)+q1(j,l)*r3(l,k)
                enddo
        enddo
enddo
!write (20,*) 'product matrix'
!do j=1,nr
!write(20,*)(prod(j,k),k=1,nc)
!enddo
do j=1,nr
        do k=1,nr
                do l=1,nr
```

```
                qqt1(j,k)=qqt1(j,k)+q1(j,l)*qt2(l,k)
                enddo
        enddo
enddo
!write (20,*) 'qqt matrix'
!do j=1,nr
!write(20,*)(qqt(j,k),k=1,nr)
!enddo
return
end
!*************************************************************************
subroutine gauss(nz,nm,maxunpar,wjfin,errfin,delp)
real wjfin(nz,nm),errfin(nz),delp(nm)
do i=maxunpar,1,-1
        if (i.eq. maxunpar)then
        delp(i)=-errfin(i)/wjfin(i,i)
        else
        sum=-errfin(i)
                do j=maxunpar,i+1,-1
                sum=sum-wjfin(i,j)*delp(j)
                enddo
        delp(i)=sum/wjfin(i,i)
        endif
enddo
return
end
```

APPENDIX-III

NUMERICAL MODEL FOR SIMULATING VIRUS TRANSPORT IN UNSATURATED ZONE

```
! Depth =Total depth
        ! delt    `=small time step
        !thetas   =saturated water content
        !thetar   =residual water content
        !alpha    = van genucheten parameter, is a measure of the first moment of the pore size density
function
        !ks       =saturated hydraulic conductivity
        !n        =inverse measure of the second moment of the pore size density function
        !hold     =old value of pressure head
        !hnew     =new value of the pressure head
        !hassum   =assumed value of the pressure head
        !thetaold=old value of the water content
        !k        =hydraulic conductivity
        !sc       =specific water capacity
        !z        =define the coordinate of the grid
        !Nnode    =total number of nodes
        !a,b,c,d,f,g=coefficient of banded coefficient matrix
        parameter(Nnode=251)
        real
z(Nnode),k(Nnode),hnew(Nnode),hold(Nnode),a(Nnode),b(Nnode),c(Nnode),d(Nnode),theta(Nnode)
        real ae(Nnode),be(Nnode),ce(Nnode),re(Nnode),variable(Nnode),v1(Nnode)
        real
hassum(Nnode),f(Nnode),f1(Nnode),g(Nnode),thetaold(Nnode),sc(Nnode),r(Nnode),rold(Nnode),v(Nn
ode)
        real
cold(Nnode),cpre(Nnode),cadv(Nnode),delc(Nnode),ai(Nnode),bi(Nnode),fl1(Nnode),fl2(Nnode)
        real cnew(Nnode),cdis(Nnode),dis(Nnode),lamda(Nnode), r1(Nnode)
        real t1(87000),cnew1(87000),theta1(87000),hnew1(87000)
        real m,n,alpha,thetas,thetar,ks,max,pe,hini,htop,c0,row,kd,coef1,coef2

        open(5,file="dn.in")


!       open(10,file="dn5.xls")
        open(11,file="dn.xls")
        open(12,file="final.xls")
        open(14,file="velocity.xls")
!       open(15,file="adv.xls")
        open(16,file="cnew.xls")
        read (5,*) depth,delt,thetas,thetar,alpha,n,ks,timemax,pe,hini,htop,c0,row,kd,coef1,coef2

        !************DEFINE THE COORDINATES OF EACH GRID**********************
        delz=depth/(Nnode-1)
        z(1)=0.0
        z(Nnode)=depth
   do j=2,Nnode-1
          z(j)=z(1)+(j-1)*delz
        enddo
```

```
        do j=2,Nnode
        cold(j)=0.0
        enddo
        cold(1)=c0
        !************INITIAL AND BOUNDARY CONDITIONS************************
        i1=1
        do j=1,Nnode
        hold(j)=hini
        enddo
        !************CALCULATION OF thetaold VALUE FROM hold VALUE*************
        m=1-(1/n)
        do j=1,Nnode
                if (hold(j).gt.0.0)then
                thetaold(j)=thetas
                else
                rold(j)=(1+(alpha*abs(hold(j)))**n)
                thetaold(j)=((thetas-thetar)*(1/(rold(j)**m))+ thetar)
                endif
        enddo
        time=delt
        !***********************TIME ITERATION STARTS*****************************
        do while(time .le. timemax)
                do j=1,Nnode
                hassum(j)=hold(j)
                enddo
  error=1
        itr=0
        !********TO MAKE MASS CONSERVATION ERROR CALCULATION STARTS******
                do while(error .gt. 0.001)
                        do j=1,(Nnode)
                        r(j)=(1+(alpha*abs(hassum(j)))**n)
                                if (hassum(j).gt.0.0)then
                                sc(j)=0.0
                                theta(j)=thetas
                                else
                                sc(j)=((thetas-thetar)*(alpha*m*n)*((alpha*abs(hassum(j)))**(n-
1)))/r(j)**(m+1)
                                theta(j)=((thetas-thetar)*(1/(r(j)**m))+ thetar)
                                endif
                        k(j)=ks*((1-(1-((theta(j)-thetar)/(thetas-thetar))**(1/m))**m)**2)*((theta(j)-
thetar)/(thetas-thetar))**0.5
                        enddo
                        do j=2,Nnode-1
                        a(j)=(k(j)+k(j-1))/(2*delz**2)
                        c(j)=(k(j)+k(j+1))/(2*delz**2)
                        d(j)=-(k(j+1)-k(j-1))/(2*delz)
                        f(j)=sc(j)/delt
                        f1(j)=f(j)*hassum(j)
                        g(j)=(theta(j)-thetaold(j))/delt
                        b(j)=-(a(j)+c(j)+f(j))
                        enddo
                        do j=1,Nnode-1
                                if (j.eq.1) then
                                ae(j)=0
                                be(j)=1
                                ce(j)=0
                                re(j)=hini
                                else
```

```
                        ae(j)=a(j)
                        be(j)=b(j)
                        ce(j)=c(j)
                        re(j)=d(j)+g(j)-f1(j)
                        endif
                enddo
                ae(Nnode)=0
                be(Nnode)=1
                ce(Nnode)=0.0
                re(Nnode)=htop
                call tdma(ae,be,ce,re,variable,Nnode)
                        do j=1,Nnode
                        hnew(j)=variable(j)
                        enddo
        !       write(10,20)(z(j),hassum(j),hnew(j),k(j),j=1,Nnode)
        !       20 format(1x,f8.3,f11.4,f11.4,f11.4)
                error=abs(hnew(1)-hassum(1))
                do j=2,Nnode
                        if (error .lt. (abs(hnew(j)-hassum(j)))) then
                        error=abs(hnew(j)-hassum(j))
                        endif
                enddo
                print *,error
                do j=1,Nnode
                hassum(j)=hnew(j)
                enddo
                itr=itr+1
        enddo
!***************************** ERROR CALCULATION ENDS*************************
        print *,'total no of iteration is', itr
        do j=1,Nnode
                if (hnew(j).gt.0.0)then
                theta(j)=thetas
                else
                r(j)=(1+(alpha*abs(hnew(j)))**n)
                theta(j)=((thetas-thetar)*(1/(r(j)**m))+ thetar)
                endif
        enddo
!       write(11,30)(z(j),hnew(j),theta(j),k(j),j=1,Nnode)
!       30 format(1x,f8.3,f11.4,f10.7,f10.5)
!*************************TIME ITERATION ENDS*********************************


!**********************VELOCITY CALCULATION STARTS**************************
  v(1)=(-k(1)/theta(1))*(((hnew(2)-hnew(1))/(delz))+1)
        do j=2,Nnode-1
        v(j)=(-k(j)/theta(j))*(((hnew(j+1)-hnew(j-1))/(2*delz))+1)
        enddo
v(Nnode)=(-k(Nnode)/theta(Nnode))*(((hnew(Nnode)-hnew(Nnode-1))/(delz))+1)

        do j=1,Nnode
        v1(j)=v(Nnode+1-j)
        enddo
        do j=1,Nnode
        v(j)=(-1)*v1(j)
        enddo
```

```
!****************CONCENTRATION CALCULATION STARTS************************
      do j=1, Nnode
      r1(j)=1+(row*kd/theta(j))
      enddo
      cold(1)=c0
      delc(1)=0.0
            do j=2,(Nnode-1)
            ai(j)=cold(j)-cold(j-1)
            bi(j)=cold(j+1)-cold(j)
                  if (ai(j) .ge. 0.0)then
                  sign=1
                  else
                  sign=-1
                  endif


                  if((ai(j)*bi(j)) .gt.0.0)then
                              if (abs (ai(j)).ge. abs(bi(j))) then
                              max=abs(ai(j))
                              else
                              max=abs(bi(j))
                              endif
                              if(2*abs(ai(j)) .lt. 2*abs(bi(j)))then
                                    if(2*abs(ai(j)) .lt. max)then
                                    delc(j)=sign*2*abs(ai(j))
                                    else
                                    delc(j)=sign*max
                                    endif
                              else
                                    if(2*abs(bi(j)) .lt. max)then
                                    delc(j)=sign*2*abs(bi(j))
                                    else
                                    delc(j)=sign*max
                                    endif
                              endif
                        else
                        delc(j)=0
                        endif

            cpre(1)=cold(1)
            cpre(j)=cold(j)-0.5*(delt/delz)*(v(j))*delc(j)

                  if (v(j).gt.0.0)then
                  fl1(j)=cpre(j)+0.5*delc(j)
                  fl2(j)=cpre(j-1)+0.5*delc(j-1)
                  cadv(j)=cold(j)-(delt/delz)*(v(j))*(fl1(j)-fl2(j))
                  else
                  fl1(j)=cpre(j+1)-0.5*delc(j+1)
                  fl2(j)=cpre(j)-0.5*delc(j)
                  cadv(j)=cold(j)-(delt/delz*(r1(j)))*(v(j+1))*(fl1(j)-fl2(j))
                  endif
            enddo
            cadv(1)=cold(1)

!     write(15,60)(z(j),cadv(j),j=1,Nnode)
!     60 format(1x,f8.3,f11.4)

            do j=1,Nnode
```

```
                dis(j)=(v(j)*delz/pe)
                lamda(j)=(dis(j)*delt/((delz**2)*r1(j)))
                enddo
                do j=1,Nnode-1
                        if (j.eq.1)  then
                        ae(j)=0
                        be(j)=1
                        ce(j)=0
                        re(j)=cadv(1)
                        else
                        ae(j)=    -lamda(j)
                        be(j)=     1+(2*lamda(j))
                        ce(j)=    -lamda(j)
                        re(j)=cadv(j)
                        endif
                enddo
                ae(Nnode)=        0
                be(Nnode)=         1
                ce(Nnode)=0.0
                re(Nnode)=cadv(Nnode)

                call tdma(ae,be,ce,re,variable,Nnode)
                        do j=1,Nnode
                        cdis(j)=variable(j)
                        enddo
                        do j=1,Nnode
                        cnew(j)=cdis(j)-((coef1+(coef2*kd*row/theta(j)))*cdis(j)*delt/r1(j))
                        enddo

                cnew(1)=1.0
                        do j=1,Nnode
                        cold(j)=cnew(j)
                        hold(j)=hnew(j)
                        thetaold(j)=theta(j)
                        enddo
                t1(i1)=time
                cnew1(i1)=cnew(26)
                theta1(i1)=theta(26)
                hnew1(i1)=hnew(26)
                i1=i1+1
                time=time+delt
                enddo
        write(12,40)(z(j),hnew(j),theta(j),k(j),j=1,Nnode)
        40 format(1x,f8.3,f11.4,f10.7,f10.5)
        write(14,50)(z(j),v(j),j=1,Nnode)
        50 format(1x,f8.3,f11.4)
        write (11,*)(t1(i1),cnew1(i1),theta1(i1),hnew1(i1),i1=1,86400,4)
        write(16,70)(z(j),cnew(j),v(j),j=1,Nnode)
        70 format(1x,f8.3,f11.4,f11.4)
 stop
 end

!*****************************THOMAS ALGORITHM*******************************

subroutine tdma(ae,be,ce,re,variable,Nnode)
        real ae(Nnode),be(Nnode),ce(Nnode),re(Nnode),variable(Nnode)
  real alpha(Nnode),bita(Nnode),y(Nnode),cte(Nnode)
                alpha(1)=be(1)
```

```
        do j=1,Nnode-1
        bita(j)=ce(j)/alpha(j)
        alpha(j+1)=be(j+1)-(ae(j+1)*bita(j))
        enddo
        y(1)=re(1)/alpha(1)

        do j=2,(Nnode)
        y(j)=((re(j)-ae(j)*y(j-1))/alpha(j))
        enddo

        cte(Nnode)=y(Nnode)
        do j=(Nnode-1),1,-1
        cte(j)=y(j)-(bita(j)*cte(j+1))
        enddo
        do j=1,(Nnode)
        variable(j)=cte(j)
        enddo
return
end
```

APPENDIX- IV

NUMERICAL MODEL FOR ESTIMATING VIRUS TRANSPORT PARAMETERS IN UNSATURATED ZONE

```
        ! Depth =Total depth
        ! delt    `=small time step
        !thetas   =saturated water content
        !thetar   =residual water content
        !alpha    = van genucheten parameter, is a measure of the first moment of the pore size density
function
        !ks              =saturated hydraulic conductivity
        !n               =inverse measure of the second moment of the pore size density function
        !hold     =old value of pressure head
        !hnew     =new value of the pressure head
        !hassum =assumed value of the pressure head
        !thetaold=old value of the water content
        !k                       =hydraulic conductivity
        !sc                      =specific water capacity
        !z                       =define the coordinate of the grid
        !nz              =total number of nodes
        !a,b,c,d,f,g=coefficient of banded coefficient matrix

        !*************************************************************************

parameter(nz=101)
parameter(nm=7)
real cobs(nz,nm),z(nz),b(nm),paracode(nm),cpred(nz,nm),cori(nz,nm),timeobs(nm)
real ccal(nz,nm),nelt(nz),zobs(nz),p(nm),plast(nm),err(nz),wj(nz,nm),d(nm),dold(nm)
real qt1(nz,nz),wjnew(nz,nz),errnew(nz),qt2(nz,nz),wjfin(nz,nz),errfin(nz),delp(nm),ratio(nm)
real depth,row,kd,coef1,coef2
real n,alpha,thetas,thetar,ks
integer maxpar,maxunpar,codeobj
open(5,file="dn.in")
open(6,file="cnew.xls")
open(11,file="cobs.xls")
open(12,file="cpred.xls")
open(13,file="cori.xls")
open(14,file="err.xls")
open(15,file="jac.xls")
open(16,file="householder.xls")
open(17,file="wjnew.xls")
open(18,file="given.xls")
open(19,file="wjfinal.xls")
open(20,file="error.xls")
open(21,file="check.out")

read(5,*) maxpar,maxunpar,depth,row,delt,timemax,maxtobs,maxzobs,almb,codeobj,hini,htop,pe
read(5,*) (paracode(i),i=1,maxpar)
read(5,*) (b(i),i=1,maxpar)
read(5,*) (timeobs(i),i=1,maxtobs)
read(5,*) (zobs(i),i=1,maxzobs)
read(5,*) (p(i),i=1,maxpar)
```

```
read(5,*) omega1,omega2,tol1,tol2
read(5,*) n,alpha,thetas,thetar,ks
read(5,*) ((cobs(i,j),j=1,maxtobs),i=1,maxzobs)
delz=depth/(nz-1)
z(1)=0.0
do j=2,nz
        z(j)=z(1)+(j-1)*delz
enddo
i=1
k=1
z1=0
itotal=maxzobs*maxtobs
do while(i .le. maxzobs)
        if (zobs(i) .gt. z1)then
        k=k+1
        z1=z1+delz
        else
        nelt(i)=k-1
        diff=zobs(i)-z1
        i=i+1
        endif
enddo

iout=1
ioutconv=0
do while (ioutconv .eq. 0)
        do ipar=1,(maxunpar+1)
                if (ipar .eq. 1)then
                        do j=1,maxpar
                        plast(j)=p(j)
                        enddo
                icountpar=1
                icpar=0
                else
                ikode=0
                        do while (ikode .eq. 0)
                                if (paracode(icountpar) .eq. 1)then
                                ikode=1
                                icpar=icountpar
                                icountpar=icountpar+1
                                else
                                icountpar=icountpar+1
                                endif
                        enddo
                endif
                do j=1,maxpar
                        if (j .eq. icpar)then
                        p(j)=plast(j)+0.01*plast(j)
                        else
                        p(j)=plast(j)
                        endif
                enddo
                do j=1,maxpar
                        if(j .eq.1)then
                        coef1=p(1)
                        else if(j .eq.2)then
                        coef2=p(2)
                        else if(j .eq.3)then
```

```
                    kd=p(3)
                    endif
                enddo
        call
transport(z,nz,nm,delz,delt,timemax,coef1,coef2,kd,row,timeobs,maxtobs,ccal,maxzobs,nelt,zobs,cobs,
objf,codeobj,n,alpha,thetas,thetar,ks,hini,htop,pe)
        write(21,*)objf
                if (ipar .eq. 1)then
                objfpre=objf
                    do j=1,maxzobs
                        do k=1,maxtobs
                        cori(j,k)=ccal(j,k)
                        enddo
                    enddo
        !           write(13,30)((cori(j,k),k=1,maxtobs),j=1,maxzobs)
        !           30 format(1x,5f10.5)
                else
                    do j=1,maxzobs
                        do k=1,maxtobs
                        cpred(j,k)=ccal(j,k)
                        enddo
                    enddo
        !           write(12,20)((cpred(j,k),k=1,maxtobs),j=1,maxzobs)
        !           20 format(1x,5f10.5)
                endif
                if (ipar .eq. 1)then
                    if (codeobj .eq. 1)then
                        do j=1,maxtobs
                        ierr=(j-1)*maxzobs
                            do i=1,maxzobs
                            err(ierr+i)=cobs(i,j)-cori(i,j)
                            enddo
                        enddo
                    else if (codeobj .eq. 2)then
                        do j=1,maxtobs
                        ierr=(j-1)*maxzobs
                            do i=1,maxzobs
                            err(ierr+i)=(cobs(i,j)-cori(i,j))/cobs(i,j)
                            enddo
                        enddo
                    else if (codeobj .eq. 3)then
                        do j=1,maxtobs
                        ierr=(j-1)*maxzobs
                            do i=1,maxzobs
                            err(ierr+i)=(cobs(i,j)-cori(i,j))/cori(i,j)
                            enddo
                        enddo
                    endif

        !           write (14,40) (err(i),i=1,itotal)
        !           40 format(1x,f10.5)
                else
                    if (codeobj .eq. 1)then
                        do j=1,maxtobs
                        ierr=(j-1)*maxzobs
                            do i=1,maxzobs
                            wj(ierr+i,ipar-1)=(cpred(i,j)-cori(i,j))/(p(icpar)-
plast(icpar))
```

```
                                        wj(ierr+i,ipar-1)=-wj(ierr+i,ipar-1)
                                        enddo
                                enddo
                        else if (codeobj .eq. 2)then
                                do j=1,maxtobs
                                ierr=(j-1)*maxzobs
                                        do i=1,maxzobs
                                        wj(ierr+i,ipar-1)=(cpred(i,j)-cori(i,j))/(p(icpar)-
plast(icpar))
                                        wj(ierr+i,ipar-1)=-wj(ierr+i,ipar-1)/cobs(i,j)
                                        enddo
                                enddo
                        else if (codeobj .eq. 3)then
                                do j=1,maxtobs
                                ierr=(j-1)*maxzobs
                                        do i=1,maxzobs
                                        wj(ierr+i,ipar-1)=(cpred(i,j)-cori(i,j))/(p(icpar)-
plast(icpar))
                                        wj(ierr+i,ipar-1)=-wj(ierr+i,ipar-1)*cobs(i,j)/(cori(i,j)**2)
                                        enddo
                                enddo
                        endif
                endif
        enddo
!write (15,50) ((wj(i,j),j=1,maxunpar),i=1,itotal)
!50 format(1x,3f10.3)
do i=1,maxunpar
sum=0
        do j=1,itotal
        sum=sum+wj(j,i)*wj(j,i)
        enddo
d(i)=abs(sqrt(sum))
enddo
if (iout .eq. 1)then
        do i=1,maxunpar
        dold(i)=d(i)
        enddo
else
        do i=1,maxunpar
                if (dold(i) .gt. d(i))then
                d(i)=dold(i)
                dold(i)=d(i)
                else
                dold(i)=d(i)
                endif
        enddo
endif
call householder(wj,qt1,nz,nm,maxunpar,itotal)
write(16,60)((qt1(i,j),j=1,itotal),i=1,itotal)
60 format(1x, 55f15.8)
mtot=itotal+maxunpar
almbcode=1.0
do while(almbcode .eq. 1)
write(21,*)almb
do i=1,mtot
        do j=1,maxunpar
        wjnew(i,j)=0
        enddo
```

```
enddo
do i=1,mtot
errnew(i)=0
enddo
do i=1,mtot
        do j=1,maxunpar
                if (i .le. itotal)then
                        do k=1,itotal
                        wjnew(i,j)=wjnew(i,j)+qt1(i,k)*wj(k,j)
                        errnew(i)=errnew(i)+qt1(i,k)*err(k)
                        enddo
                else
                lll=i-itotal
                        if(j .eq.lll)then
                        wjnew(i,j)=almb*d(j)
                        else
                        wjnew(i,j)=0
                        endif
                errnew(i)=0
                endif
        enddo
enddo
!write(17,70)((wjnew(i,j),j=1,maxunpar),i=1,mtot)
!70     format(1x,f15.5)
!write(17,71)(errnew(i),i=1,mtot)
!71     format(1x,f10.5)

call givrot(itotal,maxunpar,nz,nm,wjnew,qt2)
!write(18,80)((qt2(i,j),j=1,mtot),i=1,mtot)
!80     format(1x,56f10.5)
do i=1,mtot
        do j=1,maxunpar
        wjfin(i,j)=0
        enddo
        enddo
do i=1,mtot
errfin(i)=0
enddo
do i=1,mtot
        do j=1,maxunpar
                do k=1,mtot
                wjfin(i,j)=wjfin(i,j)+qt2(i,k)*wjnew(k,j)
                errfin(i)=errfin(i)+qt2(i,k)*errnew(k)
                enddo
        enddo
enddo
!write(19,90)((wjnew(i,j),j=1,maxunpar),i=1,mtot)
!90 format(1x,3f15.5)
!write(19,91)(errnew(i),i=1,mtot)
!91 format(1x,f10.5)
ffin=0
do j=1,maxunpar
ffin=ffin+errfin(j)*errfin(j)
enddo
gfin=0
do j=maxunpar+1,mtot
gfin=gfin+errfin(j)*errfin(j)
enddo
```

```
call gauss(nz,nm,maxunpar,wjfin,errfin,delp)
print*,(delp(j),j=1,maxunpar)
icount=1
do i=1,maxpar
        if(paracode(i) .eq. 1)then
        con=delp(icount)/plast(i)
                if((con .ge. -0.2) .and. (con .le. 0.5))then
                p(i)=plast(i)+delp(icount)
                elseif (con .lt. (-0.2))then
                delp(icount)=plast(i)*(-0.2)
                p(i)=plast(i)+delp(icount)
                elseif (con .gt. (0.5))then
                delp(icount)=plast(i)*(0.5)
                p(i)=plast(i)+delp(icount)
                endif
        ratio(icount)=abs(delp(icount))/(plast(i)+10.0**(-20))
        icount=icount+1
        endif
enddo

do j=1,maxpar
                if(j .eq.1)then
                coef1=p(1)
                elseif(j .eq.2)then
                coef2=p(2)
                elseif(j .eq.3)then
                kd=p(3)
                endif
        enddo


call
transport(z,nz,nm,delz,delt,timemax,coef1,coef2,kd,row,timeobs,maxtobs,ccal,maxzobs,nelt,zobs,cobs,
objf,codeobj,n,alpha,thetas,thetar,ks,hini,htop,pe)
write(21,*)iout,objf,objfpre

if (objf .gt. objfpre)then
almb=almb*omega1
almbcode=1
else
almbcode=0
endif
enddo
ratio1=gfin/ffin
sump1=sqrt(1.0+objfpre)
if(ratio1 .ge. 0.75)then
almb=almb*omega2
else if(ratio1 .le. 0.25)then
almb=almb*omega1
endif
if(ffin .le. (sump1*tol1))then
ioutconv=1
endif
great=ratio(1)
if(maxunpar .ge.2)then
        do i=1,maxunpar
                if (ratio(i) .gt.great)then
                great=ratio(i)
```

```
                endif
        enddo
endif
if (great .le. tol2)then
ioutconv=1
endif
write(20,*) iout,objf
iout=iout+1
enddo
print*,p
 stop
 end

!******************CONCENTRATION CALCULATION STARTS**********************

subroutine
transport(z,nz,nm,delz,delt,timemax,coef1,coef2,kd,row,timeobs,maxtobs,ccal,maxzobs,nelt,zobs,cobs,
objf,codeobj,n,alpha,thetas,thetar,ks,hini,htop,pe)

        real ae(nz),be(nz),ce(nz),re(nz),cdis(nz),cold(nz)
        real cpre(nz),cadv(nz),cobs(nz,nm),delc(nz),ai(nz),bi(nz),fl1(nz),fl2(nz)
        real cnew(nz),ccal(nz,nm),timeobs(nm),nelt(nz),zobs(nz)
        integer codeobj
        real k(nz),hnew(nz),hold(nz),a(nz),b(nz),c(nz),d(nz),theta(nz)
        real variable(nz),v1(nz)
        real hassum(nz),f(nz),f1(nz),g(nz),thetaold(nz),sc(nz),r(nz),rold(nz),v(nz)
        real r1(nz),dis(nz),lamda(nz),z(nz)
        real kd,row,m,ks,n
 !*****************************************************************************
        open(30,file="dn5.out")
        open(31,file="dn.out")
        open(32,file="final.out")
        open(33,file="velocity.out")
        open(34,file="adv.out")
        open(35,file="cnew.out")
!*****************************************************************************
        do  j=2,(nz)
        cold(j)=0.0
        enddo
        cold(1)=1.0
   time=0.0
         ll2=1

        !************INITIAL AND BOUNDARY CONDITIONS************************

        do j=1,nz
        hold(j)=hini
        enddo
        !************CALCULATION OF thetaold VALUE FROM hold
VALUE*****************************
        m=1-(1/n)
        do j=1,nz
                if (hold(j).gt.0.0)then
                thetaold(j)=thetas
                else
                rold(j)=(1+(alpha*abs(hold(j)))**n)
                thetaold(j)=((thetas-thetar)*(1/(rold(j)**m))+ thetar)
                endif
```

```
        enddo
        time=0.0
        !************************TIME ITERATION STARTS*****************************
        do while(time .lt. timemax)
                do j=1,nz
                hassum(j)=hold(j)
                enddo
  error=1
        itr=0
        !******TO MAKE MASS CONSERVATION ERROR CALCULATION STARTS********
                do while(error .gt. 0.001)
                        do j=1,(nz)
                        r(j)=(1+(alpha*abs(hassum(j)))**n)
                                if (hassum(j).gt.0.0)then
                                sc(j)=0.0
                                theta(j)=thetas
                                else
                                sc(j)=((thetas-thetar)*(alpha*m*n)*((alpha*abs(hassum(j)))**(n-
1)))/r(j)**(m+1)
                                theta(j)=((thetas-thetar)*(1/(r(j)**m))+ thetar)
                                endif
                        k(j)=ks*((1-(1-((theta(j)-thetar)/(thetas-thetar))**(1/m))**m)**2)*((theta(j)-
thetar)/(thetas-thetar))**0.5
                        enddo
                        do j=2,nz-1
                        a(j)=(k(j)+k(j-1))/(2*delz**2)
                        c(j)=(k(j)+k(j+1))/(2*delz**2)
                        d(j)=-(k(j+1)-k(j-1))/(2*delz)
                        f(j)=sc(j)/delt
                        f1(j)=f(j)*hassum(j)
                        g(j)=(theta(j)-thetaold(j))/delt
                        b(j)=-(a(j)+c(j)+f(j))
                        enddo
                        do j=1,nz-1
                                if (j.eq.1)  then
                                ae(j)=0
                                be(j)=1
                                ce(j)=0
                                re(j)=hini
                                else
                                ae(j)=a(j)
                                be(j)=b(j)
                                ce(j)=c(j)
                                re(j)=d(j)+g(j)-f1(j)
                                endif
                        enddo
                        ae(nz)=0
                        be(nz)=1
                        ce(nz)=0.0
                        re(nz)=htop
                        call tdma(ae,be,ce,re,variable,nz)
                                do j=1,nz
                                hnew(j)=variable(j)
                                enddo
                        error=abs(hnew(1)-hassum(1))
                        do j=2,nz
                                if (error  .lt. (abs(hnew(j)-hassum(j)))) then
                                error=abs(hnew(j)-hassum(j))
```

```
                    endif
                enddo
                print *,error
                do j=1,nz
                hassum(j)=hnew(j)
                enddo
                itr=itr+1
        enddo
!************************ ERROR CALCULATION ENDS*****************************
        print *,'total no of iteration is', itr
        do j=1,nz
                if (hnew(j).gt.0.0)then
                theta(j)=thetas
                else
                r(j)=(1+(alpha*abs(hnew(j)))**n)
                theta(j)=((thetas-thetar)*(1/(r(j)**m))+ thetar)
                endif
        enddo

!************************TIME ITERATION ENDS*********************************

!       write(32,220)(z(j),hnew(j),theta(j),k(j),j=1,nz)
!       220 format(1x,f8.3,f11.4,f10.7,f10.5)

!***********************VELOCITY CALCULATION STARTS************************
        v(1)=(-k(1)/theta(1))*(((hnew(2)-hnew(1))/(delz))+1)
                do j=2,nz-1
                v(j)=(-k(j)/theta(j))*(((hnew(j+1)-hnew(j-1))/(2*delz))+1)
                enddo
        v(nz)=(-k(nz)/theta(nz))*(((hnew(nz)-hnew(nz-1))/(delz))+1)

                do j=1,nz
                v1(j)=v(nz+1-j)
                enddo
                do j=1,nz
                v(j)=(-1)*v1(j)
                enddo

!       write(33,230)(z(j),v(j),j=1,nz)
!       230 format(1x,f8.3,f11.4)

!*******************CONCENTRATION CALCULATION STARTS**********************
                do j=1,nz
                r1(j)=1+((row*kd)/theta(j))
                enddo
        cold(1)=1.0
        delc(1)=0.0
                do j=2,(nz-1)
                ai(j)=cold(j)-cold(j-1)
                bi(j)=cold(j+1)-cold(j)
                        if (ai(j) .ge. 0.0)then
                        sign=1
                        else
                        sign=-1
                        endif

                        if ((ai(j)*bi(j)) .gt.0.0)then
                                if (abs (ai(j)).gt. abs(bi(j))) then
```

```
                delc(j)=sign*abs(bi(j))
                else
                delc(j)=sign*abs(ai(j))
                endif
        else
        delc(j)=0.0
        endif
 enddo
cpre(1)=1.0
do j=2,nz-1
cpre(j)=cold(j)-0.5*(delt/delz)*(v(j))*delc(j)
enddo

do j=1,nz-1
        if (v(j).gt.0.0)then
        fl1(j)=cpre(j)+0.5*delc(j)
        fl2(j)=cpre(j-1)+0.5*delc(j-1)
        cadv(j)=cold(j)-(1/r1(j))*(delt/delz)*(v(j))*(fl1(j)-fl2(j))
        else
        fl1(j)=cpre(j+1)-0.5*delc(j+1)
        fl2(j)=cpre(j)-0.5*delc(j)
        cadv(j)=cold(j)-(1/r1(j))*(delt/delz)*(v(j+1))*(fl1(j)-fl2(j))
        endif
enddo
cadv(1)=1.0

do j=1,nz
dis(j)=(v(j)*delz/pe)
lamda(j)=(dis(j)*delt/(delz**2*r1(j)))
enddo
do j=1,nz-1
        if (j.eq.1)  then
        ae(j)=0
        be(j)=1
        ce(j)=0
        re(j)=cadv(1)
        else
        ae(j)=    -lamda(j)
        be(j)=     1+(2*lamda(j))
        ce(j)=    -lamda(j)
        re(j)=cadv(j)
        endif
enddo
ae(nz)=  0
be(nz)=   1
ce(nz)=0.0
re(nz)=cadv(nz)
call tdma(ae,be,ce,re,variable,nz)

do j=1,(nz)
cdis(j)=variable(j)
enddo
do j=1,nz
cnew(j)=cdis(j)-((coef1+(coef2*kd*row/theta(j)))*cdis(j)*delt/r1(j))
enddo
cnew(1)=1.0
if (((time-delt) .le. timeobs(ll2)) .and. (timeobs(ll2) .lt. time))then
do j=1,maxzobs
```

```
                cdiff1=cnew(nelt(j))-cold(nelt(j))
                tdiff1=timeobs(ll2)+delt-time
                c1=cold(nelt(j))+((cdiff2/delt)*tdiff1)
                cdiff2=cnew(nelt(j)+1)-cold(nelt(j)+1)
                tdiff2=timeobs(ll2)+delt-time
                c2=cold(nelt(j)+1)+((cdiff2/delt)*tdiff2)
                if (nelt(j) .eq. 1)then
                cdiff=c2-c1
                zdiff=zobs(j)
                ccal(j,ll2)=c1+(cdiff*zdiff)/delz
                else
                cdiff=c2-c1
                zdiff=zobs(j)-((nelt(j)-1)*delz)
                ccal(j,ll2)=c1+(cdiff*zdiff)/delz
                endif
                enddo
                ll2=ll2+1
                endif
                do j=1,nz
          cold(j)=cnew(j)
                hold(j)=hnew(j)
                thetaold(j)=theta(j)
                enddo
                time=time+delt
        enddo
!       write(35,250)(z(j),cnew(j),j=1,nz)
!       250 format(1x,f8.3,f11.4)

        sum=0
if (codeobj .eq. 1)then
        do i=1,maxzobs
                do j=1,maxtobs
                sum=sum+(cobs(i,j)-ccal(i,j))**2
                objf=sum
                enddo
        enddo
else if(codeobj .eq. 2)then
        do i=1,maxzobs
                do j=1,maxtobs
                cnum=(cobs(i,j)-ccal(i,j))
                cden=cobs(i,j)
                sum=sum+((cnum/cden)**2)
                objf=sum
                enddo
        enddo
else if(codeobj .eq. 3)then
        do i=1,maxzobs
                do j=1,maxtobs
                cnum=(cobs(i,j)-ccal(i,j))
                cden=ccal(i,j)
                sum=sum+((cnum/cden)**2)
                objf=sum
                enddo
        enddo
endif
print*,objf
return
end
```

```
!**************************************************************************
!program for QR decomposition using House holder transformation for an unsymmetric matrice
subroutine householder(wj,qt1,nz,nm,maxunpar,itotal)
real wj(nz,nm),h(nz,nz),q(nz,nz),r2(nz,nm),w(nz),temp1(nz,nm),temp2(nz,nz)
real prod(nz,nm),qqt(nz,nz),qt1(nz,nz),r1(nz,nm)

nr=itotal
nc=maxunpar
do ii=1,nr
        do jj=1,nc
        r2(ii,jj)=0
        enddo
enddo
do ii=1,nr
        do jj=1,nc
        prod(ii,jj)=0
        enddo
enddo
do ii=1,nr
        do jj=1,nr
        qqt(ii,jj)=0
        enddo
enddo
do i=1,nc
sum=0
        do k=i,nr
                if(i .eq. 1)then
                sum=sum+wj(k,i)*wj(k,i)
                else
                sum=sum+r2(k,i)*r2(k,i)
                endif
        enddo
s=sum**(0.5)
        if(i .eq. 1)then
        den=(2.0*s*(s+abs(wj(i,i))))**(0.5)
        else
        den=(2*s*(s+abs(r2(i,i))))**(0.5)
        endif
        do k=1,nr
                if(i .eq. 1)then
                        if(k .lt. i)then
                        w(k)=0
                        else if(k .eq. i)then
                                if (wj(k,i) .gt. 0)then
                                w(k)=(wj(k,i)+s)/den
                                else if(wj(k,i) .eq. 0)then
                                w(k)=(wj(k,i))/den
                                else
                                w(k)=(wj(k,i)-s)/den
                                endif
                        else
                        w(k)=wj(k,i)/den
                        endif
                else
                        if (k .lt. i)then
                        w(k)=0
                        else if(k .eq. i)then
                                if(r2(k,i) .gt. 0)then
```

```
                                w(k)=(r2(k,i)+s)/den
                                else if(r2(k,i) .eq. 0)then
                                w(k)=r2(k,i)/den
                                else
                                w(k)=(r2(k,i)-s)/den
                                endif
                        else
                        w(k)=r2(k,i)/den
                        endif
                endif
        enddo

        !calculation of house holder matrix
        do j=1,nr
                do k=1,nr
                        if(k .eq. j)then
                        h(j,k)=1.0-2.0*w(j)*w(k)
                        else
                        h(j,k)=-2.0*w(j)*w(k)
                        endif
                enddo
        enddo

!       calculation of r matrix
        do ii=1,nr
                do jj=1,nc
                temp1(ii,jj)=0
                enddo
        enddo
        do j=1,nr
                do k=1,nc
                        do l=1,nr
                                if(i .eq. 1)then
                                temp1(j,k)=temp1(j,k)+h(j,l)*wj(l,k)
                                else
                                temp1(j,k)=temp1(j,k)+h(j,l)*r2(l,k)
                                endif
                        enddo
                enddo
        enddo
        do j=1,nr
                do k=1,nc
                r2(j,k)=temp1(j,k)
                enddo
        enddo
!       calculation of q matrix
        do ii=1,nr
                do jj=1,nr
                temp2(ii,jj)=0
                enddo
        enddo
        if (i .eq. 1)then
                do j=1,nr
                        do k=1,nr
                        temp2(j,k)=h(j,k)
                        enddo
                enddo
        else
```

```
                do j=1,nr
                        do k=1,nr
                                do l=1,nr
                                temp2(j,k)=temp2(j,k)+q(j,l)*h(l,k)
                                enddo
                        enddo
                enddo
        endif
        do j=1,nr
                do k=1,nr
                q(j,k)=temp2(j,k)
                enddo
        enddo
enddo
do i=1,nr
        do j=1,nr
        qt1(j,i)=q(i,j)
        enddo
enddo
do ii=1,nr
        do jj=1,nc
                do kk=1,nr
                r1(ii,jj)=r1(ii,jj)+qt1(ii,kk)*wj(kk,jj)
                enddo
        enddo
enddo
do ii=1,nr
        do jj=1,nc
                do kk=1,nr
                prod(ii,jj)=prod(ii,jj)+q(ii,kk)*r2(kk,jj)
                enddo
        enddo
enddo
do ii=1,nr
        do jj=1,nr
                do kk=1,nr
                qqt(ii,jj)=qqt(ii,jj)+q(ii,kk)*qt1(kk,jj)
                enddo
        enddo
enddo
return
end
!*******************************************************************************

 !program for Given's rotation
  subroutine givrot(itotal,maxunpar,nz,nm,wjnew,qt2)
real g1(nz,nz),wjnew(nz,nm),r3(nz,nm),temp3(nz,nm),temp4(nz,nz)
real q1(nz,nz),prod1(nz,nz),qt2(nz,nz),qqt1(nz,nz)

nr=itotal+maxunpar
nc=maxunpar
nin=itotal
do ii=1,nr
        do jj=1,nr
        prod1(ii,jj)=0
        qqt1(ii,jj)=0
        enddo
enddo
```

```
do ii=1,nr
        do jj=1,nc
        r3(ii,jj)=0
        enddo
enddo
do i=1,nc
        if (i .eq. 1)then
        den=sqrt(wjnew(i,i)*wjnew(i,i)+wjnew(i+nin,i)*wjnew(i+nin,i))
        cos=wjnew(i,i)/den
        sin=wjnew(i+nin,i)/den
        else
        den=sqrt(r3(i,i)*r3(i,i)+r3(i+nin,i)*r3(i+nin,i))
        cos=r3(i,i)/den
        sin=r3(i+nin,i)/den
        endif
        do j=1,nr
                do k=1,nr
                        if(j .eq. i)then
                                if(j .eq. k)then
                                g1(j,k)=cos
                                else if(k .eq. (j+nin))then
                                g1(j,k)=sin
                                else
                                g1(j,k)=0
                                endif
                        else if(j .eq. (i+nin))then
                                if(k .eq. (j-nin))then
                                g1(j,k)=-sin
                                else if(j .eq. k)then
                                g1(j,k)=cos
                                else
                                g1(j,k)=0
                                endif
                        else
                                if(j .eq. k)then
                                g1(j,k)=1
                                else
                                g1(j,k)=0
                                endif
                        endif
                enddo
        enddo
        do j=1,nr
                do k=1,nc
                temp3(j,k)=0
                enddo
        enddo
        do j=1,nr
                do k=1,nc
                        do l=1,nr
                                if(i .eq. 1)then
                                temp3(j,k)=temp3(j,k)+g1(j,l)*wjnew(l,k)
                                else
                                temp3(j,k)=temp3(j,k)+g1(j,l)*r3(l,k)
                                endif
                        enddo
                enddo
        enddo
```

```
        do j=1,nr
                do k=1,nc
                r3(j,k)=temp3(j,k)
                enddo
        enddo
        write(6,700)((r3(j,k),k=1,nc),j=1,nr)
        700 format (1x,3f15.5)
!       write (20,*) 'R2 matrix'
!       do j=1,nr
!       write(20,*)(r2(j,k),k=1,nc)
!       enddo
        do j=1,nr
                do k=1,nr
                temp4(j,k)=0
                enddo
        enddo
        if(i .eq. 1)then
                do j=1,nr
                        do k=1,nr
                        temp4(j,k)=g1(j,k)
                        enddo
                enddo
        else
                do j=1,nr
                        do k=1,nr
                                do l=1,nr
                                temp4(j,k)=temp4(j,k)+qt2(j,l)*g1(l,k)
                                enddo
                        enddo
                enddo
        endif

        do j=1,nr
                do k=1,nr
                qt2(j,k)=temp4(j,k)
                enddo
        enddo
!       write (20,*) 'QT given matrix'
!       do j=1,nr
!       write(20,*)(qt2(j,k),k=1,nr)
!       enddo
enddo
do ii=1,nr
        do jj=1,nr
                q1(ii,jj)=qt2(jj,ii)
        enddo
enddo
!write (20,*) 'Q matrix'
!do j=1,nr
!write(20,*)(q(j,k),k=1,nr)
!enddo
do j=1,nr
        do k=1,nc
                do l=1,nr
                prod1(j,k)=prod1(j,k)+q1(j,l)*r3(l,k)
                enddo
        enddo
enddo
```

```
!write (20,*) 'product matrix'
!do j=1,nr
!write(20,*)(prod(j,k),k=1,nc)
!enddo
do j=1,nr
        do k=1,nr
                do l=1,nr
                qqt1(j,k)=qqt1(j,k)+q1(j,l)*qt2(l,k)
                enddo
        enddo
enddo
!write (20,*) 'qqt matrix'
!do j=1,nr
!write(20,*)(qqt(j,k),k=1,nr)
!enddo
return
end
!**************************************************************************
subroutine gauss(nz,nm,maxunpar,wjfin,errfin,delp)
real wjfin(nz,nm),errfin(nz),delp(nm)
do i=maxunpar,1,-1
        if (i.eq. maxunpar)then
        delp(i)=-errfin(i)/wjfin(i,i)
        else
        sum=-errfin(i)
                do j=maxunpar,i+1,-1
                sum=sum-wjfin(i,j)*delp(j)
                enddo
        delp(i)=sum/wjfin(i,i)
        endif
enddo
return
end

!*****************************THOMAS ALGORITHM*********************************
subroutine tdma(ae,be,ce,re,variable,nz)
        real ae(nz),be(nz),ce(nz),re(nz),variable(nz)
  real alpha1(nz),bita(nz),y(nz),cte(nz)
                alpha1(1)=be(1)
                do j=1,nz-1
                bita(j)=ce(j)/alpha1(j)
                alpha1(j+1)=be(j+1)-(ae(j+1)*bita(j))
                enddo
                y(1)=re(1)/alpha1(1)

                do j=2,(nz)
                y(j)=((re(j)-ae(j)*y(j-1))/alpha1(j))
                enddo

                cte(nz)=y(nz)
                do j=(nz-1),1,-1
                cte(j)=y(j)-(bita(j)*cte(j+1))
                enddo
                do j=1,(nz)
                variable(j)=cte(j)
                enddo
        return
        end
```