# Development of an Equivalent Dispersion Coefficient for a Complex Contaminant Transport model

BY

Abid Maqsood Ahmad

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

## KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

# MASTER OF SCIENCE

In

CIVIL ENGINEERING

NOVEMBER 2002

UMI Number: 1414979

# UMI®

# KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

## DHAHRAN 31261, SAUDI ARABIA

## DEANSHIP OF GRADUATE STUDIES

This thesis, written by

### Abid Maqsood Ahmad

under the direction of his Thesis Advisor and approved by his Thesis Committee,

has been presented to and accepted by the Dean of the College of Graduate Studies,

in partial fulfillment of the requirements for the degree of

### MASTER OF SCIENCE IN CIVIL ENGINEERING

Thesis Committee

Dr. M. S. Al Suwaiyan (Chairman)

Dr. Rashid I. Allalya (Member)

Dr. Alladin A. Bukhari (Member)

Department Chairman
(Prof. H.I. Al-Abdul Wahhab)

Dean of Graduate Studies
(Prof. Osama Ahmad Jannadi)

Date

Dedicated to

My Parental & Maternal Grand Mothers

and Loving Parents

# Acknowledgements

All praise to Allah, the most beneficent and the most merciful, Who bestowed me with the strength to complete this work. I make a humble effort to thank Allah for his endless blessings on me, as his infinite blessings can not be thanked for. Then, I pray Allah to bestow peace on his last prophet Muhammed (*Sal-allah-'Alaihe-Wa-Sallam*) and all his righteous followers till the day of judgement.

# Contents

# List of Tables

# List of Figures

# THESIS ABSTRACT

**Name:**        Abid Maqsood Ahmad

**Title:**        Development of An Equivalent Dispersion Coefficient
for a Complex Contaminat Transport Model

**Degree:**        Master of Science

**Major Field:**     Civil Engineering

**Date of Degree:** November 2002

*The ability of artificial intelligence systems to facilitate the generation of solutions for difficult problems in civil engineering that require symbolic reasoning and efficient manipulation of diverse knowledge has generated a considerable interest recently. The process of contaminant transport in the subsurface environment is a very complex problem, which can benefit from such system. Several sophisticated theoretical models have been developed to predict the process of solute transport in porous media. Unknown parameters, which are extremely difficult to determine in the laboratory, are introduced with each developed model creating an added difficulty. In this study the parameters of a complex contaminant transport model are related to a single parameter named the Equivalent Dispersion Coefficient (EDC). This study also presents the development and design of artificial neural network (ANN) model that is able to predict the EDC for a contaminant transport model. The work consists of several tasks starting with generation of data from numerical simulations. Next the data is used to train the neural network models. Two learning algorithms, Back Propagation Algorithm and Levenberg Marquardt Algorithm are used to train the neural network models on a specific range of data. Once the training is complete these models are validated using synthetic as well as experimental data. The trained models give the EDC values when it receives the dimension less parameter data as their input. The EDC is used in linear equilibrium advective despersive equation to predict solute concentrations. The results obtained from this new approach are in good agreement with the previous studies. In addition to this, the advantage of the current approach is the automated process of obtaining EDC values.*

<div align="center">

**Master of Science Degree**
**King Fahd University of Petroleum and Minerals, Dhahran.**
**November 2002**

</div>

# " الملخــص "

الأسم: عابد مقصود أحمد

العنوان: تطوير معامل التشتت المكافئ لنموذج إنتقال الملوثات

الدرجة: الماجستير

المجال: الهندسة المدنية

تاريخ الرسالة: نوفمبر ٢٠٠٢ م


إن قدرة نظم الذّكاء الصّناعيّ لتسهيل إيجاد الحلول للمشاكل الصّعبة في الهندسة المدنيّة التي تتطلب التدبّر الرّمزيّ و المعالجة الفعّالة للمعرفة المختلفة قد نالت الكثير من الإهتمام مؤخراً. إن عمليّة إنتقال الملوث تحت سطح الأرض مشكلة معقدة جدًا ويمكنها أن تستفيد من مثل هذا النظام . طوّرَتْ عدة نماذج نظريّة متطوّرة للتّنبؤ بعمليّة نقل المواد المذابة في البيئة المسامية. مع كل نموذج متطور تظهر عوامل إضافية يكون تحديدها صعبا في المعمل. في هذه الدّراسة تم ربط عناصر نموذج نقل الملوث المعقد بمعامل وحيد يسمى معامل التسرب المتكافئ (EDC). هذه الدراسة تقدم أيضاً تطوير وتصميم شبكة عصبية صناعية و قادرة على التنبؤ بـ (ANN) لنموذج نقل الملوث. يكون العمل من عدة مهام تبدأ بجمع البيانات من محاكات الرقمية، بعد ذلك استخدمت البيانات لتدريب نموذج الشبكات العصبية باستخدام (BPA) و (LMA). بمجرد انتهاء عملية التدريب اجيزت النماذج باستخدام بيانات تجريبية وصناعية تعطي النماذج المدربة قيم (EDC) عندما تتلقى بيانات حدودها أقل من دخلها. يستخدم (EDC) في معادلة التأفك والإنتشار للتنبؤ بتركيز المواد المذابة. أظهرت النتائج التي احرزت من هذه الطريقة الجديدة نتائج قريبة من الدراسات السابقة. بالإضافة لهذا فإن ميزة الطريقة الحالية هي العمليّة الآلية للحصول على قيم (EDC).


**درجة الماجستر في العلوم**
**جامعة الملك فهد للبترول والمعادن**
**الظهران– المملكة العربية السعودية**
**نوفمبر– ٢٠٠٢م**

# Chapter 1

# INTRODUCTION

Water, is the lifeblood of every living creature on earth, covers approximately 70 percent of the earth's surface. Only about three percent of the world's water is fresh water, which is needed for most human activities and two third of that is frozen forming the polar ice caps and glaciers. The remaining one percent of the total world water supply is freshwater available as either surface water or groundwater. Groundwater accounts for more than two-third of this amount [12].

Groundwater is often wrongly thought of as an underground river or lake. which is only true in caves or within lava flows. In reality groundwater is usually held in porous soil or rocks, much the same way water is held in a sponge. When rain falls on the ground some of the water flows along the surface in streams or into lakes, some of it used by plants, some evaporate and returns to the atmosphere, and some sinks into the ground filling cracks and voids in the sub-surface and forming what is called the saturated zone. The top of this zone is called the water table. The water

may be only a foot below the ground's surface or it may be hundreds of feet deep. In arid regions like Saudi Arabia where surface water is scarce, groundwater becomes the main source of freshwater. The recent growth in municipal, agricultural and industrial sectors resulted in a huge increase in water demand, which is currently provided by the costly desalinated water as well as by mining groundwater.

Subsurface contamination is a serious problem that modern societies often face. Models that solve both flow and contaminant transport equations are frequently used to evaluate groundwater quality changes with space and time. To properly assess and predict groundwater contamination at a given site, detailed information about the nature of the suspected contaminants, the volume/mass of contaminants disposed or released into the surface, the time period over which contaminants were released and the areas on which contaminants were released is needed. The transport and fate of contaminants in groundwater are affected by a site specific physical, chemical, biological processes. These processes dictate the rate at which a chemical constituent spreads through soil. These mechanisms often act simultaneously on the chemical and may include such processes as advection, diffusion and dispersion, adsorption and reaction.

Traditionally, a two parameter partial differential equation has been used to model the one dimensional advective dispersive transport of chemicals in field soil [15]. However when comparing model predictions with experimental data some inadequacies were observed in the model predictions. Due to this inadequacy, more complex

conceptual models had to be introduced in order to better represent the real world system. These models are all based on the assumption that, either for physical or chemical reasons, adsorption does not proceed at an equal rate in all parts of the soil medium. The resulting transport equation contains several parameters which must be quantified before actual predictions can be made.

## 1.1 Literature Review

Concern about the fate of chemicals introduced into soil-water systems has recently intensified. It is often necessary to estimate the behaviour of a chemical in the field without substantial knowledge of the interaction of the chemical with the solid phase or its degradation rate. As new chemicals are proposed for future use, or as closer scrutiny is given to those already in use, it is necessary to utilize scientifically sound, comprehensive tools to evaluate the potential behaviour of these chemicals in the environment. Well-constructed tools in the form of models describing transport in soil water systems also serve the complementary purpose of increasing our understanding of basic processes affecting chemical fate.

Lapidus and Amundson [15], Brenner [4], Cleary and Adrian [6], Gershon and Nir[9], Lindstorm and Stone[16, 17], Lindstorm et al. [18], Marino [19, 20] show that much has been learned about the effects of diffusion, dispersion, advection and adsorption on the transport of chemicals in soils. Numerous models have been developed in order to describe the one-dimensional transport of chemicals in laboratory columns, as

well as in field soils. Many others will undoubtedly follow, such models are important because they continuously increase our understanding of the basic transport mechanisms involved and consequently improve our ability to predict the fate in field soils of diverse chemicals as nitrates, fertilizers, pesticides, heavy metals and radioactive waste materials. The models can provide valuable information about both the quality of groundwater and prediction of possible quality changes in the future.

## 1.1.1 Advection Dispersion Model for the Solute Transport

The equation most widely used to describe one-dimensional displacement through a non-sorbing medium is

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} \qquad (1.1)$$

where

$$C = \text{Solute Concentration, (ML}^{-3})$$

$$D = \text{Dispersion Coffecient, (L}^2\text{T}^{-1})$$

$$V = \text{Pore water velocity, (LT}^{-1})$$

$$t = \text{time, (T)}$$

$$x = \text{distance, (L)}$$

The dispersion coefficient D characterizes the dispersion of the solute during its movement and includes the effect of both molecular diffusion and mechanical dispersion. When chemical adsorption is considered, an additional term must be added

to equation (1.1) to account for the interaction between the chemical and the solid phase. This is accomplished by redefining equation (1.1) as;

$$\frac{\partial C}{\partial t} + \frac{\rho \partial s}{\theta \partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} \qquad (1.2)$$

where

$s$ = absorbed Concentration, $(MM^{-1})$

$\theta$ = Volumetric water content, $(L^3 L^{-3})$

$\rho$ = bulk density, $(ML^{-3})$

The solution of equation (1.2) will depend upon the relationship between the absorbed concentration s, and the solution concentration C.

## Advection

Advection is the mass transport due to the flow of water in which the solute is dissolved. The direction and rate of transport coincides with the groundwater. The velocity of advective transport is described by the Darcy equation

$$V = -\frac{K}{n_e}\frac{\partial h}{\partial l} \qquad (1.3)$$

where

$V$ = average linear velocity, $(L/T)$

$K$ = hydraulic conductivity, $(L/T)$

$n_e$ = effective porosity

$dh/dl$ = hydraulic gradient, $(L/L)$

## Hydrodynamic Dispersion

Dispersion occurs in porous medium because of two processes

1. Molecular Diffusion

2. Mechanical Disperison

$$D_L = \alpha_L V_i + D^* \qquad (1.4)$$

$$D_T = \alpha_T V_i + D^* \qquad (1.5)$$

where

$D_L$ = Longitudinal Hydrodynamic dispersion coefficient

$D_T$ = Transverse Hydrodynamic dispersion coefficient

$\alpha_L$ = Longitudinal dynamic dispersivity

$\alpha_T$ = Transverse dynamic dispersivity

A solute in water will move from an area of higher concentration to an area of lower concentration. This process is known as molecular diffusion. Diffusive flux is related to the concentration gradient as predicted by Fick's law which is expressed for a simple aqueous nonporous system as:

$$\widehat{J} = -D_d \, \widehat{\nabla}(C) \qquad (1.6)$$

$J$ = Chemical mass flux, $\dfrac{\text{moles}}{\text{L}^2\text{T}}$

$D_d$ = Diffusion Cofficient, $\dfrac{\text{L}^2}{\text{T}}$

$C$ = Concentration, $\dfrac{\text{moles}}{\text{L}^3}$

In porous media, diffusion cannot proceed as fast as it can in water because the ions must follow long pathways as they travel around grains. To account for this an effective diffusion coefficient, $D^*$ must be used

$$D^* = \omega D_d \qquad (1.7)$$

where

$$\omega = \text{Coefficient relateds to tortuosity}$$

Mechanical Dispersion is mixing caused by local variation in velocity around some mean velocity of flow. There are three basic causes of this phenomena

1. As fluid moves through the pores it will move faster in the center than along the edges

2. Some fluid particles will travel along longer flow paths in the porous media than other particles to go the same linear distance

3. Some pores are larger than others, which allows the fluid flowing through these pores to move faster.

## 1.1.2   Linear equilibrium Model

The relationship between the sorbed and solution concentration is described by a linear (or linearized) isotherm of the form

$$s = K_d C \qquad (1.8)$$

Where,

$$K_d \quad = \quad \text{empirical distribution coefficient} (M^{-1}L^3)$$

Subsitution of equation (1.8) into equation (1.2) gives the transport equation

$$R\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} \tag{1.9}$$

Where,

$$R \quad = \quad 1 + \frac{\rho K_d}{\theta} \tag{1.10}$$

If there is no interaction between the chemical and the solid phase, k in equation (1.10) becomes zero and R reduces to one. In some cases R may be become less than one, indicating that only a fraction of the liquid phase participates in the transport process. This occurs when the chemical is subject to anion exclusion (e.g. chloride movement in many fine textured soils), or when immobile liquid regions are present which do not contribute to advective solute transport (e.g, water inside dense aggregates or away from liquid-filled macro pores). In this case anion exclusion (1-R) can be viewed as the relative anion exclusion volume, and ($-k_d$) in equation 1.10 as the specific anion exclusion volume (e.g, expressed in $cm^3$ water per gram of soil).

The Linear Equilibrium model including the effects for the degradation and zero order production is given by Van Genuchten[28]. The soil system for the linear equilibrium model is shown in figure 1.1

$$R\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} - \mu C + \gamma(x) \tag{1.11}$$
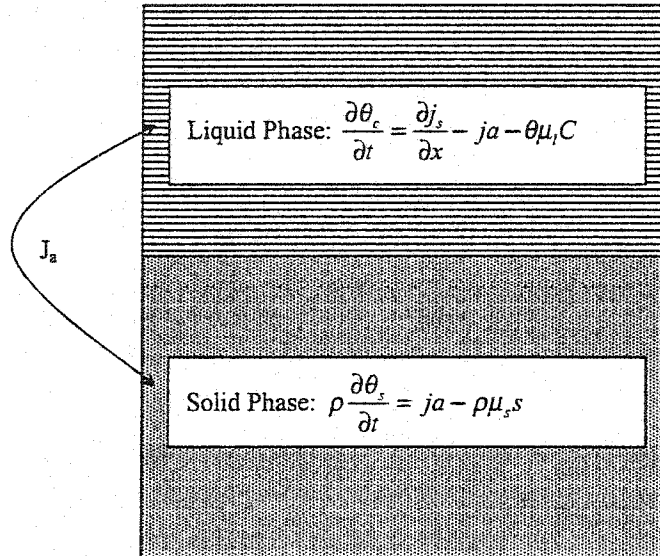
Figure 1.1: Schematic of the one-site equilibrium and kinetic sorption transport models with degradation

### 1.1.3  Physical Non-equilibrium Model

Equations (1.1) and (1.9) imply that all soil-water participates freely in the convective transport of chemicals, and that all adsorption sites are equally accessible for the solute if adsorption takes Place. Both equations furthermore, predict effluent curves which are characteristically sigmoidal or symmetrical in shape, at least for not too small values of P. Numerous experiments, both in field conditions and in laboratory, have shown serious deviations from these type of symmetrical distributions. Experimental curves frequently show a much earlier appearance of the chemical in the effluent than can be accounted for with solutions based on equation (1.1) or (1.9), while at the same time considerable more water is needed before the displacement is complete. Nielsen and Biggar [22], Desmedt and Wierenga [8] have shown that several experimental conditions seems to favor this accelerated transport followed by tailing, notable solute movement in unsaturated soils. According to Van Genuchten and Wierenga [29] solute movement through aggregated and undisturbed soils results in extreme tailing. Extreme tailing is also expected when cracked soils, or soils containing macropores, are leached under saturated conditions. Even in uniform, saturated soils, however, tailing may occur, especially when there is a strong interaction between the chemical and the solid phase.

Several attempts have been made to account for the observed asymmetry and tailing. Coats and Smith [7], Skopp and Warrick [25] have given one such approach, which involves the concept of solute transfer between mobile and immobile soil-water

phases. In this approach convective-dispersive solute transport is assumed to be confined only to mobile water phase. solute transfer between the mobile (dynamic) and immobile (stagnant) soil-water regions, furthermore, is assumed to be diffused controlled. The model discussed here is essentially is that of Coats and Smith [7], but using the notation of Genuchten and Wierenga [29]. The governing equtions for the mobile and immobile water phases, in the absence of solute adsorption, are

$$\theta_m \frac{\partial C_m}{\partial t} + \theta_{im} \frac{\partial C_{im}}{\partial t} = \theta_m D \frac{\partial^2 C_m}{\partial x^2} - \theta_{im} V_m \frac{\partial C_m}{\partial x} \tag{1.12}$$

$$\theta_{im} \frac{\partial C_{im}}{\partial t} = \alpha(C_m - C_i m) \tag{1.13}$$

Where the subscript $m$ and $im$ refer to mobile and immobile liquid regions, and where $V_m$ is the average pore water velocity in the mobile liquid phase:

$$V_m = \frac{q}{\theta_m} \tag{1.14}$$

$$V_m = \frac{V}{\phi_m} \tag{1.15}$$

In equation (1.14), q is the volumetric flux and $\phi_m$ the fraction mobile water:

$$\phi_m = \frac{\theta_m}{\theta} \tag{1.16}$$

$$\theta = \theta_m + \theta_{im} \tag{1.17}$$

The mass transfer coefficient, $\alpha$, in equation (1.13) determines the rate of exchange between the two liquid phases. The transport model assumes that this rate is proportional to the differences in concentrations between mobile and immobile soil-water phases. Equations (1.12) and (1.13) assumes that no adsorption occurs. Van

Genuchten and Wierenga [29] modified the equations to include the effects of chemical adsorption. They suggested the following set of differential equations:

$$\theta_m \frac{\partial C_m}{\partial t} + fp\frac{\partial s_m}{\partial t} + \theta_{im}\frac{\partial C_{im}}{\partial t} + (1-f)\rho\frac{\partial s_{im}}{\partial t} = \theta_m D\frac{\partial^2 C_m}{\partial x^2} - \theta_m V_m \frac{\partial C_m}{\partial x} \quad (1.18)$$

$$\theta_{im}\frac{\partial C_{im}}{\partial t} + (1-f)\rho\frac{\partial s_{im}}{\partial t} = \alpha(C_m - C_i m) \quad (1.19)$$

Where $s_m$ and $s_{im}$ are the absorbed concentrations in the dynamic and stagnant regions of the soil, both expressed per unit mass of soil assigned to these two soil regions, and where $f$ defines the mass fraction of soild phase assigned to the dynamic region.

Figure 1.2 shows schemetically the important fluxes in a "two region" soil system with decay. Equations (1.18) and (1.19) were derived with the assumption that adsorption around the larger liquid-filled pores is not necessarily the same as adsorption around the micropores in the stagnant region of the soil. When chemical moves through an unsaturated and/or aggregated soil, only part of the sorption site may be readily accessible for the chemical in the moving fluid. These sites may be located around the larger pores and in immediate contact with the mobile liquid. When an immobile liquid liquid phase is present, adsorption on the remaining part of the sorption sites can only occur after the chemical has diffused into this immobile liquid. The division of the sorption site into two fractions i.e., one fraction in close contact with the moving liquid, and one fraction away from the larger pores and in contact only with immobile (non-moving) water is characterized by the parameter

f. Total adsoption, s, is now given by

$$s = f s_m + (1 - f) s_{im} \qquad (1.20)$$

For equilibrium adsorption and assuming that the same linear equation (1.8) holds for both the dynamic and stagnant soil regions, Van Genuchten [32] gives the following set of equations:

$$(\theta_m + \rho f K)\frac{\partial C_m}{\partial t} + [\theta_{im} + (1 - f)\rho K]\frac{\partial C_{im}}{\partial t} = \theta_m D \frac{\partial^2 C_m}{\partial x^2} - \theta_m V_m \frac{\partial C_m}{\partial x} \qquad (1.21)$$

$$[\theta_{im} + (1 - f)\rho K]\frac{\partial C_{im}}{\partial t} = \alpha(C_m - C_{im}) \qquad (1.22)$$

The two region solute model including the effect of decay and production is given by Genuchten and Wagnet [33]

$$(\theta_m + f\rho K)\frac{\partial C_m}{\partial t} = \theta_m D_m \frac{\partial^2 C}{\partial x^2} - J_w \frac{\partial C_m}{\partial x} - \alpha(C_m - C_{im}) \qquad (1.23)$$

$$(\theta_m \mu_l + f\rho K \mu_{s,m})C_m + \theta_m \gamma_{l,m}(x) + f\rho \gamma_{s,m}(x)$$

$$[\theta_{im} + (1 - f)\rho K]\frac{\partial C_{im}}{\partial t} = \alpha(C_m - C_{im}) - [\theta_{im}\mu_{l,im} + (1 - f)\rho K \mu_{s,im}]$$

$$C_{im} + \theta_{im}\gamma_{l,im}(x) + (1 - f)\rho \gamma_{s,im}(x) \qquad (1.24)$$

$$\rho f \frac{\partial s_m}{\partial} = j_{a1} - \rho \mu_{sm} s_m$$

"Mobile" Soil Region

$J_{a1}$

"Mobile" Liquid Region

$$J_{a2} = \alpha(c_m - c_{im})$$

"Immobile" Liquid

$J_{a2}$

"Stagnant" Soil Region

$$\rho(1-f)\frac{\partial s_{im}}{\partial} = j_{a3} - \rho(1-f)\mu_{sim} s_{im}$$
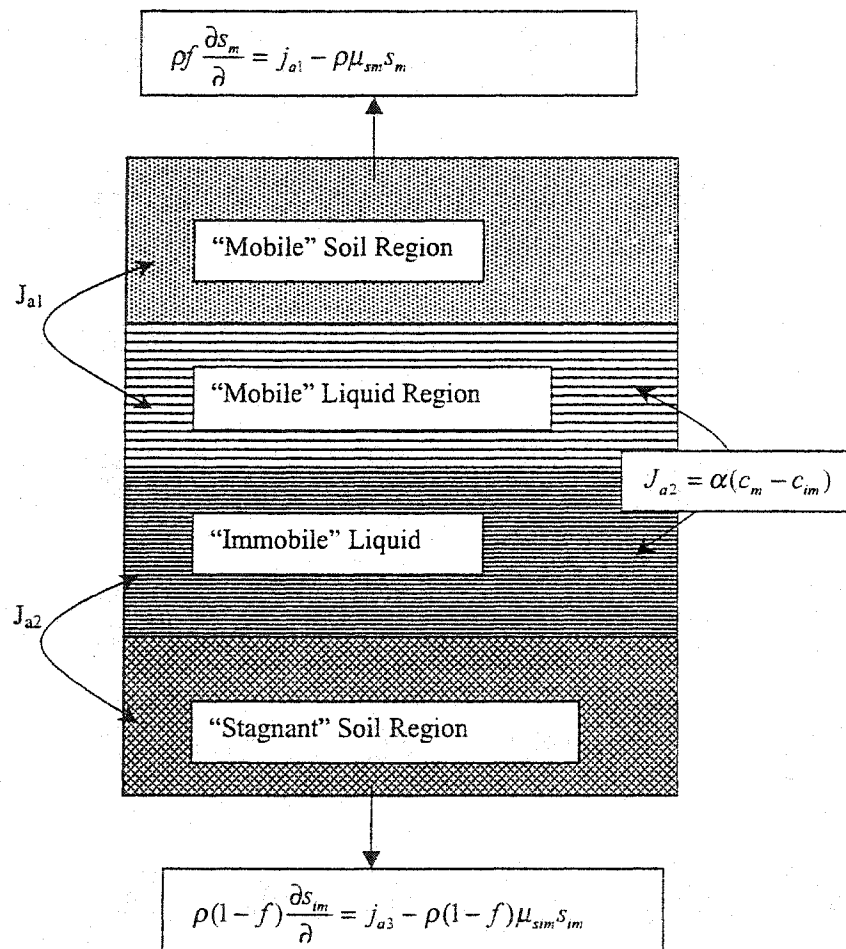
Figure 1.2: Schematic of the two-region (mobile-immobile) transport model with degradation

## 1.1.4 Physical Non-equilibrium Model and Anion Exclusion

Instead of being adsorbed, certain anions may interact with the solid phase of soil by being excluded from liquid zones adjacent to negatively charged soil particle surfaces (anion exclusion or negative adsorption). The anion exclusion model considered here differs from the physical non-equilibrium model in that the effects of anion exclusion rather than chemical adsorption are included in the governing transport equations. The soil water phase is again divided into mobile and immobile zones and anion exclusion is assumed to be restricted to the immobile water phase only i.e., to smaller-sized pores inside dense aggregates, or to immobile water along pore wall analogous to the situation described by Krupp et al. [14] Double layer theory suggests that the anion concentration within an individual pore increase roughly exponentially with the distance from the pore wall, at least for a freely extended diffuse double layer. Van Genuchten [32] assumed here that such a nonlinear concentration can be replaced by an equivalent step function which has a value of zero in the anion excluded part of the liquid phase adjacent to the pore walls, and a value equal to that of the bulk solution near the centre of the pore. This assumption leads to an equivalent exclusion distance, $d_{ex}$, near the pore walls in which concentration effectively remains zero as presented by Krupp et al.[14] The specific volume $V_{ex}$, is then simply

$$V_{ex} = d_{ex}A_o \qquad (1.25)$$

where

$$V = \text{specific volume}$$

$$A_o = \text{specific surface area,} \frac{\text{cm}^2}{\text{g}}.$$

The anion exclusion volume can also be expressed in terms of equivalent volumetric content, $\theta_{ex}$, by multiplying (1.24) with the soil bulk density, i.e:

$$\theta_{ex} = V_{ex}\rho \qquad (1.26)$$

Assuming $\theta_{ex} < \theta_{im}$, and that anion exclusion takes place only in the immobile soil water phase, then the following transport equations can be applied

$$\theta_m \frac{\partial C_m}{\partial t} + \theta_a \frac{\partial C_a}{\partial t} = \theta_m D \frac{\partial^2 C_m}{\partial x^2} - \theta_m V_m \frac{\partial C_m}{\partial x} \qquad (1.27)$$

$$\theta_a \frac{\partial C_a}{\partial t} = \alpha(C_m - C_a) \qquad (1.28)$$

Where the subscript a refers to that part of the immobile liquid phase that is not affected by anion exclusion:

$$\theta_a = \theta_{im} - \theta_{ex} \qquad (1.29)$$

equations (1.27) and (1.28) are very similar to (1.12) and (1.13) for the physical equlibrium model without adsorption, except that the immobile sink is reduced from $\theta_{im}$ to $\theta_a$.

The anion exclusion model described above is slighty different from the one given by Krupp et al. The model of Krupp et al. [14] was formulated by Van Genuchten[32] using their notations and it is given as;

$$\theta_m \frac{\partial C_m}{\partial t} + \theta_{im} \frac{\partial C_{im}}{\partial t} = \theta_m D \frac{\partial^2 C_m}{\partial x^2} - \theta_m V_m \frac{\partial C_m}{\partial x} \qquad (1.30)$$

$$\frac{\partial C_{im}}{\partial t} = K_r(C_m - \gamma C_{im}) \tag{1.31}$$

Where

$$K_r = \text{rate constant}$$

$$\gamma = \frac{\theta_{im}}{\theta_a} \tag{1.32}$$

The difference between the equation (1.27) and (1.28) and equation (1.30)and equation (1.31) is due to the fact that in the latter equations the concentrations $C_{im}$ is applied entire immobile region liquid phase, although still corrected for anion exclusion through the introduction of the parameter $\gamma$.

$$K_r = \frac{\alpha}{\theta_{im}} \tag{1.33}$$

$$C_{im} = \frac{C_a}{\gamma} \tag{1.34}$$

The anion exclusion models given in this section assume that $\theta_{ex}$ is always smaller than $\theta_{im}$, and that anion exclusion is restricted to the immobile liquid phase. Because convective transport takes place only in the mobile liquid phase which, at least in the present example, is not affected by anion exclusion, it follows that the mobile concentration, $C_m$, in the soil will never exceed the input concentration, $C_o$. This situation, however, becomes different when $\theta_{ex}$ also includes part of the mobile liquid. In that case the concentration of the non-excluded part of the liquid phase, whether it is mobile or immobile water, is likely to exceed at times $C_o$ inside the column, but not in the effluent.

## 1.1.5 Two-Site Kinetic Non-Equilibrium Model

The two site kinetic adsorption model described by Van Genuchten and Wierengaet [29] is the same model as discussed by Selim et al. [24] and Cameron and Klute [5]. Basic to the two site adsorption model is the idea that the solid phase of the soil is made up of different constituents (soil minerals, organic matter, aluminium and iron oxides), and that a chemical is likely to react with these different constituents at different rates and with different intensities. Figure1.3 shows schemitically a soil made up of the liquid pahse and solid phase. The two-site model assumes that the sorption sites can be divided into two fractions; adsorption on the one fraction (type-1 sites) is assumed to be instantaneous, while adsorption on the other fraction (type-2 sites) is thought to be time-dependent. At-equilibrium, adsorption on both types of sorption sites is described by liner equations:

$$s_1 = fK_dC \qquad (1.35)$$

$$s_2 = (1-f)K_dC \qquad (1.36)$$

Where the subscript 1 and 2 refer to type-1 and type-2 sites,respectively, and where f is the fraction of sites occupied by type-1 sorption sites. Total adsorption, s, is simply

$$s = s_1 + s_2 \qquad (1.37)$$

because type-1 sites are always at equilibrium it follows from (1.35) that

$$\frac{\partial s_1}{\partial t} = fK_d\frac{\partial C}{\partial t} \qquad (1.38)$$

The adsorption rate for the kinetic non-equilibrium (type-2) sites is given by a linear, reversible, first order rate equation of the form

$$\frac{\partial s_2}{\partial t} = \alpha(K_2 - s_2) \tag{1.39}$$

Where $\alpha$ is a first order rate coefficient. Combining equation (1.2)with the equation above leads to the transport model.

$$(1 + \frac{f\rho K_d}{\theta})\frac{\partial C}{\partial t} + \frac{\rho}{\theta}\frac{\partial s_2}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial c}{\partial x} \tag{1.40}$$

$$\frac{\partial s_2}{\partial t} = \alpha[(1 - f)K_d C - s_2) \tag{1.41}$$

The two-site non equilibrium model makes a distinction between type-1 (equilibrium) and type-2 (first order kinetic) adsorption as shown by Van Genuchten and Wagnet [33]. For a steady-state flow in a homogeneous soil, transport of linearly adsorbed solute with the effect of degradation and zero order production is given by,

$$[1 + \frac{f\rho K_d}{\theta}]\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} - \frac{\alpha\rho}{\theta}[(1-f)K_d C - s_2] \tag{1.42}$$
$$-\mu_l C - \frac{f\rho K_d \mu_{s,1} C}{\theta} + \gamma_l(x) + \frac{f\rho\gamma_{s,1}}{\theta}$$

$$\frac{\partial s_2}{\partial t} = \alpha[(1 - f)K_d C - s_2] - \mu_{s,2}s_2 + (1 - f)\gamma_{s,2}(x) \tag{1.43}$$

## 1.1.6   One-Site Kinetic Non-Equilibrium Model

The one site kinetic non-equilibrium adsorption model is a special case of the two-site adsorption model in that now all sorption sites are assumed to be time dependent

(type-2) sites . The parameter $f$ in the previous section is hence zero, and transport

equation reduces to

$$\frac{\partial C}{\partial t} + \frac{\rho}{\theta}\frac{\partial s_2}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial c}{\partial x} \qquad (1.44)$$

$$\frac{\partial s_2}{\partial t} = \alpha(K_d C - s) \qquad (1.45)$$
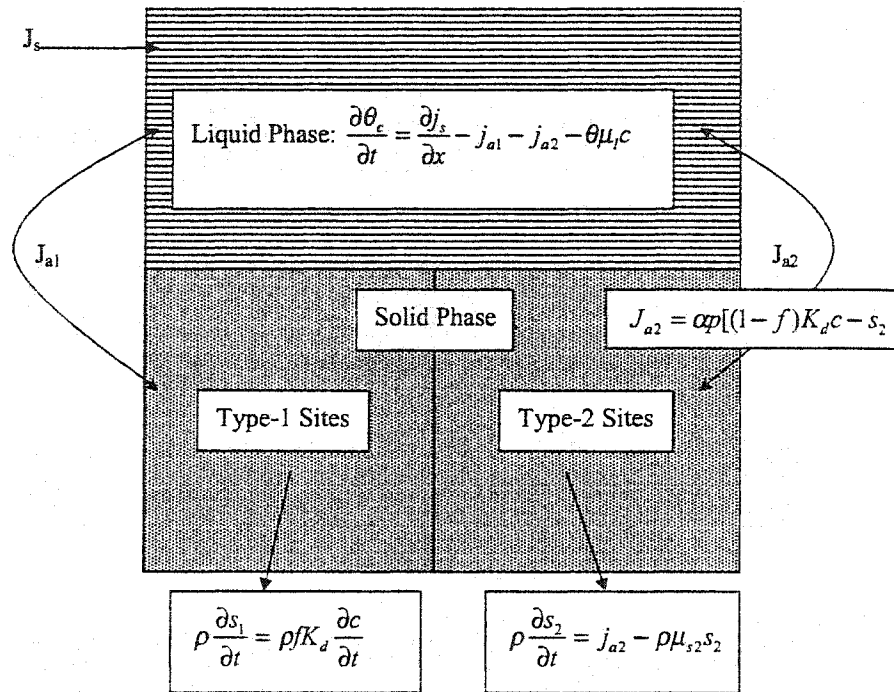
Figure 1.3: Schematic of the two-site partial equilibrium, partial kinetic sorption transport model with degradation

## 1.2 Objective of The Study

As is evident from the literature review the transport of contaminant in a sub-surface is a complex phenomena which involves several extremely difficult to quantify parameters. The objective of this work is to replace and relate the parameters of a complex transport model to a single parameter which will be called the Equivalent Dispersion coefficient (EDC). Once an EDC is obtained, one can simulate the actual field transport process using the EDC in the traditional two parameter transport model, instead of using the more complex models.

The main objectives of this work are as follows:

1. Relate the various parameters of the complex transport model to an equivalent dispersion coefficient.

2. Make use of Artificial Intelligence to come up with a model which is able to predict the values of EDC.

3. Prepare a numerical code based on the linear equilibrium model to simulate the solute transport using the EDC.

## 1.3 Approach of the Study

At the first stage of work, models/softwares will be used to generate data using the non-equilibrium contaminant transport model. The generated data will be fitted to the traditional contaminant transport model. Based on the fitted data, Equivalent

Dispersion Coefficient (EDC) for different data set will be found out. On the basis of this information, a model will be developed using Non-Linear Fitting parameter estimation procedure. The Neural Network techniques BackPropagation algorithm and Levenberg Marquardt lgorithm are used for non-linear fitting. Models will be developed in matlab, Which are capable of finding the equivalent dispersion coefficient. The developed models will be verified using numerical as well as experimental data. A numerical code is also developed to simulate the solute transport based on linear equilibrium model using EDC.

# Chapter 2

# MODEL DEVELOPMENT

This chapter presents the theoretical basis on which models based on Neural Network are developed. In the first section, the development of extensive tailing caused by solute moving through a non-sorbing porous media is discussed. This tailing is due to three main factors; Un-saturated flow, Aggregated Media, Pore Water Velocity. Later in this chapter, the complex contaminant transport models are divided into two groups, these models are introduced to better predict solute transport. One group explain tailing on the basis of physical processes whereas the other group describes it on the basis of chemical processes. Complex model parameters are introduced along with each complex contaminant transport model. These complex transport model parameters are described in dimensional and dimensionless form. In the end, a numerical model/technique is developed to simulate the solute transport using linear equilibrium model .

# 2.1 Solute Transport Through a Porous Media

## 2.1.1 Non-Sorbing Porous Media

The equation most widely used to describe one-dimensional displacement through a non-sorbing medium is [15]

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x}$$

(2.1)

where

$C$ = Solute Concentration, $(ML^{-3})$

$D$ = Dispersion Coffecient, $(L^2T^{-1})$

$V$ = Pore water velocity, $(LT^{-1})$

$t$ = time, $(T)$

$x$ = distance, $(L)$

Although analytical as well as numerical solutions of equation (2.1) predict symmetrical Breakthorugh curves (BTC). Many experimental studies have shown that the BTC will have some degree of skewness. Tailing has been observed under one or more of the following conditions:

1. **Un-saturated Flow:**

   Nielsen and Biggar [1] noted considerable tailing with decreasing water content at approximately the same flow velocity. They argued that under unsaturated

conditions the larger pores are eliminated for transport and the proportion of water, which does not readily move within soil, is increased. This water has been identified as dead, stagnant or immobile water as suggested by Coats and Smith [7]. Decreasing the water content increases the amount of air filled macro-pores which are dependent upon side wards diffusion in order to become saturated with the displacing solution. The more dead or stagnant water that is created, more tailing will occur. The idea of dead spaces, or "pockets", distributed uniformly along the flowing channels, has lead to Turner's model [27]. This model assumes that no advection occurs in the pockets and that soulte transfer in and out of these dead spaces occur by molecular diffusion only. In order to explain tailing during unsaturated flow, Biggar and Nielsen [3] suggested diffusion in and out of the immobile water films covering the porous material. They reasoned that since the transfer of the chemical into these films was by diffusion, it is slow and likely to cause tailing. This effect is called stagnant film effect. Since the relative amount of film water increases with decreasing water content. However, Coats and Smith [7] calculated that even for high flow velocities, diffusion into film covering the porous medium would be an instantaneous type of process, incapable of yielding significant tailing.

2. **Aggregated Media:**

Soils are composed of slowly and rapidly conducting pore sequences. Ag-

gregates have many micro-pores in which displacement is dependent upon diffusion, while advection in the smaller pores is usually negligible. This result in slow and incomplete mixing and hence tailing, even under saturated conditions.

3. **Pore-Water Velocity:**

The pore-water velocity also appears to influence the amount of tailing. Some experiments indicate that tailing effects become more pronounced with decreasing velocity. Biggar and Nielsen [2] observed a small effect of pore water velocity on the degree of tailing with both glass beads and Aiken clay loam, using choloride tracer.

From the discussion above it is evident that for more soils the simple advective dispersive equation (2.1), yielding nearly sigmoid or symmetrial breakthrough curves, will not give an accurate description of the experimental data.

## 2.1.2  Sorbing Porous Media

When a sorbing porous media is considered, an additional term must be added to left hand side of equation (2.1) in order to account for the interaction between the chemical and the medium.

$$\frac{\partial C}{\partial t} + \frac{\rho \partial s}{\theta \partial t} = D \frac{\partial^2 C}{\partial x^2} - V \frac{\partial C}{\partial x} \tag{2.2}$$

Equation (2.2) is an extension of equation (2.1) in which only a sorption term is added and as such it may be expected to have limitations when describing the

solute movement through an un-saturated, aggregated sorbing porous medium as equation.

As discussed in the previous chapter, most of the studies on the solute transport, especially when applied to pesticide movement, were based on linear equilibrium equations as given by Kay and Elrick [13]. In nearly all of these studies, serious deviations were observed between calculated and experimental effluent curves. It was often not possible to predict the early arrival of the applied chemical in the effluent as well as the generally skewed shape of the observed curves. The introduction of experimentally determined nonlinear equilibrium adsorption relation did, at least in some cases lead to better predictions, especially when observed hysteresis phenomena in the adsorption-desorption isotherms were taken into account by Van Genuchten et al. [30]. In general, however, predictions based on equilibrium adsorption models were found to be inadequate.

In attempts to improve the predictions, equilibrium and kinetic non-equilibrium models were introduced. Although the kinetic non-equilibrium models resulted in some improvement in predictive capabilities at relatively low pore-water velocity. In the last few years, several studies have focused on the description of asymmetrical (skewed) and non-sigmoidal concentration distribution as explained by Van Genuchten and Wierenga [29]. At least two groups of models have been proposed to explain and predict tailing.

In one group of models, Van Genuchten and Wierenga [29], Skopp and Warrick

[25] explained tailing on the basis of physical processes such as the presence of distinct mobile and immobile water soil-water region. Convective solute transport in these models is assumed to occur only in mobile soil-water phase, while adsorption in a stagnant region of the soil is controlled by diffusion through the immobile (non-moving) fraction of the soil-water phase. This situation is primarily a physical problem insofar as the physical make-up of the soil is responsible for the presence of the relatively immobile water. Nielsen and Biggar [22] explained that the physical non-equilibrium situation can also occur in systems where the chemical is not subject to adsorption, notable in highly aggregated soils or soils that contain many liquid-filled macropores. In other group of models, Selim et al. [24], Cameron and Klute [5] explained tailing in effluent curves on basis of chemical processes by assuming the presence of a two-site adsorption mechanism. In this approach adsorption on one fraction of the sorption sites is assumed to be instantaneous, while adsorption on the remaining sites is thought to be time-dependent.

## 2.2   Complex Model Parameters

With the introduction of more complex models, aimed to better simulate the transport process, a new problem arises which is the estimation of various new parameters which now appear in the governing transport equation. For example, the more involved physical non-equilibrium and two-site adsorption models each contain parameters i.e. mobile water content ($\theta_m$), immobile water content ($\theta_{im}$), fraction of

a site $(f)$ and mass transfer coefficient $(\alpha)$ which must be quantified before the transport equations can be used to simulate solute movement. The complex model parameters are shown in table 2.1. They are usually presented in a dimensionless form in the complex models and are shown in table 2.2. Peclet number $(P)$ can be viewed approximately as the ratio of the residence times for diffusive $(L^2/Dm$ or $L^2/D)$ and advective transport $(L/V_m$ or $L/V)$. The retardation factor $(R)$ reflects the effects of adsorption during the transport through the soil. The parameter $(\beta)$ describes the maximum degree of non-equilibrium in the system, either in a physical or chemical-kinetic sense. Finally, the mass transport coefficient $(\omega)$ describes the rate at which equilibrium is obtained from the initial non-equilibrium situation. [23].

As was indicated earlier the purpose of this work is to present a technique/model

| Parameter | Two-Site Model | Two Region Model |
|-----------|----------------|------------------|
| $D$ | $VL/P$ | $VL/P$ |
| $K_D$ | $[\theta(R-1)]/\rho_b$ | $[\theta(R-1)]/\rho_b$ |
| $\phi_m = \theta_m/\theta$ | | $\beta R - f(R-1)$ |
| $f$ | $(\beta R - \phi_m)/(R-1)$ | $(\beta R - \phi_m)/(R-1)$ |
| $\alpha$ | $(\omega v)/[(1-\beta)RL]$ | $\omega q/L$ |

Table 2.1: Expression for the original parameters in terms of the dimensionless parameters P,R,$\beta$, $\omega$

such that the parameters of complex contaminant transport model are related to a single parameter called the equivalent dispersion coefficient (EDC). A neural network approach to this problem seems very natural since the neural networks are an ideal tool for solving problems where we know that a certain set of starting data produces particular results, but we have no knowledge of the actual relations (i.e. a

| Parameter | Two-Site Model | Two Region Model |
|-----------|----------------|------------------|
| $T$ | $Vt/L$ | $Vt/L$ |
| $Z$ | $x/L$ | $x/L$ |
| $P$ | $VL/D$ | $V_m L/D_m = VL/D$ |
| $R$ | $1 + \rho K_D/\theta$ | $1 + \rho K_D/\theta$ |
| $\beta$ | $\theta + f\rho K_d/\theta + \rho K_d$ | $\theta_m + f\rho K_d/\theta + \rho K_d$ |
| $\omega$ | $\alpha(1-\beta)RL/V$ | $\alpha L/\theta V$ |

Table 2.2: Expression for the dimensionless parameters P,R,$\beta$, $\omega$ for the non-equilibrium ADE

detailed physical model) that connect them.

## 2.3  Numerical Modeling

The Linear Equilibrium Model for a solute transport in soil is given by equation 2.1. Which assumes steady state water flux, a constant soil water content, and no interaction between the chemical and the solid phase. When chemical adsorption is considered an additional term must be added as shown in equation 2.2, Which account for interaction between the chemical and solid phase. The initial condition for this study is

$$C(x,0) = C_i \tag{2.3}$$

For the upper Boundary of the soil column (x = 0): a third type, constant flux boundary condition of the form is applied

$$\left(-D\frac{\partial C}{\partial x} + VC\right)\big|_{x=0} = VC_0 \tag{2.4}$$

Where $C_0$ is the concentration of the input solution. For the lower boundary, the following condition can be applied

$$\frac{\partial C}{\partial x}(\infty, t) = 0 \qquad (2.5)$$

The condition assumes the presence of a semi-infinite column.

## 2.3.1 Numerical Technique

A preliminary idea about the task of a numerical method can be obtained by considering a flow situation. A grid is drawn to cover the flow domain as shown in figure 2.1. With a sufficiently fine grid, the complete distribution of the relevant variables can be expressed in terms of their values at the grid points. Thus, the task of a numerical method is to evaluate concentration, velocity, etc. at the chosen grid points. From the differential equations algebraic equations are derived for the grid-point values of the variables.

## 2.3.2 Control Volume Approach

The particular practice that was chosen here for the derivation of the discretization equations is the control volume approach. The calculation domain is divided into sub-domains or control volumes such that there is one control volume around a grid point. The differential equation is integrated over a control volume to yield the discretization equation. Thus, the discretization equation represents the same conservation principle over a finite region as the differential equation over an in-
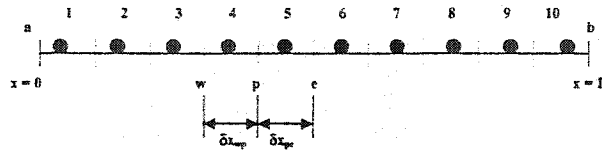
Figure 2.1: Computational grid

finitesimal region. This direct interpretation of the discretization equation makes

the method easy to understand in physical terms; the coefficients in the equation can

be identified even when they appear in the computer program, as familiar quantities

such as concentration, mass fraction etc[34].

## 2.3.3  Discretization

The partial differential equation is integrated over the control volume, with the aid

of assumptions about the relations between the nodal values at P and the rates of

production/decay of this entity within the cells and its transport by advection and

dispersion across the cell boundaries. For the purpose of solution the flow domain

is overlaid with a grid whose centre points or nodes denote the location at which all

variables are calculated. The nodes of a typical grid are labeled as P, (e;w) which lies on boundary of constant y or x.

## 2.3.4   Steady Advection Dispersion Equation

The general partial advection dispersion equation governing the solute transport is presented by the following advective dispersive equation:

$$\frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}(\Gamma \frac{\partial \phi}{\partial x}) + S_\phi \tag{2.6}$$

Formal integration over a control volume gives

$$\int_A n.(\rho u \phi) dA = \int_A n.(\Gamma \frac{\partial \phi}{\partial x}) dA + \int_{CV} S_\phi dV \tag{2.7}$$

The equation represents flux balance in a control volume. The left hand side gives the net advective flux and the right hand side contains the net dispersive flux and the production or decay of the property $\phi$ within the control volume.

## 2.3.5   Transient Advection Dispersion Equation

The conservation law for the transport of a unsteady advective dispersive flow has the general form:

$$\frac{\partial}{\partial t}(\rho \phi) + \frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}(\Gamma \frac{\partial \phi}{\partial x}) + S_\phi \tag{2.8}$$

The first term of the equation represents the rate of change and is zero for steady flows. The finite volume integration of equation 2.8 over a control volume (CV) must be augmented with a further integration over a finite time step ($\delta t$). By replacing

the volume integrals of the advective and dispersive terms with surface integrals and changing the order of integration in the rate of change term, the equation is written in this form:

$$\int_{CV}(\int_t^{t+\delta t} \frac{\partial}{\partial t}(\rho\phi)dt)dV + (\int_t^{t+\delta t}(\int_A n(\rho u\phi)dA)dt \qquad (2.9)$$
$$= (\int_t^{t+\delta t}(\int_A n(\Gamma\frac{\partial\phi}{\partial x})dA)dt + \int_t^{t+\delta t}\int_{CV} S_\phi dV dt$$

It can be written in this form:

$$\int_{CV}(\int_t^{t+\Delta t} R\frac{\partial}{\partial t}(\rho\phi)dt)dV + (\int_t^{t+\Delta t}(\int_A n(\rho u\phi)dA)dt \qquad (2.10)$$
$$= (\int_t^{t+\Delta t}(\int_A n(\Gamma\frac{\partial\phi}{\partial x})dA)dt + \int_t^{t+\Delta t}\int_{CV} S_\phi dV dt$$

The fully implicit discretisation equation is

$$a_p\dot\phi_p = a_w\phi_w + a_e\phi_e + a_p^o\phi_p^o + S_u \qquad (2.11)$$

where

$$a_p = a_w + a_w + a_p^o + \Delta F - S_p \qquad (2.12)$$

$$a_p^o = \frac{R\rho_p^o\Delta V}{\Delta t} \qquad (2.13)$$

$$S\Delta V = S_u + S_p\phi_p \qquad (2.14)$$

In our case we have the equation in the from of Mass fraction "m", and the equation is written as

$$\rho R\frac{\partial m}{\partial t} = \frac{\partial}{\partial x}(\rho D_{AB}\frac{\partial m}{\partial x}) - \rho u\frac{\partial m}{\partial x} \qquad (2.15)$$

A central differencing scheme is used to fully discretized the equation 2.15. It is convenient to define variables "F" and "D" to represent the advective mass flux per

unit area and dispersive conductance at cell faces:

$$F = \rho u \ and \ D = \Gamma/\delta x \qquad (2.16)$$

$$F_w = (\rho u)_w \ and \ F_e = (\rho u)_e \qquad (2.17)$$

$$D_w = \Gamma_w/\delta x_{wp} \ and \ D_e = \Gamma_e/\delta x_{pe} \qquad (2.18)$$

The integrated advection-dispersion equation can be written as:

$$\frac{R\rho(m_P - m_P^0)\Delta V}{\Delta t} + F_e m_e - F_w m_w = D_e(m_e - m_p) - D_w(m_p - m_w) \qquad (2.19)$$

For the first Node:

$$[a_E + a_P^o + (F_e - F_w) - \{-(D_a + F_w)\}]m_p = a_e\phi_e + a_p^o\phi_p^o + (D_a + F_a)m_a \qquad (2.20)$$

For the Last Node:

$$[a_w + a_P^o + (F_e - F_w) - (F_e - F_w)\}]m_p = a_w\phi_w + a_p^o\phi_p^o + F_b m_b \qquad (2.21)$$

| Node | $a_W$ | $a_P$ | $S_P$ | $S_U$ |
|---|---|---|---|---|
| First | 0 | $D_e - F_e/2$ | $-(2D+F)$ | $(2D+F)m_a$ |
| Internal | $(D_w + F_w)/2$ | $(D_e + F_e)/2$ | 0 | 0 |
| Last | $(D_w + F_w)/2$ | 0 | $(F_e - F_w) = 0$ | $F_b m_b$ |

Table 2.3: The coefficients at the first, internal and last node determined using method of central difference

## 2.3.6    Solute Transport Numerical Algorithm

To simulate the solute transport, the following steps are followed and a program was developed whose flow chart is shown in fig 2.2. The program code was attached in appendix D.

1. Define the Variables

2. Determine the grid points

3. All the grid points of discretized advection dispersion equation 2.11 are calculated using the control volume approach.

4. Using discretized advection dispersion equation 2.11 initial and boundary conditions mass fraction is calculated at each node for next time step.

5. In this manner we can keep advancing and can calculate mass fraction history.
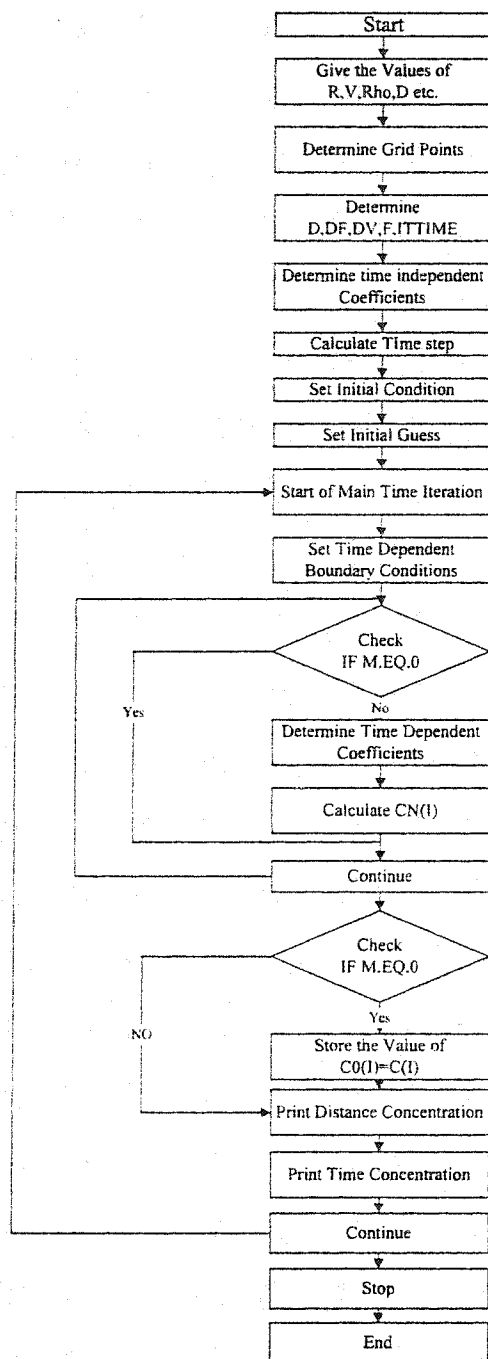
Figure 2.2: Flow chart for solute transport algorithm

# Chapter 3

# NEURAL NETWORKS

Artificial intelligence in general and neural networks specifically can be used to solve problems in civil engineering. The key in using artificial neural networks in civil engineering, or in any other discipline for that matter, is to observe, recognize and define problems in a away that they will be addressable by neural nets. It is obvious that the neural network is not magic-potion for the civil and environmental engineers, but it very well may help solve problems that conventional computing has not been successful in solving.

## 3.1   Short History of Neural Networks

The history of artificial neural networks is filled with colorful, creative individuals from many different fields, many of whom struggled for decades to develop concepts that we now take for granted. Neural Networks research can be tracked back to a

1940s paper by McCulloch and Pitts [21]. They showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function. Their work is often acknowledged as the origin of the neural net-work field. In 1957, Rossenblatt [21] invented the perceptron. The perceptron is the simplest form of a neural network used for the classification of a special type of patterns said to be lineraly seprable. The single-layer perceptron shown in figure 3.1 has a single neuron. Such
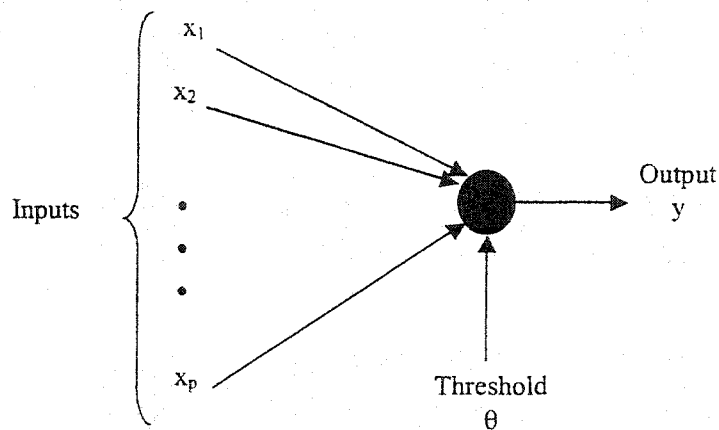


Figure 3.1: Single layer perceptron

a perceptron is limited to performing pattern classification with only two classes. He proved that, given linearly separable classes, a perceptron would, in a finite number of training trials, develop a weight vector that separates the classes (a pattern classification task). He also showed that starting values weight does not affect his proof. At approximately the same time, widrow [21] developed a similar network called

adeline. Minskey and Papert [21] pointed out that the perceptron theorem obviously applies to those problems that the structure is capable of computing. They showed that elementary calculation, such as exclusive or problems, cannot be solved by single-layer perceptrons. Rosenblatt [21] also studied structures with more layers and believed that they could overcome the limitations of simple perceptrons. However, no learning algorithm was known that could determine the weights necessary to implement a given calculation. Minskey and Papert [21] doubted that one could be found, and recommend that other approaches to artificial intelligence should be pursued.

Following this discussion, most of the computer science community left the neural-network paradigm for twenty (20) years. In the early 1980s Hopfield [21] revived neural network research. His efforts coincided with the development of new learning algorithms, such as backpropagation. The growth in neural-network research and applications has been phenomenal since the revival. Many of the advances in neural networks have had to do with new concepts, such as innovative architecture and training rules. Just as important has been the availability of powerful new computers on which to test these new concepts.

## 3.2 Applications

Neural networks have been applied in many fields since they were first invented. In the aerospace industry their application encompasses high performance aircraft

auto-pilots, flight path simulations, aircraft control systems, auto-pilot enhancements, aircraft component fault detectors. In the banking systems neural nets have been extensively used to read checks, documents, and use for credit application evaluation. The neural network application in the defense system has been phenomenal especially in weapon steering, target tracking, and object discrimination. In oil and gas industry it has been applied in the exploration field. The money that has been invested in neural network software and hardware, and the depth and breadth of interest in these devices have been growing rapidly.

## 3.2.1 Neural Network Application in Civil Engineering

The usual civil engineering tasks are: analysis, design, system identification, diagnosis, prediction/estimation, control, planning and scheduling. However all these problems appear to be one of two more abstract types of problem:

1. Casual modeling (mapping from cause to effect for estimation and prediction).

2. Inverse mapping from effects to possible causes.

Engineers have in the past used variety of tools for performing both casual modeling and inverse mapping. This set of tools includes statistics, regression, probabilities, optimization, rules of thumb and knowledge-based systems, and others. The very nature of neural network is to map from one space patterns (i.e., input patterns) to a space of output patterns (i.e., the output patterns). As such, an artificial neural network is thus another tool that can be used for determining such casual models

and inverse mappings.

## 3.3 Methodology

This section covers the historical background of technology, provides definitions of virtual intelligence and artificial neural networks, and offer more general information on the nature and mechanism of the artificial neural network and its relation to biological networks.

Virtual intelligence has been referred to by different names. Among these are artificial intelligence, computational intelligence, and soft computing. There seems to be no uniformly acceptable name for this collection of analytic tools among the researchers and practitioners of the technology. Of these, artificial intelligence is used the least as an umbrella term because artificial intelligence has historically been referred to as rule-based expert systems and today is used synonymously with expert systems. Expert systems made many promises of delivering intelligent computers and programs but did not fulfill these promises. Many believe that soft computing is the most appropriate term to use and that virtual intelligence is a subset of soft computing. [21].

Virtual intelligence may be defined as a collection of new analytic tools that attempt to imitate life [21]. Virtual intelligence techniques exhibit an ability to learn and deal with new situations. Artificial neural networks, evolutionary programming, and fuzzy logic are among the paradigms that are classified as virtual intelligence. These

techniques possess one or more attribute of reason such as generalization, discovery, association, and abstraction [21]. In the last decade, virtual intelligence has matured to a set of analytic tools that facilitate solving problems that were previously difficult or impossible to solve. The trend now seems to be the integration of these tools with each other as well as with conventional tools, such as statistical analysis, to build sophisticated systems that can solve challenging problems.

An artificial neural network is an information processing system that has certain performance characteristics in common with biological neural networks. Therefore at first biological neural network is described before offering a detailed definition of neural networks.

## 3.3.1 Biological Inspiration

This section briefly describes those characteristics of brain function that have inspired the development of artificial neural networks. The brain consists of a large number (approximately $10^{11}$) of highly connected elements (approximately $10^4$ per element) called neurons. For our purposes these neurons have three principle components;

1. The dendrites

2. cell body

3. axon

The dendrites are tree-like receptive networks of nerve fibers that carry electrical signals into cell body. The cell body effectively sums and threshold these incoming signals. The axon is a single long fiber that carries the signal from the cell body out to other neuron. The point of contact between axon of one cell and dendrite of another cell is called a synapse. It is the arrangement of neurons and the strengths of the individual synapses, determined by a complex chemical process, that establish the function of neural network. Figure 3.2 is a simplified schematic diagram of two biological neurons. Some of the neural structure is defined at birth. Other parts are developed through learning, as new connections are made and others waste away. This development is most noticeable in the early stages of life. For example it has been shown that if a young cat is denied use of one eye during a critical window of time, it will never develop normal vision in that eye. Neural structure continue to change throughout life. These later changes tend to consist mainly of strengthening or weakening of synaptic junctions. For instance, it is believed that new memories are formed by modification of these synaptic strengths. Thus, the process of learning a new friends face consists of altering various synapses. Artificial neural networks do not approach the complexity of the brain. There are, however, two key similarities between biological and artificial neural networks.

1. The building blocks of both networks are simple computational devices (although artificial neural networks are much simpler than biological neurons) that are highly interconnected.
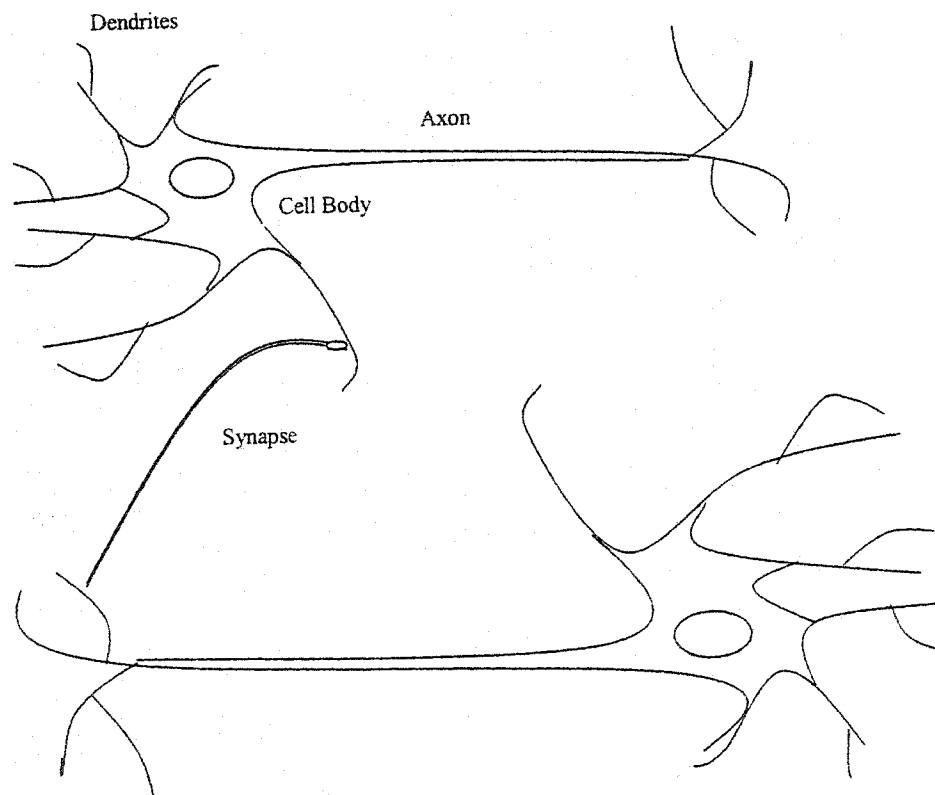
Figure 3.2: Diagram of two biological neurons

2. The connections between neurons determine the function of the network.

It is worth noting that even though biological neurons are very slow when compared to electrical circuits ($10^{-3}$ compared to $10^{-9}$sec), the brain is able to perform many tasks much faster than any conventional computer. This is in part because of massively parallel structure of biological neural networks: all of the neurons are operating at the same time. Artificial neural networks share this parallel structure. Even though most artificial neural networks are currently implemented on conventional digital computers, their parallel structure makes them ideally suited to implementation using Very large scale integration chips (VLSI), optical devices and parallel processes.

Artificial neural networks are information-processing systems that are a rough approximations and simplified simulation of this biological process and have performance characteristics similar to those of biological neural networks. They have been developed as generalizations of mathematical models of human cognition or neural biology based on the following assumptions:

1. Information processing occurs in many simple elements that are called neurons (processing elements).

2. Signal are passed between neurons over connecting links.

3. Each connecting link has an associated weight, which , in a typical neural network, multiplies the signal being transmitted.

4. Each neuron applies an activation function (usually nonlinear) to its net input
   to determine its output.

Figure 3.3 is a schematic of a typical neuron (processing element) in an artificial
neural network.

## 3.4  Mechanics of Neural Network Operation

An artificial neural network is a collection of neurons that are arranged in specific
formations. Neurons are grouped into layers. A multi-layer network usually consists
of an input layer, one or more hidden layers, and an output layer. The number of
neurons in the input layer corresponds to the number of parameters that are being
presented to the network as input. The same is true for the output layer. Neural-
network analysis is not limited to a single output and that neural nets can be trained
to build neuron-models with multiple outputs. The neurons in the hidden layer
or layers are responsible primarily for feature extraction. They provide increased
dimensionality and accommodate such tasks as classification and prediction. Figure
3.3 is a schematic diagram of a fully connected, three layer neural network. Neural-
network scientists and practitioners have classified the many kinds of neural networks
that exist. One of the most popular classifications is based on training methods:
supervised and un-supervised. Un-supervised neural networks, also known as self
organizing maps, are mainly clustering and classification algorithms. Supervised
learning requires training data which has been labeled with the desired outcome for
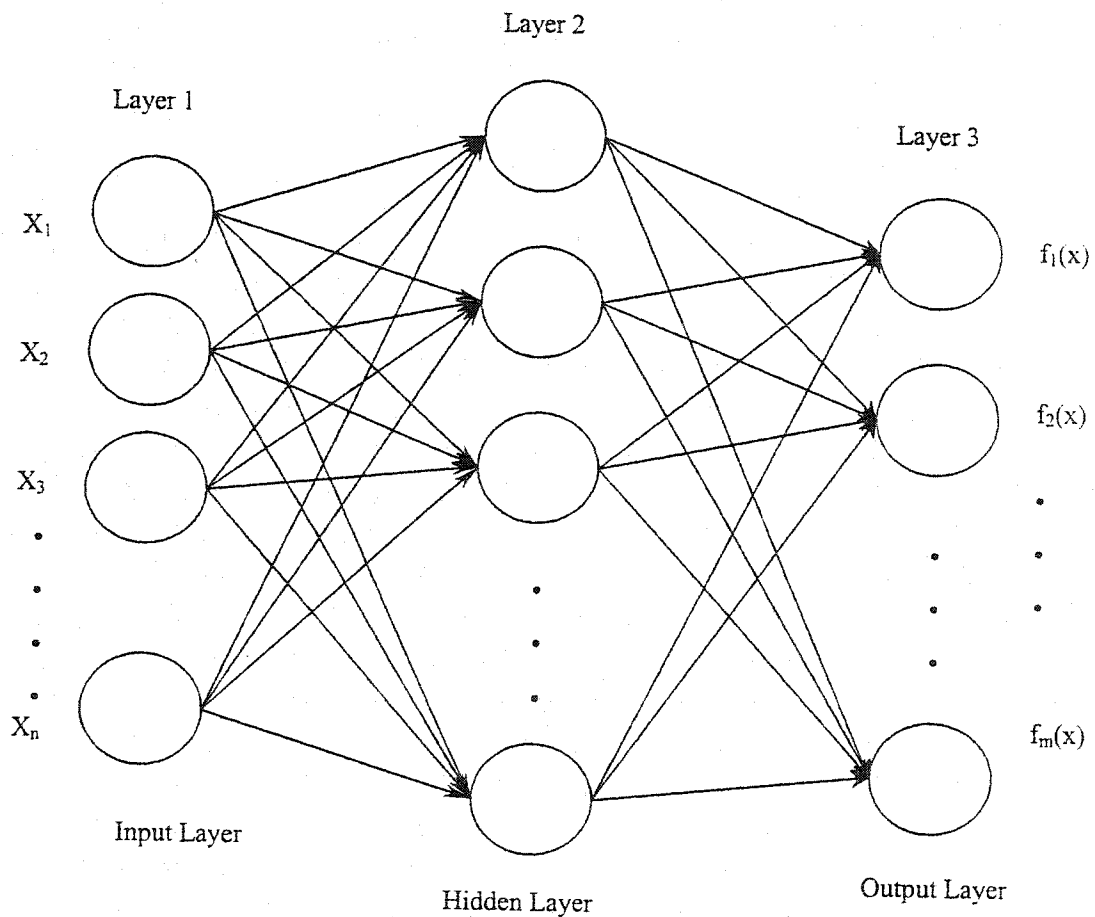
Figure 3.3: Topology of a multi-layer feed forward neural network. The topology of the network is commonly designated as (n-j-k) where n, j, and k represent the number of elements in the first, second and third layer in the network respectively

each pattern of inputs. Unlike traditional statistical classification techniques, such as discriminant analysis, neural networks can output both *discrete* and *continuous* data.

## 3.5  Learning Algorithm

### 3.5.1  Backpropagation Algorithm

The backpropagation (BP) algorithm is the most widely used learning procedure for supervised neural nets. Before beginning training, some small random numbers are usually used to initialize each weight on each connection. BP requires preexisting training patterns, and involves a forward-propagation step followed by a backward-propagation step. The forward-propagation step begins by sending the input signals through the nodes of each layer. A nonlinear activation function, called the sigmoid function, is usually used at each node for the transformation of the incoming signals to an output signal. This process repeats until the signals reach the output layer and an output vector is calculated. The backward-propagation step calculates the error vector by comparing the calculated and target outputs. New sets of weights are iteratively calculated, by modifying the existing weights, based on these error values until a minimum overall error, or global error, is obtained. The mean-square error (MSE) is usually used as a measure of the global error [11] which can he defined

as

$$MSE = \sum_{j=1}^{N_o} (d_i - y_i)^2,\qquad(3.1)$$

where $N_o$, is the number of output nodes, $y$ and $d$ are the output and target signals respectively.

$$\underbrace{w^{(m)}(K+1)}_{\text{New weights}} = \underbrace{w^{(m)}(K)}_{\text{Old weights}} + \gamma \underbrace{w^{(m)}(K) - w^{(m)}(K-1)}_{\text{Old change in weights}} + (1-\gamma)\underbrace{\alpha S^m (a^{m-1})^T}_{\text{Sensitivity}}.\quad(3.2)$$

Performance of the trained network can be evaluated by some simple statistical functions such as recognition rate (i.e., percentage of the total number of correctly classified outcomes over the number of sample points, or simply %Reco) and mean-square-error (MSE). If the error value on the test data set begins to increase, training is halted and the results are examined to determine whether they are acceptable. If the results are unacceptable, then it is possible to retrain the network, by either modifying some network parameters (e.g., the seed value for the random number generator, and the number of nodes in the middle layer), or increasing or decreasing the variations present in the training patterns. Once an acceptable error value is obtained during the test stage, the network is ready for solving real problems, such as prediction or classification problems. Figure 3.4 shows the steps involved in the application of backpropagation algorithm.

## 3.5.2   Levenberg Marquardt Algorithm

The Levenberg-Marquardt algorithm is a variation of Newton's method that was designed for minimizing functions that are sum of squares of other nonlinear func-
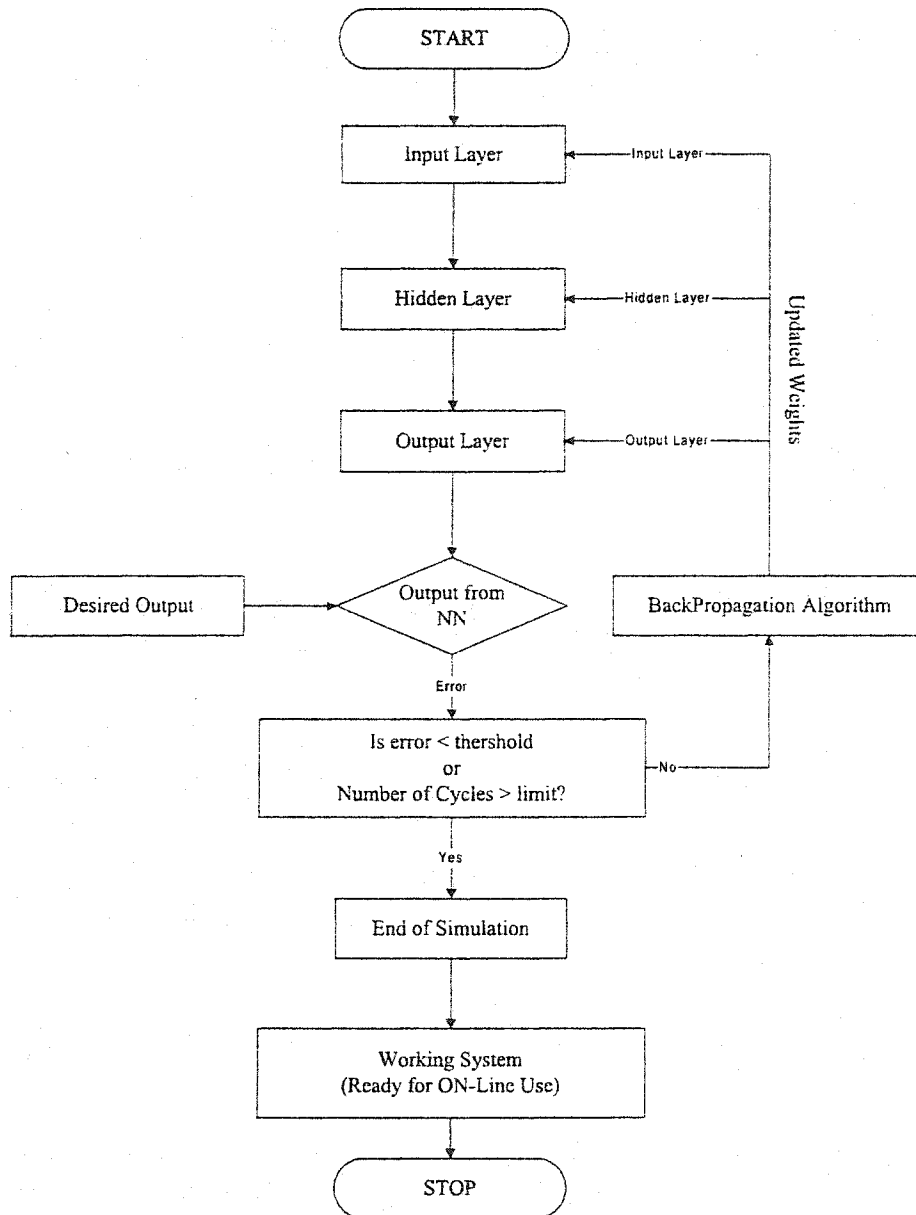
Figure 3.4: Flow chart of Backpropagation algorithm

tions.

In case of Levenberg-Marquardt algorithm the sum square error (SSQ) is usually the measure of the global error which can be defined as

$$F(x) = \sum_{q=1}^{Q}(t_q - a_q)^T(t_q - a_q) = \sum_{q=1}^{Q} e_q^T e_q = \sum_{q=1}^{Q}\sum_{j=1}^{S^M}(e_{j,q})^2 \qquad (3.3)$$

where $e_{j,q}$ is the jth element of the error for the qth input/target pair. The levenberg-Marquardt algorithm is described as

$$X_{k+1} = X_K - [J^T(X_k)J(X_k) + \mu I]^{-1}J^T(X_k)v(X_k) \qquad (3.4)$$

$$\triangle X_K = -[J^T(X_k)J(X_k) + \mu I]^{-1}J^T(X_k)v(X_k) \qquad (3.5)$$

This algorithm has a very useful feature that as $\mu_k$ is increase it approaches the steepest descent algorithm with small learning rate:

$$X_{k+1} = X_K - \frac{1}{\mu_k}(X_k)v(X_k) = X_K - \frac{1}{2\mu_k}\nabla F(x) \qquad (3.6)$$

While as $\mu_k$ is decreased to zero the algorithm becomes Gauss-Newton. Figure 3.5 shows the steps of levenberg-marquardt algorithm.

## 3.6 Activation Functions

An Activation function is used to transform the activation level of a unit (neuron) into an output signal. Typically, activation functions have a "squashing" effect i.e. it limits the permissible output range to some finite values[11].

$$Z_k = \sum_{j=1}^{m} w_{kj}x_j \qquad (3.7)$$

$$Y = \phi(Z_k) = \phi[w_{kj}x_j] \qquad (3.8)$$
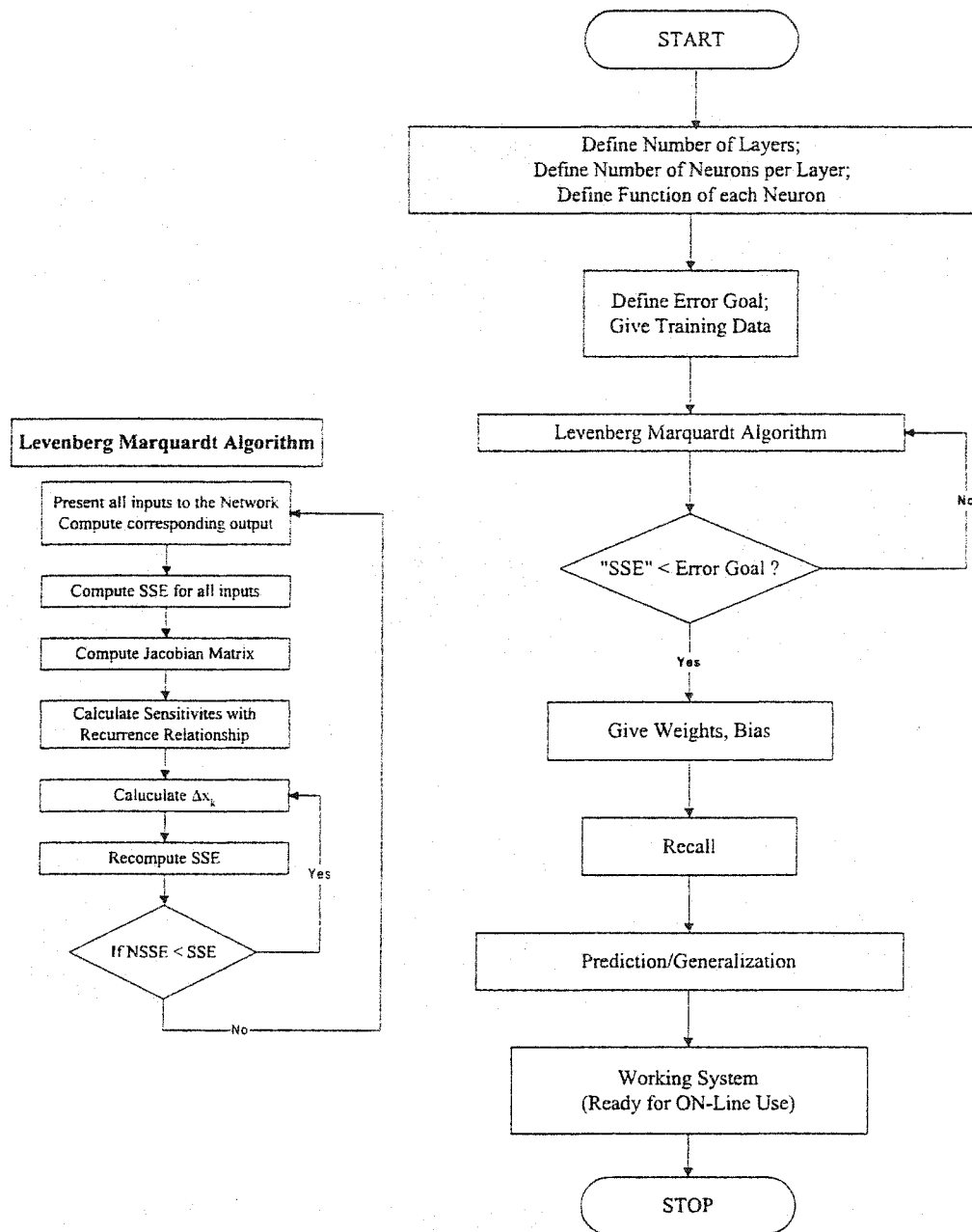
Figure 3.5: Flow chart of Levenberg-Marquardt algorithm

where $x_1, x_2,...$ are the input signals; $w_{k1}, w_{k2}, ...$ are synaptic weights of neuron $k$ and $\phi$ is the activation function.

$$\phi(z, \lambda) = \text{tansig}(z, \lambda) = \frac{1 - e^{-\lambda z}}{1 + e^{-\lambda z}} \quad \lambda \in \mathbf{Z}^+. \quad (3.9)$$

$$\phi(z, \lambda) = \text{logsig}(z, \lambda) = \frac{1}{1 + e^{-\lambda z}} \quad \lambda \in \mathbf{Z}^+. \quad (3.10)$$

$$\phi(z, \alpha, \lambda) = \text{sinmul}(z, \alpha, \lambda) = z + \frac{\alpha}{\lambda \pi} \sin(\lambda \pi z), \quad \lambda \in \mathbf{Z}^+, \ 0 < \alpha < 1. \quad (3.11)$$

$$\phi(z) = \text{piecewise}(z) = \begin{cases} -1 & \text{if} \quad z < -1, \\ z & \text{if} \quad -1 < z < 1, \\ 1 & \text{if} \quad z > 1. \end{cases} \quad (3.12)$$

Activation function for the hidden units are needed to introduce nonlinearity into the networks. Nonlinearity make the multilayer networks more powerful. For backpropagation learning the activation function must be differentiable. The more common activation functions are sigmoidals (log and tangent), piecewise-linear and sinusoidal functions, as depicted in Fig. 3.6. For hidden units, sigmoidal functions are usually preferable. With sigmoid units,a very small change in the weights will usually produce a change in the outputs, which makes it possible to tell that whether that change in weights is good or bad.

## 3.7 Data Generation

A data generation program is designed to generate data for the training of neural network models as shown in the figure 3.7.
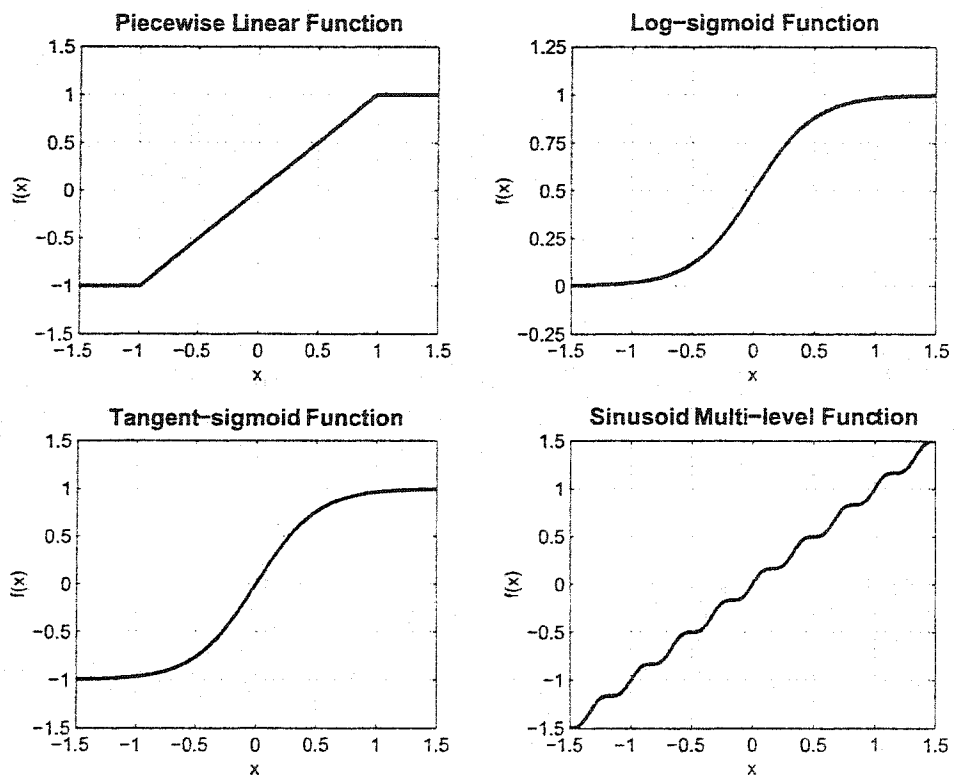
Figure 3.6: Commonly used activation functions.
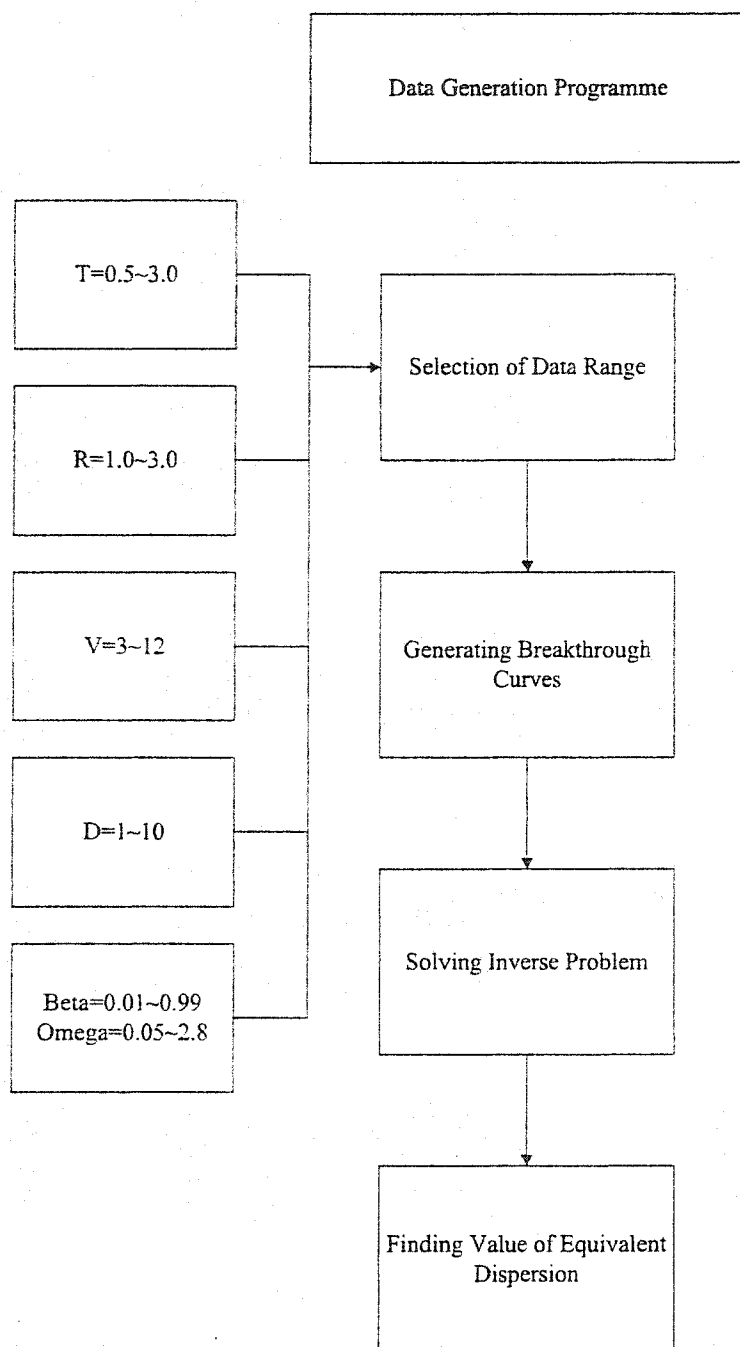
Figure 3.7: Flow chart of data generation programme

## 3.7.1 Data Range

In the first stage of the work, different ranges of data are selected for which the models are going to be trained. The selection of the data range depends upon different factors which include;

1. The model is prepared for lab scale studies

2. The values are chosen such that they cover both conservative and non-conservative tracers

3. Availability of experimental data

For pore volume $(T)$ data ranged from 0.5 to 3.1. In case of retardation factor $(R)$ three values are selected which are 1.026, 2.2 and 3.1. Pore water velocity $(V)$ has its range of values from 3 to 12 cm/day. Data range for dispersion coefficient is from 1 to 10 $cm^2/day$ for $\beta$ 0.01 to 0.99 and for $\omega$ it is from 0.05 to 2.8. Table 3.1 shows the complete data set. The procedure for the selection of data for a particular model is shown in figure 3.8. In table 3.2 a particular model for a pore water velocity $(V)$ is shown. A complete data set for a particular value of pore volume $(T)$ is shown in the appendix D.

## 3.7.2 Generating Breakthrough Curves

Table 3.3 shows few values of particular data set which are used to generate breakthrough curves. Breakthrough curves are generated using CXTFIT 2.1 [26]. The

| No. | $T = 0.5,\ 1.0,\ 1.8,\ 2.2,\ 3.1$ | | |
|---|---|---|---|
| 1. | $R = 1.026$ | $V = 3$<br>$V = 7$<br>$V = 10$<br>$V = 12$ | $D = 1, 5, 7, 10$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |
| 2. | $R = 2.2$ | $V = 3$<br>$V = 7$<br>$V = 10$<br>$V = 12$ | $D = 1, 5, 7, 10$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |
| 3. | $R = 3.0$ | $V = 3$<br>$V = 74$<br>$V = 10$<br>$V = 12$ | $D = 1,5,7,10$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |

Table 3.1: Complete data set for the generation of data

| No. | $T = 0.5, R = 1.026\ and\ V = 3\ cm/day$ | | |
|---|---|---|---|
| 1. | $R = 1.026$ | $V = 3$ | $D = 1$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |
| 2. | $R = 1.026$ | $V = 3$ | $D = 5$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |
| 3. | $R = 1.026$ | $V = 3$ | $D = 7$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |
| 4. | $R = 1.026$ | $V = 3$ | $D = 10$<br>$Beta = 0.0 - 0.99$<br>$Omega = 0.0 - 2.8$ |

Table 3.2: Complete data set for a particular value of pore water velocity associated with retardation factor and pore volume
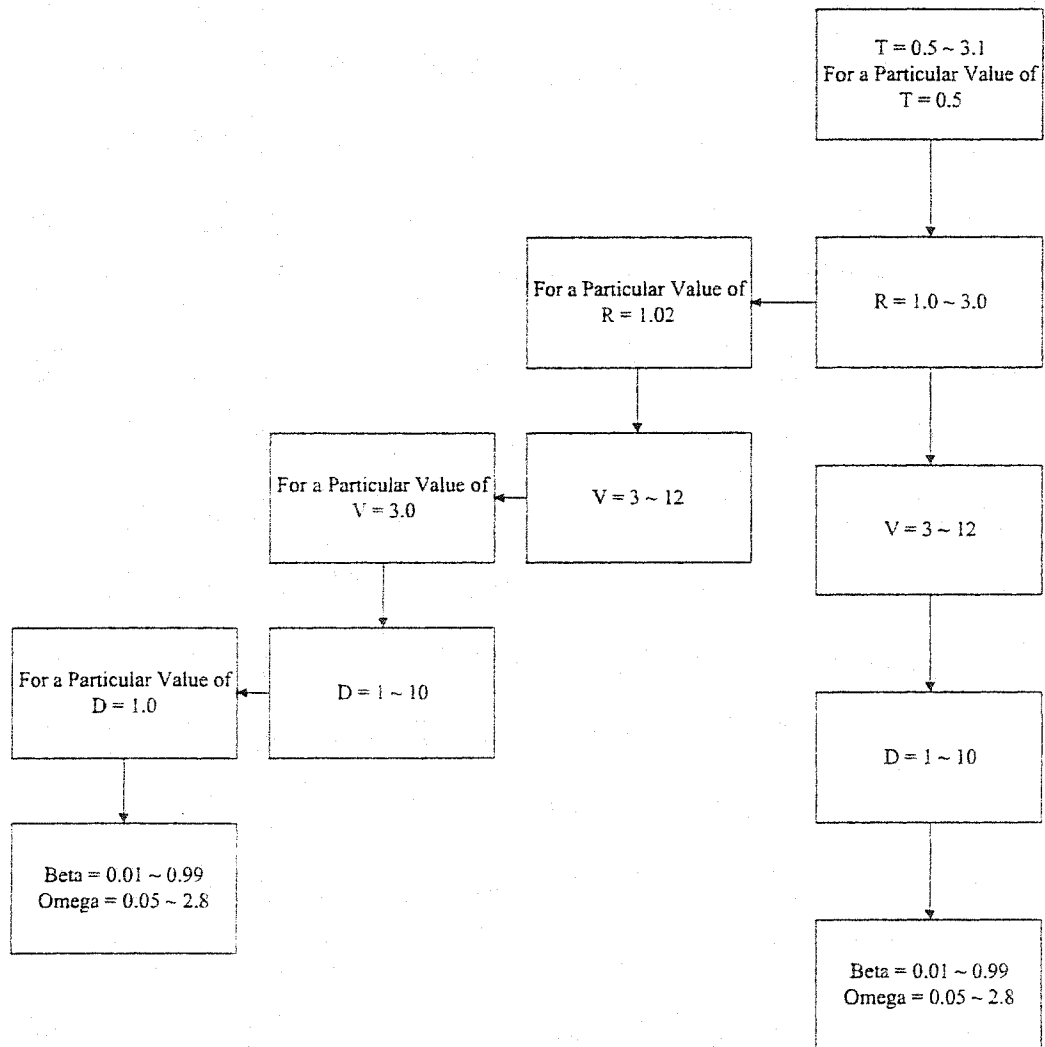
Figure 3.8: Flow chart for the selection data range for different variables

CXTFIT 2.1 code has its limitation and it can not run more than twelve 12 data files at a time. To prepare a data input file for CXTFIT 2.1, a programme was written in java language and a batch file is prepared for running the programme. These input data files are used to generate the breakthrough curves.

## 3.7.3    Solving Inverse Problem

The generated breakthrough curves are used to come up with the input files for solving the inverse problem. Another programme is written in java language which reads the breakthrough curves from output file from the direct problem and converts the data into an input file for the inverse problem. The inverse problem is solved considering linear equilibrium model to determine the value of equivalent dispersion coefficient (EDC). Table 3.4 shows the few values of final data set for a particular model for which the neural network models are trained. A complete data set is shown in appendix D.

| Pore volume | Pore water velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1 | 0.1 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 0.2 |

Table 3.3: Data used to generate Breakthrough curves for T=0.5, R=1.026, V=3cm/day

| Pore volume | Pore water velocity | Retardation factor | Dispersion coefficient | Beta | Omega | EDC |
|---|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1.00 | 0.1 | 0.2 | 221.12 |
| 0.5 | 3 | 1.026 | 1.00 | 0.2 | 0.2 | 198.47 |
| 0.5 | 3 | 1.026 | 1.00 | 0.3 | 0.2 | 178.40 |
| 0.5 | 3 | 1.026 | 1.00 | 0.4 | 0.2 | 160.56 |
| 0.5 | 3 | 1.026 | 1.00 | 0.5 | 0.2 | 163.21 |
| 0.5 | 3 | 1.026 | 1.00 | 0.6 | 0.2 | 91.99 |
| 0.5 | 3 | 1.026 | 1.00 | 0.7 | 0.2 | 55.08 |
| 0.5 | 3 | 1.026 | 1.00 | 0.8 | 0.2 | 28.38 |
| 0.5 | 3 | 1.026 | 1.00 | 0.9 | 0.2 | 12.25 |
| 0.5 | 3 | 1.026 | 1.00 | 0.99 | 0.2 | 1.08 |

Table 3.4: Data used for training the neural network algorithms for T=0.5, R=1.026, V=3 cm/day

# Chapter 4

# TRAINING

## 4.1   Training Set Development

A major aspect of the neural net approach is the phase of convergence. Convergence

in the Backpropagation (BP) and Levenberg Marquardt (LM) algorithm means

that the global minimum (smallest error) of the error function is obtained in a

reasonable amount of iterations. The iterative process may require long training

times of the order of several hundreds of thousands of iterations. Sometimes the

network may get stuck in a local minimum during training which means that the

network has failed to learn acceptably and gives large errors [10]. The development

of faster learning algorithms, local minimum detection and avoidance methods are

active areas of research on neural nets [35]. Careful examination of the data is

crucial before the training process starts. Sets of training patterns which do not

adequately distinguish between different facies groups will invariably result in either

slow convergence or non-convergence[36] . Developing training data usually includes the choice of training variables and their respective sample and data. In this work, the numerically generated data is used to train the neural network model. The training data does not contain any outliers.The neural network training can be made more efficient if certain preprocessing steps are performed on the network inputs and targets[37]. This can be done by normalizing the whole data set with respect to mean and standard deviation. For this purpose, each data set is normalized in two steps; first the mean of the data is subtracted from all the data points in the set and then this set is divided by the standard deviation of the data set in this way the inputs and targets will have zero means and unity standard deviation.

## 4.2   Training

The final development step is training and testing the network. Training is the stage when the net learns the recognition task by adjusting the weights in the links between the nodes created by processing representative examples (input and output pairs). Each pass through the training data is called an epoch, and the neural network learns through the overall change in weights accumulating over many epochs. Training continues until the values of the weights cause the network to map the input patterns to appropriate results [35]. All the programs for normalization input data and for the training of the neural networks was written in MATLAB. The program listings are given in Appendix B.

## 4.2.1 BackPropagation Neural Network Training

In this approach, the neural network design consists of only one net. The input layer consists of 60 nodes the middle layer contains 20 nodes and the output layer contains a single node as there is only one output. In this approach the whole net was trained using the pattern Mode[1]. It is also known as online training where a single example $\{Z^t, D^t\}$ is chosen (e.g. randomly) from the training set at each iteration t. An estimate of the true gradient is then computed based on the error $\{E^t\}$ of that example. and then the weights are updated:

$$\underbrace{w^{(m)}(K+1)}_{\text{New weights}} = \underbrace{w^{(m)}(K)}_{\text{Old weights}} + \gamma \underbrace{w^{(m)}(K) - w^{(m)}(K-1)}_{\text{Old change in weights}} + (1 - \gamma) \underbrace{\alpha S^m (a^{m-1})^T}_{\text{Sensitivity}}.$$

Each neural network model was trained for 60, 000 iterations for which they took about 12 minutes each, and the maximum error was found to be less than $10^{-2}$ at the output layer.

## 4.2.2 Levenberg Marquardt Neural Network Training

The levenberg marquardt algorithm is applied using the batch mode, where the weights are updated after a complete sweep through a training set. The levenberg marquardt algorithm use the jacobi approximation and most important they work only for mean square error loss functions. In this approach, the neural network design consists of only one net. The input layer consists of 20 nodes the middle layer contains 10 nodes and the output layer contains a single node as there is only

---

[1]In Pattern Mode learning, weight are updated after the presentation of each data point

one output. Neural network model training was stopped after the sum square error is less than a predefined value.

## 4.2.3   Training of Networks

Based on the data range one hundred and sixty neural networks models are trained. Both Backpropagation (BP) and Levenberg Marquardt (LM) algorithms are used for the training of neural networks. The neural networks training for specific values of pore volume are presented. There are four data sets for pore volume $T = 0.5$ and retardation factor $R = 1.026$. Each data set has a different value of pore water velocity and has been trained using both algorithms. The network training for both algorithms are shown in Figure 4.1 - 4.15. The continuous line as shown in these figure represents the data used to train the network and the dotted line is the network prediction for each output. Figure 4.2 - 4.16 shows an enlarge view of training for both above mentioned algorithms. Similarly for pore volume $T = 1.0, 1.8, 2.2, 3.1$ and retardation factor $R = 1.026$ neural networks training for both algorithms are shown in Figure 4.17- 4.48.
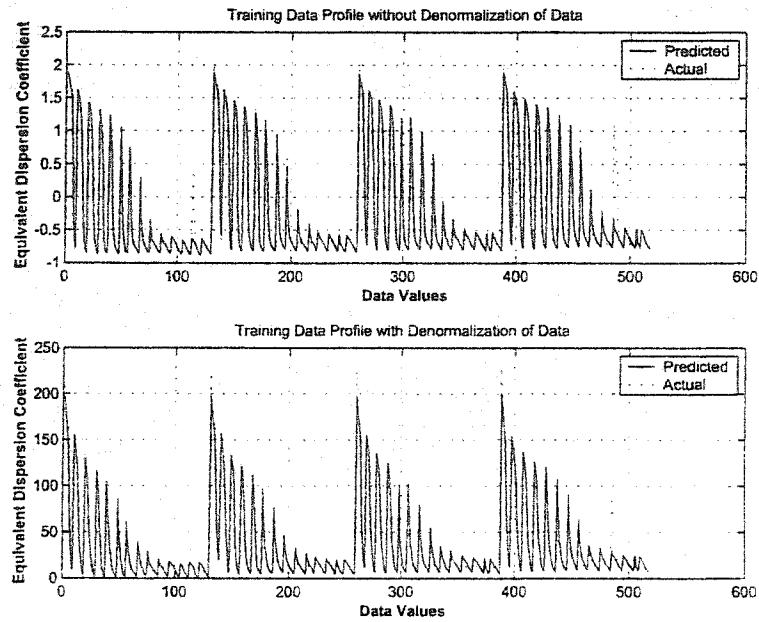
Figure 4.1: Neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=3 cm/day
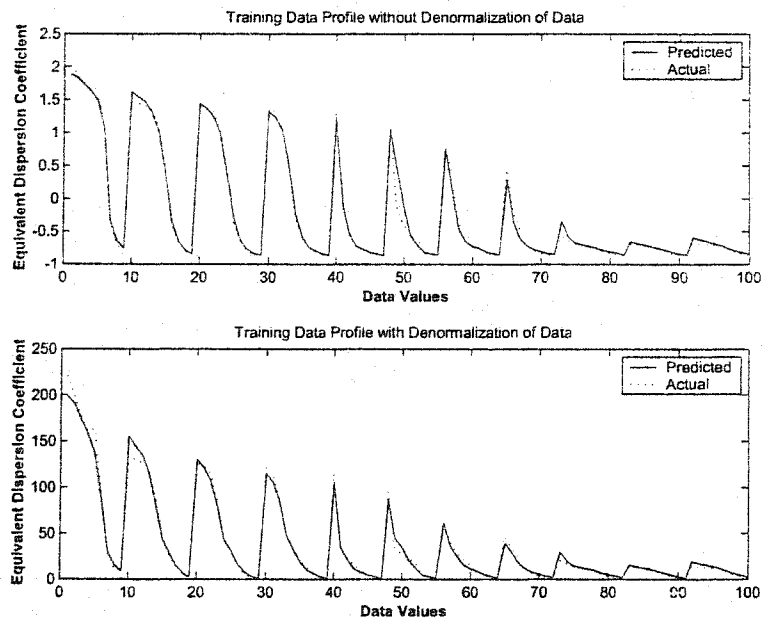


Figure 4.2: Enlarge view of neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=3 cm/day
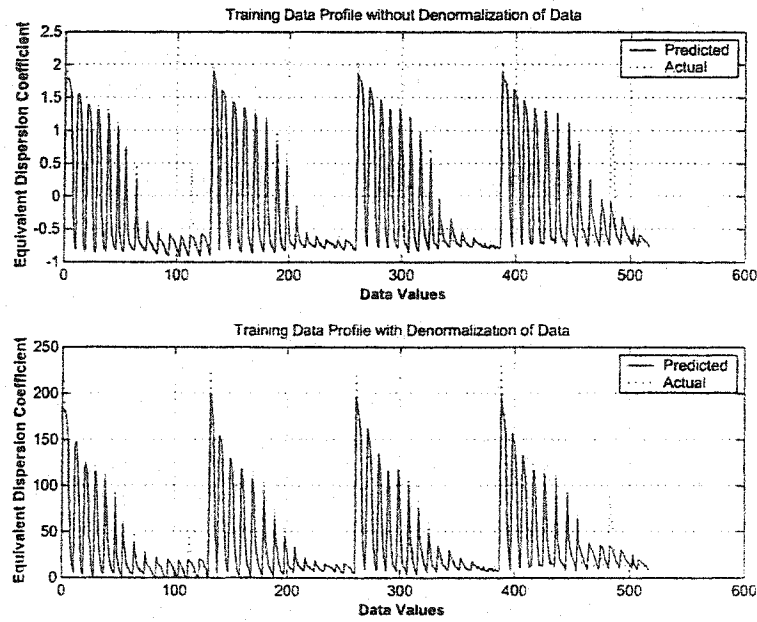
Figure 4.3: Neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=3 cm/day
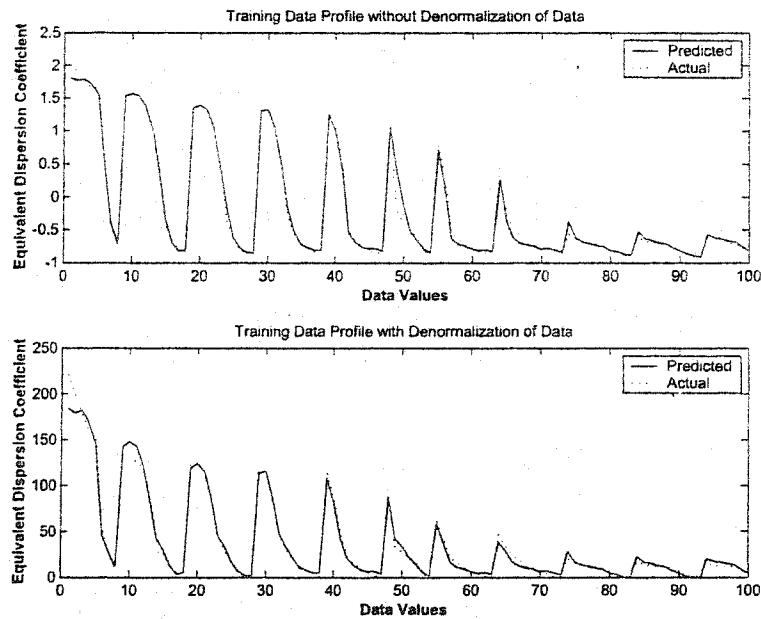


Figure 4.4: Enlarge view of neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=3 cm/day
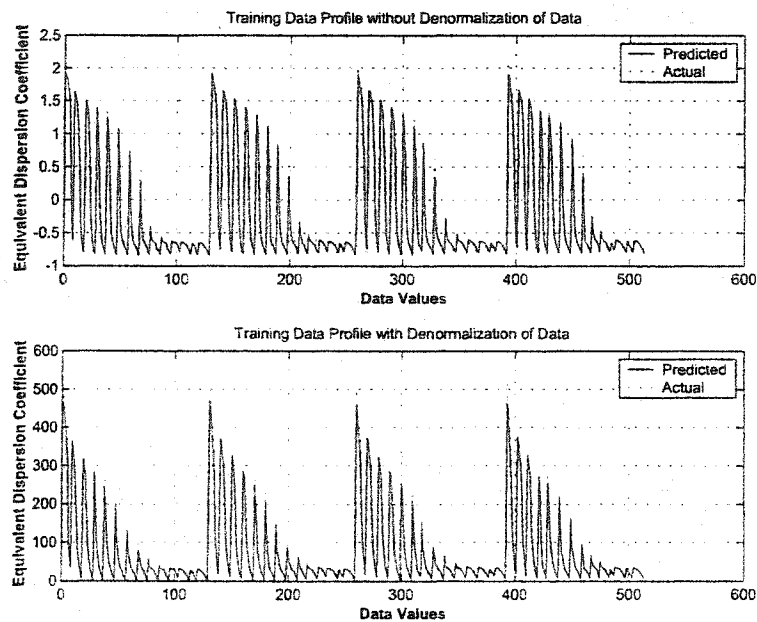
Figure 4.5: Neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=7 cm/day
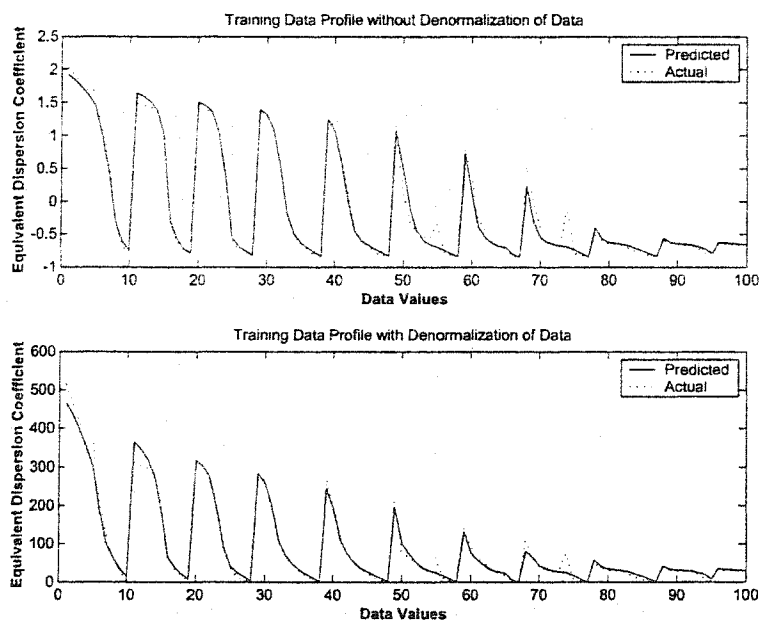


Figure 4.6: Enlarge view of neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=7 cm/day
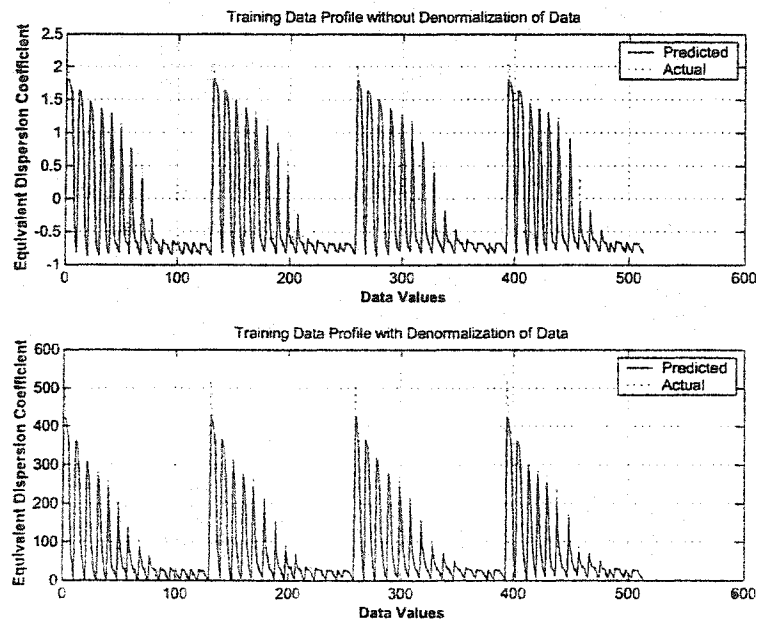
Figure 4.7: Neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=7 cm/day
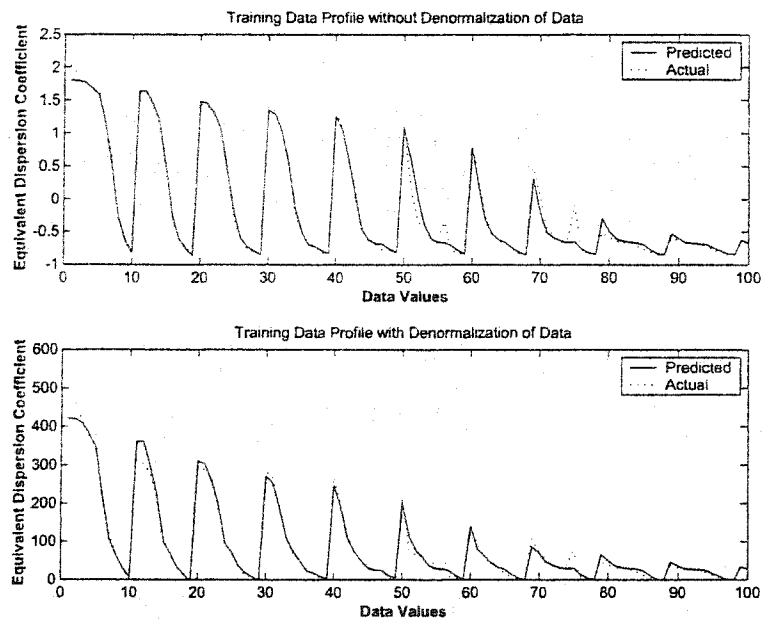


Figure 4.8: Enlarge view of neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=7 cm/day
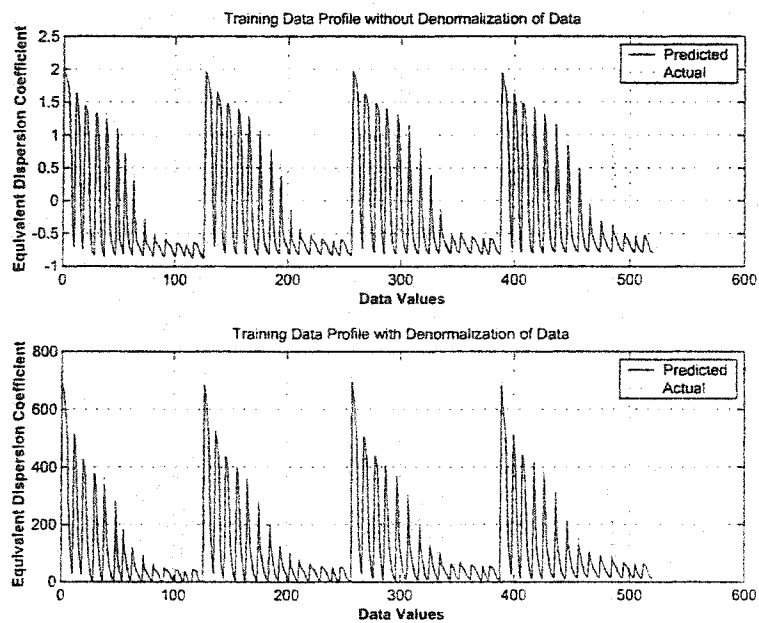
Figure 4.9: Neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=10 cm/day
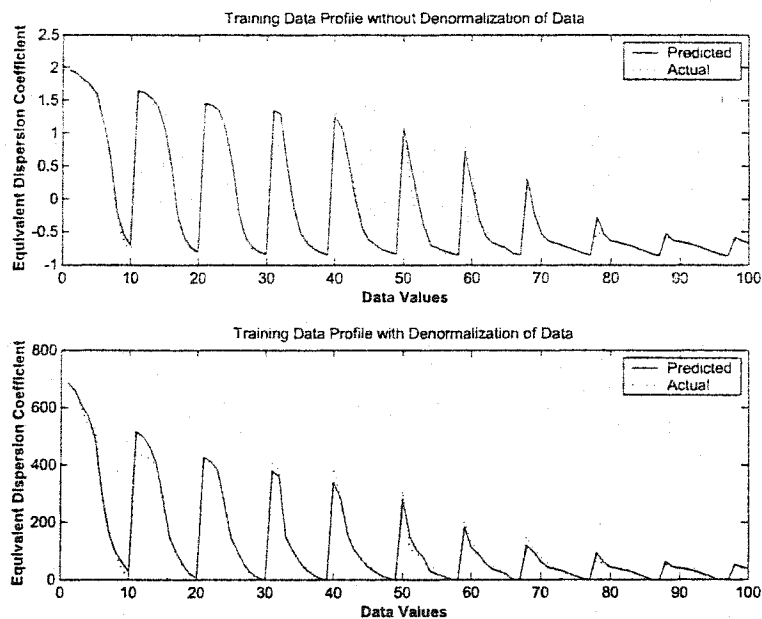


Figure 4.10: Enlarge view of neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=10 cm/day
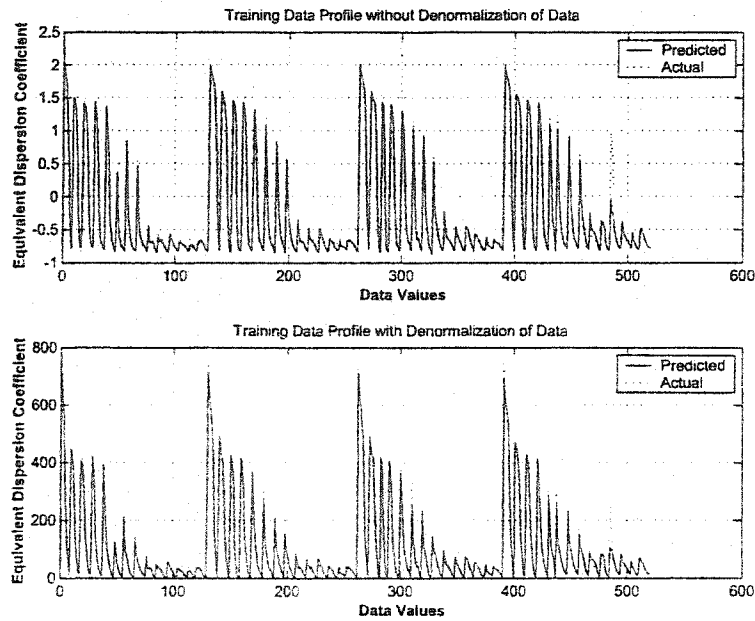
Figure 4.11: Neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=10 cm/day
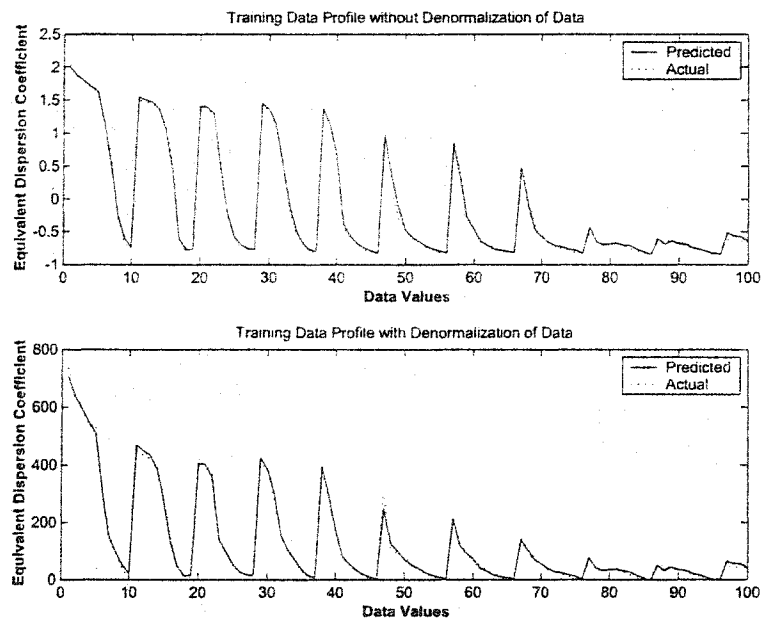


Figure 4.12: Enlarge view of neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=10 cm/day
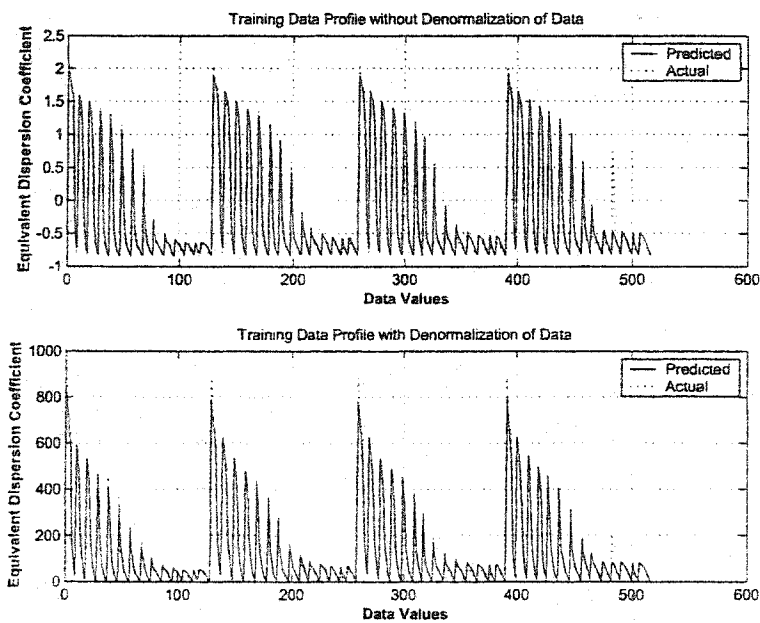
Figure 4.13: Neural network model training using Backpropagation algorithm for T=0.5, R=1.026, V=12 cm/day
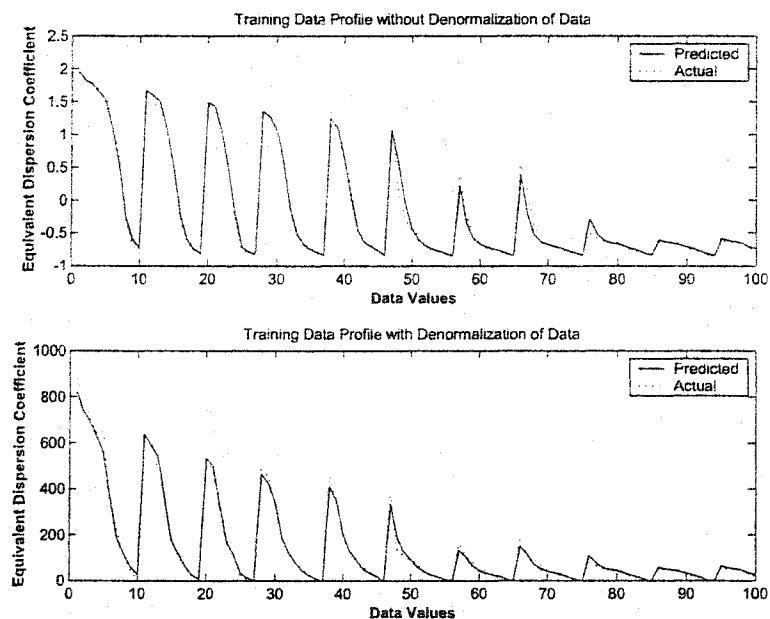


Figure 4.14: Enlarge view of neural network model training using using Backpropagation algorithm for T=0.5, R=1.026, V=12 cm/day
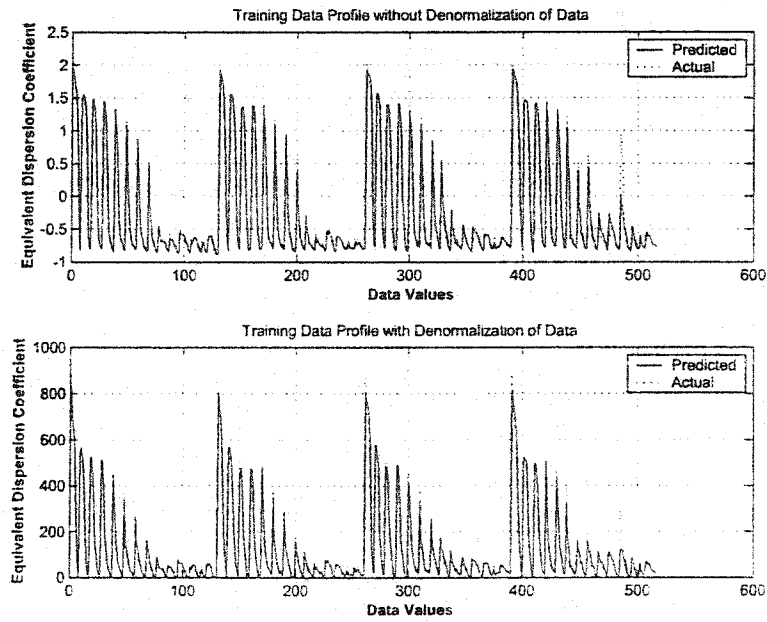
Figure 4.15: Neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=12 cm/day
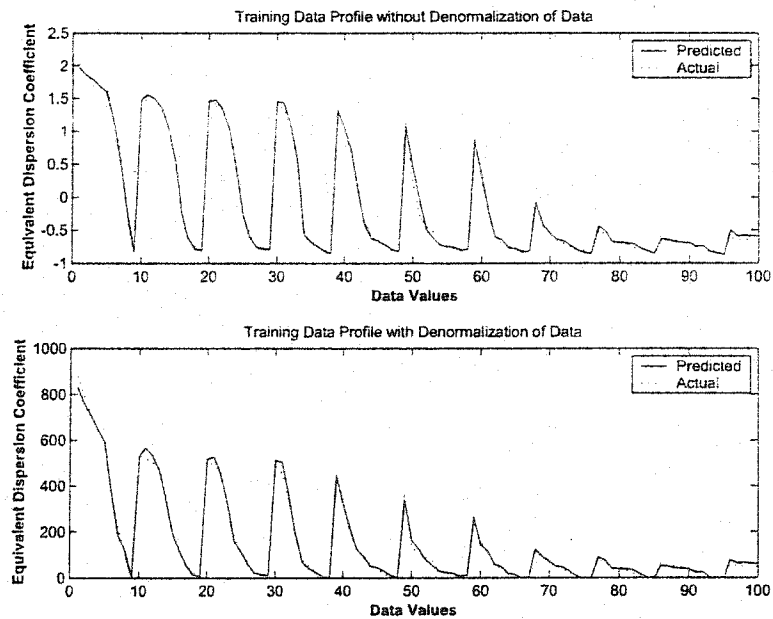


Figure 4.16: Enlarge view of neural network model training using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=12 cm/day
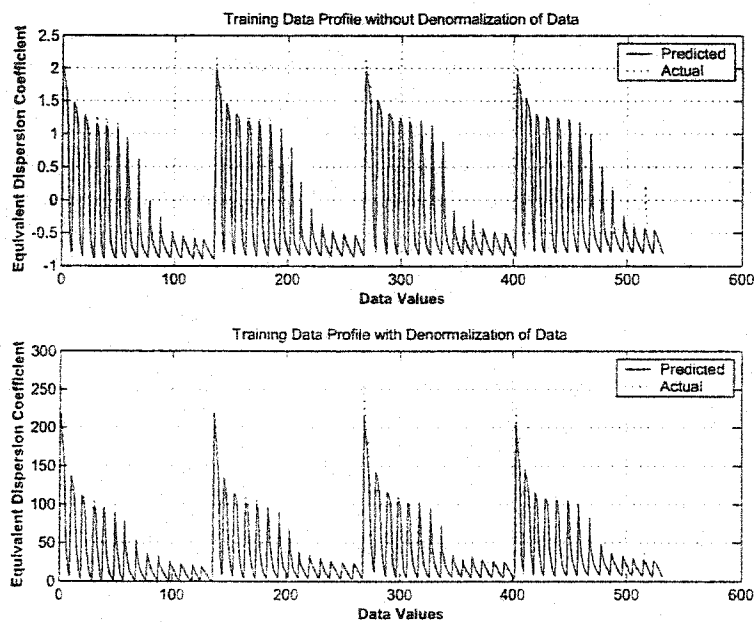
Figure 4.17: Neural network model training using Backpropagation algorithm for T=1.0, R=1.026, V=3 cm/day
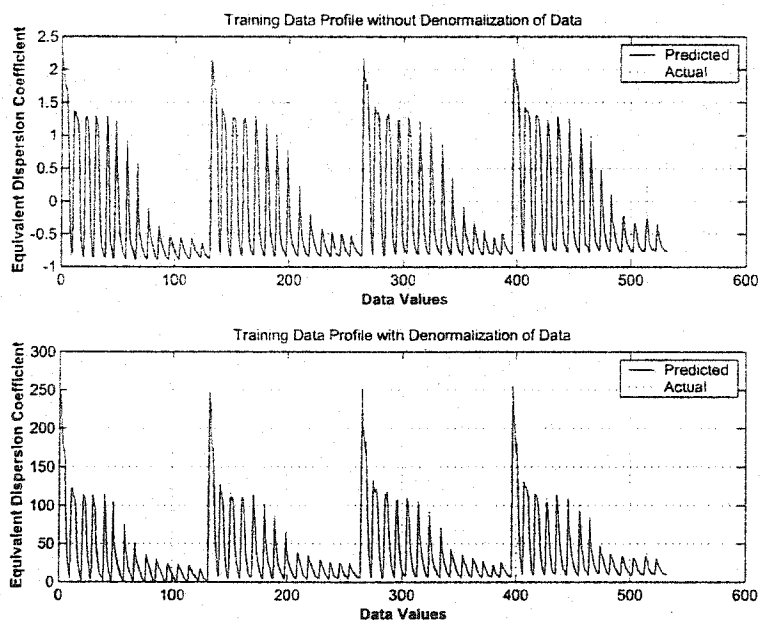


Figure 4.18: Neural network model training using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=3 cm/day
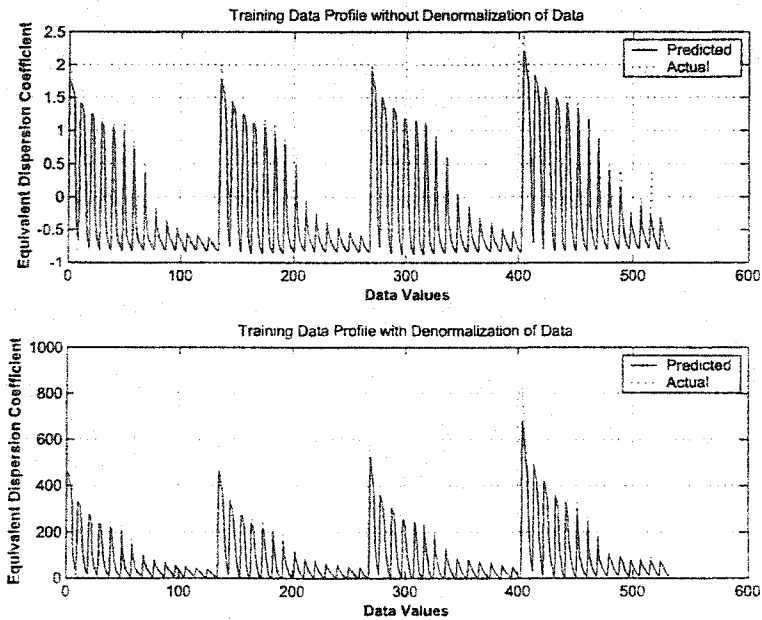
Figure 4.19: Neural network model training using Backpropagation algorithm for T=1.0, R=1.026, V=7 cm/day
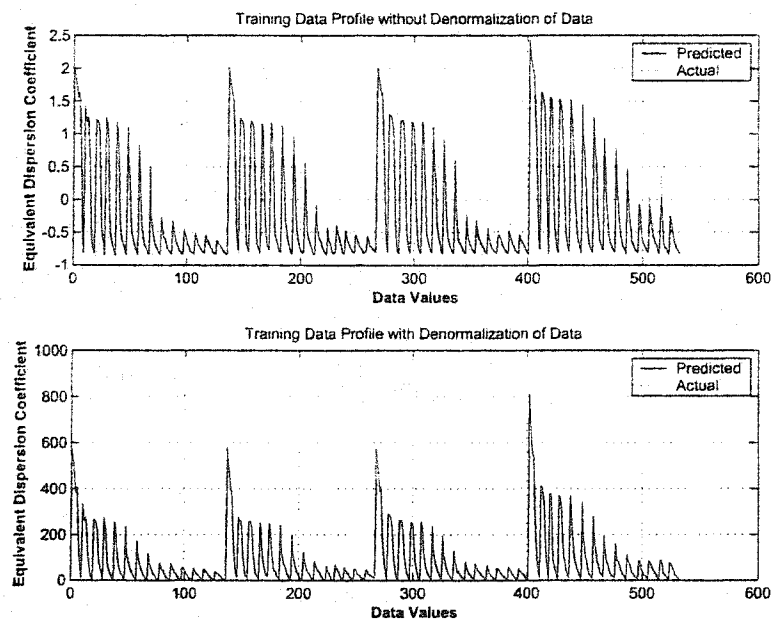


Figure 4.20: Neural network model training using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=7 cm/day
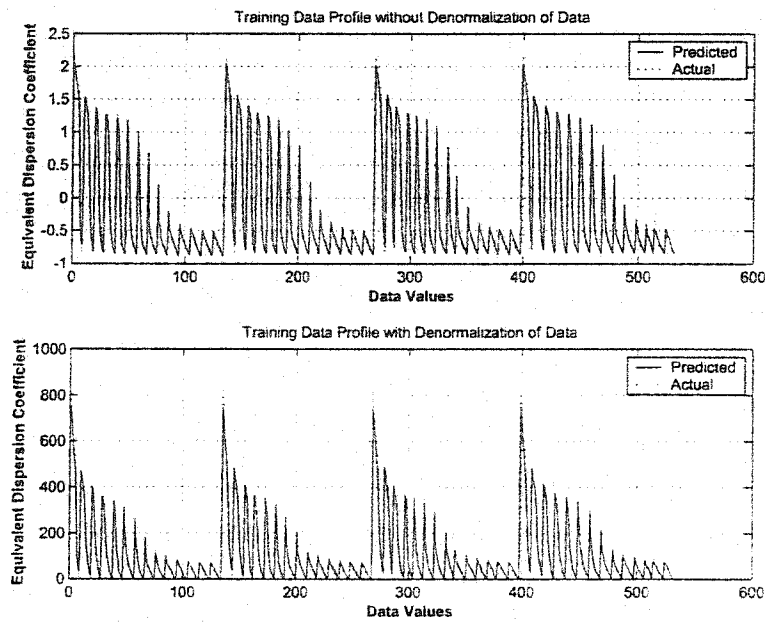
Figure 4.21: Neural network model training using Backpropagation algorithm for T=1.0, R=1.026, V=10 cm/day
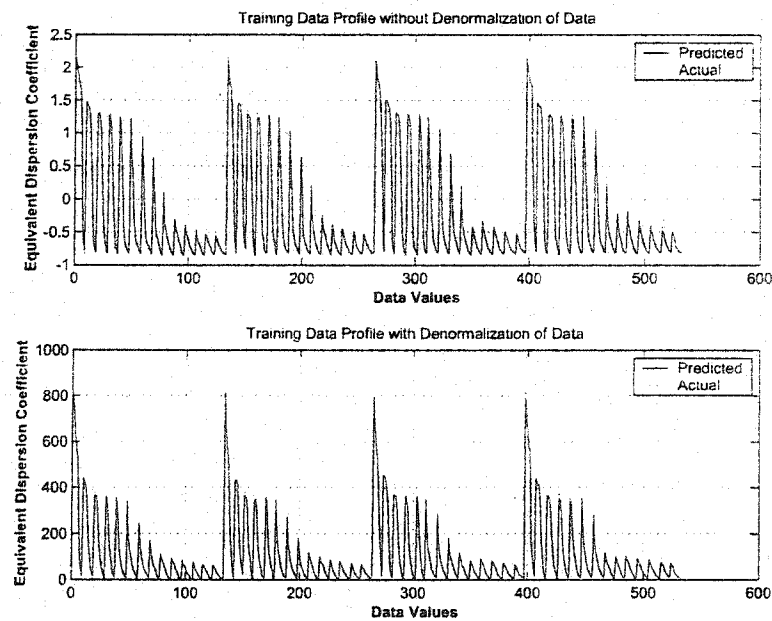


Figure 4.22: Neural network model training using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=10 cm/day
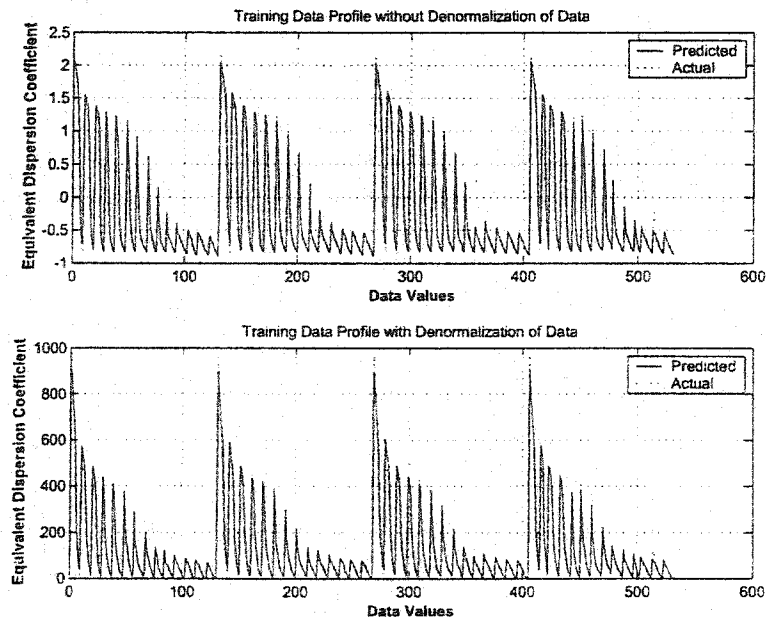
Figure 4.23: Neural network model training using Backpropagation algorithm for T=1.0, R=1.026, V=12 cm/day
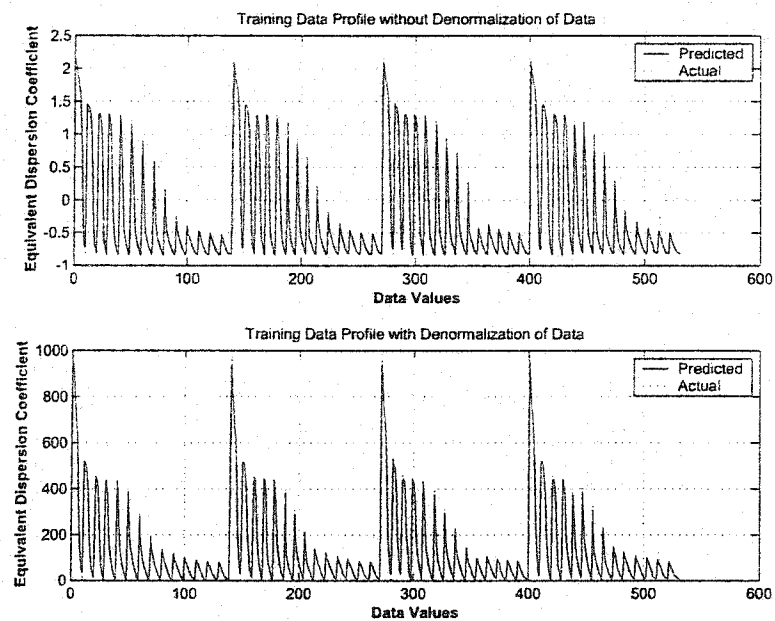


Figure 4.24: Neural network model training using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=12 cm/day
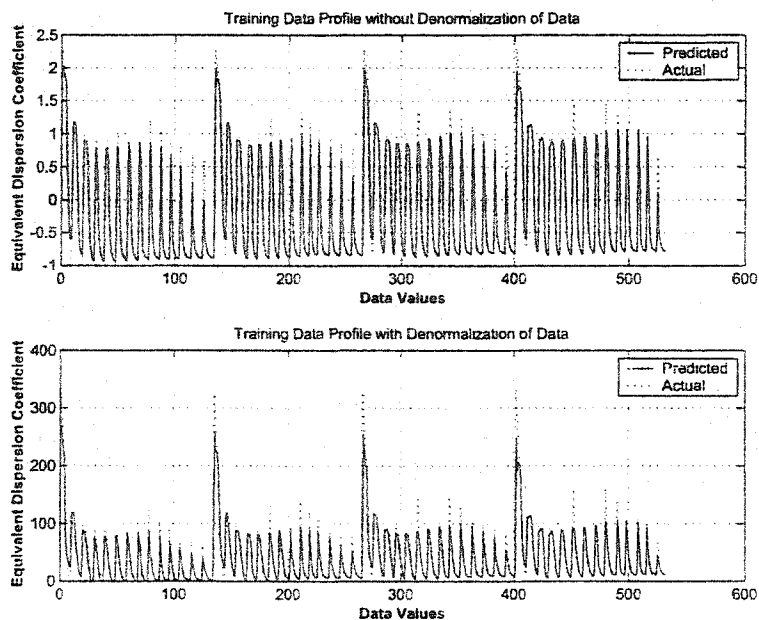
Figure 4.25: Neural network model training using Backpropagation algorithm for T=1.8, R=1.026, V=3 cm/day
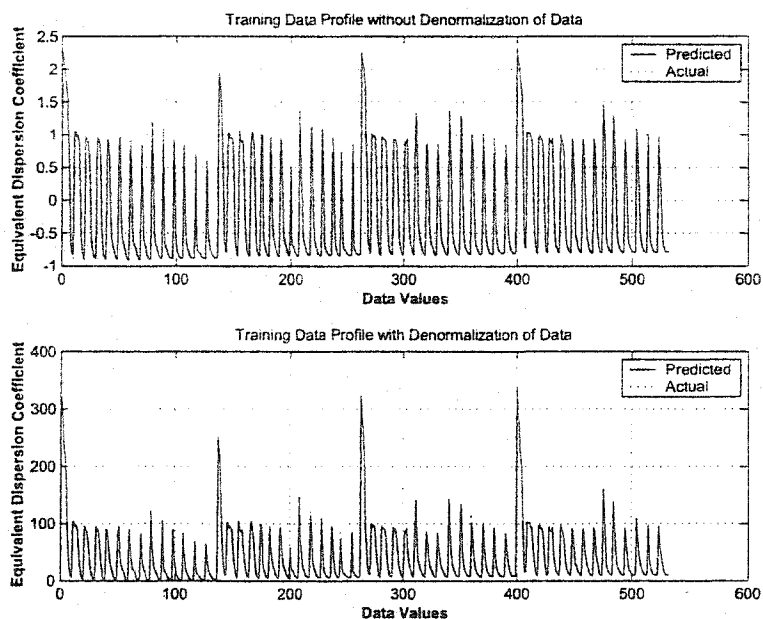


Figure 4.26: Neural network model training using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=3 cm/day
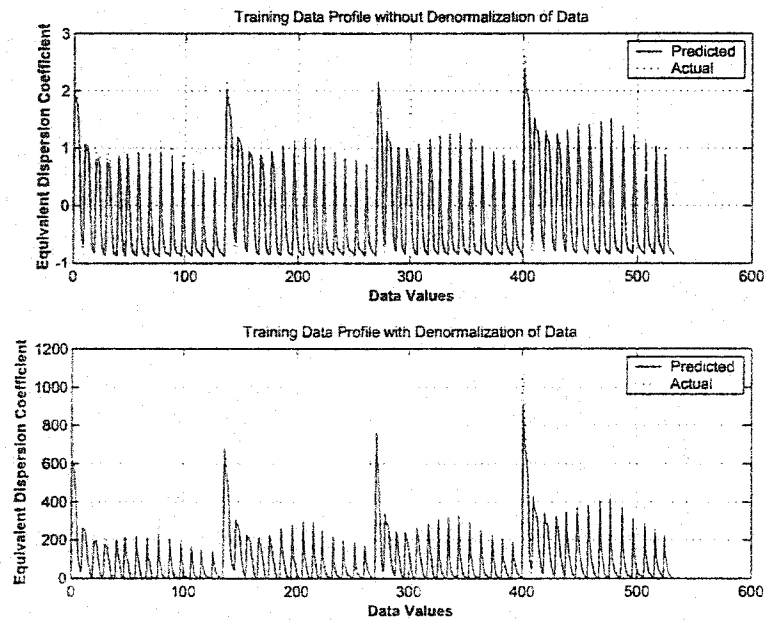
Figure 4.27: Neural network model training using Backpropagation algorithm for T=1.8, R=1.026, V=7 cm/day
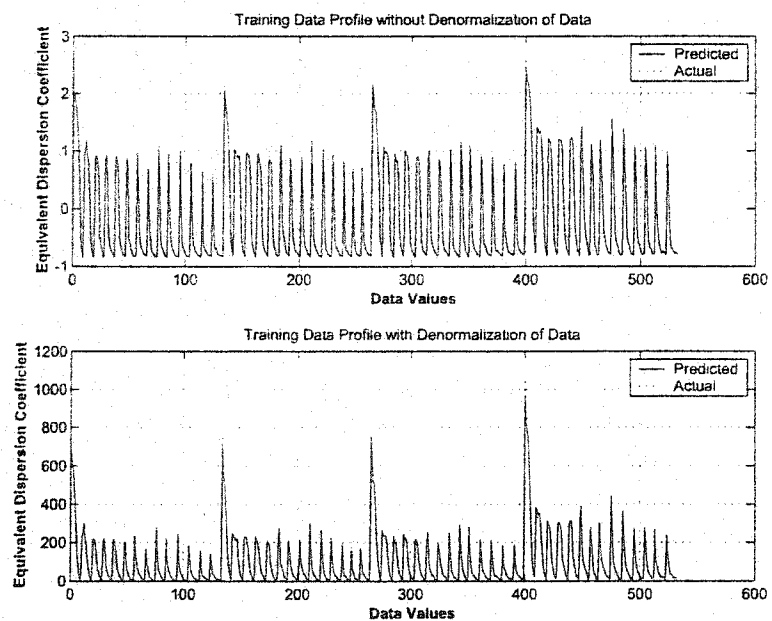


Figure 4.28: Neural network model training using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=7 cm/day
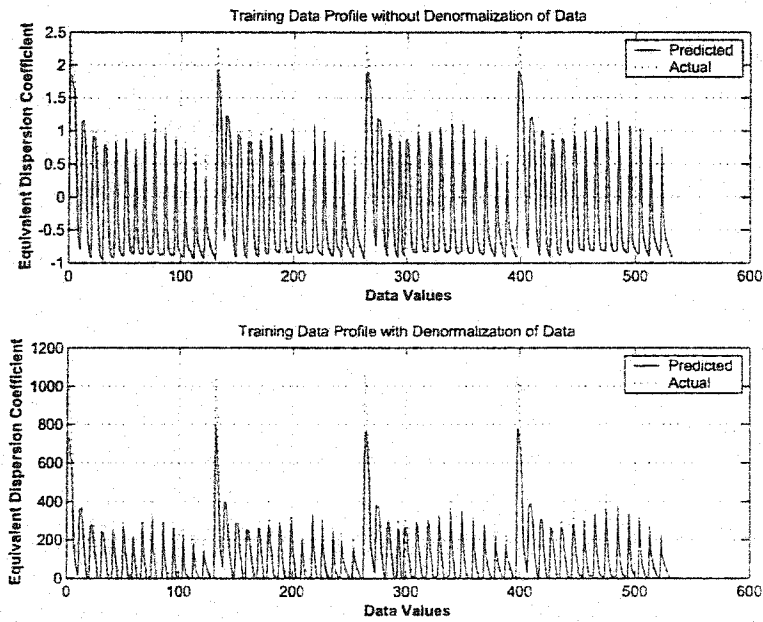
Figure 4.29: Neural network model training using Backpropagation algorithm for T=1.8, R=1.026, V=10 cm/day
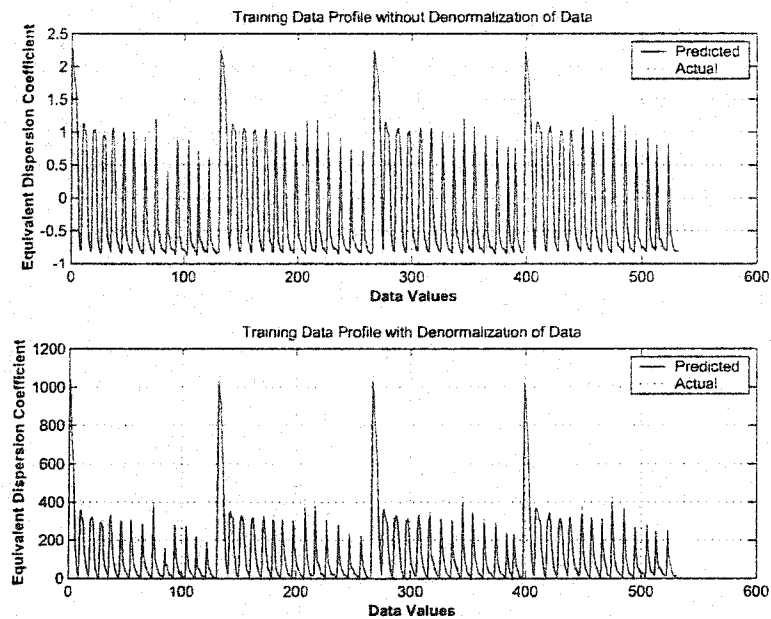


Figure 4.30: Neural network model training using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=10 cm/day
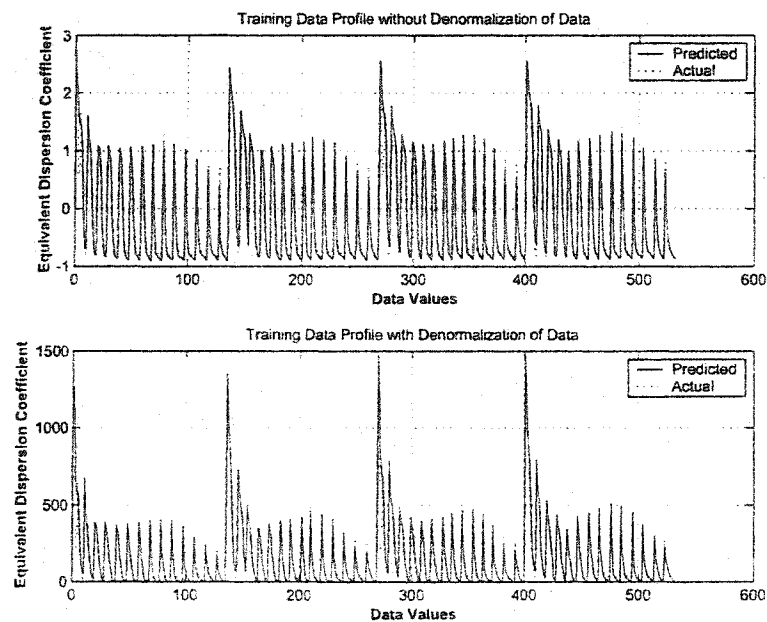
Figure 4.31: Neural network model training using Backpropagation algorithm for T=1.8, R=1.026, V=12 cm/day
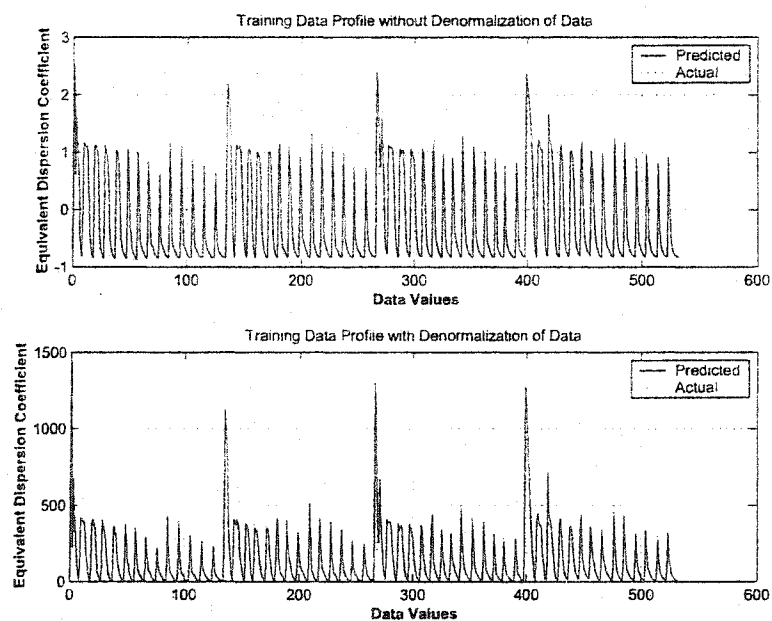


Figure 4.32: Neural network model training using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=12 cm/day
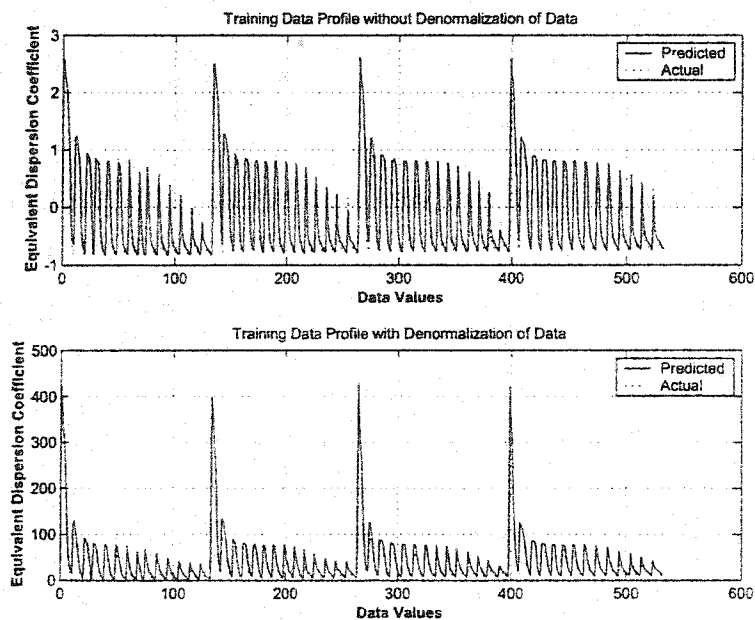
Figure 4.33: Neural network model training using Backpropagation algorithm for T=2.2, R=1.026, V=3 cm/day
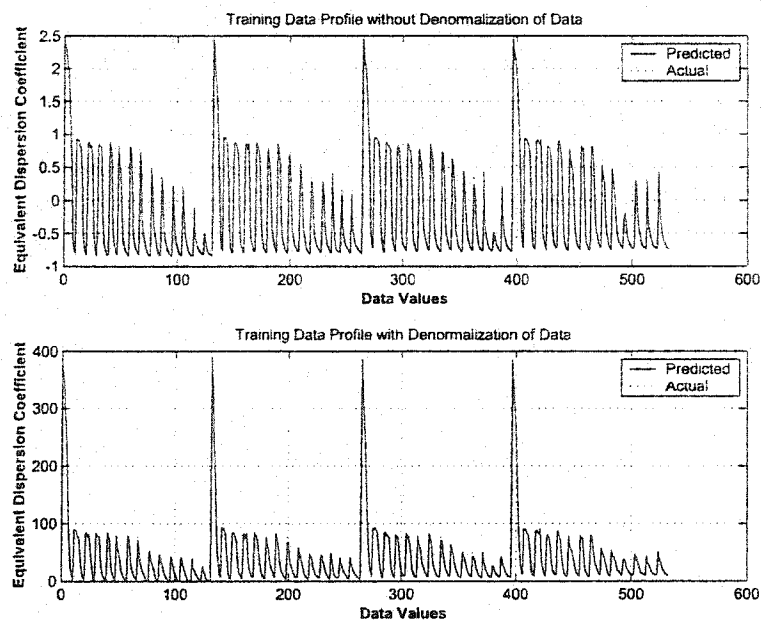


Figure 4.34: Neural network model training using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=3 cm/day
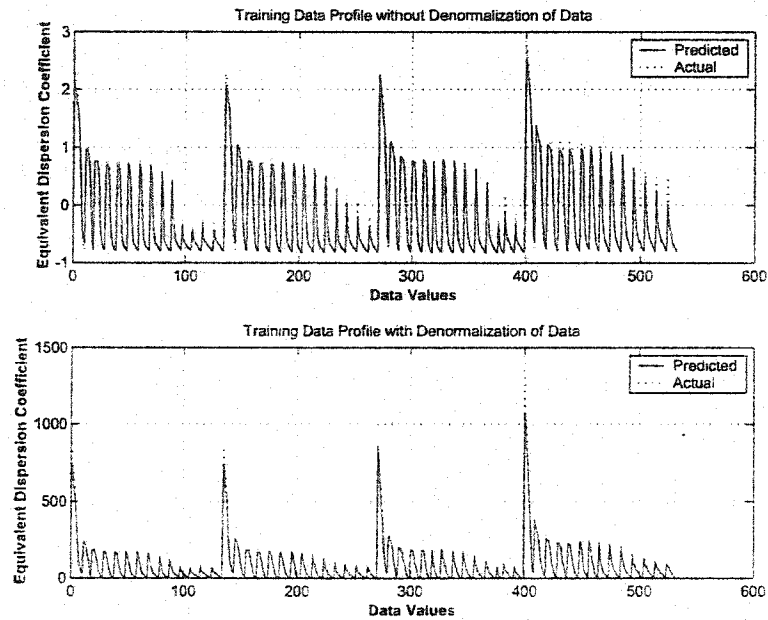
Figure 4.35: Neural network model training using Backpropagation algorithm for T=2.2, R=1.026, V=7 cm/day
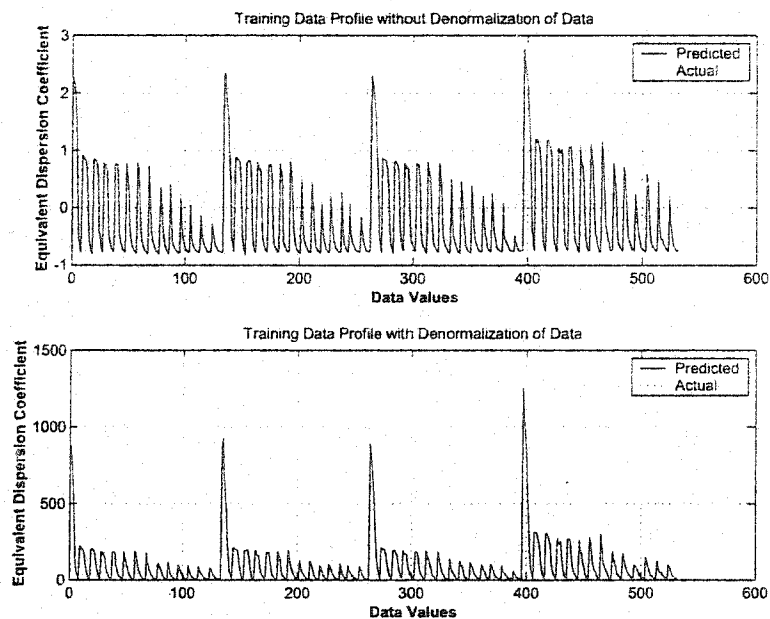


Figure 4.36: Neural network model training using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=7 cm/day
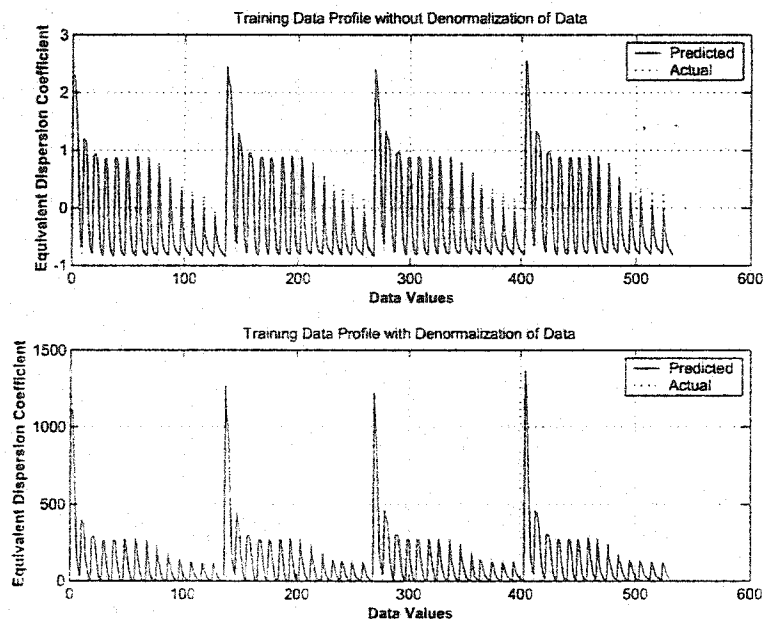
Figure 4.37: Neural network model training using Backpropagation algorithm for T=2.2, R=1.026, V=10 cm/day
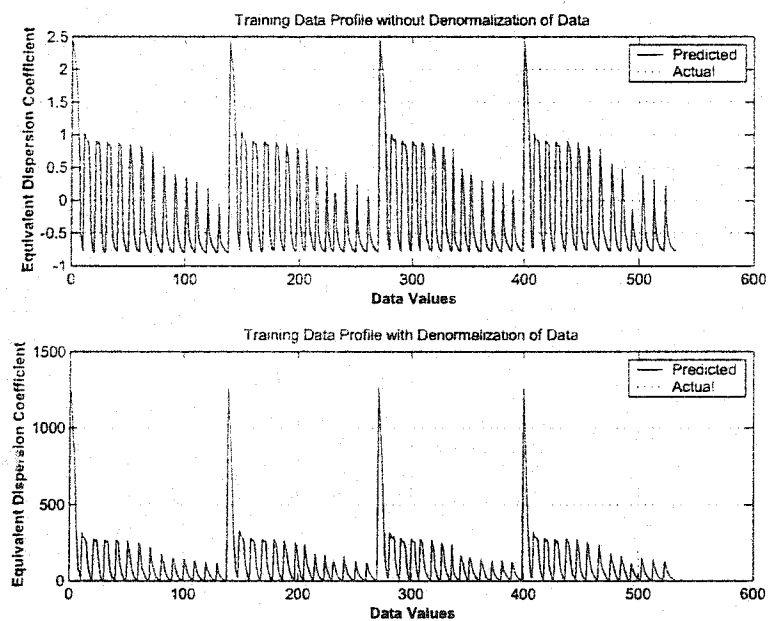


Figure 4.38: Neural network model training using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=10 cm/day
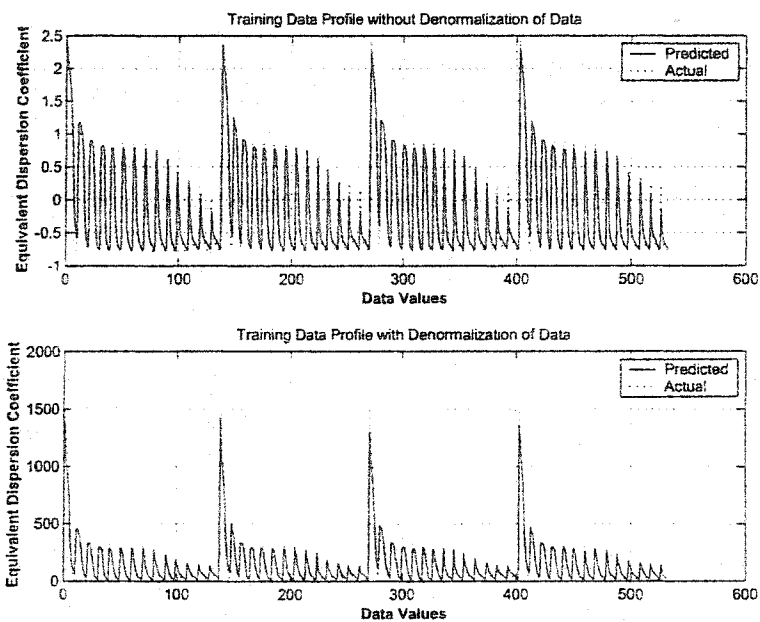
Figure 4.39: Neural network model training using Backpropagation algorithm for T=2.2, R=1.026, V=12 cm/day
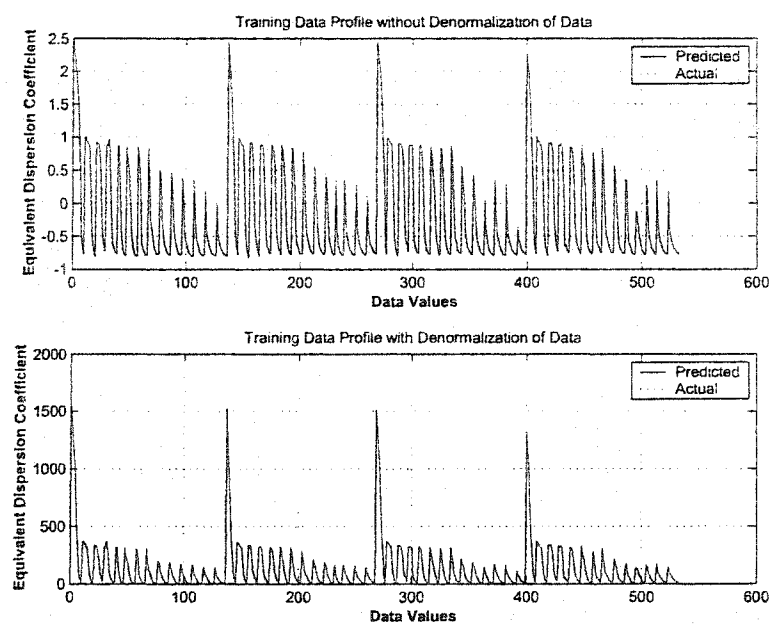


Figure 4.40: Neural network model training using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=12 cm/day
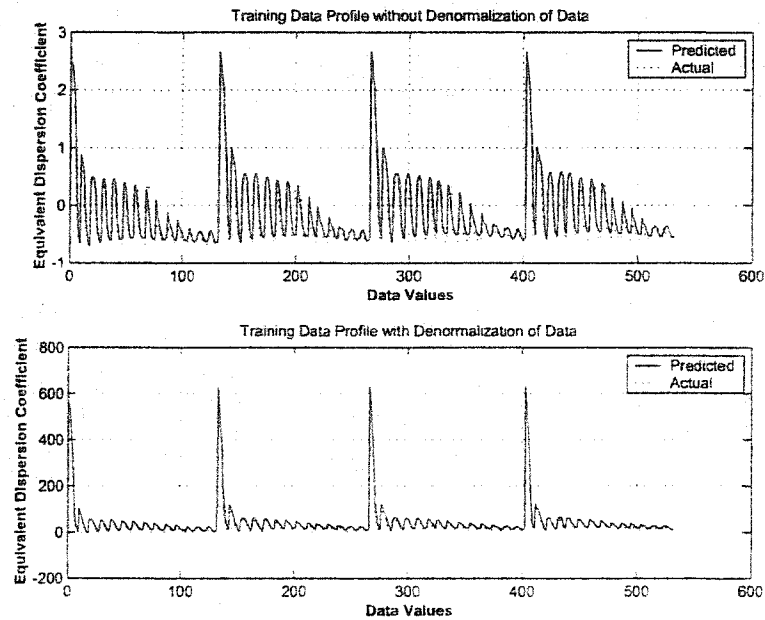
Figure 4.41: Neural network model training using Backpropagation algorithm for T=3.1, R=1.026. V=3 cm/day
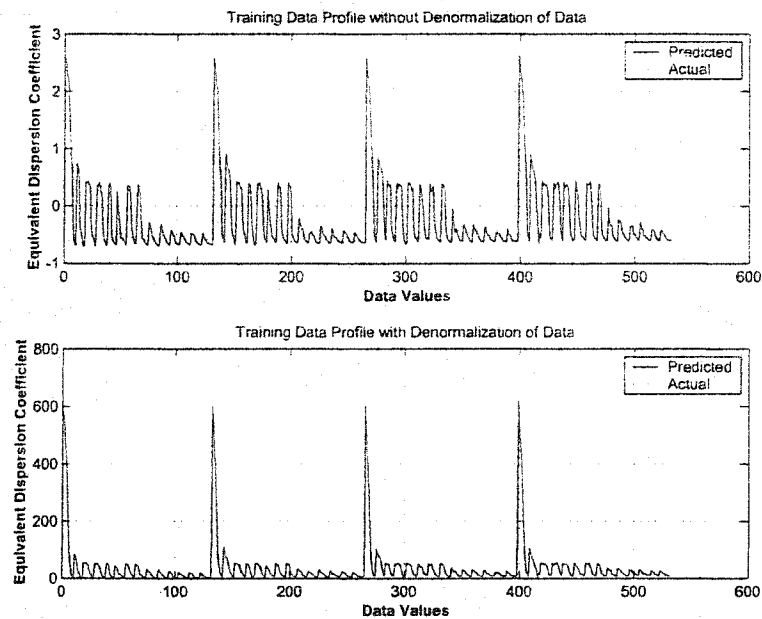


Figure 4.42: Neural network model training using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=3 cm/day
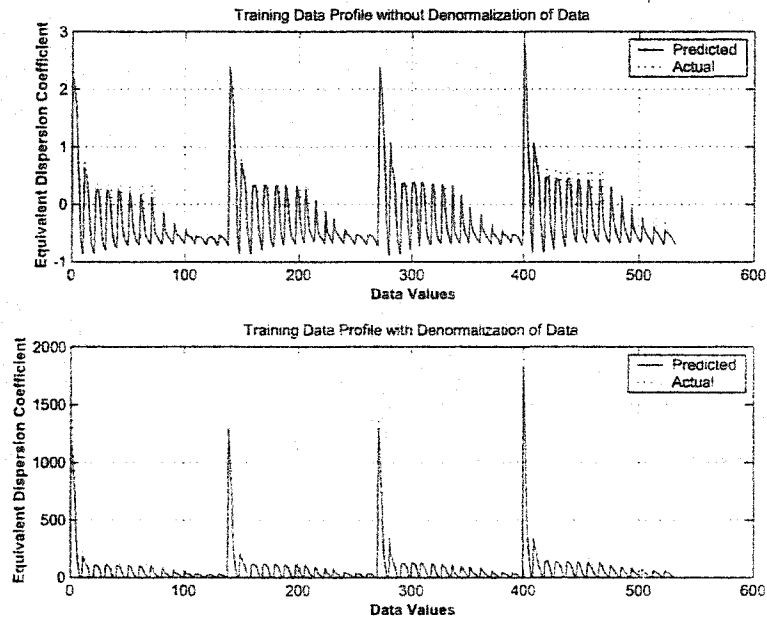
Figure 4.43: Neural network model training using Backpropagation algorithm for T=3.1, R=1.026, V=7 cm/day
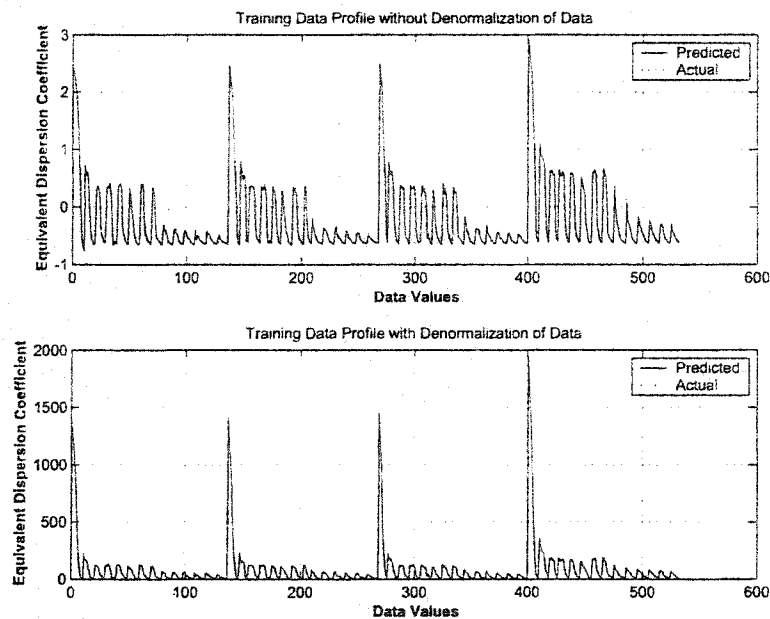


Figure 4.44: Neural network model training using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=7 cm/day
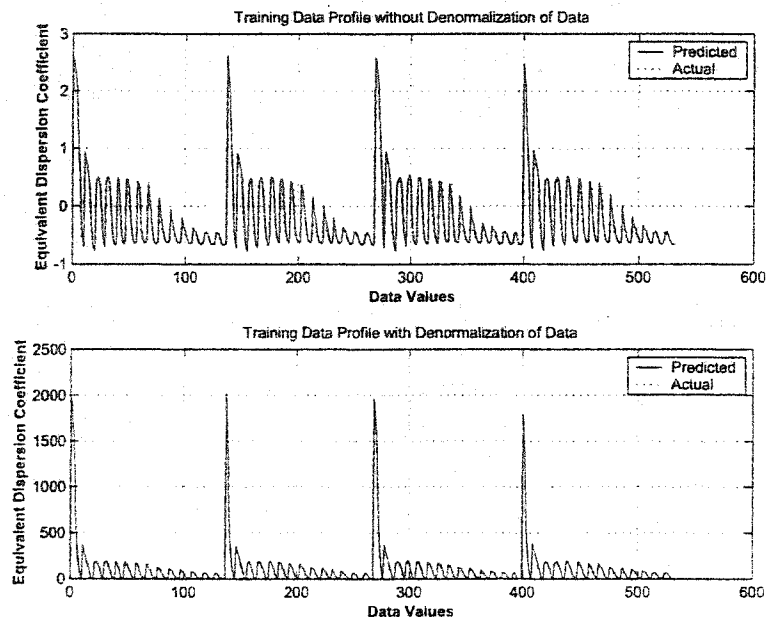
Figure 4.45: Neural network model training using Backpropagation algorithm for T=3.1, R=1.026, V=10 cm/day
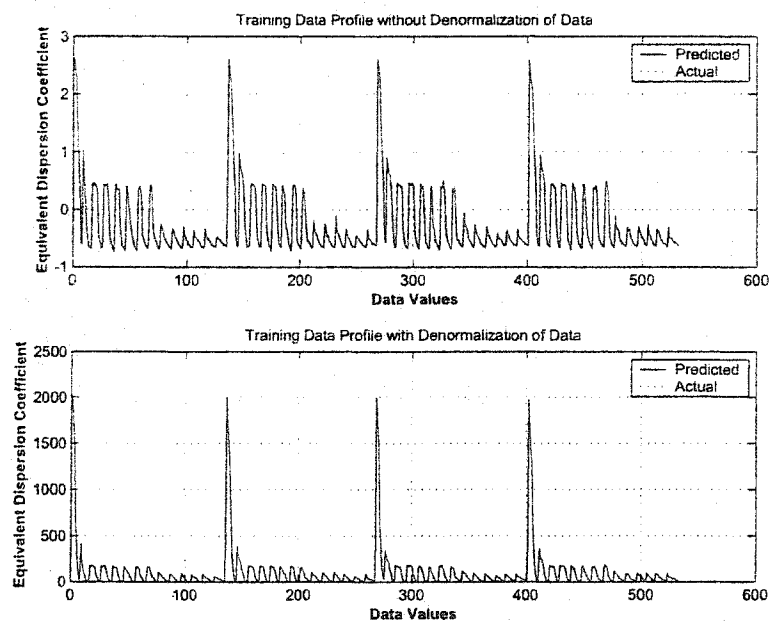


Figure 4.46: Neural network model training using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=10 cm/day
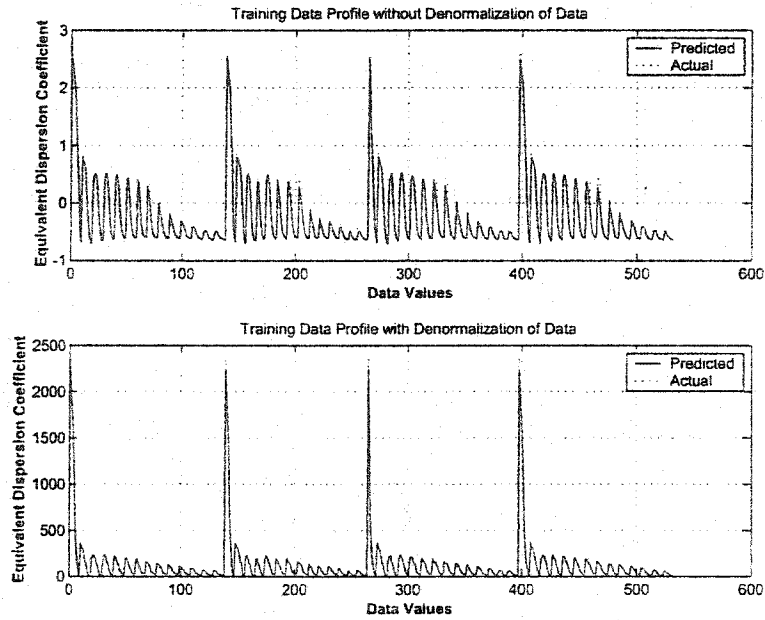
Figure 4.47: Neural network model training using Backpropagation algorithm for T=3.1, R=1.026, V=12 cm/day
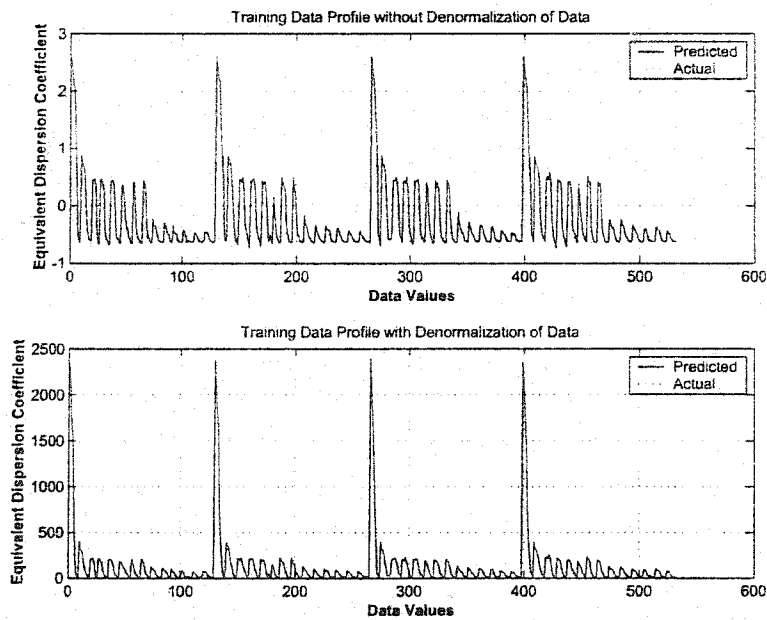


Figure 4.48: Neural network model training using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=12 cm/day

# Chapter 5

# VALIDATION

To test how well artificial neural networks (ANNs) models were trained, it was important to perform validation. Validation is a process by which the performance of a trained network is tested against examples that were not included in the training set.

## 5.1 Validation Using Synthetic Data

Both models are validated using both algorithms and using data which has not been seen by the ANNs. The validation of ANNs for both algorithms are shown in Figure 5.1 - 5.40 for particular values of pore volume. The EDC values predicted by ANNs are found to be within the tolerance limits. It is obserevd that LM ANNs performs much better as compared to the BP ANNs.
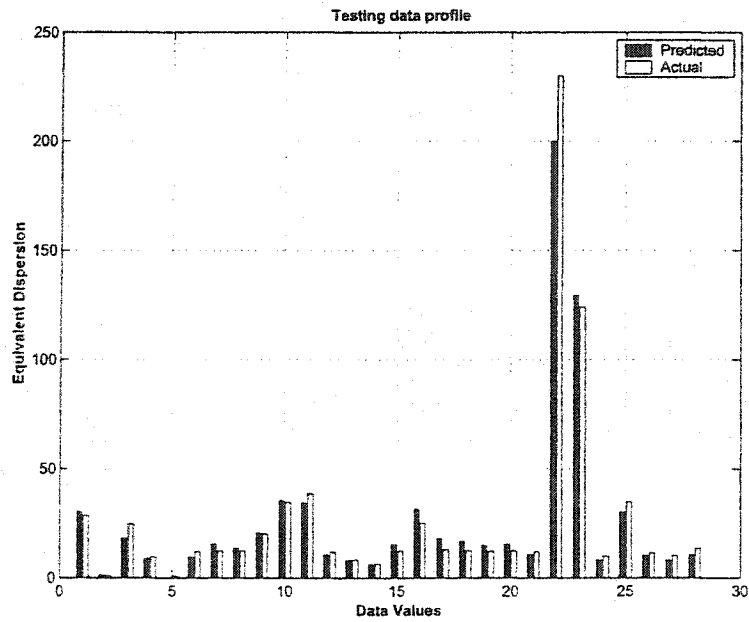
Figure 5.1: Validation of neural network model using Backpropagation algorithm for T=0.5, R = 1.026, V=3 cm/day
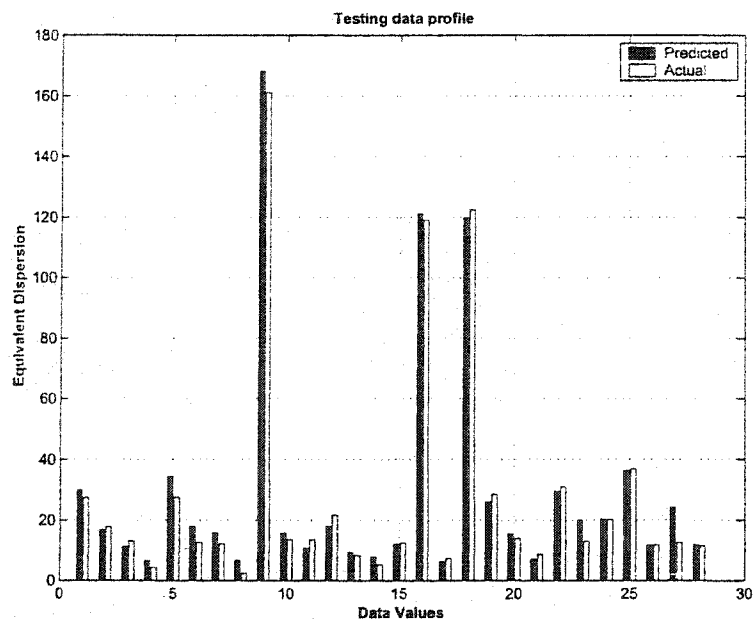


Figure 5.2: Validation of neural network model using Levenberg Marquardt algorithm for T=0.5, R = 1.026, V=3 cm/day
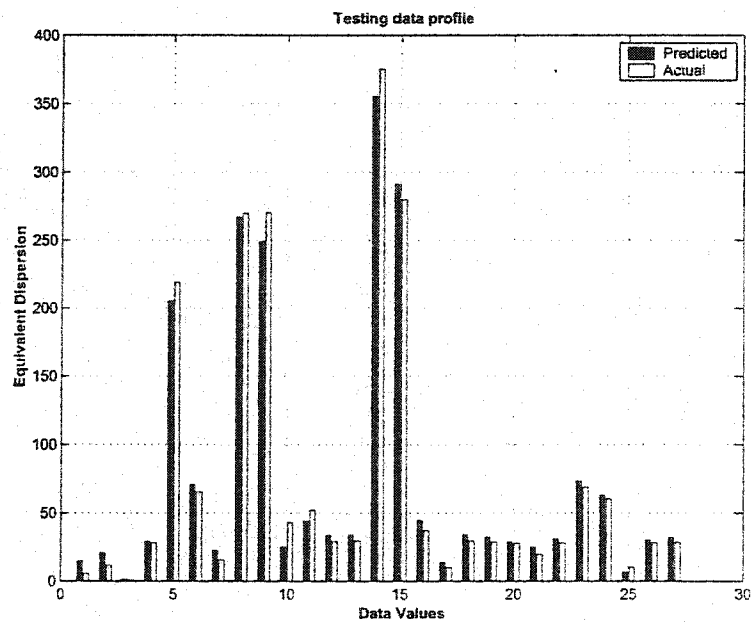
Figure 5.3: Validation of neural network model using Backpropagation algorithm for T=0.5, R = 1.026, V=7 cm/day
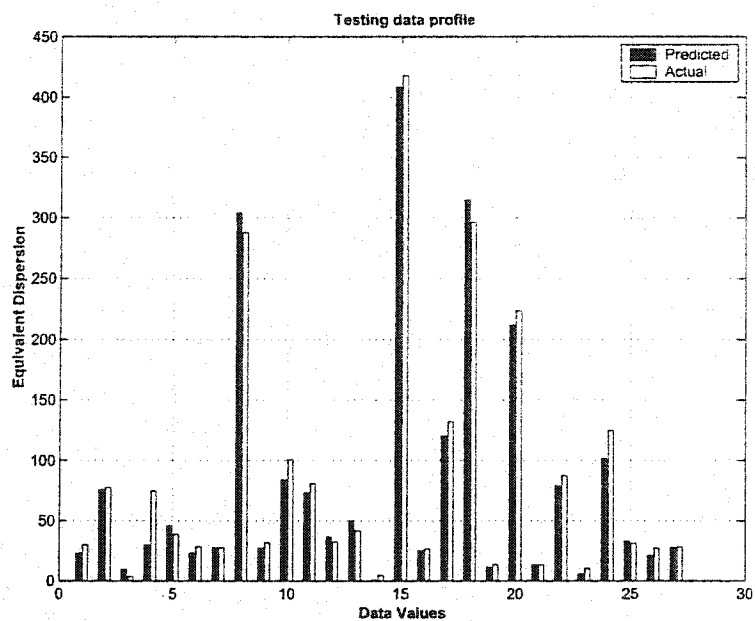


Figure 5.4: Validation of neural network model using Levenberg Marquardt algorithm for T=0.5, R = 1.026, V=7 cm/day
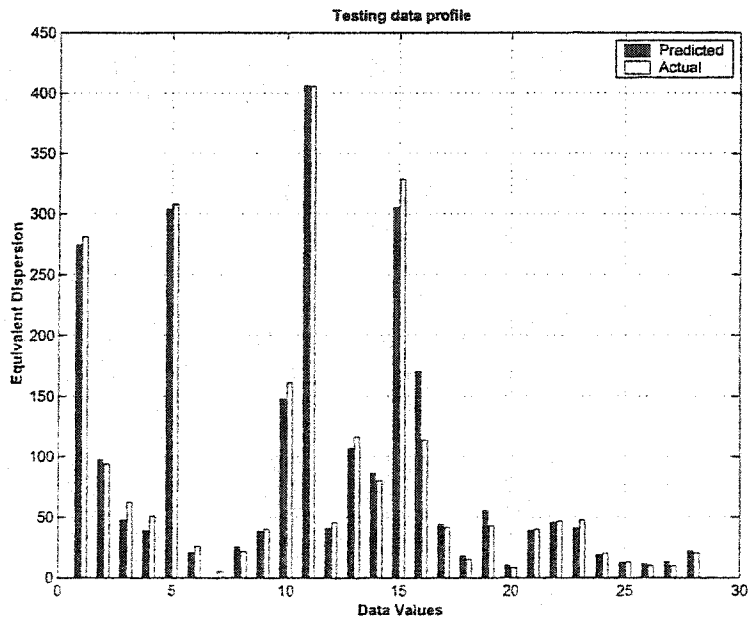
Figure 5.5: Validation of neural network model using Backpropagation algorithm for T=0.5, R=1.026, V=10 cm/day
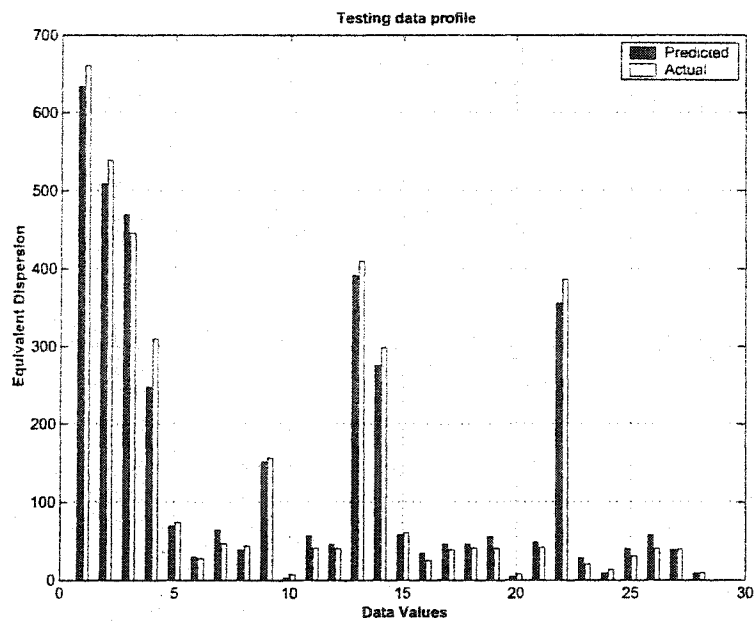


Figure 5.6: Validation of neural network model using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=10 cm/day
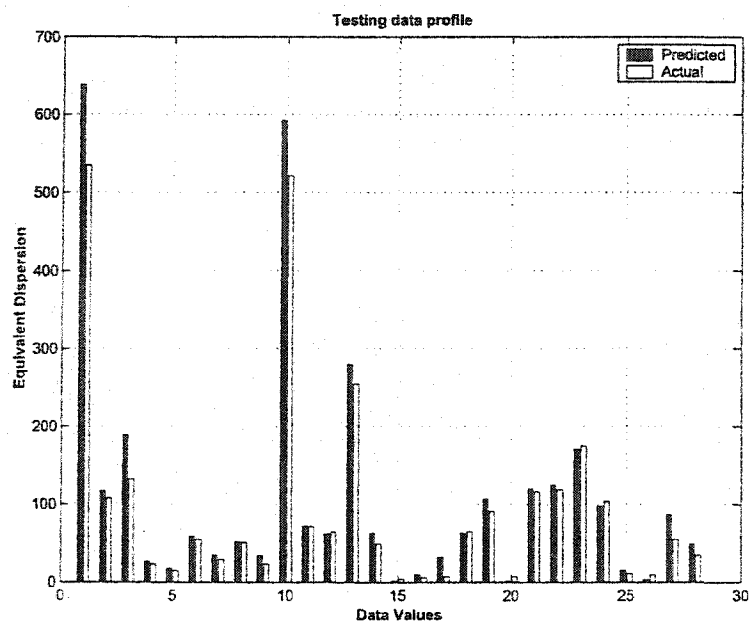
Figure 5.7: Neural neural network model Training using Backpropagation algorithm for T=0.5, R=1.026, V=12 cm/day
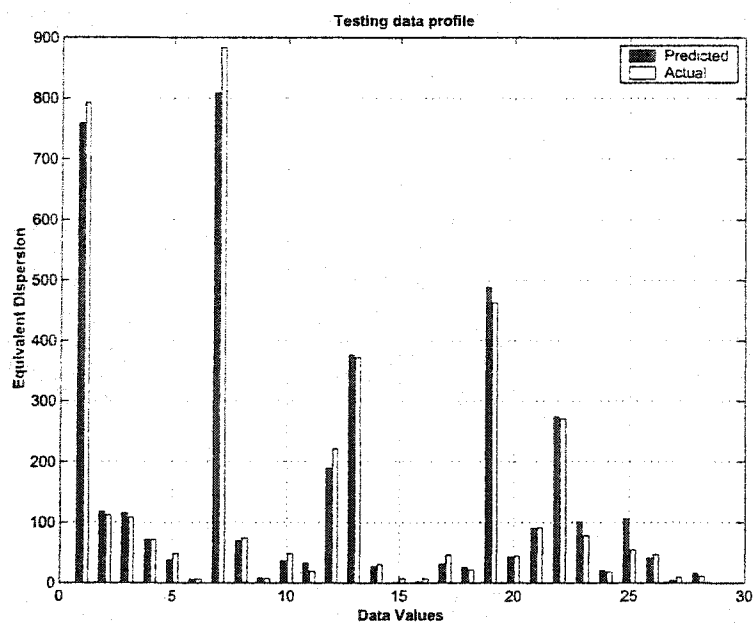


Figure 5.8: Validation of neural network model using Levenberg Marquardt algorithm for T=0.5, R=1.026, V=12 cm/day
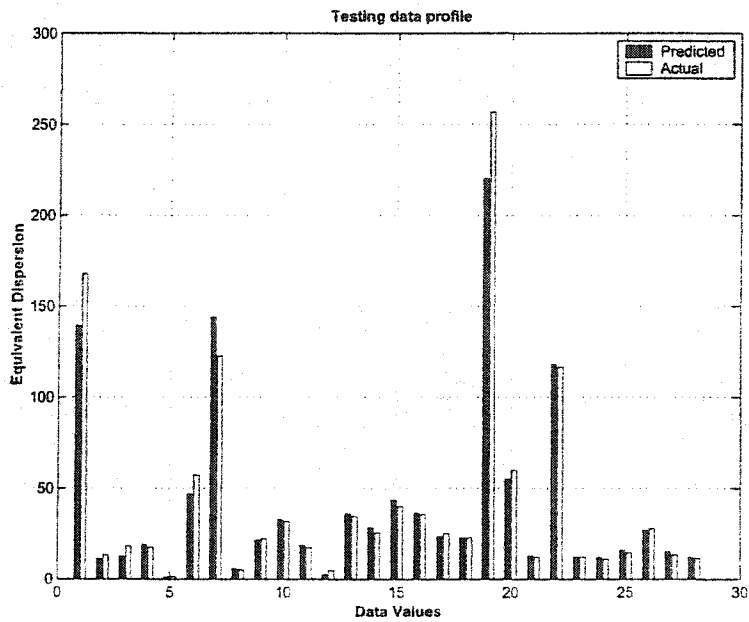
Figure 5.9: Validation of neural network model using Backpropagation algorithm for T=1.0, R=1.026, V=3 cm/day
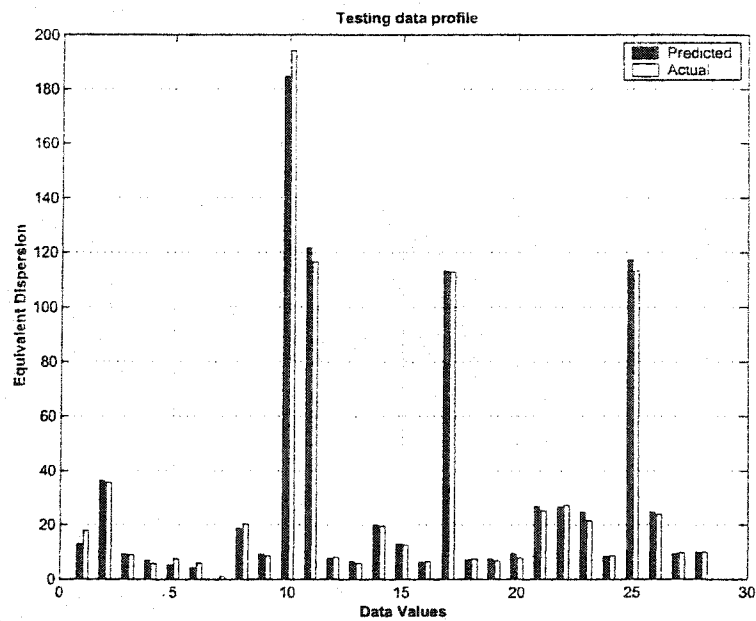


Figure 5.10: Validation of neural network model using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=3 cm/day
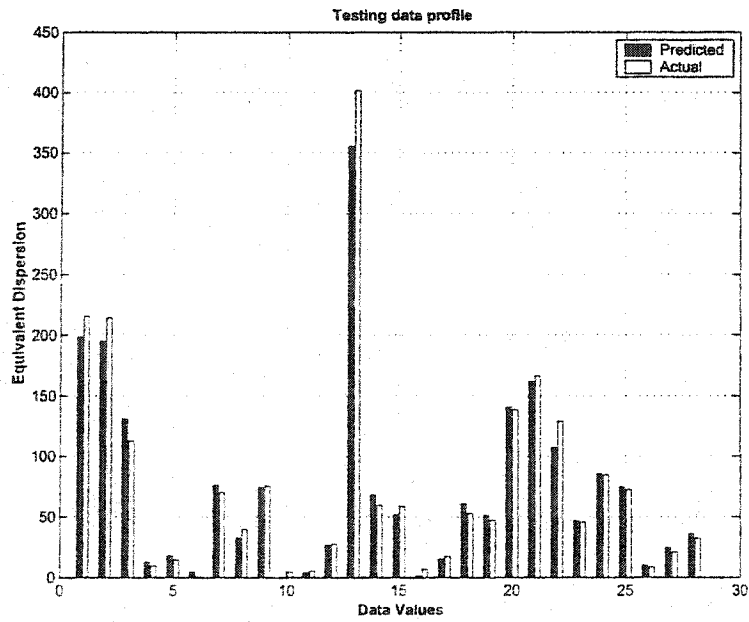
Figure 5.11: Validation of neural network model using Backpropagation algorithm for T=1.0, R=1.026, V=7 cm/day
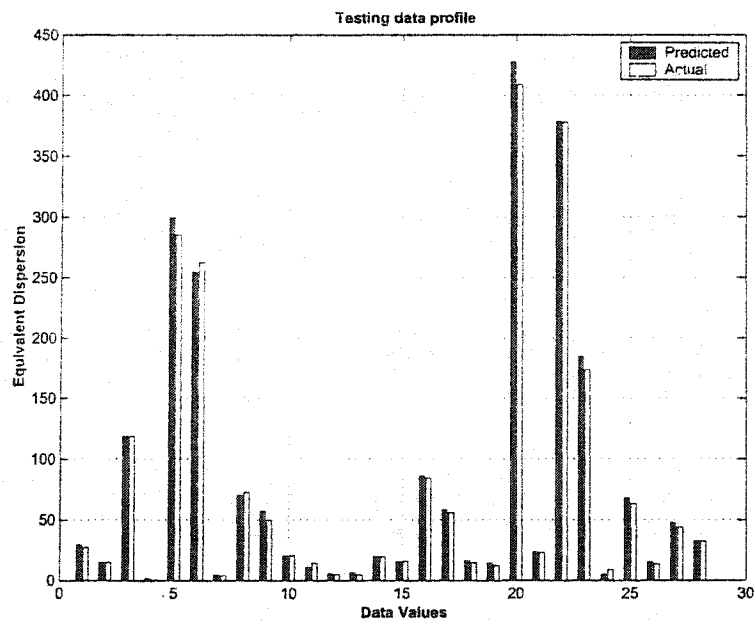


Figure 5.12: Validation of neural network model using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=7 cm/day
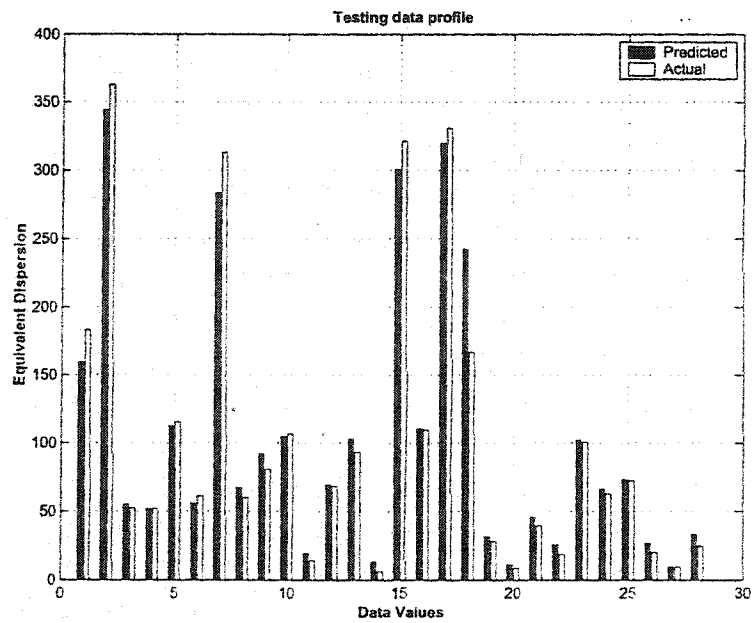
Figure 5.13: Validation of neural network model using Backpropagation algorithm for T=1.0, R=1.026, V=10 cm/day
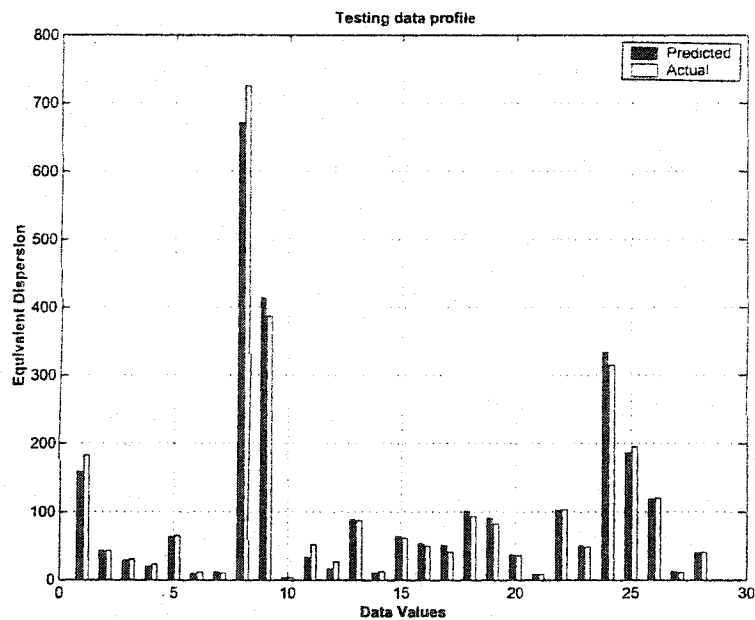


Figure 5.14: Validation of neural network model using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=10 cm/day
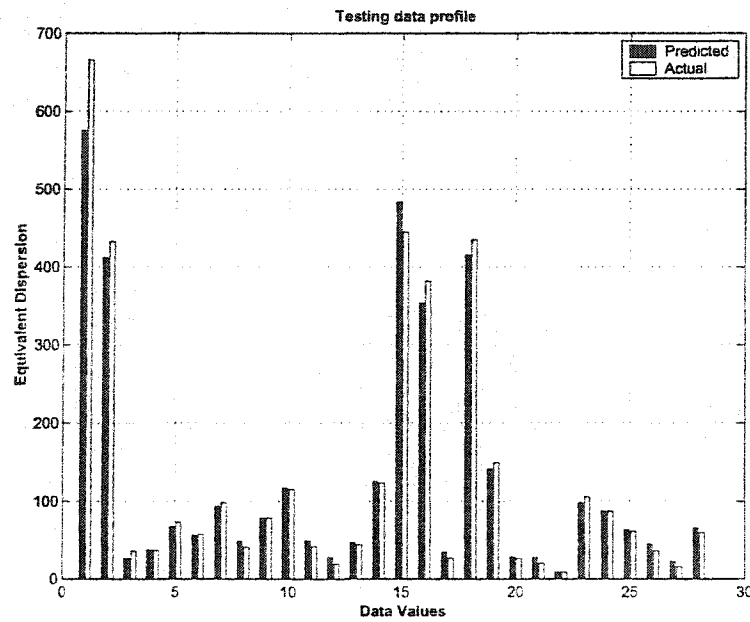
Figure 5.15: Validation of neural network model using Backpropagation algorithm for T=1.0, R=1.026, V=12 cm/day
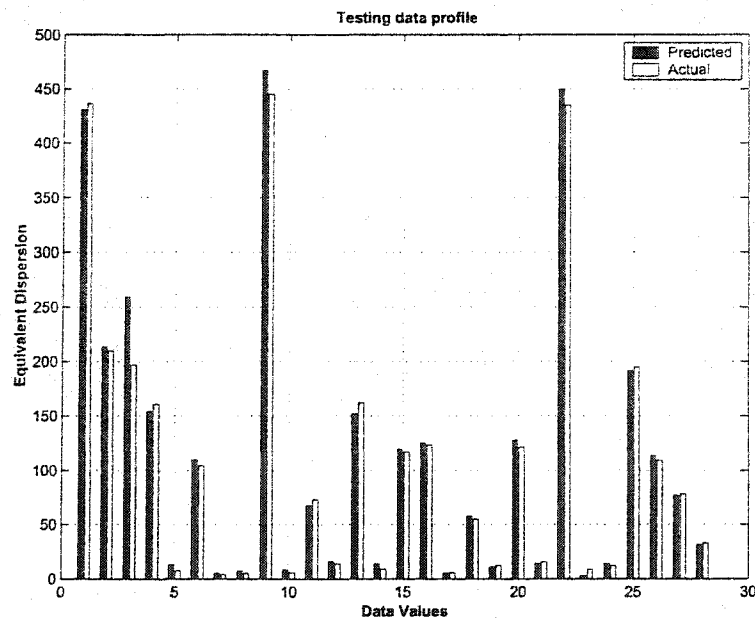


Figure 5.16: Validation of neural network model using Levenberg Marquardt algorithm for T=1.0, R=1.026, V=12 cm/day
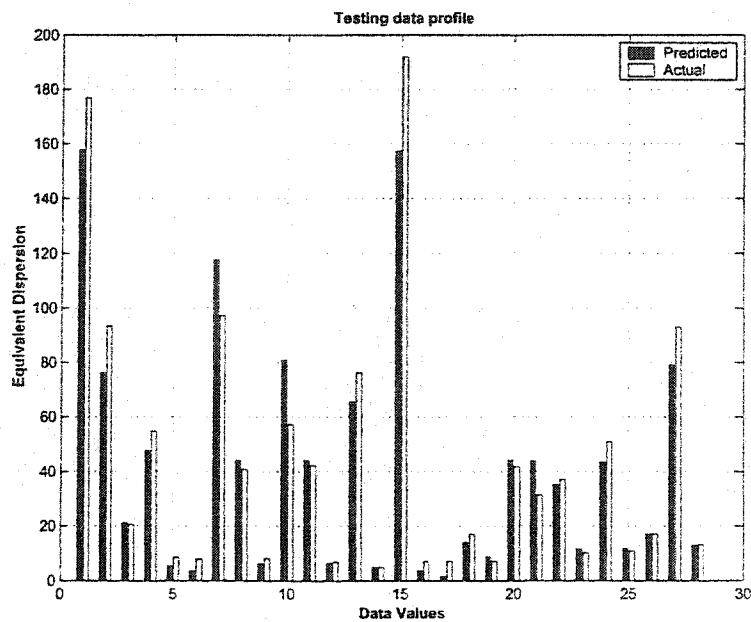
Figure 5.17: Validation of neural network model using Backpropagation algorithm for T=1.8, R=1.026, V=3 cm/day
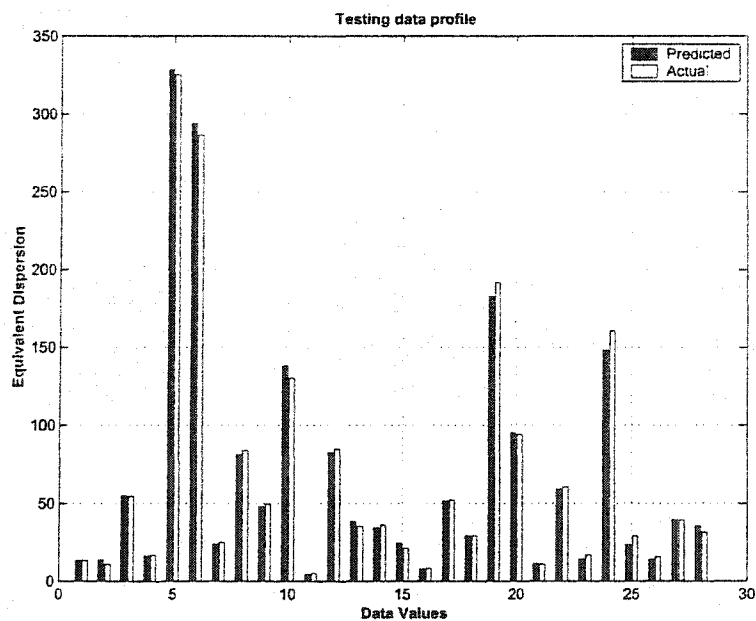


Figure 5.18: Validation of neural network model using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=3 cm/day
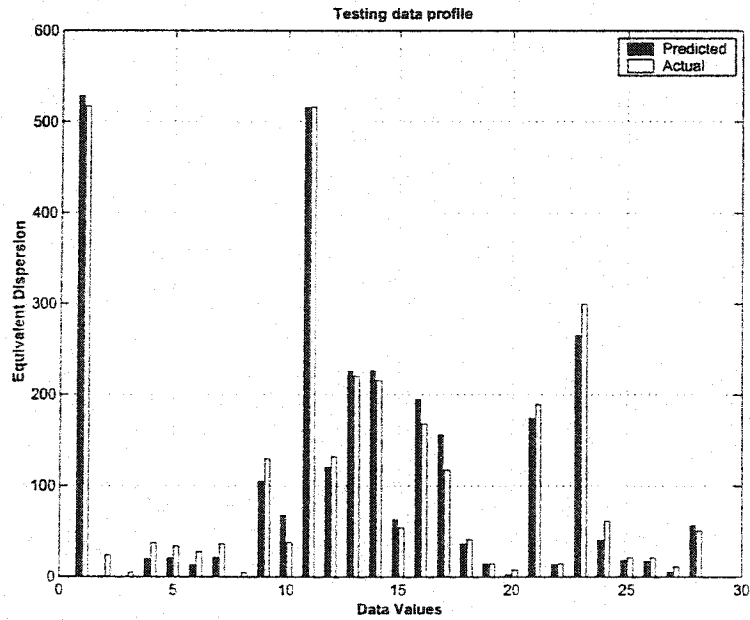
Figure 5.19: Validation of neural network model using Backpropagation algorithm for T=1.8, R=1.026, V=7 cm/day
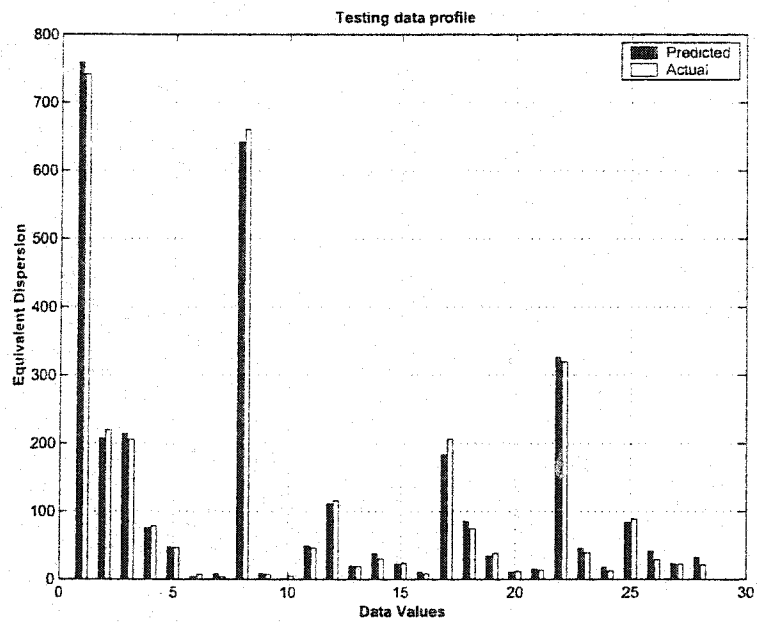


Figure 5.20: Validation of neural network model using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=7 cm/day
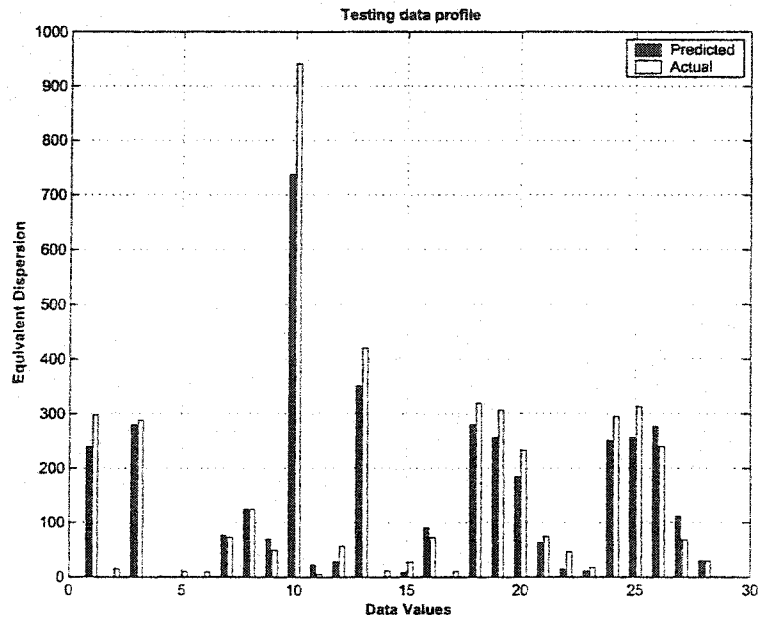
Figure 5.21: Validation of neural network model using Backpropagation algorithm for T=1.8, R=1.026, V=10 cm/day
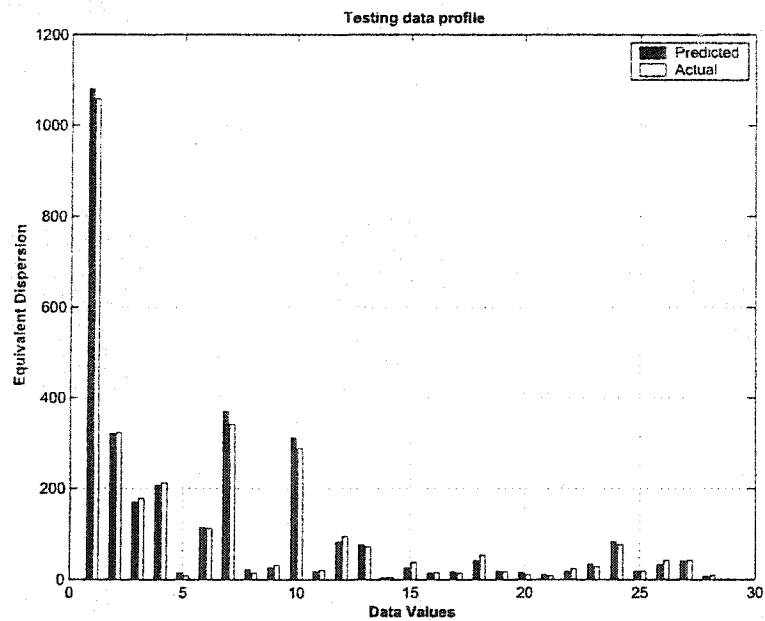


Figure 5.22: Validation of neural network model using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=10 cm/day
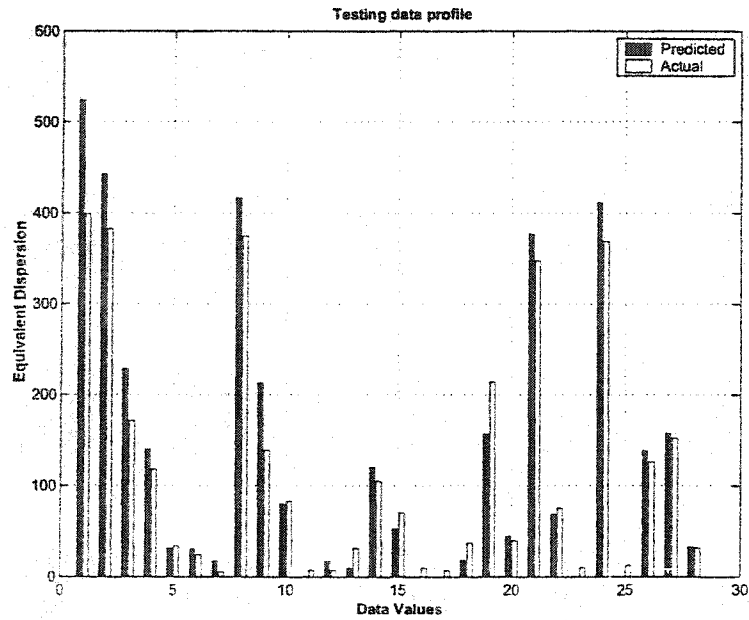
Figure 5.23: Validation of neural network model using Backpropagation algorithm for T=1.8, R=1.026, V=12 cm/day
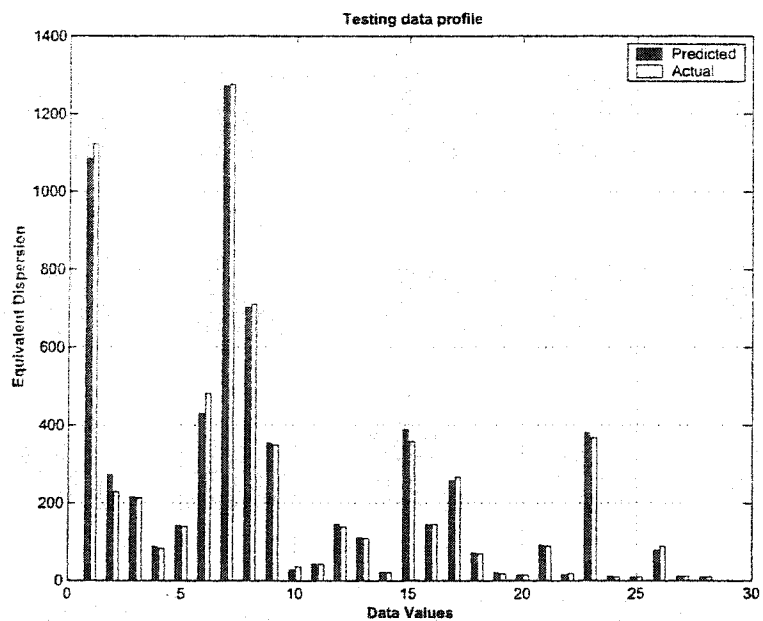


Figure 5.24: Validation of neural network model using Levenberg Marquardt algorithm for T=1.8, R=1.026, V=12 cm/day
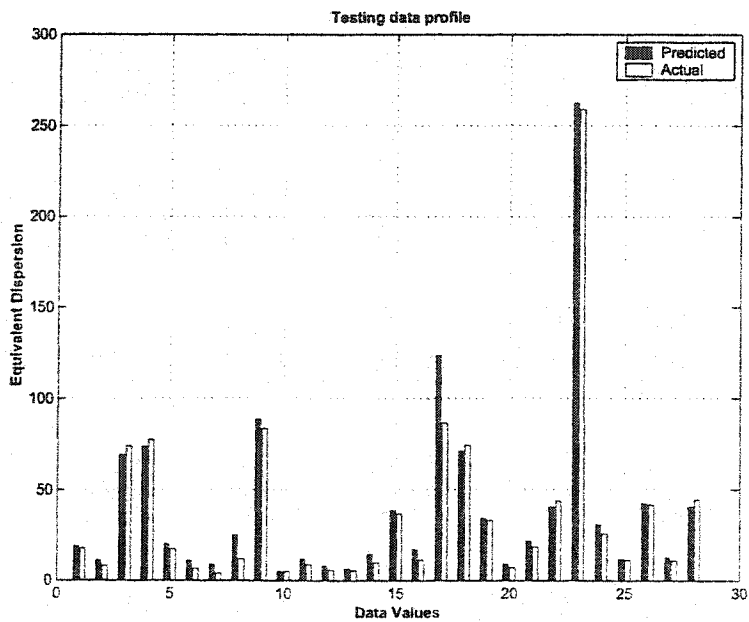
Figure 5.25: Validation of neural network model using Backpropagation algorithm for T=2.2, R=1.026, V=3 cm/day
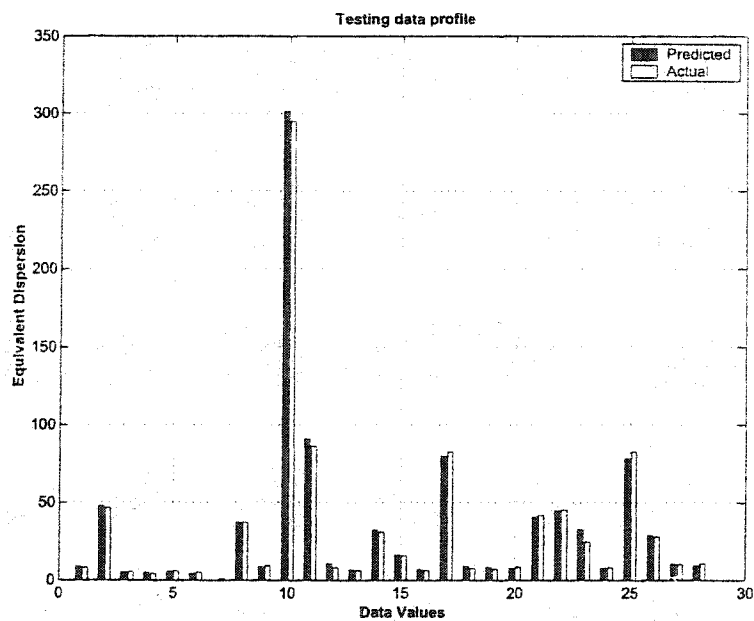


Figure 5.26: Validation of neural network model using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=3 cm/day
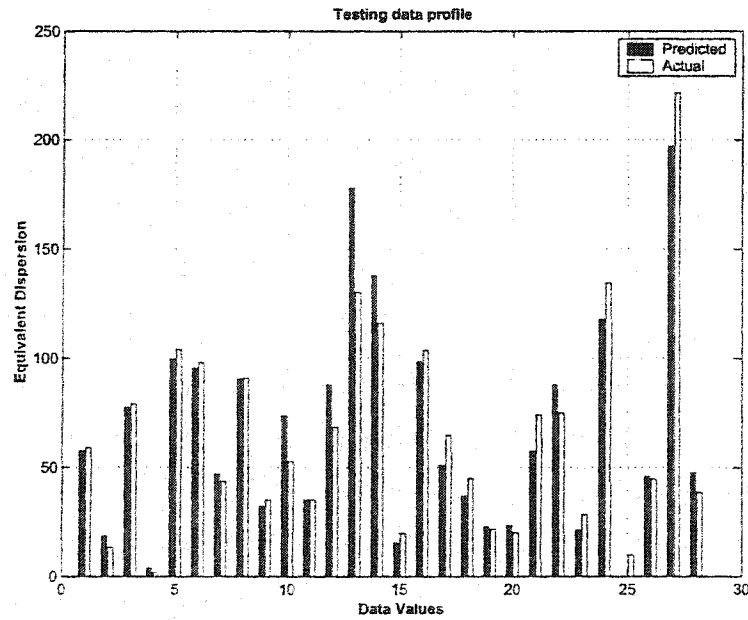
Figure 5.27: Validation of neural network model using Backpropagation algorithm for T=2.2, R=1.026, V=7 cm/day
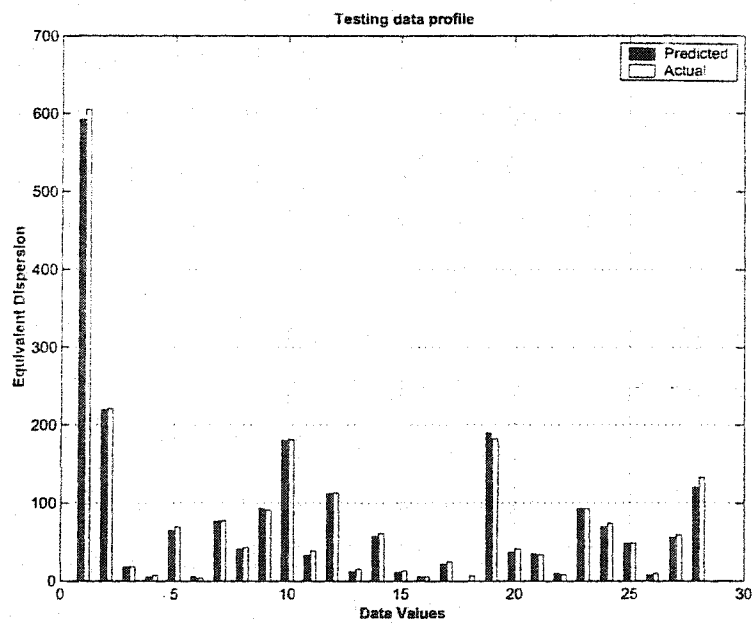


Figure 5.28: Validation of neural network model using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=7 cm/day
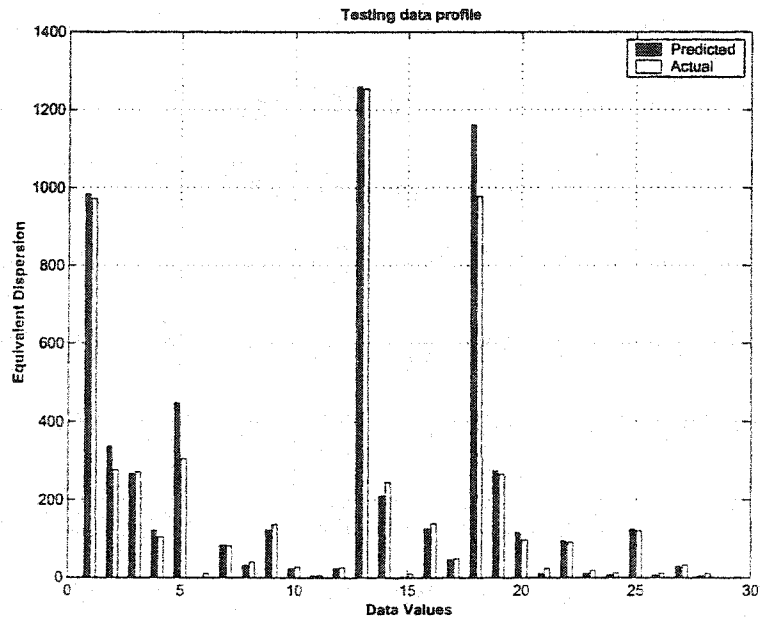
Figure 5.29: Validation of neural network model using Backpropagation algorithm for T=2.2, R=1.026, V=10 cm/day
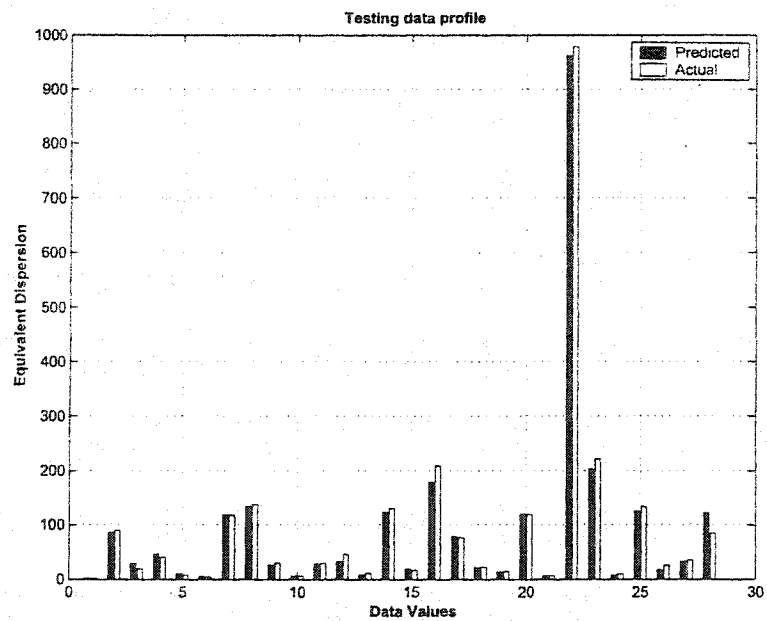


Figure 5.30: Validation of neural network model using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=10 cm/day
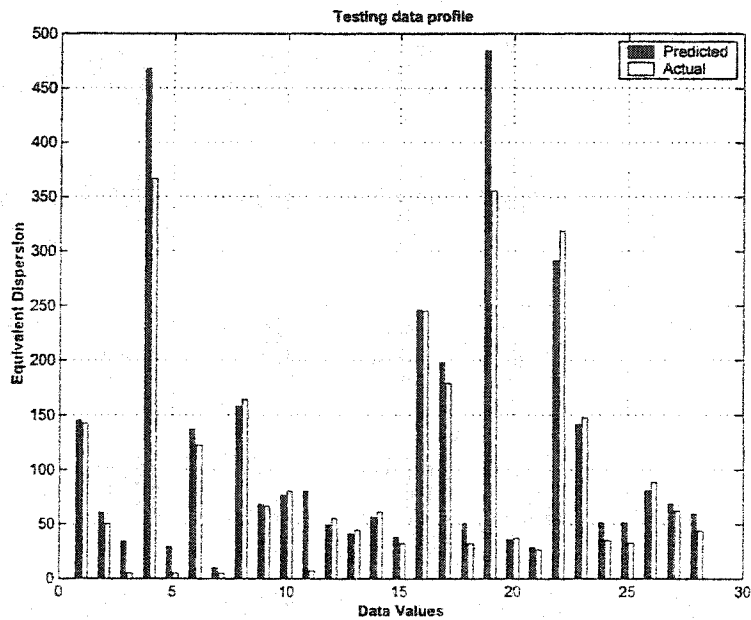
Figure 5.31: Validation of neural network model using Backpropagation algorithm for T=2.2, R=1.026, V=12 cm/day
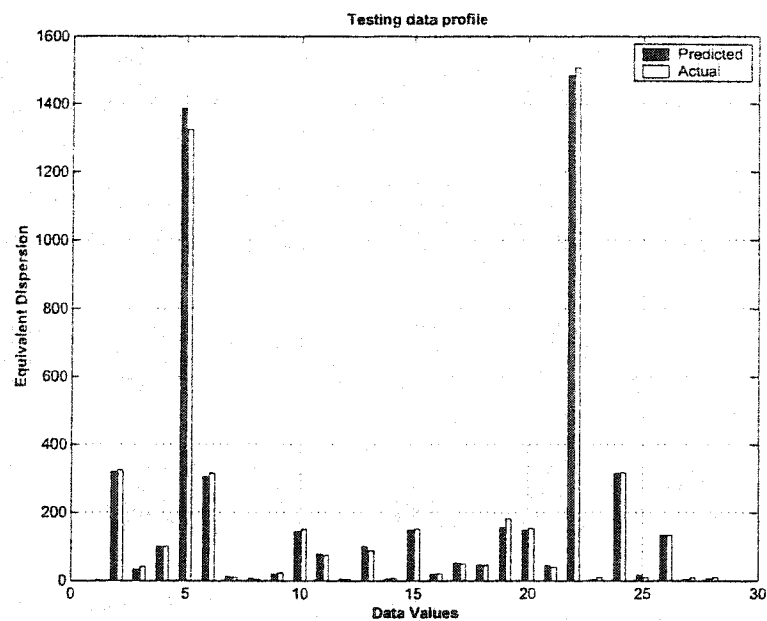


Figure 5.32: Validation of neural network model using Levenberg Marquardt algorithm for T=2.2, R=1.026, V=12 cm/day
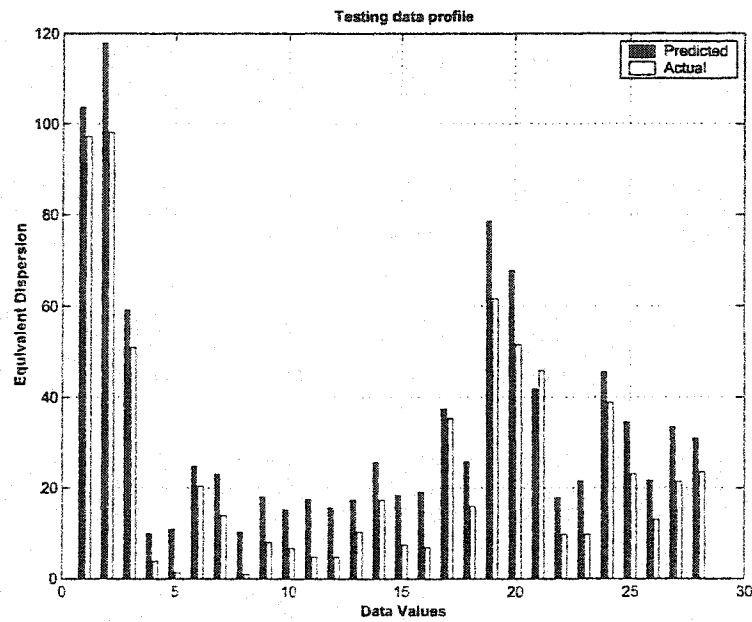
Figure 5.33: Validation of neural network model using Backpropagation algorithm for T=3.1, R=1.026, V=3 cm/day
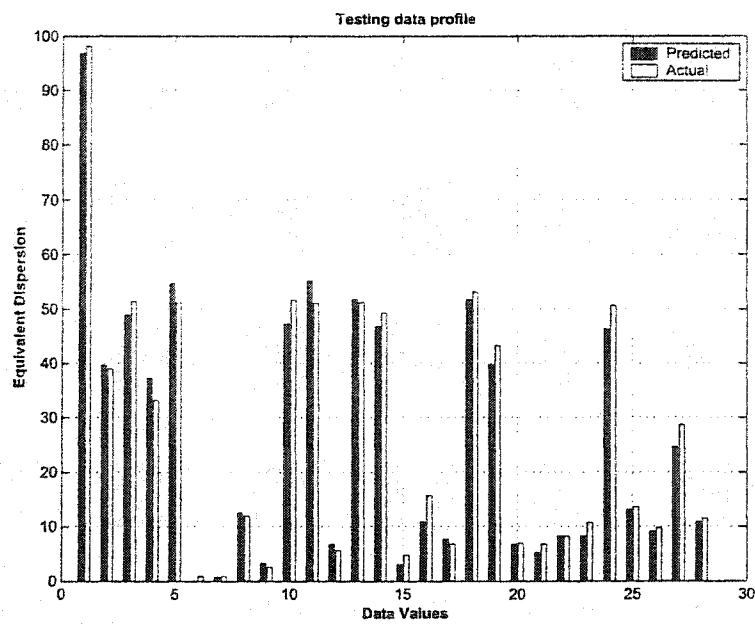


Figure 5.34: Validation of neural network model using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=3 cm/day
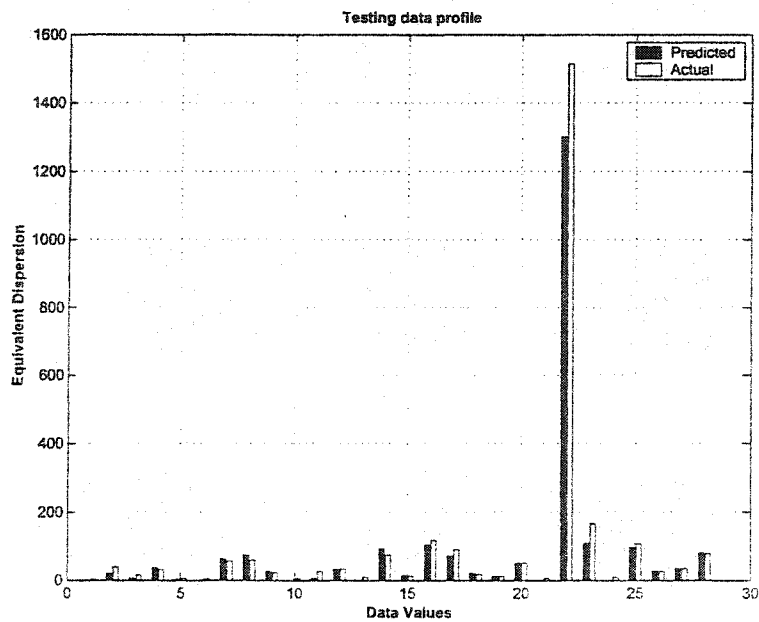
Figure 5.35: Validation of neural network model using Backpropagation algorithm for T=3.1, R=1.026, V=7 cm/day
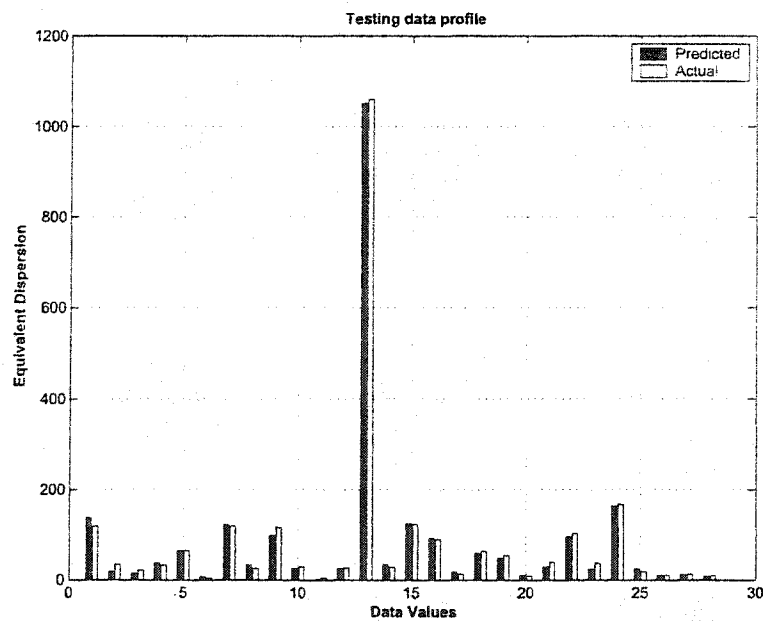


Figure 5.36: Validation of neural network model using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=7 cm/day
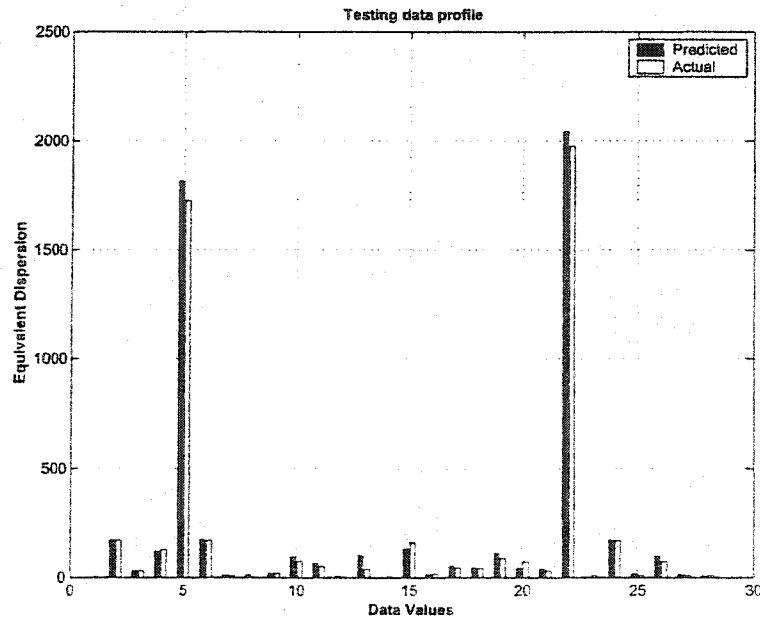
Figure 5.37: Validation of neural network model using Backpropagation algorithm for T=3.1, R=1.026, V=10 cm/day
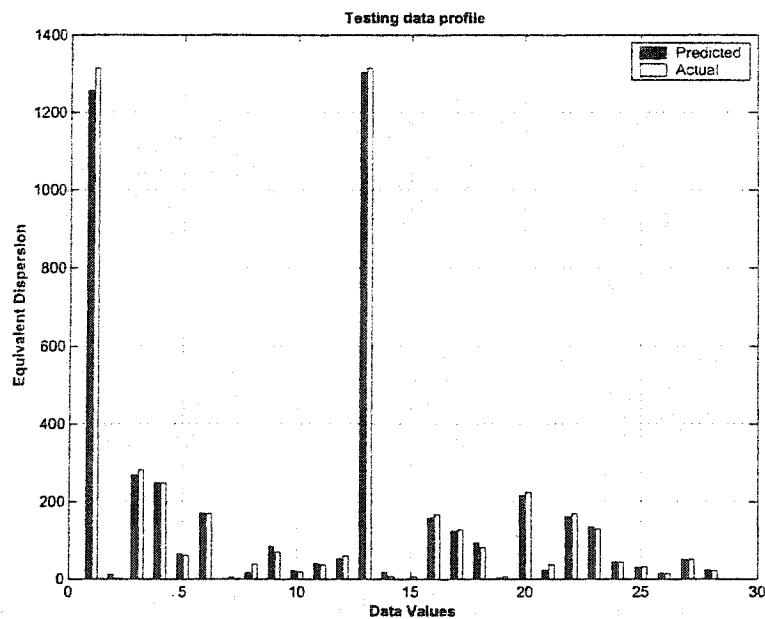


Figure 5.38: Validation of neural network model using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=10 cm/day
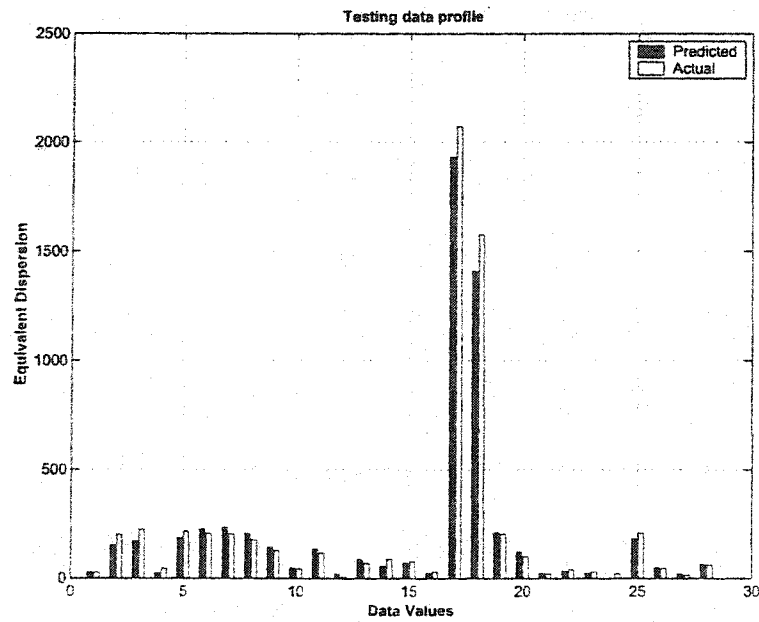
Figure 5.39: Validation of neural network model using Backpropagation algorithm for T=3.1, R=1.026, V=12 cm/day
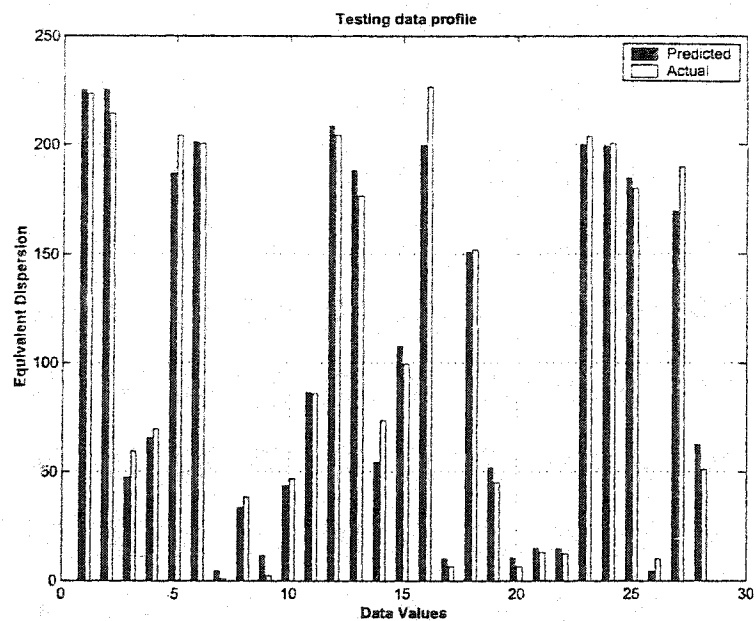


Figure 5.40: Validation of neural network model using Levenberg Marquardt algorithm for T=3.1, R=1.026, V=12 cm/day

## 5.2  Validation using Experimental data

The trained ANNs models are also validated using experimental data.

## 5.3  Experimental Data 1

A column study was conducted by Van Genuchten [31] to understand the movement of contaminants within porous media. These experiments are used here for validation.

### 5.3.1  Glendale clay Loam Soil

Glendale clay loam soil was used in that study. It is a sub sample of the calcareous Glendale series, a fine-silty, mixed thermic type torrifluvent. The Tables 5.1 and 5.2 summarize some physical and chemical properties of the Glendale clay loam.

| Analysis | Amount |
|---|---|
| Sand (percentage) | 38.8 |
| Silt (percentage) | 29.8 |
| Clay (percentage) | 31.4 |
| CEC meq/100g | 31.1 |
| pH | 7.7 |
| $CaCo_3$ equivalent (percentage) | 7.0 |
| Gypsum (percentage) | 0.0 |

Table 5.1: Physical and chemical properties of Glendale clay loam Soil

### 5.3.2  Tracers and their analyis

The following tracer was used in this study:

| Analysis | Amount |
|----------|--------|
| Ca | 22.0 |
| Mg | 3.6 |
| Na | 0.8 |
| K | 1.5 |
| NH$_4$ | 1.2 |

Table 5.2: Ex-changeable (meq/100g) ions of Glendale clay loam Soil

## Tritium

Ten milli curie (mC) of tritiated water was added to each liter of 0.01 N CaCl$_2$ to obtain a final activity of 23.000 cpm.

## 2,4,5-Trichlorophenoxyacetic acid

Ten ppm herbicide solutions were used in the column experiments. Radioactive $2, 4, 5, -T$ with a specific activity of 4.93 $\mu Ci/mg$, labelled at the carboxyl carbon position was added to unlabelled solution making up the 10 ppm, to aid in the herbicide analysis.

## 5.3.3  Column Studies

Soil columns were prepared by carefully packing air-dried Glendale clay loam soil material into plexiglass cylinders of 5.4 cm inner diameter. Before introducing the first tracer, the columns were leached with deliberate amount of 0.01 N $CaCl_2$ to assure a physical equilibrium and as much as possible a chemical equilibrium. After establishing steady state flow conditions, the leaching solution was replaced with a solution containing one of the two tracers used in the study. The soil physical data

for the different experiments are given in table 5.3. In table 5.4 the various tritium experiments are summarized together with the values of different parameters.

The trained ANNs models are used for validation. The LM ANNs proved to be more effective than the BP ANNs. The EDC values obtained from these ANNs model are then used in the linear equilibrium model to simulate the solute transport. Figure 5.41 - 5.50 show the comparison of breakthrough curves obtained from both ANNs, inverse problem and two region model. The validation results are given in table 5.5.

| Experiment No. | Tracer | Bulk Density $\rho$ $(g/cm^3)$ | Water Content $\theta$ $cm^3/cm^3$ | Flux q (cm/day) | Pulse Period $T_1$ (-) | Pulse Period $t_1$ (days) | Largest Aggregate Size (mm) |
|---|---|---|---|---|---|---|---|
| 1 | $^3H$ | 1.360 | 0.4605 | 5.093 | 2.2 | 5.60 | 2.0 |
| 2 | $^3H$ | 1.360 | 0.4605 | 5.093 | 0.512 | 1.389 | 2.0 |
| 3 | $^3H$ | 1.360 | 0.4601 | 4.334 | 2.2 | 7.007 | 2.0 |
| 4 | $^3H$ | 1.361 | 0.4637 | 4.552 | 2.2 | 6.723 | 2.0 |
| 5 | $^3H$ | 1.361 | 0.4675 | 4.567 | 0.5 | 1.535 | 2.0 |
| 6 | $^3H$ | 1.361 | 0.4670 | 2.749 | 1.5 | 7.645 | 2.0 |
| 7 | $^3H$ | 1.222 | 0.4452 | 4.508 | 1.8 | 5.332 | 6.3 |
| 8 | $^3H$ | 1.222 | 0.4337 | 2.008 | 1.80 | 9.653 | 6.3 |
| 9 | $^3H$ | 1.309 | 0.4333 | 4.040 | 1.80 | 6.021 | 6.3 |
| 10 | $^3H$ | 1.126 | 0.3933 | 2.760 | 3.1 | 13.25 | 6.3 |

Table 5.3: Soil physical data for various displacements through Glendale clay loam

| Experiment No. | Tracer | Velocity $v$ (cm/day) | $\beta$ (-) | $R$ (-) | $P$ (-) | $\omega$ (-) | $\phi$ (-) | $v_m$ (cm²/day) | $D$ (1/day) | $\alpha$ (mm) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $^3H$ | 11.0597 | 0.926 | 1.0266 | 95 | 1.473 | 0.940 | 11.765 | 3.71 | 0.250 |
| 2 | $^3H$ | 11.0597 | 0.926 | 1.0266 | 95 | 1.473 | 0.940 | 11.765 | 3.21 | 0.250 |
| 3 | $^3H$ | 9.419 | 0.928 | 1.0266 | 95 | 0.564 | 0.942 | 10.00 | 3.15 | 0.048 |
| 4 | $^3H$ | 9.8166 | 0.850 | 1.0264 | 45 | 1.496 | 0.862 | 11.388 | 7.59 | 0.227 |
| 5 | $^3H$ | 9.7689 | 0.887 | 1.0264 | 45 | 1.496 | 0.862 | 10.854 | 7.25 | 0.227 |
| 6 | $^3H$ | 5.886 | 0.830 | 1.0262 | 45 | 1.017 | 0.841 | 7.00 | 7.78 | 0.080 |
| 7 | $^3H$ | 10.126 | 0.872 | 1.0247 | 56 | 0.722 | 0.884 | 12.00 | 6.138 | 0.101 |
| 8 | $^3H$ | 5.074 | 0.717 | 1.0246 | 56 | 1.960 | 0.725 | 7.00 | 3.73 | 0.086 |
| 9 | $^3H$ | 9.323 | 0.798 | 1.0272 | 57 | 0.928 | 0.788 | 11.83 | 6.23 | 0.125 |
| 10 | $^3H$ | 7.017 | 0.694 | 1.0258 | 35 | 1.460 | 0.702 | 10.00 | 8.56 | 0.111 |

Table 5.4: Summary of various tritium ($^3H_2O$) displacements through Glendale clay loam

| No. | Tracer | EDC from Inverse | EDC from Backpropagation | EDC from Levenberg Marquardt |
|---|---|---|---|---|
| 1 | $^3H_2O$ | 5.10 | 7.87 | 7.29 |
| 2 | $^3H_2O$ | 5.16 | 5.54 | 5.92 |
| 3 | $^3H_2O$ | 5.43 | 8.22 | 9.354 |
| 4 | $^3H_2O$ | 12.81 | 11.53 | 10.95 |
| 5 | $^3H_2O$ | 11.03 | 9.60 | 10.33 |
| 6 | $^3H_2O$ | 13.54 | 11.62 | 10.79 |
| 7 | $^3H_2O$ | 12.44 | 40.19 | 36.76 |
| 8 | $^3H_2O$ | 12.07 | 12.96 | 12.82 |
| 9 | $^3H_2O$ | 19.04 | 34.136 | 26.46 |
| 10 | $^3H_2O$ | 26.26 | 25.2 | 26.46 |

Table 5.5: Values of EDC obtained by solving Inverse problem and from both algorithms
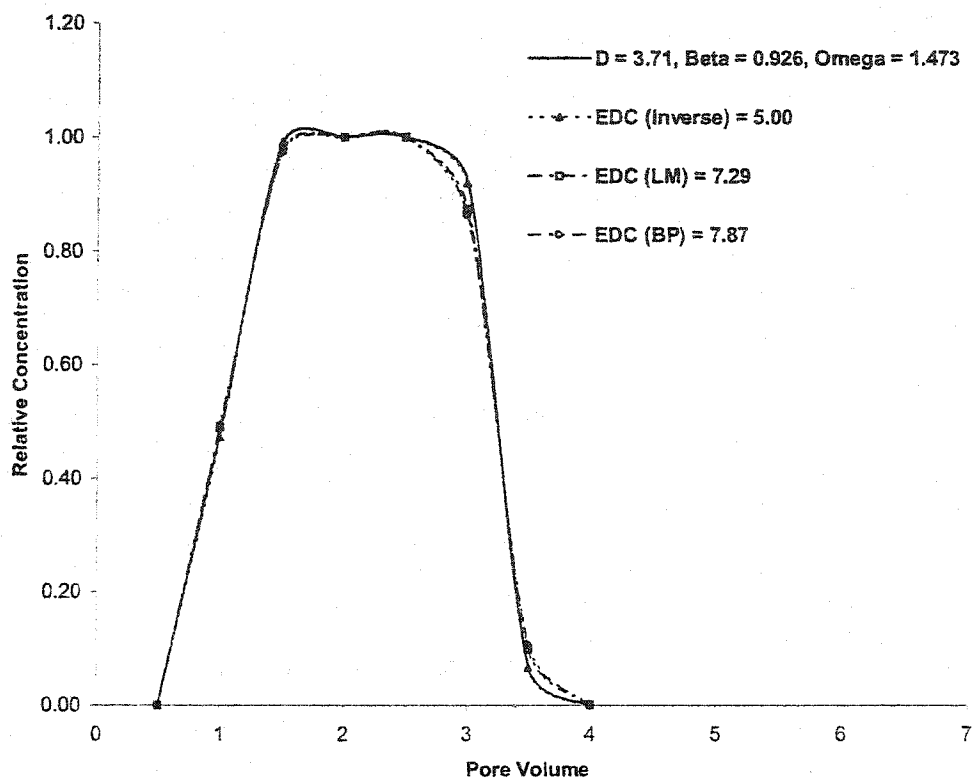
Figure 5.41: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
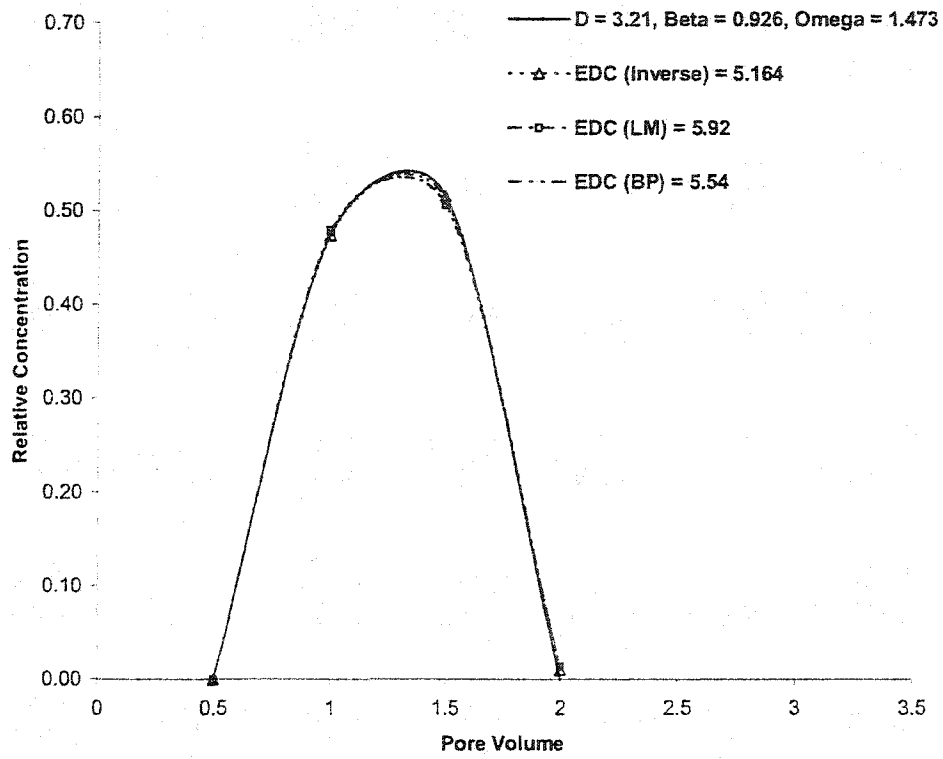
Figure 5.42: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
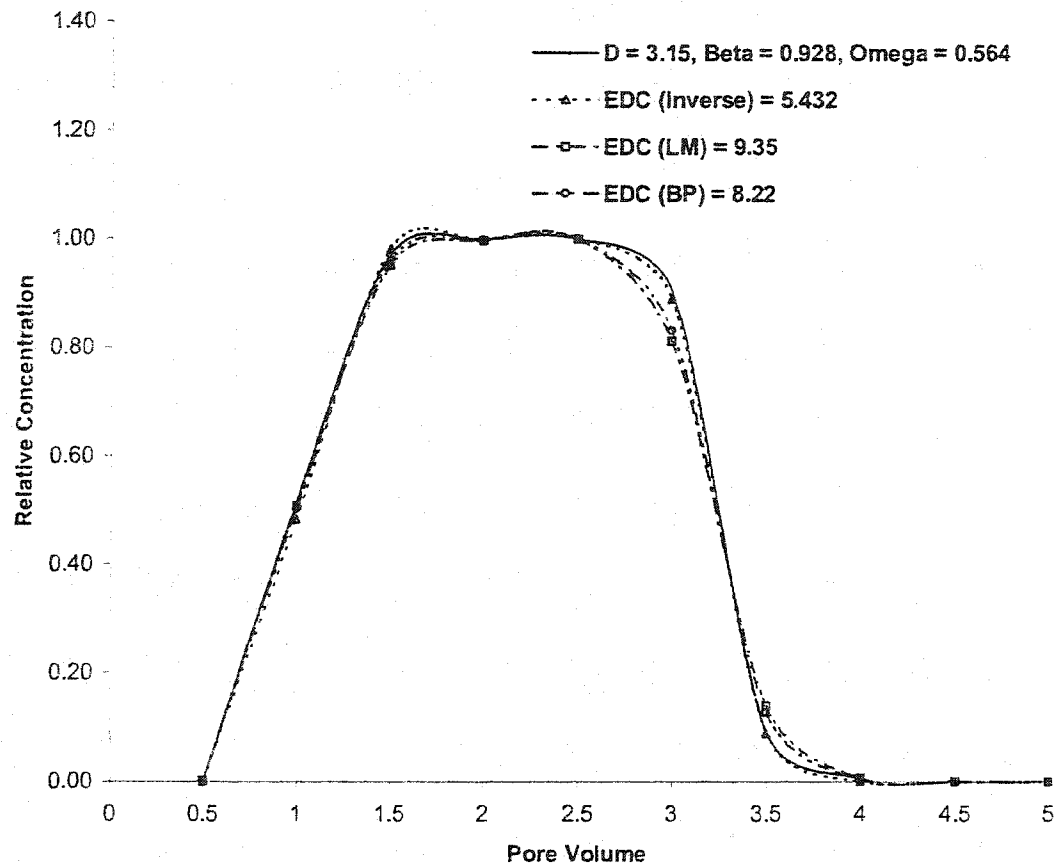
Figure 5.43: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
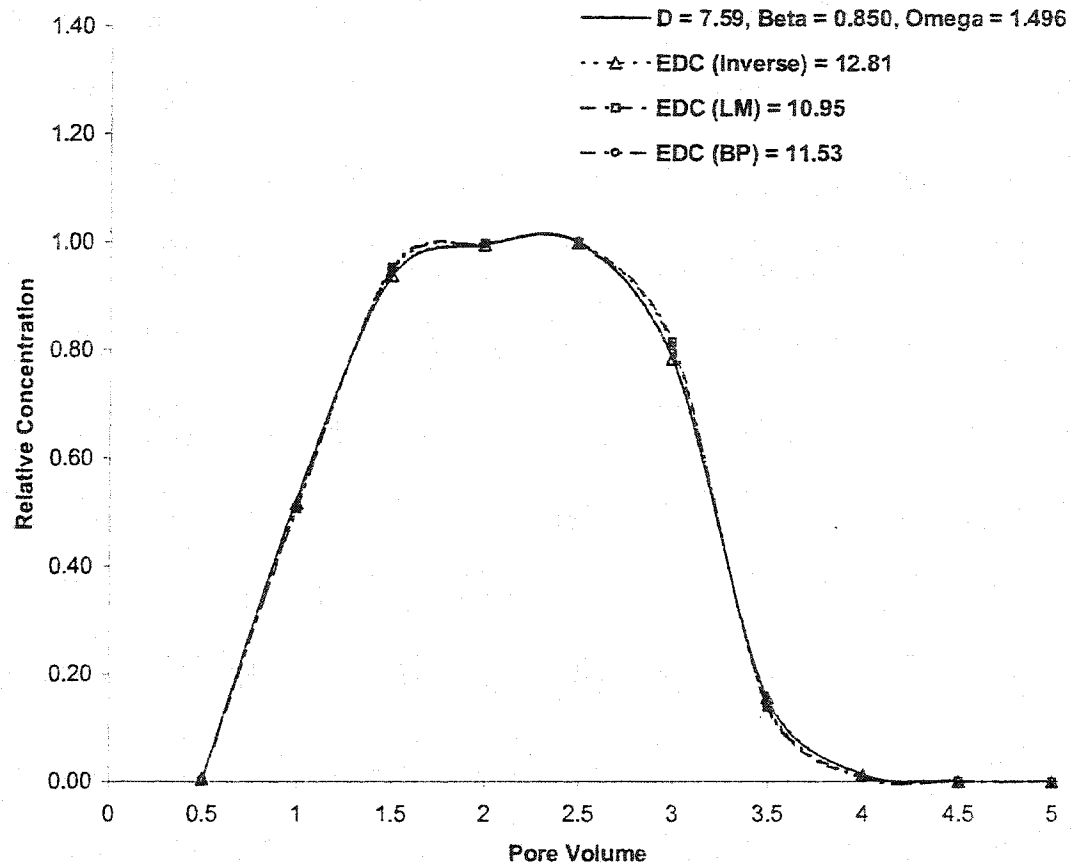
Figure 5.44: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
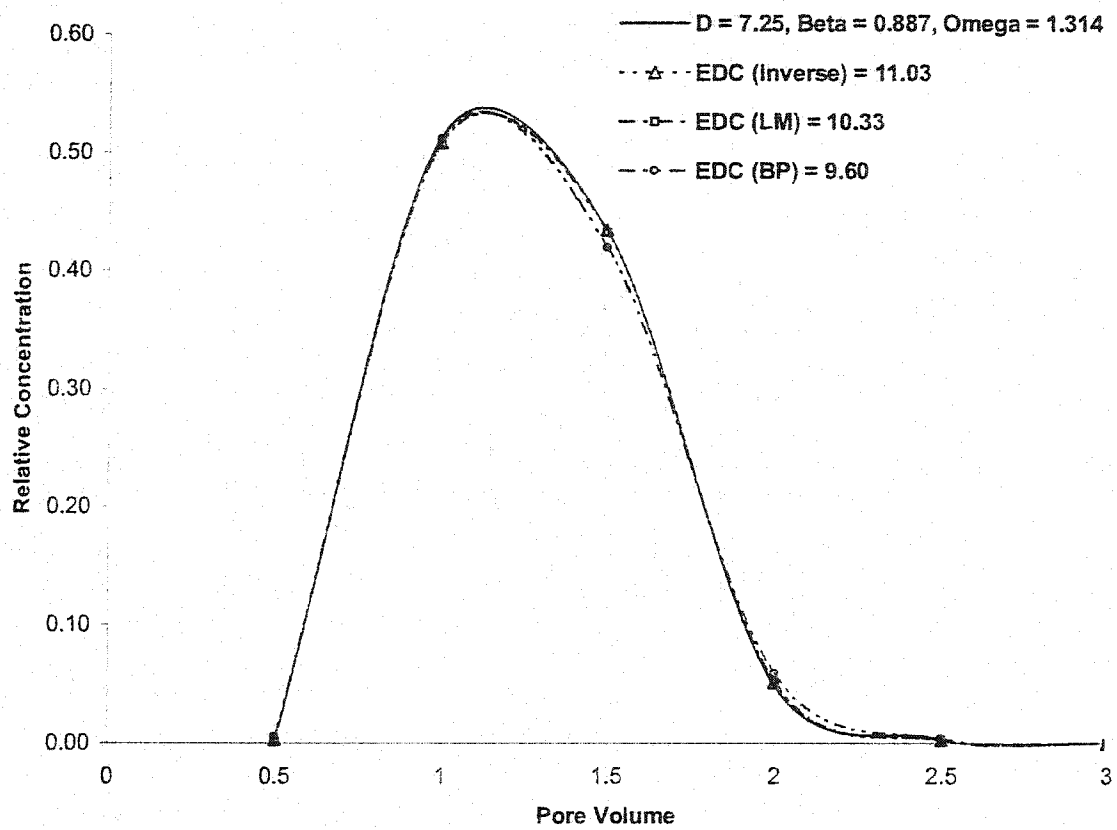
Figure 5.45: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
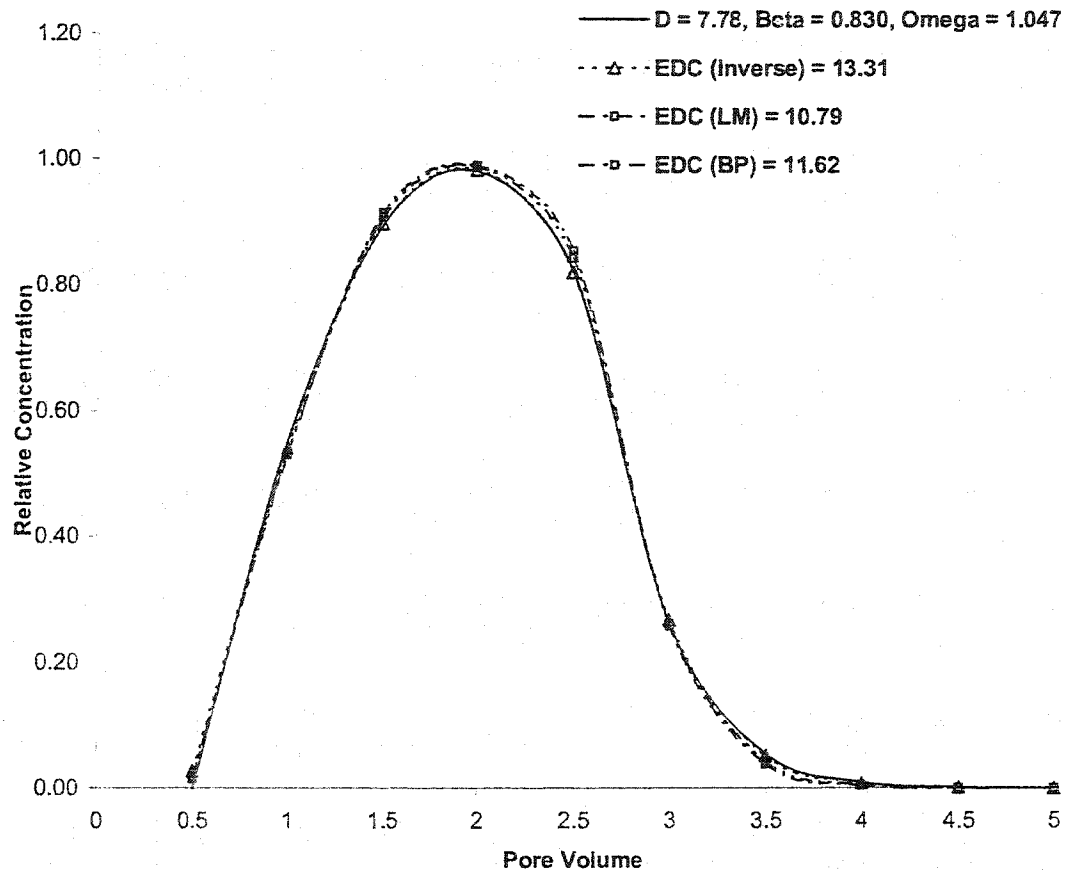
Figure 5.46: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
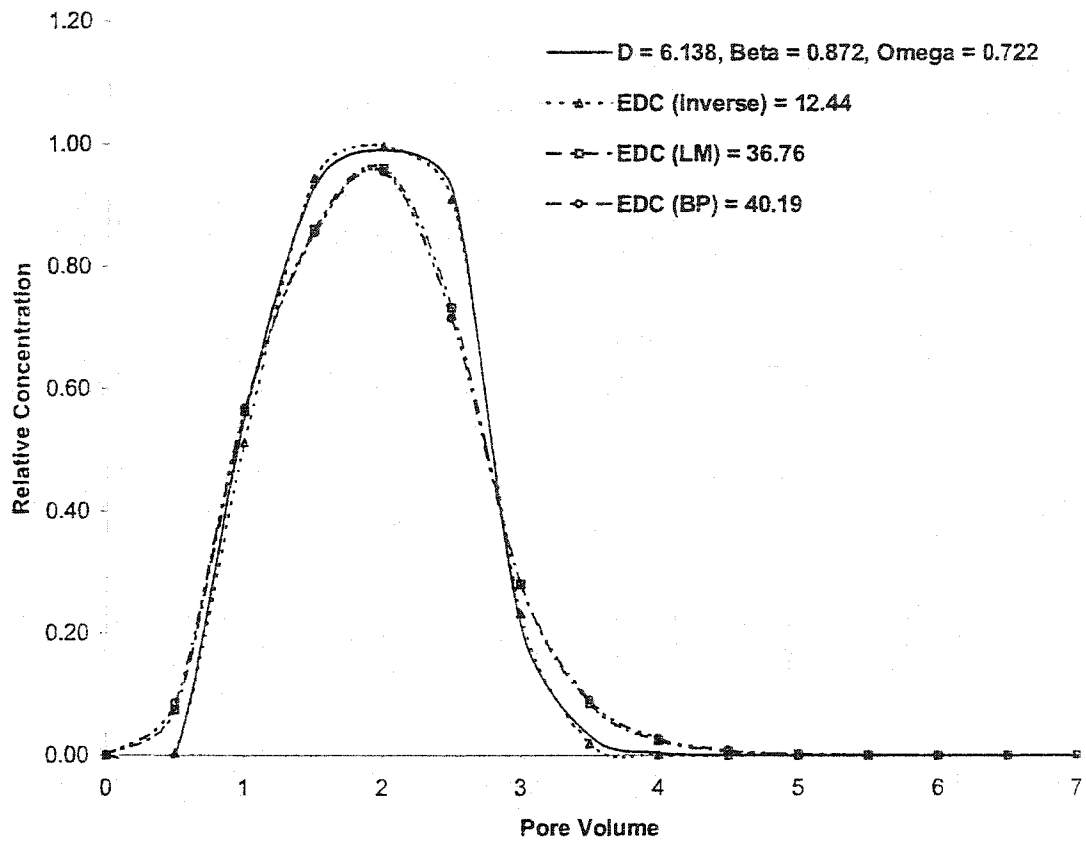
Figure 5.47: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
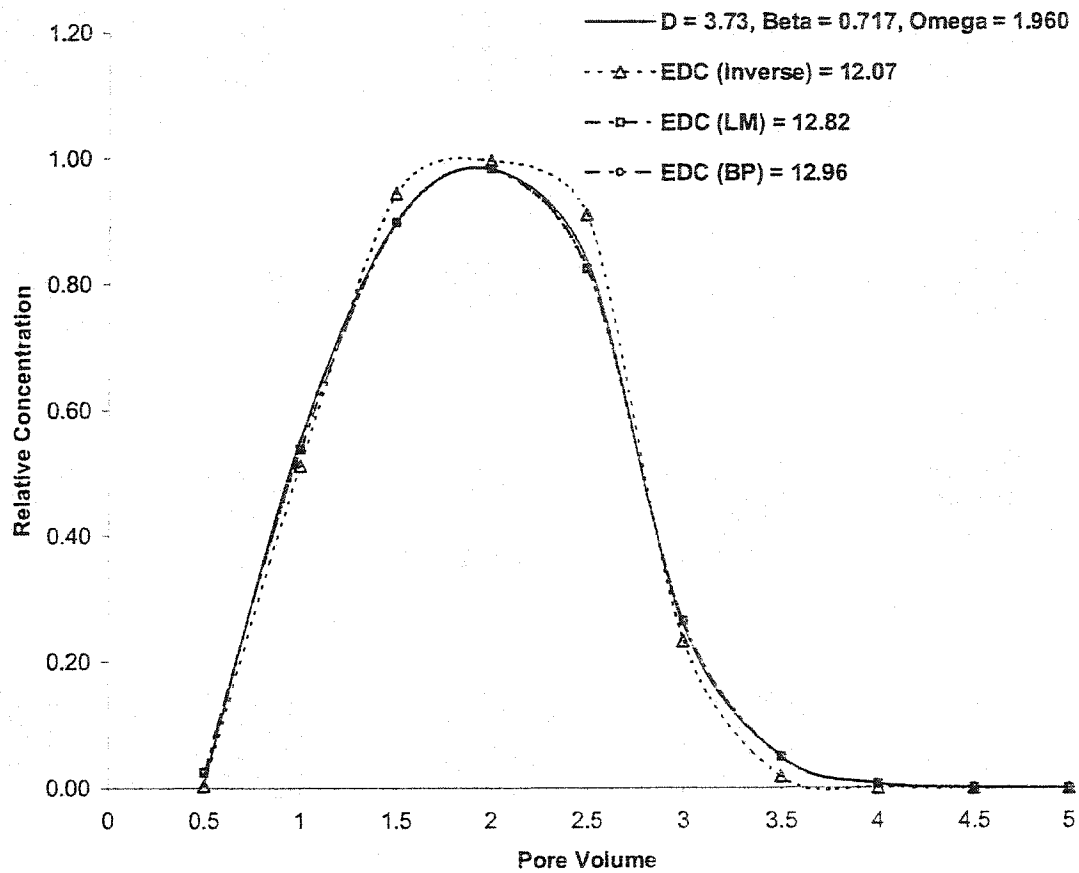
Figure 5.48: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
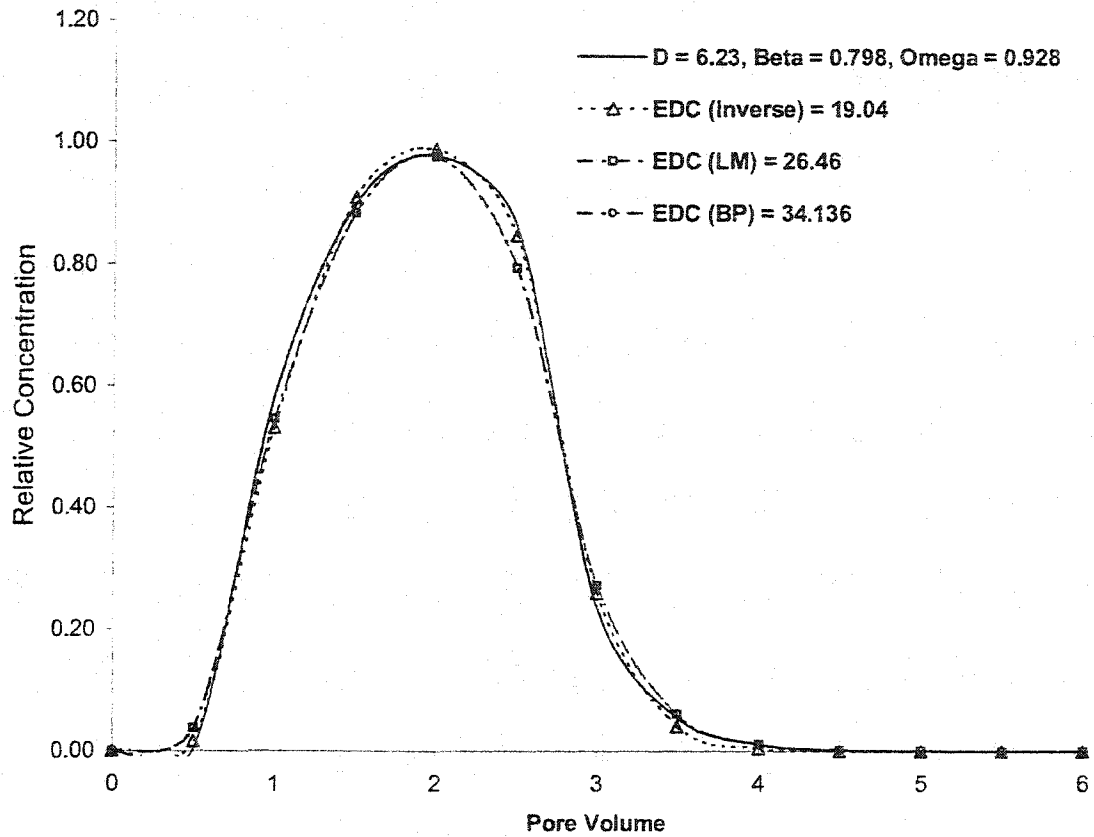
Figure 5.49: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
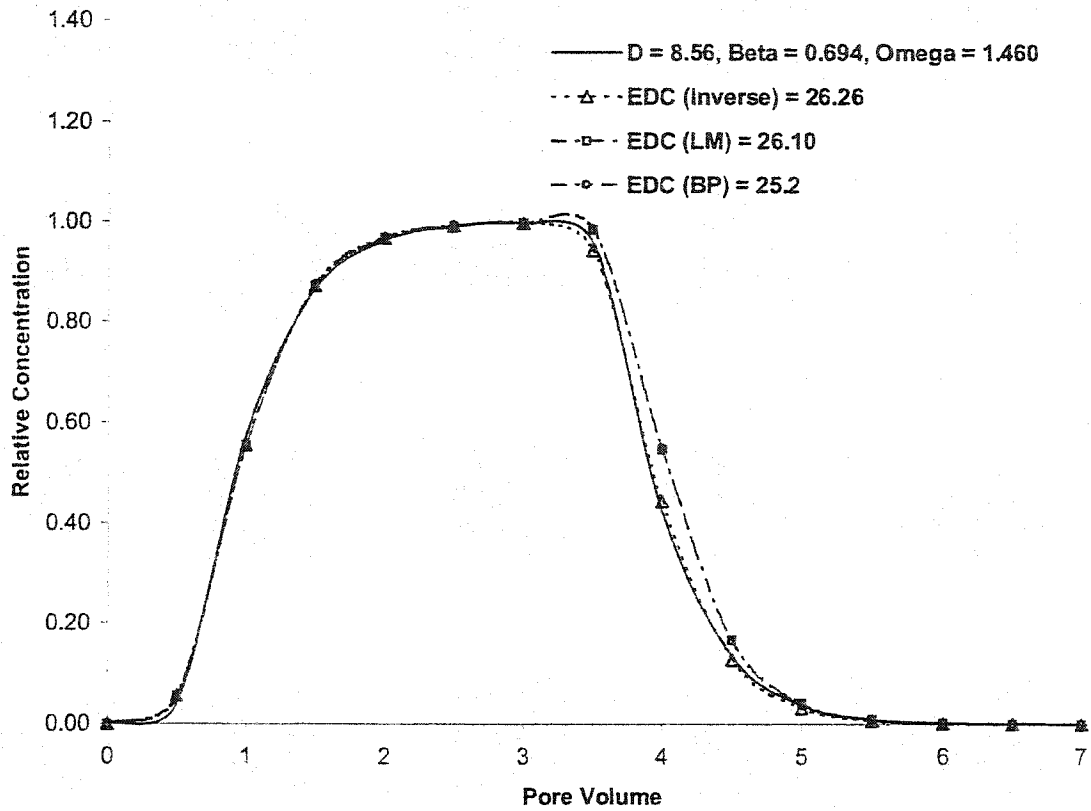
Figure 5.50: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem

# 5.4 Experimental Data 2

Nkeddi-Kizza et al. [23] conducted a series of experiments to study the contaminant transport. The second set of breakthrough curves is taken from there study for validation purposes.

## 5.4.1 Aggregated Oxisol

Aggregated Oxisol material was used in that study. It has kaolinite as the predominant clay mineral. Tables 5.6 summarizes some physical and chemical properties aggregated Oxisol.

| Analysis | Amount |
|---|---|
| aggregates (fractions) | 0.5-4.0 |
| $Fe_2O_3(Percentage)$ | 6.5 |
| CEC meq/100g | 2 |
| pH | 3.7 |

Table 5.6: Physical and chemical properties of Ione Oxisol

## 5.4.2 Tracers and their analysis

The following tracers ware used in this study:

$^{45}Ca, ^{36}Cl, and ^{3}H_2O$

The pulse solutions of $CaCl_2$ were given spiked with $^{45}Ca, ^{35}Cl$, and $^{3}H_2O$, each giving about 5 nCi/ml.

## 5.4.3 Column Studies

Breakthrough curves (BTC) of $^{45}$Ca,$^{36}$Cl,and$^{3}$H$_2O$ applied together as a pulse to soil columns packed with an aggregated oxisol, were measured under water-saturated conditions. The soil within the column was adjusted to a pH of 4 to 7 and separated into aggregates fractions of 0.5 − 1.0 and 2.0 − 4.0 mm in diameter. Each aggregate fraction was first saturated with 0.1 N $CaCl_2$ and then packed separately into plexiglas cylinders, 45 $cm^2$ in cross-sectional area and 5 cm long. Table 5.7 summarizes column data for various experiments. Table 5.8 shows the different column experiments and table 5.9 gives the comparisons of EDC values as determined by both algorithms and by solving inverse problem. Figure 5.51 - 5.54 show the Breakthrough curves obtained from using the EDC values.

| Experiment No. | Tracer | Bulk Density | Water Content | Flux | Pulse Period | | Aggregate Diameter |
|---|---|---|---|---|---|---|---|
| | | $\rho$ | $\theta$ | q | $T_1$ | $t_1$ | |
| | | $(g/cm^3)$ | $cm^3/cm^3$ | $(cm/hr)$ | (-) | (hr) | (mm) |
| 1 | $^{36}Cl$ | 1.18 | 0.52 | 0.221 | 0.5 | 1.383 | 0.5-1 |
| 2 | $^{45}Ca$ | 1.24 | 0.58 | 0.1702 | 0.5 | 8.517 | 0.5-1 |
| 3 | $^{45}Ca$ | 1.24 | 0.58 | 0.2416 | 0.5 | 6.000 | 0.5-1 |
| 4 | $^{45}Ca$ | 1.25 | 0.58 | 0.26 | 0.5 | 8.530 | 2-4.7 |

Table 5.7: Soil physical data for various displacements through Ione Oxisol

| Experiment No. | Tracer | Velocity | $\beta$ | R | P | $\omega$ | D |
|---|---|---|---|---|---|---|---|
| | | v | | | | | |
| | | (cm/day) | (-) | (-) | (-) | (-) | ($cm^2/day$) |
| 1 | $^{36}Cl$ | 10.00 | 0.73 | 2.20 | 13 | 0.69 | 3.92 |
| 2 | $^{45}Ca$ | 7.044 | 0.67 | 3.00 | 8 | 1.74 | 4.4025 |
| 3 | $^{45}Ca$ | 10.00 | 0.67 | 2.20 | 8 | 1.55 | 6.24 |
| 4 | $^{45}Ca$ | 10.00 | 0.5 | 1.50 | 4 | 0.59 | 12.50 |

Table 5.8: Summary of various $^{45}Ca$, $^{36}Cl$ displacements through Ione Oxisol

| No. | Tracer | EDC from Inverse | EDC from Backpropagation | EDC from Levenberg Marquardt |
|-----|--------|------------------|-------------------------|------------------------------|
| 1 | $^{36}Cl$ | 8.53 | 20.14 | 12.69 |
| 2 | $^{45}Ca$ | 7.577 | 15.42 | 15.83 |
| 3 | $^{45}Ca$ | 11.22 | 13.36 | 9.801 |
| 4 | $^{45}Ca$ | 41.78 | 121.53 | 42.18 |

Table 5.9: Values of EDC obtained by solving inverse problem and from both algorithms
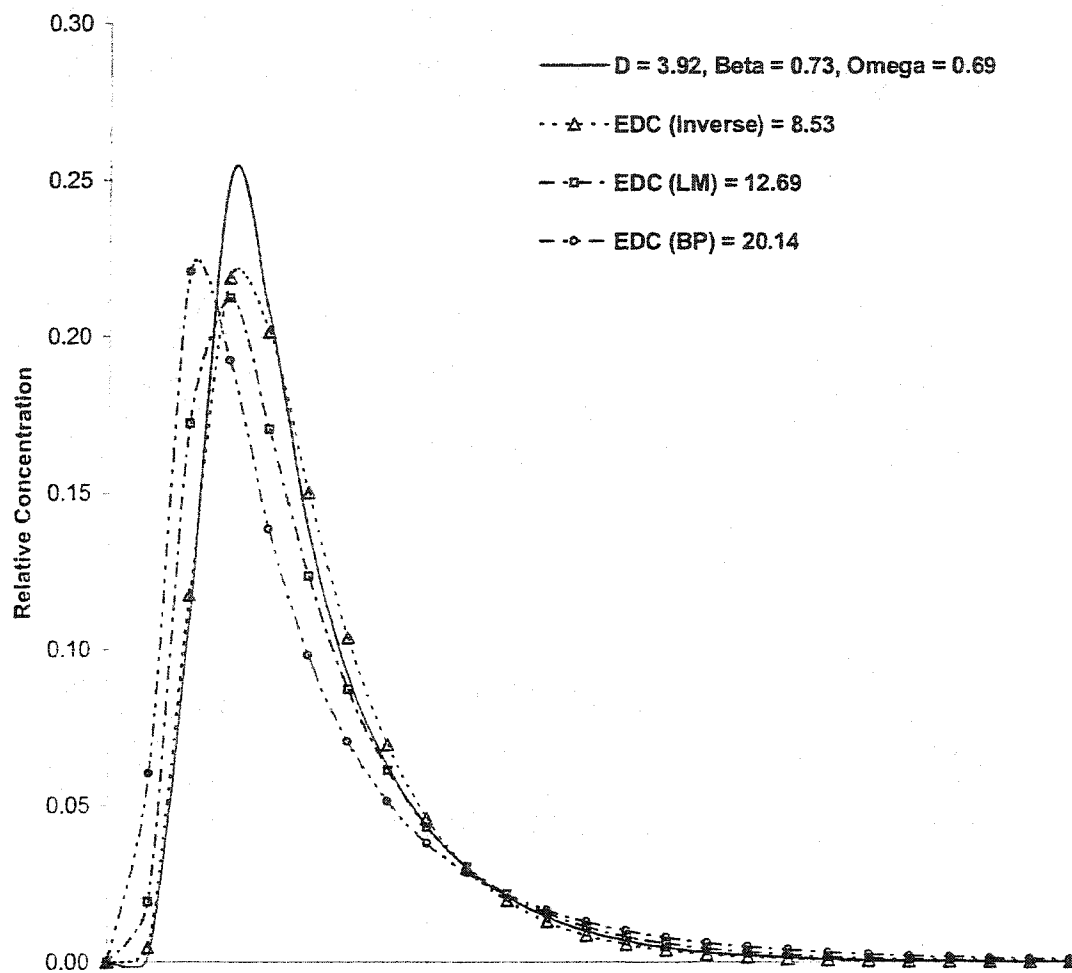
Figure 5.51: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
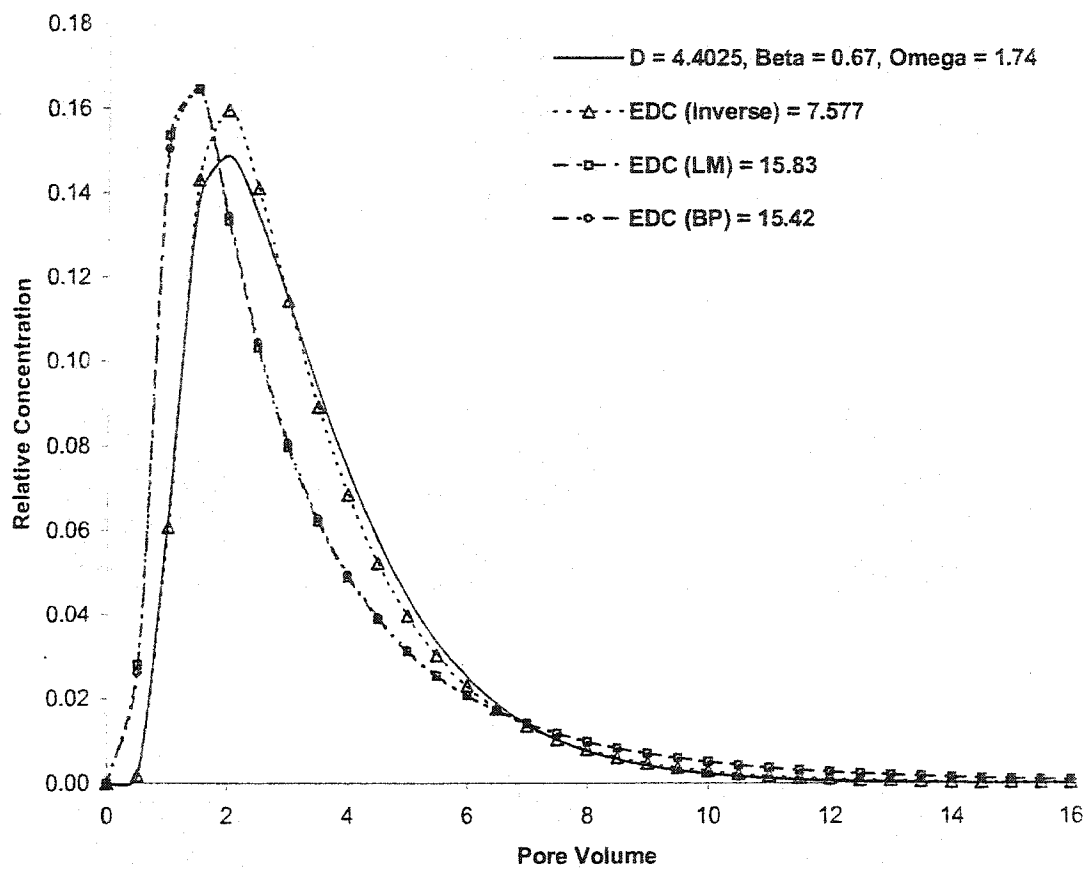
Figure 5.52: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
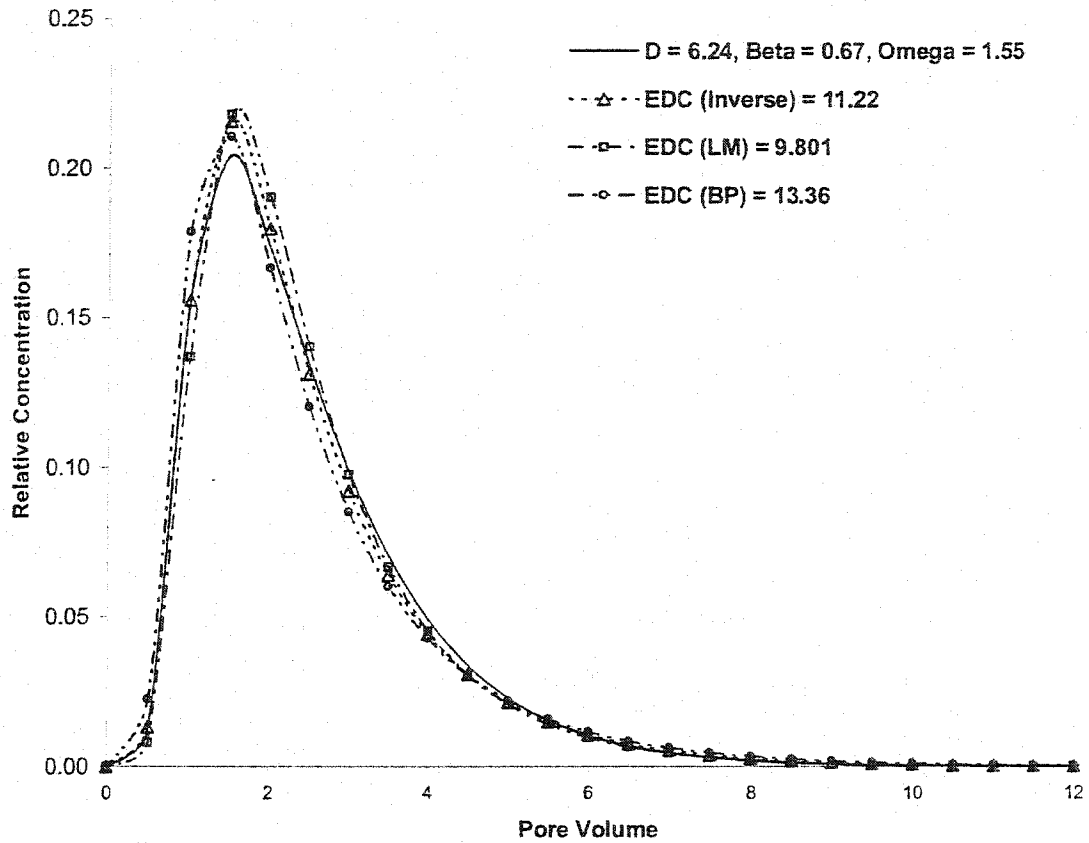
Figure 5.53: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem
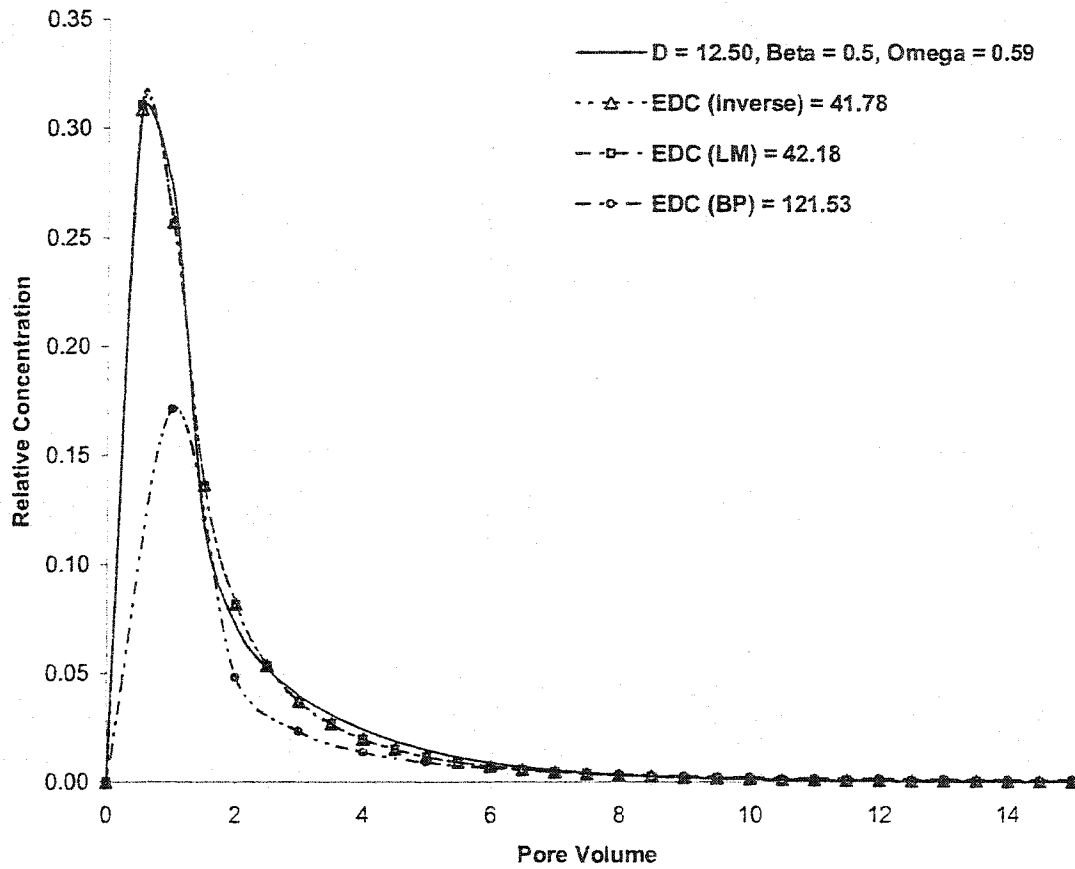
Figure 5.54: Breakthrough curves showing comparison of EDC values obtained from both ANNs model and inverse problem

# Chapter 6

# SUMMARY AND

# CONCLUSIONS

Groundwater, which is a major source of drinking water around the world is often the most significant water resource in many countries. Quality deterioration of this source reduces its ability to furnish communities with the water needed for the daily activities. The continuous rise in population, expanding agricultural, industrialization and higher living standards create an added demand on good quality water but contamination limit its availability. Sub-surface contamination can be caused by leakage from ponds and lagoons which are widely used as components of large waste-disposal systems, and by leaching of animal wastes, fertilizers and pesticides from agricultural soils. Contaminants that have entered the groundwater can move horizontally or vertically, depending on contaminant density and the natural flow pattern of the water in the aquifer. A contaminant on an average spreads as

plume, contamination concentration could be reduced with time and distance due to mechanisms like adsorption, ion exchange, dispersion, and decay. The rate of attenuation or reduction in concentration is a function of the type of contaminant and of the local hydrogeologic framework, but decades and even centuries are required for the process to become completely effective. Under the right conditions, and given enough time, contaminating fluids invading a natural body of groundwater can move great distances, hidden from view but still toxic. The eventual point of discharge of the contaminated groundwater may be a drinking water well or surface water body. Often by the time sub-surface contamination is conclusively identified it is too late to apply remedial measures that would be of much benefit. The results of the study will contribute to a great extent in providing automated reliable system which can be used by scientists and engineers involved in the planning and management of groundwater resources, in predicting the fate of contaminant that enter groundwater flow systems. The results will help them to come up with reliable predictions of the transport of contaminants within the flow system. It also gives them some insight into the physical and chemical factors that influence the sub-surface migration of dissolved contaminants.

Models that solve both flow and contaminant transport equations are frequently used to evaluate groundwater quality changes with space and time. Traditionally a two parameter partial differential equation has been used to describe contaminant transport. However some inadequacies were observed in this transport model when comparing model predictions with the experimental data. Due to these inadequacies

more complex conceptual models have been introduced in order to represent the real world systems. These models are based on the assumption that for either physical or chemical reasons, adsorption does not proceed at an equal rate in all parts of the soil medium. The resulting transport equations for these models contain several parameters.

In this study the parameters of a complex contaminant transport model were related to a single parameter named as Equivalent Dispersion Coefficient (EDC). Two different algorithms i.e. backpropagation algorithm and Levenberg Marquardt algorithm were used to come up with neural network based models. A total of one hundred and sixty (160) neural network models were trained. These models were trained on numerically generated data. The data was generated using CXTFIT 2.1. For this purpose a data generation program was designed. First the direct problem was solved using complex model parameters to generate breakthrough curves. Next inverse problem was solved considering linear equilibrium model to come up with the values of EDC for corresponding values of complex model parameters. Training was a process by which the network learns the recognition task by adjusting the weights in the links between the nodes created by processing input and output pairs. Training was continued until the values of weights cause the network to map the input patterns on an appropriate degree. Before training starts certain preprocessing steps were performed on the networks inputs and targets to make the neural network training more efficient. For this purpose the whole data set was normalized with respect to mean and standard deviation. In this way the inputs and targets

will have zero mean and unity standard deviation. After the training was complete neural network models were validated using numerically generated data which were not seen by the artificial neural network models before. The models predict the values of EDC when they receive the values of complex model parameters as an input. Once the validation process was complete the models were tested using the experimental data.

In the second part of this study, the EDC values obtained from the experimental data were then used in linear equilibrium advective despersive model. The linear equilibrium advective despersive model is numerically solved to predict the solute concentration. The breakthrough curves obtained from this approach and the ones obtained from the complex model were compared and found to be within tolerable limits. When EDC was used in linear equilibrium model the solute transport was successfully simulated.

Based on the results of the study the following conclusions can be drawn

1. For each data set the value of EDC moves from a maximum to a value equal to the dispersion coefficient as the model moves from a non-equilibrium two region model to the linear equilibrium model at the same time the value of mobile water content ($\theta_m$) vary from minimum to a value equal to water content ($\theta$). The value of fraction of adsorption site ($f$) changes from a minimum to a value of one (1) and the value of alpha ($\alpha$) moves from zero (0) towards infinity ($\infty$) as the model moves from physical non equilibrium model to a linear equilibrium model .

2. For a particular value of pore volume (T) the EDC value increases as the value of retardation factor (R) increases. Similarly for a particular value of pore volume (T) and retardation factor (R) the value of EDC shows an increase as the value of velocity increases.

3. Both algorithms i.e. backpropagation algorithm and Levenberg Marquardt algorithm are able to predict the values within tolerable limits.

4. For conservative contaminant both algorithm performs reasonably well however for non-conservative contaminant Levenberg Marquardt Algorithm gives better results.

5. The developed model works on the specific range of data however this range is flexible and can be increased.

6. Neural nets have the potential to be developed into more sensitive prediction tools. They obviate the need to specify the forms of correlations.

7. The neural network based models can be taught to identify patterns between input and target values and can subsequently predict out comes from fresh input conditions.

Based on the study following recommendation are made:

1. Extension of the current work is suggested in future by extending the data range.

2. The current model must be converted in the form of a software to make it easy to use by engineers and operators in the lab.

3. The ANNs models developed must be coupled with the numerical solution of linear equilibrium advective dispersive model to make a more comprehensive software. This software will give the breakthrough curve for the given value of parameters after calculating the value of EDC.

4. Mathematical relationships can be developed between the parameters of a contaminant transport model and EDC.

# Appendix A

# MATLAB Code

## A.1  BackPropagation Algorithm Code

A Program to determine the equivalent dispersion coefficient for contaminant transport model using backpropagation approach

```
function [Result,SE]=bpnnvalid1(fn)
clc;
global Total DATA slope1 slope2 n1 N M L K LR MOM alpha1 beta

if nargin < 1
fn='file.txt';  % file to read data from.
end

DATA = readdata(fn);
[DATA,mean_std]= datanormalize(DATA);
mean_std Total=max(size(DATA)); ITER = 60000; TestNo =
ceil(0.05*size(DATA,1));

Initialization OK=0;
while OK==0
Test=setdiff(ceil(rand(1,TestNo)*Total),0);
if
length(Test)==TestNo
OK=1;
end
end

Train=setdiff([1:Total],Test);
Testdata=DATA(Test,:);
DATA=DATA(Train,:);
```

```
Network parameters
N=size(DATA,2)-1;        % Nos of data nodes in input layer
M=60;                    % Neurons in first hidden layer
L=30;                    % Neurons in second hidden layer
K=1;                     % Neurons in output layer
LR=0.01;                 % Learning rate
MOM=0.5;                 % Momentum constant

Activation function parameters
alpha1=0.9;              % Slope parameters of AF in first hidden layer
n1=10;                   % Nos of levels in AF of first hidden layer
slope1=0.5;
slope2=2;
beta=1;

Weights
p=10;                    % Fixing the variance of the weights
W1=randn(M,N+1)/sqrt(p*N);
W2=randn(L,M+1)/sqrt(p*M);
W3=randn(K,L+1)/sqrt(p*L);
Weights for momentum
MW1=randn(M,N+1)/sqrt(p*N);
MW2=randn(L,M+1)/sqrt(p*M);
MW3=randn(K,L+1)/sqrt(p*L);

Training Starts
Result=[]; SE=[];
for iter=1:ITER
disp(['Running iteration = ' num2str(iter)])
k=ceil(rand*(Total-TestNo));
[Y1,Y2,X1,X2,Y3,E]=findSSE(W1,W2,W3,k);

Error and its gradients
S3=E;
f2=diag([ourdtansig(slope2,Y2(1:end-1));1]);
S2=f2*W3'*S3;
f1=diag([beta+alpha1*cos(n1*pi*X1)+...
((ourlogsig(slope1,X1)-ourlogsig(slope1,X1).^2)*slope1)   ;1]);
S1=f1*W2'*S2(1:end-1,1);
BW1=W1;
BW2=W2;
BW3=W3;
```

```
Weight Update
W1=W1+MOM*MW1+(1-MOM)*LR*S1(1:end-1,1)*[DATA(k,1:N) 1];
W2=W2+MOM*MW2+(1-MOM)*LR*S2(1:end-1,1)*Y1';
W3=W3+MOM*MW3+(1-MOM)*LR*S3*Y2';
MW1=W1-BW1;
MW2=W2-BW2;
MW3=W3-BW3;
SE=[SE E.^2];
end
save tempv7 W1 W2 W3 N M L K slope1 slope2 n1 alpha1 SE beta

Plotting Data Before Denormalization
for kk=1:Total-TestNo
[Y1,Y2,X1,X2,Y3,E]=findSSE(W1,W2,W3,kk);
Result=[Result; Y3 DATA(kk,end)];
end
figure;semilogy(SE);grid on;

Sum Square Error Plot
SE=[SE zeros(1,abs(ITER-floor(ITER/Total)*Total-Total))];
SSE=reshape(SE,length(SE)/Total,Total); SSE=sum(SSE);
figure;plot(SSE);grid on; title('Sum Square Error')

Plotting Data After Denormalization
Result(:,1) = denormalize(Result(:,1), mean_std(:,end));
Result(:,2) = denormalize(Result(:,2), mean_std(:,end));
figure;plot(Result);grid on;
title('Training data profile with denormalization')
legend('Predicted','Actual');

Histogram
[hp,xp]=hist(Result(:,2),30);
gg=find(hp==0);
hp(gg)=[];
xp(gg)=[];
xxp = min(xp):min(xp)/5:max(xp);
hhp = spline(xp,hp,xxp);
figure;plot(xxp,hhp) [hn]=hist(Result(:,1),xp);
hhn = spline(xp,hn,xxp);
hold on;
plot(xxp,hhn,'k');
title('BP');
legend('Actual','Predicted');
```

For Validation

```
Result=[]; DATA=Testdata; for kk=1:TestNo
[Y1,Y2,X1,X2,Y3,E]=findSSE(W1,W2,W3,kk); Result=[Result; Y3
DATA(kk,end)]; end figure;plot(Result);grid on;
title('Testing data profile without denormalization')
Result(:,1)= denormalize(Result(:,1), mean_std(:,end));
Result(:,2) =denormalize(Result(:,2), mean_std(:,end));
figure;plot(Result);grid on; title('Testing data profile with
denormalization'); legend('Predicted','Actual');
figure;bar(Result);grid on; title('Testing data profile with
denormalization'); colormap(cool); legend('Predicted','Actual');
return

function [Y1,Y2,X1,X2,Y3,E]=findSSE(W1,W2,W3,k)
global DATA slope1 slope2 n1 N alpha1 beta
X1=W1*[DATA(k,1:N) 1]';
Y1=0.5*X1+alpha1*1/(n1*pi)*sin(n1*pi*X1)+...
ourlogsig(slope1,X1);
Y1=[Y1 ;1]; X2=W2*Y1; Y2=ourtansig(slope2,X2); Y2=[Y2 ;1];
X3=W3*Y2; Y3=X3; E=DATA(k,end)-Y3;
return

function y = ourlogsig(slope,x)
y = 1 ./ (1 + exp(-slope*x)); i = find(~finite(y)); y(i) =
sign(x(i));
% derivative of the above is (x-x.^2)*slope
return

function y = ourtansig(slope,x)
y = 2./(1 + exp(-x*slope)) - 1; i = find(~finite(y)); y(i) =
sign(x(i));
% derivative of the above is (1-x.^2)*slope/2
return

function y = sinmul(beta,alpha,n,x)
y=beta*x+alpha*1/(n*pi)*sin(n*pi*x);
% derivative of the above is beta+alpha*cos(n*pi*x)
return

function y = ourdtansig(slope,x)
y=(1-x.^2)*slope/2; return
```

```
function y = dsinmul(beta,alpha,n,x)
y=beta+alpha*cos(n*pi*x); return


function [data,mean_std] = datanormalize(data)
mean_std=[]; for h=1:size(data,2)
MEAN=mean(data(:,h));
STD=std(data(:,h));
data(:,h)=(data(:,h)-MEAN)/STD;
mean_std=[mean_std [MEAN; STD]];
end


data(:,end)=abs(data(:,end)).^(1/2).*sign(data(:,end));
return
function [data] = denormalize(data,mean_std)
data(:,end)=(abs(data(:,end)).^2).*sign(data(:,end));
for h=1:size(data,2)
MEAN = mean_std(1,h);
STD  = mean_std(2,h);
data(:,h)=data(:,h)*STD+MEAN;
end return


Read Data From Text File function DATA = readdata(fn)
fid=fopen(fn,'r');          % open disk file with read-only access
DATA=[];


Start reading the proper data
while 1
lineread=fgetl(fid) ;          %
if  findstr('EOF',lineread);  % Read lines until EOF encountered
break; end
dataline=deblank(lineread);    % Remove trailing blanks
DATA=[DATA ;str2num(dataline)];
end
fclose(fid) return
```

# A.2 Levenberg Marquardt Algorithm Code

A program to determine the equivalent dispersion coefficient for the contaminat transport model using Levenberg Marquardt Approach

```
function [Result,SSEF]=Lmbg_final(fn)
clc;
global Total DATA slope1 slope2 n1 N M L K v alpha1 beta
if nargin < 1
fn='3.txt';  % file to read data from.
end
DATA = readdata(fn); [DATA,mean_std]= datanormalize(DATA);
mean_std; Total=max(size(DATA)); TestNo=ceil(0.05*size(DATA,1));


Initialization
OK=0; while OK==0
Test=setdiff(ceil(rand(1,TestNo)*Total),0);
if length(Test)==TestNo
OK=1;
end
end
Train=setdiff([1:Total],Test); Testdata=DATA(Test,:);
DATA=DATA(Train,:);


Network parameters
N=size(DATA,2)-1;      %Nos of data nodes in input layer
M=20;                  %Neurons in first hidden layer
L=10;                  %Neurons in second hidden layer
K=1;                   %Neurons in output layer
mu=1;
v=10;
T=1;


Activation function parameters
alpha1=0.8;            % Slope parameters of AF in first hidden layer
beta=1;
n1=10;                 % Nos of levels in AF of first hidden layer
slope1=0.05;
slope2=2;
Sther=2;


Weights
p=10;                  % Fixing the variance of the weights
W1=randn(M,N+1)/sqrt(p*N);
```

```
W2=randn(L,M+1)/sqrt(p*M);
W3=randn(K,L+1)/sqrt(p*L);

Training Starts
cond_err=1; SSEF =[];
while cond_err == 1
J=[];
V=[];
for k=1:Total-TestNo
[tempo,Y1,Y2,X1,X2,E,Y3]=findSSE(W1,W2,W3,k);
SE(k)=tempo;
[V]=[V;E];
[J]=findJACOB(X1,X2,Y1,Y2,W2,W3,k,J);
end
SSE=sum(SE);
SW1=W1; SW2=W2; SW3=W3;
cond_mu = 1;
while cond_mu == 1 % 2nd while
Dx=-inv([J'*J+mu*eye(size(J,2))])*J'*V;
W1=SW1+reshape(Dx(1:M*(N+T)),N+T,M)';
W2=SW2+reshape(Dx(M*(N+T)+1:M*(N+T)+L*(M+T)),M+T,L)';
W3=SW3+reshape(Dx(M*(N+T)+L*(M+T)+1:M*(N+T)+L*(M+T)+K*(L+T)),L+T,K)';

for kk=1:Total-TestNo
[tempo]=findSSE(W1,W2,W3,kk);
SEnew(kk)=tempo;
end
SSEnew=sum(SEnew);
if SSEnew > SSE
mu = mu*v
else
mu = mu/v
cond_mu = 0;
end
end % 2nd while ends
SSEF = [SSEF SSEnew] save tempb3 W1 W2 W3 N M L K mu v slope1
slope2 n1 alpha1 SSEF beta
if SSEnew < Sther
cond_err = 0;
end
end % 1st while ends

figure;plot(SSEF);grid on;title('Sum Square Error')
```

```
Plotting Data Before Denormalization
Result=[]; for kk=1:Total-TestNo
[SE,Y1,Y2,X1,X2,E,Y3]=findSSE(W1,W2,W3,kk);
Result=[Result; Y3 DATA(kk,end)];
end figure;plot(Result);grid on;title('Training Data Profile
without Denormalization')

Plotting Data After Denormalization
Result(:,1) = denormalize(Result(:,1), mean_std(:,end));
Result(:,2) = denormalize(Result(:,2), mean_std(:,end));

figure;plot(Result);grid on;title('Training data profile with
denormalization')

HISTOGRAM
[hp,xp]=hist(Result(:,2),30); gg=find(hp==0); hp(gg)=[];
xp(gg)=[]; xxp = min(xp):min(xp)/5:max(xp); hhp =
spline(xp,hp,xxp); figure;plot(xxp,hhp) [hn]=hist(Result(:,1),xp);
hhn = spline(xp,hn,xxp); hold on;plot(xxp,hhn,'k');
title('Levenberg Marquardt Neural Network');
legend('Actual','Predicted');

FOR VALIDATION
Result=[]; DATA=Testdata; for kk=1:TestNo
[SE,Y1,Y2,X1,X2,E,Y3]=findSSE(W1,W2,W3,kk);
Result=[Result; Y3 DATA(kk,end)];
end

figure;plot(Result);grid on;title('Testing data profile without
denormalization')
Result(:,1) = denormalize(Result(:,1), mean_std(:,end));
Result(:,2) = denormalize(Result(:,2), mean_std(:,end));

figure;plot(Result);grid on;title('Testing data profile with
denormalization') figure;bar(Result);grid on;
title('Testing data profile with denormalization');
colormap(bone);
legend('Predicted','Actual'); return

function [SE,Y1,Y2,X1,X2,E,Y3]=findSSE(W1,W2,W3,k)
global DATA slope1 slope2 n1 N alpha1 beta
X1=W1*[DATA(k,1:N) 1]';
Y1=[beta*X1+alpha1*1/(n1*pi)*sin(n1*pi*X1)]; Y1=[Y1 ;1]; X2=W2*Y1;
```

```
Y2=[ourtansig(slope2,X2)]; Y2=[Y2 ;1]; X3=W3*Y2; Y3=X3;
E=DATA(k,end)-Y3; SE=E.^2;
return


function a = ourtansig(slope,n) a = 2 ./ (1 + exp(-n*slope)) - 1;
% derivative of the above is (1-a^2)*slope/2
return


function [J]=findJACOB(X1,X2,Y1,Y2,W2,W3,k,J)
global DATA slope1 slope2 n1 alpha1 N beta
S3=-1;
f2=diag([(1-Y2(1:end-1).^2)*slope2/2; 1]); S2=f2*W3'*S3;
f1=diag([beta+alpha1*cos(n1*pi*X1); 1]); S1=f1*W2'*S2(1:end-1,1);
G1=S1(1:end-1,1)*[DATA(k,1:N) 1];G1=G1';G1=G1(:)';
G2=S2(1:end-1,1)*Y1';G2=G2';G2=G2(:)'; G3=S3*Y2';G3=G3';G3=G3(:)';
J=[J;G1 G2 G3]; return


function [data,mean_std] = datanormalize(data)
mean_std=[]; for h=1:size(data,2)
MEAN=mean(data(:,h));
STD=std(data(:,h));
data(:,h)=(data(:,h)-MEAN)/STD;
mean_std=[mean_std [MEAN; STD]];
end


data(:,end)=abs(data(:,end)).^(1/2).*sign(data(:,end)); return


function [data] = denormalize(data,mean_std)
data(:,end)=(abs(data(:,end)).^2).*sign(data(:,end)); for
h=1:size(data,2) MEAN = mean_std(1,h); STD  = mean_std(2,h);
data(:,h)=data(:,h)*STD+MEAN; end return
```

Read Data from Text File

```
function DATA = readdata(fn)
fid=fopen(fn,'r');              % open disk file with read-only access
DATA=[];
Start reading the proper data while 1
lineread=fgetl(fid);
if  findstr('EOF',lineread);   % Read lines until EOF encountered
break;                         %
end
dataline=deblank(lineread);    % Remove trailing blanks
```

```
DATA=[DATA ;str2num(dataline)];
end
fclose(fid)
return
```

# Appendix B

# FOTRAN Code

A Finite Difference scheme to simulate the solute transport in the porous media

```
      DIMENSION DXP(0:200),AE(0:200),AW(0:200),AP(0:200),
     *CO(0:200),CN(0:200),C(0:200),SU(0:200),SP(0:200),APO(0:200)
      OPEN(1,FILE='DATA1')
      OPEN(2,FILE='DATA2')

      DL=0.3
      DX=.01
      TTIME=432000
      DT=30
      DAB=3.4722e-9
      NC=30
      R=1
      V=3.4722e-7
      RHO=1126
      F=RHO*V
      D=DAB*RHO/DX
      DF=F-F
      DV=DX

C LENGTH OF CELLS
      DXP(0)=0.0
      DXP(1)=0.005
      DO 05 I=2,NC
      DXP(I)=DXP(I-1)+DX
05    CONTINUE

C  LENGTH OF THE LAST CELL
      DXP(NC+1)=0.3
```

```
C  INITIALIZATION OF VALUES FOR THE  SIMULATION
   DO 10 I=1,NC
   IF(I.EQ.1)THEN
   AW(I)=0.0
   ELSEIF(I.EQ.NC)THEN
   AW(I)=D + F/2
   ELSE
   AW(I)=D + F/2
   ENDIF

   IF(I.EQ.1)THEN
   AE(I)=D - F/2
   ELSEIF(I.EQ.10)THEN
   AE(I)=0.0
   ELSE
   AE(I)=D- F/2
   ENDIF

   IF(I.EQ.1)THEN
   SP(I)=-(2*D+F)
   ELSEIF(I.EQ.10)THEN
   SP(I)=0.0
   ELSE
   SP(I)=0.0
   ENDIF

   IF((I.GT.1).AND.(I.LT.10))THEN
   SU(I)=0.0
   ENDIF
   APO(I)=(R*RHO*DV)/(DT)
   IF(I.EQ.1)THEN
   AP(I)=AW(I)+AE(I)+APO(I)+DF-SP(I)
   ELSEIF(I.EQ.10)THEN
   AP(I)=AW(I)+AE(I)+APO(I)+DF-SP(I)
   ELSE
   AP(I)=AW(I)+AE(I)+APO(I)+DF-SP(I)
   ENDIF
10 CONTINUE

C  TIME STEP
   ITTIME=TTIME/DT
   NIT=10000
```

```
C   INITIAL CONDITION:
    DO 25 I=0,NC+1
    CO(I)=0.0001
25  CONTINUE

C   INITIAL GUESS:
    DO 26 I=1,NC+1
    C(I)=0.5
26  CONTINUE

C   ITERATION
    TIME=0.0
    DO 30 I1=1,ITTIME
    IF(I1.LE.8)THEN
    CEN=0.0001
    C(0)=CEN
    ELSEIF((I1.GT.8).AND.(I1.LE.200))THEN
    CEN=1.0
    C(0)=CEN
    ELSE
    CEN=0.0001
    C(0)=CEN
    ENDIF
    M=1
    DO 35 I2=1,NIT
    IF(M.NE.0)THEN
    M=0
    DO 40 I3=1,50
    DO 45 I=1,NC
    IF(I.EQ.1)THEN
    SU(I)=(2*D + F)*CEN
    ENDIF
    IF(I.EQ.10)THEN
    SU(I)=F*C(NC+1)
    ENDIF
    CN(I)=(1/AP(I))*(AW(I)*C(I-1)+AE(I)*C(I+1)+APO(I)*CO(I)+SU(I))
45  CONTINUE
    CN(NC+1)=CN(NC)
    DO 50 I=1,NC
    ERR=ABS(1-C(I)/CN(I))
    IF(ERR.GT..0001)THEN
    M=M+1
    ENDIF
```

```
50   CONTINUE
     DO 55 I=1,NC+1
     C(I)=CN(I)
55   CONTINUE
40   CONTINUE
     ENDIF
35   CONTINUE

     IF(M.EQ.0)THEN
     DO 60 I=0,NC+1
     CO(I)=C(I)
60   CONTINUE
     ENDIF
     TIME=TIME + DT
     WRITE(1,*) TIME
     DO   65 I=0,NC+1
     WRITE(1,'(F10.4,3X,F10.4)')DXP(I),C(I)
65   CONTINUE
     WRITE(2,'(E10.4,3X,F10.4,3X,F10.8)') TIME,C(10),DXP(10)
30   CONTINUE
     STOP
     END
```

# Appendix C

# Data Generation Tables

## C.1 One complete Data set used to generate Break-thorugh curevs for T=0.5, R=1.026, V=3cm/day

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1 | 0.1 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 0.2 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 0.4 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 0.6 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1 | 0.4 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 0.6 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 0.8 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 1 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 1.2 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 1.4 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1 | 0.6 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 1.4 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 1.6 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 1.8 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 2 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 2.2 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 1 | 0.6 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 2.2 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 2.4 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 2.6 |
| 0.5 | 3 | 1.026 | 1 | 0.1 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.2 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.3 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.4 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.5 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.6 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.7 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.8 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.9 | 2.8 |
| 0.5 | 3 | 1.026 | 1 | 0.99 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 0.2 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 5 | 0.6 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 0.2 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 0.4 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 0.6 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 0.8 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 1 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 5 | 0.6 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 1 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 1.2 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 1.4 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 1.6 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 1.8 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 5 | 0.6 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 1.8 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 2 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 2.2 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 2.4 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 2.6 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 5 | 0.9 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 2.6 |
| 0.5 | 3 | 1.026 | 5 | 0.1 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.2 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.3 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.4 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.5 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.6 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.7 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.8 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.9 | 2.8 |
| 0.5 | 3 | 1.026 | 5 | 0.99 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 0.2 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0 3 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 0.4 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 0.6 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 7 | 0.9 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 0.6 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 0.8 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 1 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 1.2 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 1.4 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 7 | 0.9 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 1.4 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 1.6 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 1.8 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 2 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 2.2 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 7 | 0.9 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 2.2 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 2.4 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 2.6 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 2.6 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 2.6 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 2.6 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 2.6 |
| 0.5 | 3 | 1.026 | 7 | 0.1 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.2 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.3 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.4 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.5 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.6 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.7 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.8 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.9 | 2.8 |
| 0.5 | 3 | 1.026 | 7 | 0.99 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0 3 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 0.2 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 0.4 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 10 | 0.4 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 0.4 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 0.6 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 0.8 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 1 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 1.2 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 10 | 0.4 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 1.2 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 1.4 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 1.6 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 1.8 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 2 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 10 | 0.4 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 2 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 2.2 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 2.4 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 2.6 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 2.6 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 2.6 |
| 0.5 | 3 | 1.026 | 10 | 0.9 | 2.6 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 2.6 |
| 0.5 | 3 | 1.026 | 10 | 0.1 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.2 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.3 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.4 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.5 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.6 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.7 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.8 | 2.8 |

| Pore Volume | Pore Velocity | Retardation factor | Dispersion Coefficient | Beta | Omega |
|---|---|---|---|---|---|
| 0.5 | 3 | 1.026 | 10 | 0.9 | 2.8 |
| 0.5 | 3 | 1.026 | 10 | 0.99 | 2.8 |

## C.2 Data Used for Training the Neural Network Algorithms for T=0.5, R=1.026, V=3cm/day

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 1 | 0.1 | 0.2 | 221.12 |
| 1 | 0.2 | 0.2 | 198.47 |
| 1 | 0.3 | 0.2 | 178.40 |
| 1 | 0.4 | 0.2 | 160.56 |
| 1 | 0.5 | 0.2 | 163.21 |
| 1 | 0.6 | 0.2 | 91.99 |
| 1 | 0.7 | 0.2 | 55.08 |
| 1 | 0.8 | 0.2 | 28.38 |
| 1 | 0.9 | 0.2 | 12.25 |
| 1 | 0.99 | 0.2 | 1.08 |
| 1 | 0.1 | 0.4 | 133.76 |
| 1 | 0.2 | 0.4 | 130.10 |
| 1 | 0.3 | 0.4 | 126.48 |
| 1 | 0.4 | 0.4 | 119.23 |
| 1 | 0.5 | 0.4 | 91.36 |
| 1 | 0.6 | 0.4 | 47.23 |
| 1 | 0.7 | 0.4 | 27.44 |
| 1 | 0.8 | 0.4 | 15.13 |
| 1 | 0.9 | 0.4 | 6.38 |
| 1 | 0.99 | 0.4 | 0.84 |
| 1 | 0.1 | 0.6 | 126.45 |
| 1 | 0.2 | 0.6 | 123.07 |
| 1 | 0.3 | 0.6 | 116.21 |
| 1 | 0.4 | 0.6 | 86.38 |
| 1 | 0.5 | 0.6 | 46.39 |
| 1 | 0.6 | 0.6 | 28.65 |
| 1 | 0.7 | 0.6 | 17.79 |
| 1 | 0.8 | 0.6 | 4.52 |
| 1 | 0.9 | 0.6 | 2.72 |
| 1 | 0.99 | 0.6 | 1.04 |
| 1 | 0.1 | 0.8 | 121.59 |
| 1 | 0.2 | 0.8 | 115.20 |
| 1 | 0.3 | 0.8 | 86.06 |
| 1 | 0.4 | 0.8 | 47.98 |
| 1 | 0.5 | 0.8 | 30.85 |
| 1 | 0.6 | 0.8 | 20.45 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 1 | 0.7 | 0.8 | 13.12 |
| 1 | 0.8 | 0.8 | 7.58 |
| 1 | 0.9 | 0.8 | 2.40 |
| 1 | 0.99 | 0.8 | 1.03 |
| 1 | 0.1 | 1 | 114.67 |
| 1 | 0.2 | 1 | 89.67 |
| 1 | 0.3 | 1 | 51.32 |
| 1 | 0.4 | 1 | 33.71 |
| 1 | 0.5 | 1 | 23.24 |
| 1 | 0.6 | 1 | 15.92 |
| 1 | 0.7 | 1 | 10.41 |
| 1 | 0.8 | 1 | 4.37 |
| 1 | 0.9 | 1 | 2.14 |
| 1 | 0.99 | 1 | 1.03 |
| 1 | 0.1 | 1.2 | 95.26 |
| 1 | 0.2 | 1.2 | 33.47 |
| 1 | 0.3 | 1.2 | 27.65 |
| 1 | 0.4 | 1.2 | 22.52 |
| 1 | 0.5 | 1.2 | 18.47 |
| 1 | 0.8 | 1.2 | 4.05 |
| 1 | 0.9 | 1.2 | 1.95 |
| 1 | 0.99 | 1.2 | 0.90 |
| 1 | 0.1 | 1.4 | 62.98 |
| 1 | 0.2 | 1.4 | 41.39 |
| 1 | 0.3 | 1.4 | 29.51 |
| 1 | 0.4 | 1.4 | 21.56 |
| 1 | 0.5 | 1.4 | 15.67 |
| 1 | 0.6 | 1.4 | 11.09 |
| 1 | 0.7 | 1.4 | 7.41 |
| 1 | 0.8 | 1.4 | 3.74 |
| 1 | 0.9 | 1.4 | 1.81 |
| 1 | 0.99 | 1.4 | 1.03 |
| 1 | 0.1 | 1.6 | 46.48 |
| 1 | 0.2 | 1.6 | 33.18 |
| 1 | 0.3 | 1.6 | 24.58 |
| 1 | 0.4 | 1.6 | 18.34 |
| 1 | 0.5 | 1.6 | 12.42 |
| 1 | 0.6 | 1.6 | 10.72 |
| 1 | 0.7 | 1.6 | 5.98 |
| 1 | 0.8 | 1.6 | 3.85 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 1 | 0.9 | 1.6 | 1.65 |
| 1 | 0.99 | 1.6 | 1.03 |
| 1 | 0.1 | 1.8 | 20.70 |
| 1 | 0.2 | 1.8 | 16.55 |
| 1 | 0.3 | 1.8 | 13.56 |
| 1 | 0.4 | 1.8 | 12.58 |
| 1 | 0.5 | 1.8 | 12.16 |
| 1 | 0.6 | 1.8 | 10.28 |
| 1 | 0.7 | 1.8 | 5.61 |
| 1 | 0.8 | 1.8 | 3.22 |
| 1 | 0.9 | 1.8 | 1.61 |
| 1 | 0.99 | 1.8 | 1.03 |
| 1 | 0.1 | 2 | 16.98 |
| 1 | 0.2 | 2 | 13.93 |
| 1 | 0.3 | 2 | 12.74 |
| 1 | 0.4 | 2 | 12.33 |
| 1 | 0.5 | 2 | 12.10 |
| 1 | 0.6 | 2 | 9.60 |
| 1 | 0.7 | 2 | 5.25 |
| 1 | 0.8 | 2 | 3.01 |
| 1 | 0.9 | 2 | 1.42 |
| 1 | 0.99 | 2 | 0.82 |
| 1 | 0.1 | 2.2 | 14.22 |
| 1 | 0.2 | 2.2 | 12.97 |
| 1 | 0.3 | 2.2 | 12.46 |
| 1 | 0.4 | 2.2 | 12.20 |
| 1 | 0.5 | 2.2 | 12.02 |
| 1 | 0.6 | 2.2 | 8.80 |
| 1 | 0.7 | 2.2 | 4.74 |
| 1 | 0.8 | 2.2 | 3.04 |
| 1 | 0.9 | 2.2 | 1.34 |
| 1 | 0.99 | 2.2 | 0.82 |
| 1 | 0.1 | 2.4 | 13.30 |
| 1 | 0.2 | 2.4 | 12.64 |
| 1 | 0.3 | 2.4 | 12.31 |
| 1 | 0.4 | 2.4 | 12.12 |
| 1 | 0.5 | 2.4 | 11.97 |
| 1 | 0.6 | 2.4 | 8.09 |
| 1 | 0.7 | 2.4 | 4.63 |
| 1 | 0.8 | 2.4 | 2.83 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 1 | 0.9 | 2.4 | 1.28 |
| 1 | 0.99 | 2.4 | 0.82 |
| 1 | 0.1 | 2.6 | 12.83 |
| 1 | 0.2 | 2.6 | 43.42 |
| 1 | 0.3 | 2.6 | 30.95 |
| 1 | 0.4 | 2.6 | 19.89 |
| 1 | 0.5 | 2.6 | 9.80 |
| 1 | 0.6 | 2.6 | 5.61 |
| 1 | 0.7 | 2.6 | 4.25 |
| 1 | 0.8 | 2.6 | 2.35 |
| 1 | 0.9 | 2.6 | 1.27 |
| 1 | 0.99 | 2.6 | 0.90 |
| 1 | 0.1 | 2.8 | 12.57 |
| 1 | 0.2 | 2.8 | 12.35 |
| 1 | 0.3 | 2.8 | 12.19 |
| 1 | 0.4 | 2.8 | 12.08 |
| 1 | 0.5 | 2.8 | 11.90 |
| 1 | 0.6 | 2.8 | 6.97 |
| 1 | 0.7 | 2.8 | 4.13 |
| 1 | 0.8 | 2.8 | 2.49 |
| 1 | 0.9 | 2.8 | 1.18 |
| 1 | 0.99 | 2.8 | 0.90 |
| 5 | 0.1 | 0.2 | 224.97 |
| 5 | 0.2 | 0.2 | 201.22 |
| 5 | 0.3 | 0.2 | 179.94 |
| 5 | 0.4 | 0.2 | 161.14 |
| 5 | 0.5 | 0.2 | 170.94 |
| 5 | 0.6 | 0.2 | 95.28 |
| 5 | 0.7 | 0.2 | 56.93 |
| 5 | 0.8 | 0.2 | 30.29 |
| 5 | 0.9 | 0.2 | 13.51 |
| 5 | 0.99 | 0.2 | 5.22 |
| 5 | 0.1 | 0.4 | 134.30 |
| 5 | 0.2 | 0.4 | 130.53 |
| 5 | 0.3 | 0.4 | 127.13 |
| 5 | 0.4 | 0.4 | 120.95 |
| 5 | 0.5 | 0.4 | 99.83 |
| 5 | 0.6 | 0.4 | 52.03 |
| 5 | 0.7 | 0.4 | 30.03 |
| 5 | 0.8 | 0.4 | 16.91 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 5 | 0.9 | 0.4 | 7.98 |
| 5 | 0.99 | 0.4 | 5.32 |
| 5 | 0.1 | 0.6 | 126.81 |
| 5 | 0.2 | 0.6 | 123.55 |
| 5 | 0.3 | 0.6 | 118.38 |
| 5 | 0.4 | 0.6 | 96.87 |
| 5 | 0.5 | 0.6 | 52.14 |
| 5 | 0.6 | 0.6 | 31.85 |
| 5 | 0.7 | 0.6 | 20.03 |
| 5 | 0.8 | 0.6 | 6.86 |
| 5 | 0.9 | 0.6 | 6.70 |
| 5 | 0.99 | 0.6 | 5.18 |
| 5 | 0.1 | 0.8 | 122.16 |
| 5 | 0.2 | 0.8 | 117.34 |
| 5 | 0.3 | 0.8 | 97.66 |
| 5 | 0.4 | 0.8 | 54.88 |
| 5 | 0.5 | 0.8 | 34.73 |
| 5 | 0.6 | 0.8 | 23.18 |
| 5 | 0.7 | 0.8 | 15.28 |
| 5 | 0.8 | 0.8 | 9.54 |
| 5 | 0.9 | 0.8 | 6.55 |
| 5 | 0.99 | 0.8 | 5.18 |
| 5 | 0.1 | 1 | 117.21 |
| 5 | 0.2 | 1 | 100.99 |
| 5 | 0.3 | 1 | 59.78 |
| 5 | 0.4 | 1 | 38.39 |
| 5 | 0.5 | 1 | 26.50 |
| 5 | 0.6 | 1 | 18.48 |
| 5 | 0.7 | 1 | 12.59 |
| 5 | 0.8 | 1 | 9.42 |
| 5 | 0.9 | 1 | 6.35 |
| 5 | 0.99 | 1 | 5.18 |
| 5 | 0.1 | 1.2 | 104.92 |
| 5 | 0.2 | 1.2 | 36.36 |
| 5 | 0.3 | 1.2 | 29.98 |
| 5 | 0.4 | 1.2 | 24.05 |
| 5 | 0.5 | 1.2 | 18.54 |
| 5 | 0.6 | 1.2 | 13.54 |
| 5 | 0.7 | 1.2 | 12.04 |
| 5 | 0.8 | 1.2 | 9.57 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 5 | 0.9 | 1.2 | 6.17 |
| 5 | 0.99 | 1.2 | 4.91 |
| 5 | 0.1 | 1.4 | 76.36 |
| 5 | 0.2 | 1.4 | 48.35 |
| 5 | 0.3 | 1.4 | 34.10 |
| 5 | 0.4 | 1.4 | 25.03 |
| 5 | 0.5 | 1.4 | 18.52 |
| 5 | 0.6 | 1.4 | 13.57 |
| 5 | 0.7 | 1.4 | 9.73 |
| 5 | 0.8 | 1.4 | 9.26 |
| 5 | 0.9 | 1.4 | 6.03 |
| 5 | 0.99 | 1.4 | 5.17 |
| 5 | 0.1 | 1.6 | 55.20 |
| 5 | 0.2 | 1.6 | 38.66 |
| 5 | 0.3 | 1.6 | 28.62 |
| 5 | 0.4 | 1.6 | 21.60 |
| 5 | 0.5 | 1.6 | 12.66 |
| 5 | 0.6 | 1.6 | 12.16 |
| 5 | 0.7 | 1.6 | 11.91 |
| 5 | 0.8 | 1.6 | 8.22 |
| 5 | 0.9 | 1.6 | 5.51 |
| 5 | 0.99 | 1.6 | 5.17 |
| 5 | 0.1 | 1.8 | 23.90 |
| 5 | 0.2 | 1.8 | 19.52 |
| 5 | 0.3 | 1.8 | 15.39 |
| 5 | 0.4 | 1.8 | 13.00 |
| 5 | 0.5 | 1.8 | 12.35 |
| 5 | 0.6 | 1.8 | 12.13 |
| 5 | 0.7 | 1.8 | 11.90 |
| 5 | 0.8 | 1.8 | 8.41 |
| 5 | 0.9 | 1.8 | 5.84 |
| 5 | 0.99 | 1.8 | 5.17 |
| 5 | 0.1 | 2 | 20.24 |
| 5 | 0.2 | 2 | 16.28 |
| 5 | 0.3 | 2 | 13.49 |
| 5 | 0.4 | 2 | 12.59 |
| 5 | 0.5 | 2 | 12.26 |
| 5 | 0.6 | 2 | 12.09 |
| 5 | 0.7 | 2 | 11.86 |
| 5 | 0.8 | 2 | 8.06 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 5 | 0.9 | 2 | 6.40 |
| 5 | 0.99 | 2 | 5.28 |
| 5 | 0.1 | 2.2 | 17.25 |
| 5 | 0.2 | 2.2 | 14.16 |
| 5 | 0.3 | 2.2 | 12.86 |
| 5 | 0.4 | 2.2 | 12.41 |
| 5 | 0.5 | 2.2 | 12.20 |
| 5 | 0.6 | 2.2 | 12.05 |
| 5 | 0.7 | 2.2 | 10.59 |
| 5 | 0.8 | 2.2 | 8.31 |
| 5 | 0.9 | 2.2 | 6.33 |
| 5 | 0.99 | 2.2 | 5.28 |
| 5 | 0.1 | 2.4 | 14.98 |
| 5 | 0.2 | 2.4 | 13.22 |
| 5 | 0.3 | 2.4 | 12.59 |
| 5 | 0.4 | 2.4 | 12.31 |
| 5 | 0.5 | 2.4 | 12.16 |
| 5 | 0.6 | 2.4 | 12.03 |
| 5 | 0.7 | 2.4 | 11.77 |
| 5 | 0.8 | 2.4 | 8.28 |
| 5 | 0.9 | 2.4 | 6.26 |
| 5 | 0.99 | 2.4 | 5.28 |
| 5 | 0.1 | 2.6 | 13.72 |
| 5 | 0.5 | 2.6 | 19.79 |
| 5 | 0.6 | 2.6 | 13.99 |
| 5 | 0.7 | 2.6 | 8.09 |
| 5 | 0.8 | 2.6 | 6.27 |
| 5 | 0.9 | 2.6 | 5.22 |
| 5 | 0.99 | 2.6 | 4.90 |
| 5 | 0.1 | 2.8 | 13.18 |
| 5 | 0.2 | 2.8 | 12.61 |
| 5 | 0.3 | 2.8 | 12.36 |
| 5 | 0.4 | 2.8 | 12.19 |
| 5 | 0.5 | 2.8 | 12.08 |
| 5 | 0.6 | 2.8 | 11.99 |
| 5 | 0.7 | 2.8 | 11.58 |
| 5 | 0.8 | 2.8 | 8.18 |
| 5 | 0.9 | 2.8 | 6.14 |
| 5 | 0.99 | 2.8 | 4.90 |
| 7 | 0.1 | 0.2 | 226.97 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 7 | 0.2 | 0.2 | 202.51 |
| 7 | 0.3 | 0.2 | 180.76 |
| 7 | 0.4 | 0.2 | 161.24 |
| 7 | 0.5 | 0.2 | 175.16 |
| 7 | 0.6 | 0.2 | 97.05 |
| 7 | 0.7 | 0.2 | 57.95 |
| 7 | 0.8 | 0.2 | 31.40 |
| 7 | 0.9 | 0.2 | 14.41 |
| 7 | 0.99 | 0.2 | 7.42 |
| 7 | 0.1 | 0.4 | 134.57 |
| 7 | 0.2 | 0.4 | 130.75 |
| 7 | 0.3 | 0.4 | 127.39 |
| 7 | 0.4 | 0.4 | 121.47 |
| 7 | 0.5 | 0.4 | 103.18 |
| 7 | 0.6 | 0.4 | 54.83 |
| 7 | 0.7 | 0.4 | 31.56 |
| 7 | 0.8 | 0.4 | 18.05 |
| 7 | 0.9 | 0.4 | 9.38 |
| 7 | 0.99 | 0.4 | 7.48 |
| 7 | 0.1 | 0.6 | 126.98 |
| 7 | 0.2 | 0.6 | 123.79 |
| 7 | 0.3 | 0.6 | 119.07 |
| 7 | 0.4 | 0.6 | 101.24 |
| 7 | 0.5 | 0.6 | 55.52 |
| 7 | 0.6 | 0.6 | 33.73 |
| 7 | 0.7 | 0.6 | 21.42 |
| 7 | 0.8 | 0.6 | 8.23 |
| 7 | 0.9 | 0.6 | 9.93 |
| 7 | 0.99 | 0.6 | 7.36 |
| 7 | 0.1 | 0.8 | 122.49 |
| 7 | 0.2 | 0.8 | 118.09 |
| 7 | 0.3 | 0.8 | 102.01 |
| 7 | 0.4 | 0.8 | 58.97 |
| 7 | 0.5 | 0.8 | 36.98 |
| 7 | 0.6 | 0.8 | 24.82 |
| 7 | 0.7 | 0.8 | 16.69 |
| 7 | 0.8 | 0.8 | 10.99 |
| 7 | 0.9 | 0.8 | 9.78 |
| 7 | 0.99 | 0.8 | 7.36 |
| 7 | 0.1 | 1 | 117.93 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 7 | 0.2 | 1 | 104.80 |
| 7 | 0.3 | 1 | 64.83 |
| 7 | 0.4 | 1 | 41.11 |
| 7 | 0.5 | 1 | 28.41 |
| 7 | 0.6 | 1 | 20.05 |
| 7 | 0.7 | 1 | 14.05 |
| 7 | 0.8 | 1 | 11.92 |
| 7 | 0.9 | 1 | 9.37 |
| 7 | 0.99 | 1 | 7.35 |
| 7 | 0.1 | 1.2 | 108.33 |
| 7 | 0.2 | 1.2 | 37.92 |
| 7 | 0.3 | 1.2 | 31.35 |
| 7 | 0.4 | 1.2 | 25.18 |
| 7 | 0.5 | 1.2 | 19.37 |
| 7 | 0.6 | 1.2 | 13.88 |
| 7 | 0.7 | 1.2 | 12.23 |
| 7 | 0.8 | 1.2 | 11.91 |
| 7 | 0.9 | 1.2 | 9.03 |
| 7 | 0.99 | 1.2 | 6.89 |
| 7 | 0.1 | 1.4 | 83.83 |
| 7 | 0.2 | 1.4 | 52.45 |
| 7 | 0.3 | 1.4 | 36.74 |
| 7 | 0.4 | 1.4 | 27.05 |
| 7 | 0.5 | 1.4 | 20.25 |
| 7 | 0.6 | 1.4 | 15.16 |
| 7 | 0.7 | 1.4 | 11.35 |
| 7 | 0.8 | 1.4 | 11.86 |
| 7 | 0.9 | 1.4 | 8.76 |
| 7 | 0.99 | 1.4 | 7.35 |
| 7 | 0.1 | 1.6 | 60.41 |
| 7 | 0.2 | 1.6 | 41.82 |
| 7 | 0.3 | 1.6 | 30.95 |
| 7 | 0.4 | 1.6 | 23.51 |
| 7 | 0.5 | 1.6 | 12.98 |
| 7 | 0.6 | 1.6 | 12.29 |
| 7 | 0.7 | 1.6 | 12.05 |
| 7 | 0.8 | 1.6 | 10.96 |
| 7 | 0.9 | 1.6 | 7.66 |
| 7 | 0.99 | 1.6 | 7.35 |
| 7 | 0.1 | 1.8 | 25.56 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 7 | 0.2 | 1.8 | 21.14 |
| 7 | 0.3 | 1.8 | 16.86 |
| 7 | 0.4 | 1.8 | 13.57 |
| 7 | 0.5 | 1.8 | 12.55 |
| 7 | 0.6 | 1.8 | 12.21 |
| 7 | 0.7 | 1.8 | 12.06 |
| 7 | 0.8 | 1.8 | 11.78 |
| 7 | 0.9 | 1.8 | 8.41 |
| 7 | 0.99 | 1.8 | 7.35 |
| 7 | 0.1 | 2 | 21.96 |
| 7 | 0.2 | 2 | 17.91 |
| 7 | 0.3 | 2 | 14.41 |
| 7 | 0.4 | 2 | 12.86 |
| 7 | 0.5 | 2 | 12.36 |
| 7 | 0.6 | 2 | 12.17 |
| 7 | 0.7 | 2 | 12.03 |
| 7 | 0.8 | 2 | 11.69 |
| 7 | 0.9 | 2 | 8.60 |
| 7 | 0.99 | 2 | 7.42 |
| 7 | 0.1 | 2.2 | 18.99 |
| 7 | 0.2 | 2.2 | 15.44 |
| 7 | 0.3 | 2.2 | 13.30 |
| 7 | 0.4 | 2.2 | 12.59 |
| 7 | 0.5 | 2.2 | 12.26 |
| 7 | 0.6 | 2.2 | 12.10 |
| 7 | 0.7 | 2.2 | 13.62 |
| 7 | 0.8 | 2.2 | 10.69 |
| 7 | 0.9 | 2.2 | 8.53 |
| 7 | 0.99 | 2.2 | 7.42 |
| 7 | 0.1 | 2.4 | 16.58 |
| 7 | 0.2 | 2.4 | 13.94 |
| 7 | 0.3 | 2.4 | 12.85 |
| 7 | 0.4 | 2.4 | 12.43 |
| 7 | 0.5 | 2.4 | 12.23 |
| 7 | 0.6 | 2.4 | 12.11 |
| 7 | 0.7 | 2.4 | 11.99 |
| 7 | 0.8 | 2.4 | 10.58 |
| 7 | 0.9 | 2.4 | 8.47 |
| 7 | 0.99 | 2.4 | 7.42 |
| 7 | 0.1 | 2.6 | 14.80 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 7 | 0.7 | 2.6 | 9.94 |
| 7 | 0.8 | 2.6 | 8.20 |
| 7 | 0.9 | 2.6 | 7.18 |
| 7 | 0.99 | 2.6 | 6.88 |
| 7 | 0.1 | 2.8 | 13.80 |
| 7 | 0.2 | 2.8 | 12.88 |
| 7 | 0.3 | 2.8 | 12.46 |
| 7 | 0.4 | 2.8 | 12.29 |
| 7 | 0.5 | 2.8 | 12.14 |
| 7 | 0.6 | 2.8 | 12.08 |
| 7 | 0.7 | 2.8 | 11.94 |
| 7 | 0.8 | 2.8 | 10.36 |
| 7 | 0.9 | 2.8 | 8.36 |
| 7 | 0.99 | 2.8 | 6.88 |
| 10 | 0.1 | 0.2 | 229.97 |
| 10 | 0.2 | 0.2 | 204.62 |
| 10 | 0.3 | 0.2 | 181.86 |
| 10 | 0.4 | 0.2 | 161.77 |
| 10 | 0.5 | 0.2 | 181.66 |
| 10 | 0.6 | 0.2 | 99.85 |
| 10 | 0.7 | 0.2 | 59.60 |
| 10 | 0.8 | 0.2 | 33.30 |
| 10 | 0.9 | 0.2 | 16.18 |
| 10 | 0.99 | 0.2 | 11.32 |
| 10 | 0.1 | 0.4 | 134.73 |
| 10 | 0.2 | 0.4 | 131.09 |
| 10 | 0.3 | 0.4 | 127.50 |
| 10 | 0.4 | 0.4 | 122.34 |
| 10 | 0.5 | 0.4 | 107.44 |
| 10 | 0.6 | 0.4 | 59.58 |
| 10 | 0.7 | 0.4 | 34.17 |
| 10 | 0.8 | 0.4 | 20.13 |
| 10 | 0.9 | 0.4 | 11.97 |
| 10 | 0.99 | 0.4 | 10.31 |
| 10 | 0.1 | 0.6 | 127.46 |
| 10 | 0.2 | 0.6 | 124.20 |
| 10 | 0.3 | 0.6 | 120.01 |
| 10 | 0.4 | 0.6 | 106.26 |
| 10 | 0.5 | 0.6 | 61.29 |
| 10 | 0.6 | 0.6 | 36.89 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 10 | 0.7 | 0.6 | 23.82 |
| 10 | 0.8 | 0.6 | 9.96 |
| 10 | 0.9 | 0.6 | 12.00 |
| 10 | 0.99 | 0.6 | 11.21 |
| 10 | 0.1 | 0.8 | 123.39 |
| 10 | 0.2 | 0.8 | 119.08 |
| 10 | 0.3 | 0.8 | 107.14 |
| 10 | 0.4 | 0.8 | 65.99 |
| 10 | 0.5 | 0.8 | 40.77 |
| 10 | 0.6 | 0.8 | 27.58 |
| 10 | 0.7 | 0.8 | 19.13 |
| 10 | 0.8 | 0.8 | 13.59 |
| 10 | 0.9 | 0.8 | 11.97 |
| 10 | 0.99 | 0.8 | 11.20 |
| 10 | 0.1 | 1 | 119.13 |
| 10 | 0.2 | 1 | 109.33 |
| 10 | 0.3 | 1 | 73.49 |
| 10 | 0.4 | 1 | 45.66 |
| 10 | 0.5 | 1 | 31.61 |
| 10 | 0.6 | 1 | 22.72 |
| 10 | 0.7 | 1 | 16.61 |
| 10 | 0.8 | 1 | 12.20 |
| 10 | 0.9 | 1 | 11.92 |
| 10 | 0.99 | 1 | 11.19 |
| 10 | 0.1 | 1.2 | 111.59 |
| 10 | 0.2 | 1.2 | 40.36 |
| 10 | 0.3 | 1.2 | 33.55 |
| 10 | 0.4 | 1.2 | 27.16 |
| 10 | 0.5 | 1.2 | 21.15 |
| 10 | 0.6 | 1.2 | 15.39 |
| 10 | 0.7 | 1.2 | 12.54 |
| 10 | 0.8 | 1.2 | 12.16 |
| 10 | 0.9 | 1.2 | 11.93 |
| 10 | 0.99 | 1.2 | 9.85 |
| 10 | 0.1 | 1.4 | 94.22 |
| 10 | 0.2 | 1.4 | 59.44 |
| 10 | 0.3 | 1.4 | 41.13 |
| 10 | 0.4 | 1.4 | 30.41 |
| 10 | 0.5 | 1.4 | 23.13 |
| 10 | 0.6 | 1.4 | 17.87 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 10 | 0.7 | 1.4 | 14.10 |
| 10 | 0.8 | 1.4 | 12.09 |
| 10 | 0.9 | 1.4 | 11.89 |
| 10 | 0.99 | 1.4 | 11.19 |
| 10 | 0.1 | 1.6 | 69.33 |
| 10 | 0.2 | 1.6 | 47.10 |
| 10 | 0.3 | 1.6 | 34.79 |
| 10 | 0.4 | 1.6 | 26.69 |
| 10 | 0.5 | 1.6 | 14.10 |
| 10 | 0.6 | 1.6 | 12.61 |
| 10 | 0.7 | 1.6 | 12.24 |
| 10 | 0.8 | 1.6 | 14.62 |
| 10 | 0.9 | 1.6 | 10.81 |
| 10 | 0.99 | 1.6 | 11.19 |
| 10 | 0.1 | 1.8 | 28.05 |
| 10 | 0.2 | 1.8 | 23.63 |
| 10 | 0.3 | 1.8 | 19.38 |
| 10 | 0.4 | 1.8 | 15.39 |
| 10 | 0.5 | 1.8 | 13.07 |
| 10 | 0.6 | 1.8 | 12.41 |
| 10 | 0.7 | 1.8 | 12.18 |
| 10 | 0.8 | 1.8 | 12.05 |
| 10 | 0.9 | 1.8 | 11.86 |
| 10 | 0.99 | 1.8 | 11.19 |
| 10 | 0.1 | 2 | 24.51 |
| 10 | 0.2 | 2 | 20.53 |
| 10 | 0.3 | 2 | 16.71 |
| 10 | 0.4 | 2 | 13.76 |
| 10 | 0.5 | 2 | 12.71 |
| 10 | 0.6 | 2 | 12.31 |
| 10 | 0.7 | 2 | 12.18 |
| 10 | 0.8 | 2 | 12.07 |
| 10 | 0.9 | 2 | 11.48 |
| 10 | 0.99 | 2 | 10.26 |
| 10 | 0.1 | 2.2 | 89.99 |
| 10 | 0.2 | 2.2 | 72.06 |
| 10 | 0.3 | 2.2 | 55.99 |
| 10 | 0.4 | 2.2 | 41.94 |
| 10 | 0.5 | 2.2 | 30.86 |
| 10 | 0.6 | 2.2 | 22.98 |

| Dispersion Coefficient | Beta | Omega | Equivalent Dispersion Coefficient |
|---|---|---|---|
| 10 | 0.7 | 2.2 | 17.64 |
| 10 | 0.8 | 2.2 | 13.97 |
| 10 | 0.9 | 2.2 | 11.39 |
| 10 | 0.99 | 2.2 | 10.26 |
| 10 | 0.1 | 2.4 | 19.27 |
| 10 | 0.2 | 2.4 | 16.00 |
| 10 | 0.3 | 2.4 | 13.71 |
| 10 | 0.4 | 2.4 | 12.79 |
| 10 | 0.5 | 2.4 | 12.39 |
| 10 | 0.6 | 2.4 | 12.25 |
| 10 | 0.7 | 2.4 | 12.16 |
| 10 | 0.8 | 2.4 | 13.77 |
| 10 | 0.9 | 2.4 | 11.31 |
| 10 | 0.99 | 2.4 | 10.26 |
| 10 | 0.1 | 2.6 | 17.29 |
| 10 | 0.7 | 2.6 | 12.56 |
| 10 | 0.8 | 2.6 | 11.00 |
| 10 | 0.9 | 2.6 | 10.09 |
| 10 | 0.99 | 2.6 | 9.85 |
| 10 | 0.1 | 2.8 | 15.69 |
| 10 | 0.2 | 2.8 | 13.75 |
| 10 | 0.3 | 2.8 | 12.90 |
| 10 | 0.4 | 2.8 | 12.48 |
| 10 | 0.5 | 2.8 | 12.34 |
| 10 | 0.6 | 2.8 | 12.20 |
| 10 | 0.7 | 2.8 | 12.08 |
| 10 | 0.8 | 2.8 | 13.43 |
| 10 | 0.9 | 2.8 | 11.19 |
| 10 | 0.99 | 2.8 | 9.85 |

# Bibliography

[1] Biggar, J. W. and Nielsen, D. R. Miscible Displacement in Soils. I. Experimental Information. *Soil Science Society of America Journal*, 25:1-5, 1961.

[2] Biggar, J. W. and Nielsen, D. R. Miscible Displacement II. Behaviour of Tracers. *Soil Science Society of America Proceedings*, 26:125-128, 1962.

[3] Biggar, J. W. and Nielsen, D. R. Miscible Displacement and Leaching Phenomena. *Irrigation of Agricultural Lands. Agronomy*, 11:254-274, 1967.

[4] Brenner, H. The diffusion model of longitudinal mixing in beds of finite length numerical values. *Chemical Engineering Science*, 17:299-243, 1962.

[5] Cameron, D. A. and Klute, A. Convective-Despersive solute transport with a combined equilibrium and kinetic adsorption model. *Water Resources Research*, 13:183-188, 1977.

[6] Cleary, R. W. and Adrian, D. D. Analytical solution of the convective-despersive equation for cation adsorption in soils. *Soil Science Society of America Proceedings*, 37(2):197-199, 1973.

[7] Coats, K. H. and Smith, B. D. Dead-end pore Volume and dispersion in porous media. *Society of Petroleum Engineers Journal*, 4:73–84, 1964.

[8] De smedt, F. and Wierenga, P. J. A generalized solution for solute flow in soils with modile and immobile water. *Water Resources Research*, 15:1137–1141, 1979.

[9] Gershon, N. D. and Nir, A. Effect of boundary conditions of models on tracer distribution in flow through porous mediums. *Water Resources Research*, 5(4):830–840, 1969.

[10] Gori, M. and Tesi, A. On the problem of local Minima in Backpropagation. *IEEE Trans. Pattern Analysis Match. Intellig.*, 14:141–166, 1992.

[11] Haykin, S. *Neural Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1999.

[12] Howard, K. Groundwater flow, Contaminant transport and well protection - An overview. Technical report, Groundwater Research group, University of Toronto at Scarborough 1265 Military Trail Scarborough Ontario, 2000.

[13] Kay, B. D. and Elrick, D. E. Adsorption and movement of lindane in soils. *Soil Science*, 104:314–322, 1967.

[14] Krupp, H. K., Biggar, J. W. and Nielsen, D. R. Relative flow rates of salt and water in soil. *Soil Science Society Of American Proceeding*, 36:412–417, 1972.

[15] Lapidus, L. and Amundson N. R. Mathematics of adsorption in beds. VI. The effect of longitudinal diffusion in oion exchange and chromatographis columns. *Journal of physical chemistry*, 56:984–988, 1952.

[16] Lindstorm, F.T. and Stone, M. W. On the start or initial phase of linear mass transport of chemicals in water saturated sorbing medium, I. *Journal of Applied Mathematics*, 26(3):578–591, 1974a.

[17] Lindstorm, F.T. and Stone. M. W. On the start or initial phase of linear mass transport of chemicals in water saturated sorbing medium, II. Integral eqution approach. *Journal of Applied Mathematics*. 26(3):592–606. 1974b.

[18] Lindstorm, F.T.. Haque, R.. Freed, V.H. and Boersma, L. Theory on the movement of some herbicides in soils: Linear diffusion and convection of chemicals in soils. *Journal of Environmental Science and Technology*. 1:561–565, 1967.

[19] Marino, M.A. Numerical and analytical soultions of dispersion in a finite, adsorbing porous medium. *Water Resources Bulletin*, 10(1):81–90, 1974a.

[20] Marino, M.A. Distribution of contaminants in porous medium. *Water Resources Research*, 10(5):1013–1018, 1974b.

[21] Mohaghegh, S. Virtual-Inteeligence Applications in Petroleum Engineering: Part 1- Artificial Neural Networks. *Society of Petroleum Engineers*, pages 64–73, 2000.

[22] Nielsen, D. R. and Biggar, J. W. Miscible displacement in soils: I. Experimental information. *Soil Science Society Of America Proceedings*, 25:1–5, 1961.

[23] Nkedi-Kizza, P., Biggar, J. W., Selim, H. M. and Van Genuchten, M. Th. On the equivalence of two conceptual models for describing ion exchange during transport through an aggegated oxisol. *Water Resources Research*, 20:1123–1130, 1984.

[24] Selim, H. M., Davidson, J. M. and Mansell, R. S. Evaluation of a two-site adsorption-desorption model for describing solute transport in soils. *In Proceedings Summer Computer Simulation Conference., Washigton, D. C*, pages 444–448, 1976.

[25] Skopp, J. and Warrick, A. W. A two-phase model for the miscible displacement of reactive solutes through soils. *Soil Science Society Of American Proceedings*, 38:545–550, 1974.

[26] Toride, N., Leiji, F. J. and Van Genuchten, M. Th. The CXTFIT code for extimating transport parameters from laboratory or field tracer experiments. Technical Report 137, U. S. Salinity Laboratory, USDA-SEA-AR, Riverside, California, 1991.

[27] Turner, G. A. Miscible Displacement in Soils. I. Experimental Information. *Soil Science Society of America Journal*, 25:1–5, 1961.

[28] Van Genucheten, M. Th. Analytical solutions for chemical transport with simultaneous adsorption, zero-order production and first-order decay. *Journal of Hydrology*, 49:213–223, 1981.

[29] Van Genucheten, M. Th. and Wierenga, P. J. Mass transfer studies in sorbing porous media: I. Analytical solutions. *Soil Science Society Of America Journal*, 40:473–480, 1976.

[30] Van Genucheten, M. Th., Davidson, J. M. and Wierenga, P. J. An evaluation of kinetic and equilibrium equations for the prediction of pesticide movement through porous media. *Soil Science Society Of America Proceedings*, 38:29–35, 1974.

[31] Van Genuchten, M. Th. *Mass Transfer Studies in Sorbing Porous Media*. PhD thesis, New Mexico State University, Las Cruces, New Mexico, 1974.

[32] Van Genuchten, M. Th. Non-equlibrium transport parameters from miscible displacement experiments. Technical Report 119, U. S. Salinity Laboratory, USDA-SEA-AR, Riverside, California, 1981.

[33] Van Genuchten, M. Th. and Wagnet, R. J. Two-site/two-region models for pesticide transport and degradation: theoretical development and analytical solutions. *Soil Science Society Of America Jouranl*, 53:1303–1310, 1989.

[34] Versteg, K. H. and Malalasekera, W. *An introduction to Computational Fluid dynamics*. Longman Scientific Technical, 1995.

[35] Yann, L. C., Leon, B. Genevieve, B. O. and Klaus, R. M. Efficient BackProp. *Neural Networks: Trick of Trade*, 1998.

[36] Yu, X. H. Can Backpropagation surface not have local Minima. *IEEE Trans. Neural Networks*, 3:1019–1992, 1992.

[37] Zhang, Y. X. Fast Learning with Backpropagating algorithm with a sine type thresholding Function. *Appl. Opt.*, 31:2414-2416, 1992.

# Vitae

- **Abid Maqsood Ahmad**

- Born in Lahore, Pakistan in 1976.

- Received Bachelor of Science Degree in Civil Engineering from
  University of Engineering and Technology, Lahore, Pakistan in 1999.

- Served as Junior Engineer Structures at National Engineering Services
  of Pakistan (NESPAK), Lahore, Pakistan from 1999 to 2000.

- Joined Civil Engineering Department, KFUPM, as a Research Assistant
  in September 2000.

- Received Master of Science Degree in Civil Engineering from KFUPM,
  Dhahran, Saudi Arabia in November 2002.