

การพัฒนาโครงข่ายประสาทเทียมเพื่อการประเมินการปนเปื้อนของสารมลพิษในน้ำใต้ดินได้อย่าง
รวดเร็ว



นางสาวสายนที จำปีรัตน์

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

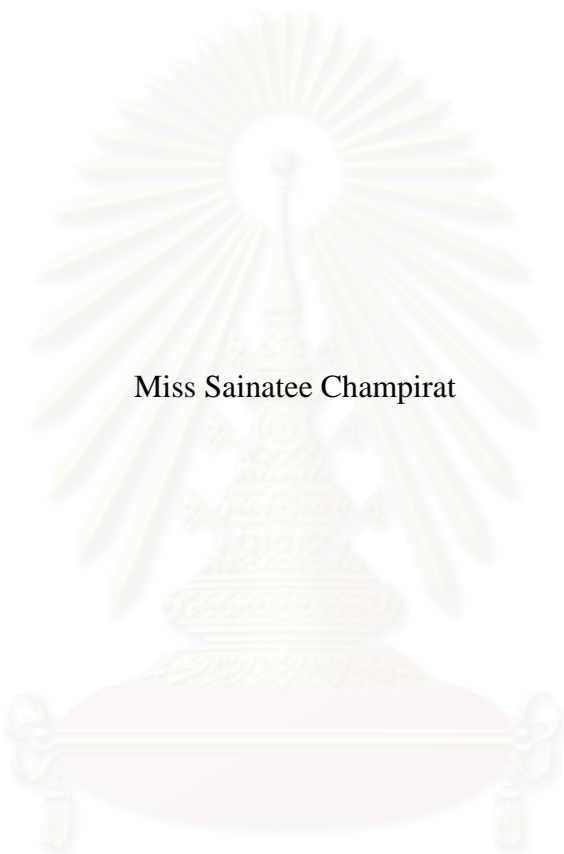
สาขาวิชาการจัดการสิ่งแวดล้อม (สหสาขาวิชา)

บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK FOR THE RAPID
ASSESSMENT OF SUBSURFACE CONTAMINATION



Miss Sainatee Champirat

A Thesis Submitted in Partial Fulfillment of the Requirements
For the Degree of Master of Science Program in Environmental Management
(Interdisciplinary Program)

Graduate School

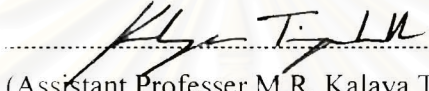
Chulalongkorn University

Academic Year 2006

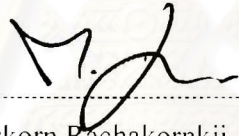
Copyright of Chulalongkorn University


Thesis Title A DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK
FOR THE RAPID ASSESSMENT OF SUBSURFACE
CONTAMINATION
By Miss Sainatee Champirat
Field of study Environmental Management
Thesis Advisor Professor Kanthasamy K. Muraleetharan, Ph.D.
Thesis Co-advisor Punjaporn Weschayanwivat, Ph.D.

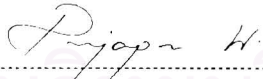
Accepted by the Graduate School, Chulalongkorn University in Partial
Fulfillment of the Requirements for the Master's Degree

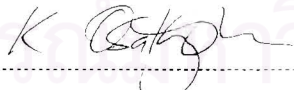
..... Dean of the Graduate School
(Assistant Professor M.R. Kalaya Tingsabadh, Ph.D.)


THESIS COMMITTEE

..... Chairman
(Manaskorn Rachakornkij, Ph.D.)

..... Thesis Advisor
(Professor Kanthasamy K. Muraleetharan, Ph.D.)

..... Thesis Co-advisor
(Punjaporn Weschayanwivat, Ph.D.)

..... Member
(Assistant Professor Khemarath Osathaphan, Ph.D.)

..... Member
(Oranuj Lorphensri, Ph.D.)

สาขานี้ จำปรีดน์ : การพัฒนาโครงข่ายประสาทเทียมเพื่อการประเมินการปนเปื้อนของสารมลพิษในน้ำใต้ดินได้อย่างรวดเร็ว. (A DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK FOR THE RAPID ASSESSMENT OF SUBSURFACE CONTAMINATION).
 อ. ที่ปรึกษา: PROFESSOR KANTHASAMY K. MURALEETHARAN, Ph.D., อ.ที่ปรึกษา
 ร่วม: อาจารย์ ดร. ปัญจพร เวชยันต์วิวัฒน์. 117 หน้า.

ปัจจุบันหลายประเทศทั่วโลกได้ให้ความสนใจและตระหนักในปัญหาที่เกิดจากการปนเปื้อนและการเคลื่อนที่ของสารมลพิษในน้ำใต้ดิน เครื่องมือที่ใช้ในการประเมินการเคลื่อนที่ของสารมลพิษในน้ำใต้ดินจึงมีความสำคัญและจำเป็นอย่างมากในการจัดลำดับความสำคัญและประเมินผลกระทบในเบื้องต้นเพื่อทำการบำบัดสารปนเปื้อนเหล่านั้น เครื่องมือเหล่านี้ควรมีความสะดวกต่อการใช้งานและมีความน่าเชื่อถือ ดังนั้นงานวิจัยนี้จึงได้ทำการศึกษาความเป็นไปได้ในการใช้โครงข่ายประสาทเทียม (Artificial Neural Network) แบบฟีดฟอรวอร์ดแบคพรอพพากชัน (Feedforward Backpropagation) เพื่อการทำนายความเข้มข้นของสารมลพิษที่ปนเปื้อนในน้ำใต้ดิน ณ จุดใดๆ ได้อย่างรวดเร็ว ฐานข้อมูลที่โครงข่ายประสาทเทียมใช้เพื่อการเรียนรู้ นั้น เป็นชุดข้อมูลที่ได้จากการจำลองสถานการณ์การเคลื่อนที่ของสารมลพิษในพื้นที่ต่างๆ กันจากโปรแกรม Visual MODFLOW โดยในฐานข้อมูลที่ใช้จะประกอบด้วย ข้อมูลความเข้มข้นของสารมลพิษ ณ พื้นที่หนึ่งๆ และเวลาหนึ่งๆ ที่ลักษณะพื้นที่ต่างๆ กัน นอกจากนี้ยังประกอบไปด้วยค่าของตัวแปรอีก 4 ตัวแปร คือ ความสามารถของดินในการให้น้ำไหลผ่าน การกระจายตัวของสารปนเปื้อนตามทิศทางการไหล ความแตกต่างของความดันน้ำคงที่ ณ บริเวณตอนเหนือและตอนท้ายแหล่งกำเนิด และอัตราการไหลลงสู่ น้ำใต้ดินของสารปนเปื้อน ผลที่ได้แสดงให้เห็นว่า โครงข่ายประสาทเทียมที่ประกอบด้วย 2 ชั้นซ่อน (hidden layers) ให้ผลการทำนายความเข้มข้นของสารมลพิษได้แม่นยำกว่าโครงข่ายประสาทเทียมที่ประกอบด้วย 1 ชั้นซ่อน โดยให้ค่าผลรวมของผลต่างกำลังสองระหว่างค่าจริงจากชุดข้อมูลที่ใช้ในการทดสอบโครงข่ายประสาทเทียมและค่าที่ได้จากการทำนายของโครงข่ายประสาทเทียมต่ำกว่า นอกจากนี้แล้ว วิธีการเรียนรู้ของโครงข่ายประสาทเทียมโดยใช้วิธีการที่เรียกว่า Levenburg-Marquardt optimization ร่วมกับ Bayesian regularization พร้อมทั้งทำการเตรียมข้อมูลก่อนนำเข้าสู่ระบบ ยังส่งผลให้การทำนายมีความแม่นยำมากยิ่งขึ้นอีกด้วย จากงานวิจัยครั้งนี้สรุปได้ว่าโครงข่ายประสาทเทียมสามารถใช้เพื่อการทำนายความเข้มข้นของสารมลพิษที่ปนเปื้อนลงสู่ น้ำใต้ดิน โดยการกำหนดเพียงเวลา สถานที่ และตัวแปรที่จำเป็นสำหรับแบบจำลองทางคณิตศาสตร์เพียงบางตัวเท่านั้น

สาขาวิชา... การจัดการสิ่งแวดล้อม... ลายมือชื่อนิสิต *Savitree Champirat*

ปีการศึกษา 2549... ลายมือชื่ออาจารย์ที่ปรึกษา *K. K. M*

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม *Pijaporn W.*

4889510320: MAJOR ENVIRONMENTAL MANAGEMENT

KEYWORD: CONTAMINANT TRANSPORT / GROUNDWATER / ARTIFICIAL NEURAL NETWORK

SAINATEE CHAMPIRAT: A DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK FOR THE RAPID ASSESSMENT OF SUBSURFACE CONTAMINATION. THESIS ADVISOR: PROFESSOR KANTHASAMY K. MURALEETHARAN, Ph.D., THESIS CO-ADVISOR: PUNJAPORN WESCHAYANWIWAT, Ph.D., 117 pp.

Contaminant transport in groundwater is a major environmental concern in many countries. It is desirable for users and regulatory agencies to have a reliable and convenient screening method to estimate the contaminant migration. Such a tool must allow users to identify critical sites and prioritize the clean-up and further investigative efforts. This study has investigated the feasibility of using an artificial neural network (ANN) as a quick predictor. A feedforward backpropagation neural network (FBNN) was used to train a data set simulated from the Visual MODFLOW program. The data set contained spatial and temporal variations of concentrations of one contaminant and the specified groundwater model parameters including hydraulic conductivity, longitudinal dispersivity, constant head difference, and recharge rate at the source of contamination under several hypothetical conditions. 1- and 2-hidden-layer FBNNs with various numbers of neurons were trained. Levenberg-Marquardt optimization with Bayesian regularization was found to be the best training algorithm for this training data set because of its capability of easing overfitting problem and improving generalization of the network. Data preprocessing prior training the network is also a crucial step for improving accuracy of prediction. The network with 2 hidden layers showed more accurate prediction than that with 1 hidden layer as indicated by the lower mean square error of the validation data set. The result indicates that this tool can be used to predict the concentration of contaminant as a function of time for given locations and model parameters.

Field of study Environmental Management Student's signature Sainatee Champirat
 Academic year 2006 Advisor's signature K.K. M
 Co-advisor's signature Pjip- H

Acknowledgements

First of all, I would like to offer my sincere thanks to Professor Kanthasamy K. Muraleetharan, Ph.D., my thesis advisor, for his encouragement, and careful and wise guidance of my thesis. Moreover, I am pleased to acknowledge the support of my co-advisor, Punjaporn Wechayaviwat, Ph.D., for her advice. Another person is Akiko Uyeda, my English Instructor, whom I am deeply indebted to for helping correcting English and revising.

Additionally, I would like to express my deep gratitude to the Environmental and Hazardous Waste Management program at Chulalongkorn University for granting me support to pursue my graduate studies. The support has given me not only a future career, but also opportunities and experiences in environmental work.

Special thanks also go to my thesis committees, Manaskorn Rachakornkij, Ph.D., Assistant Professor Khemarath Osathaphan, Ph.D., and Oranuj Lorphenstri, Ph.D., who gave insightful comments and reviewed my work on a very short notice. I thanks to staffs at NRC-EHWM, Pamornrat Waichai, Soontorn Pannok, and Walanya Kongseng, for helping with my graduation process.

I would like to thanks all my friends for helping me especially in official document works with Thai government agencies during a time I was in USA. Moreover, I would like to say thank you to N. (Ravi) Ravichandran, Ph.D. and Danny Krier, my computer lab mates, for helping me at any time I had problems. Thanks to my Thai friends at Norman, Oklahoma, for their caring. I thank to Direk Khajonrat for his constant support of my work.

Finally, I owe much to my parents and sister for always believing in me and encouraging me to achieve my goals.

Content

	Pages
Abstract (in Thai)	iv
Abstract (in English)	v
Acknowledgements	vi
Content	vii
List of Tables	ix
List of Figures	xi
Chapter I: Introduction	1
1.1 Problem.....	1
1.2 Objectives.....	2
1.3 Expected Results.....	3
Chapter II: Literature Review	4
2.1 Groundwater Contamination.....	4
2.1.1 Groundwater Hydrology.....	6
2.1.2 Contaminant Transport.....	11
2.2 Artificial Neural Network.....	18
2.2.1 Background.....	18
2.2.2 Types of Artificial Neural Networks (ANNs).....	19
2.2.3 Feedforward Neural Network Architecture.....	21
2.2.4 Feedforward Backpropagation Neural Network (FBNN).....	27
2.2.5 Bayesian Regularization: Improving Generalization.....	29
2.3 Related Research.....	31
Chapter III: Methodology	34
3.1 The Methodology Framework.....	34
3.2 The Creation of the Hypothetical Site Using MODFLOW.....	35
3.3 The Determination of the Inputs for the ANNs.....	46
3.4 The Generation of the Data.....	51

3.5	The Preparation of the Data Set.....	52
3.6	The Development of the ANN and Network Training.....	58
3.7	The Testing of Network.....	62
3.8	The Network Performance Criteria.....	63
3.9	The Numerical Experiments.....	64
3.10	The Flowchart of Methodology.....	69
Chapter IV: Results and Discussion.....		71
4.1	Phase I: Determine the Training Function.....	71
4.2	Phase II: Determine the Data Preprocessing Method.....	72
4.3	Phase III: Determine the Network Architecture.....	76
4.4	Illustration of Breakthrough Curve.....	86
4.5	Practical Application of a Developed Neural Network.....	88
4.6	Efficiency of ANN applications in different scenarios.....	89
Chapter V: Summary and Suggestion.....		90
5.1	Summary.....	90
5.2	Suggestions.....	91
Reference.....		93
Appendices.....		95
Biography.....		117

List of Tables

	Pages
Table 2-1. Estimated distribution of the world's water.....	6
Table 2-2. Saturated hydraulic conductivity (K) values found in nature.....	11
Table 3-1. Model Dimensions.....	35
Table 3-2. The varied values of hydraulic conductivity (m/s).....	37
Table 3-3. The varied values of longitudinal dispersivity (m).....	37
Table 3-4. The summary of the dispersivity variables.....	38
Table 3-5. The default property values given by MODFLOW.....	38
Table 3-6. The varied recharge rate at the source of contamination (mm/year).....	39
Table 3-7. The varied values of constant head difference (m).....	40
Table 3-8. The model transport and chemical reaction parameters.....	41
Table 3-9. The locations of the observation wells.....	44
Table 3-10. The parameters of the site_base.....	47
Table 3-11. The values of the 4 parameters at each site.....	48
Table 3-12. The summary of the variation of the 4 parameters.....	51
Table 3-13. The numbers of sites that contain each value of the 4 model parameters used in training and testing data sets.....	55
Table 3-14. Training function and its description.....	65
Table 3-15. The network architecture.....	67
Table 3-16. The summary of the phases and the numerical experiments.....	68
Table 4-1. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 1.....	73
Table 4-2. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 2.....	73

Table 4-3.	The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 3.....	74
Table 4-4.	The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase III: Numerical Experiment No. 1.....	78
Table 4-5.	The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase III: Numerical Experiment No. 2.....	79
Table 4-6.	The percentage of the number of data points in the error boundaries (Experiment No. 1).....	84
Table 4-7.	The percentage of the number of data points in the error boundaries (Experiment No. 2).....	85

List of Figures

	Pages
Fig. 2-1. Hydrologic cycle.....	5
Fig. 2-2. Aquifer.....	7
Fig. 2-3. Darcy's experiment.....	8
Fig. 2-4. Sources of groundwater contamination.....	12
Fig. 2-5. Breakthrough curve of C/C_0 and time (days) at a point of interest.....	18
Fig. 2-6. A single-layer network.....	21
Fig. 2-7. Multiple-Layer Neural Network.....	24
Fig. 2-8. Three examples of the transfer function.....	26
Fig. 2-9. The structure of a feedforward backpropagation neural network.....	28
Fig. 2-10. Gradient descent for a network.....	29
Fig. 3-1. Top view of the hypothetical site.....	36
Fig. 3-2. Side view of the hypothetical site.....	36
Fig. 3-3. Assigning hydraulic conductivity and longitudinal dispersivity.....	38
Fig. 3-4. Assigning the recharge boundaries.....	39
Fig. 3-5. Assigning the constant head boundaries.....	40
Fig. 3-6. Setup of assigning the model transport parameters.....	41
Fig. 3-7. Assigning the concentration of the contaminant at the source of contamination.....	42
Fig. 3-8. The locations of the observation wells.....	43
Fig. 3-9. Setting the WHS solver parameters.....	45
Fig. 3-10. Setting the transport simulation options.....	45
Fig. 3-11. The comparison between 7 breakthrough curves from 7 values of hydraulic conductivity.....	49
Fig. 3-12. The comparison between 6 breakthrough curves from 6 values of longitudinal dispersivity.....	49
Fig. 3-13. The comparison between 8 breakthrough curves from 8 values of constant head difference.....	50

Fig. 3-14. The comparison between 6 breakthrough curves from 6 values of recharge flux at the source of the contamination.....	50
Fig. 3-15. The combination of the 4 parameters to create one site.....	52
Fig. 3-16. An example of how data were organized for one site.....	53
Fig. 3-17. Steps of creating the data sets.....	54
Fig. 3-18. The combination of the data.....	56
Fig. 3-19. How the data sets were flipped.....	57
Fig. 3-20. Flow chart of the methodology.....	70
Fig. 4-1. The plot between the desired C/C_0 versus the predicted C/C_0 simulated from the ANN.....	75
Fig. 4-2. The plot of the desired C/C_0 versus the predicted C/C_0 simulated from the 1-hidden-layer FBNNs.....	80
Fig. 4-3. The plot of the desired C/C_0 versus the predicted C/C_0 simulated from the 2-hidden-layer FBNNs.....	82
Fig. 4-4. Breakthrough curves of the 5 observation wells of each site.....	86
Fig. 4-5. Some examples of the less accurate predictions.....	87
Fig. 4-6. Graphic User Interface.....	89

CHAPTER I

INTRODUCTION

1.1 Statement of the Problems

Groundwater, water found beneath the earth's surface, is stored in the cracks of rocks or the spaces in soil and sand. It is a renewable resource by the natural process of the hydrological cycle. Because groundwater exists everywhere, it is a major source of the world's supply of drinkable water. The availability of groundwater varies from one region to another depending on the geology of the area. The quantity available and geologic materials and conditions control the occurrence of groundwater (Chester, 2000). The movement of groundwater is slower than the flow of surface water. It is governed by hydraulic gradients and aquifer permeability.

Groundwater can be the subsurface transporting agent for dissolved chemicals, including contaminants. Materials dissolved from wastes may be transported from burial or disposal sites by groundwater flow. These chemical contaminants reduce the quality of the groundwater. Due to the slow rates of groundwater movement and the natural flushing of aquifers, contaminated areas commonly remain so for decades or longer. Most of the hazardous substances that contaminate groundwater are man-made products such as gasoline, oil, road salts, and chemicals. Some of the major sources of these products are storage tanks, septic systems, hazardous waste sites, and landfills. Because of the volumes of hazardous substances and because of their stability in groundwater, such contamination can pose a serious threat to public health, groundwater-dependent economic activities, and the long-term welfare of valuable ecosystems. If groundwater becomes polluted, it becomes unsafe and unfit for human use (Geophysics Research Forum (U.S.), Geophysics Study Committee, 1984).

Drinking or consuming contaminated groundwater can have serious health effects. Diseases or poisoning may be caused by toxins that have leached into well

water supplies. Other long-term effects, such as certain types of cancer, may result from exposure to polluted water.

One of the major environmental concerns in many countries is the problem of groundwater contamination caused by the leachate from landfills or chemical leakage from underground storage tanks. It is desirable for the users and regulatory agencies to have a reliable, rapid and easy to use screening tool for estimating contaminant migration in order to identify critical sites and prioritize the clean-up and further investigative efforts. Such a tool would preclude a reliance solely on the traditional approach that involves time-consuming groundwater quality monitoring studies and complicated, computer intensive three-dimensional numerical analyses. An artificial neural network (ANN) is an efficient statistical technique used to develop an understanding of a complex physical system when an ample amount of data about the behavior of the system is present (Haykin, 1994).

This research is a feasibility study on the utilization of an ANN for dealing with subsurface contamination problems. Visual MODFLOW will be used to create a database of spatial and temporal variations of the concentrations of typical contaminants in typical subsurface conditions. Then, the database will be used to train an ANN to predict the concentrations of a given chemical at a given location. If found to be a feasible and quick screening tool, the ANN could replace the time-consuming and complicated three-dimensional numerical analysis program during the initial monitoring phase. However, for a complex site, the ANN would probably require the combination of real data with simulations from three-dimensional groundwater contaminant transport programs.

1.2 Objectives

1. To investigate the feasibility of using an artificial neural network as a reliable, rapid, and easy to use screening tool for estimating contaminant migration at a hypothetical site for one selected contaminant.
2. To determine the network architecture and components that provide the best prediction.

1.3 Expected Results

An artificial neural network can be used as a quick screening tool to simulate the breakthrough curve of concentration versus time.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER II

LITERATURE REVIEW

2.1 Groundwater Contamination

Water is essential to life; without it, life on the earth would not be possible. Water can be found in different forms and places. The three states of water are gas (e.g. water vapor), liquid, and solid (e.g. ice). The water that is found in the atmosphere is called atmospheric water. Water that exists above the ground surface is called surface water. The last place where water can be found is below the ground surface, which known as subsurface water. Fig. 2-1 illustrates the hydrologic cycle. There are five processes involved in the movement of water through the cycle.

1. *Condensation*: The vapor water in the atmosphere condenses to form clouds.
2. *Precipitation*: The clouds release moisture in the form of rain, snow, etc. that falls onto the earth's surface.
3. *Infiltration*: The precipitated water seeps into the ground and is the major source for plants. When it seeps deeper, it becomes groundwater.
4. *Runoff*: This process occurs when the rate of precipitation on a surface exceeds the rate at which the water seeps into the ground. Thus a portion of the precipitated liquid water runs off over the land and becomes surface runoff. Runoff flows on the ground and into streams and rivers. Finally, it is stored in various sizes of reservoirs such as lakes or oceans.

5. *Evaporation and Transpiration:* In the evaporation process, liquid water changes into vapor by the motivation of the sun; whereas, transpiration occurs when water is transferred from plants into the atmosphere.

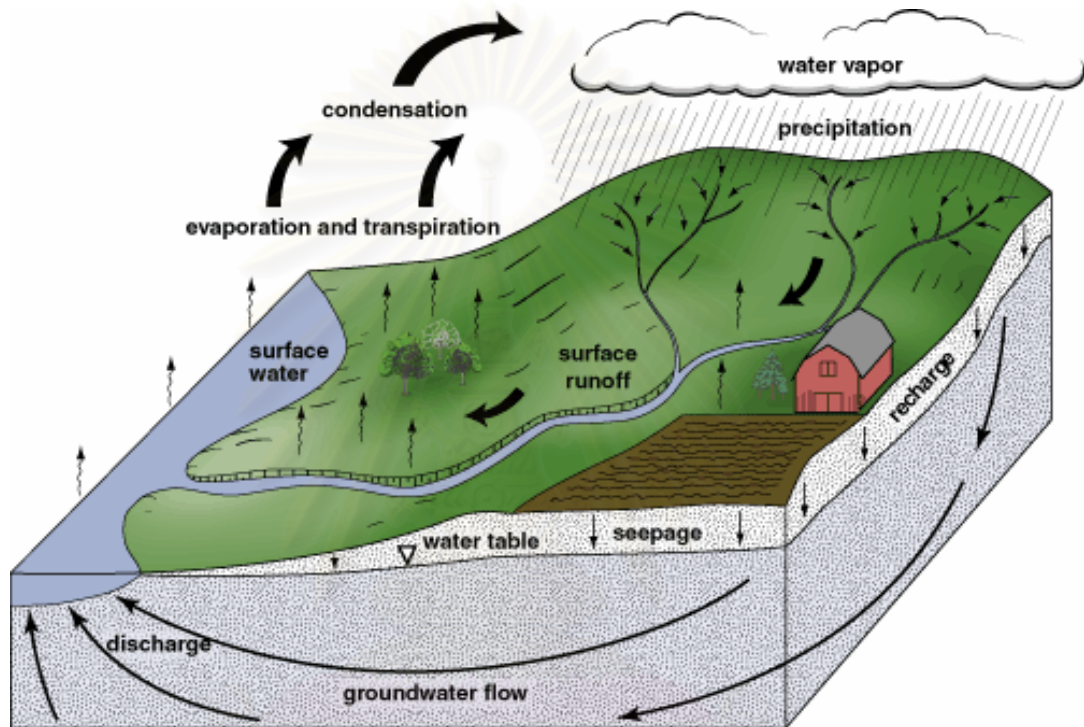


Fig. 2-1. Hydrologic cycle

(from Illinois State Geological Survey, 2004)

A huge portion of the water on the earth exists as surface water (Table 2-1). However, 97.2% of total water is salt water in oceans and cannot be used to support life. About 0.32% of total water is fresh liquid water that is readily able to be used by humans, animals, or plants. This percentage does not include the fresh water in the forms of glaciers and ice caps. Groundwater makes up about 94% of the readily used fresh liquid water (0.31% of total water). It is the largest freshwater supply source in the world, since the groundwater below a depth of 0.8 km is saline (Bouwer, 1978).

Groundwater accounts for a small portion of the world's total water, but it accounts for a major portion of the world's freshwater resources (Liu and Liptak, 2000). It has become a major source of water throughout the ages.

Table 2-1. Estimated distribution of the world's water

	Volume 1000 km ³	Percentage of Total Water
Atmospheric water	13	0.001
Surface water		
Salt water in oceans	1 320 000	97.2
Salt water in lakes and inland seas	104	0.008
Fresh water in lakes	125	0.009
Fresh water in stream channels (average)	1.25	0.0001
Fresh water in glaciers and icecaps	29 000	2.15
Water in the biomass	50	0.004
Subsurface water		
Vadose water	67	0.005
Groundwater within depth of 0.8 km	4200	0.31
Groundwater between 0.8 and 4 km depth	4200	0.31
Total (rounded)	1 360 000	100

(from Bouwer H., 1978)

2.1.1 Groundwater Hydrology

Groundwater is water found beneath the earth surface, but not all subsurface water is classified as "groundwater." The water which exists in the unsaturated zone between the ground surface and water table is called "vadose water." Vadose water is held to the soil particles by capillary forces and the forces of cohesion and adhesion. In this zone, the water pressure is less than the atmospheric pressure. The water found under the water table is "groundwater." It is the portion of subsurface water that is contained in geological formations, called "aquifers."

Aquifer

This geological formation is permeable and it transmits and yields water. Hence, it is saturated. However, the capability of an aquifer to yield water depends on the porosity of the aquifer material. Fig. 2-2 schematically shows the types of aquifers. Normally, there are two main types of aquifers: confined aquifers and unconfined aquifers. The former type is overlaid with impervious material. This material can be totally impermeable or permeable for water to transmit in a vertically direction, but not in a horizontal direction. A confined aquifer is completely filled with groundwater under a greater-than-atmospheric pressure. Therefore, it does not have a free water table, but a piezometric surface instead, which is characterized by the pressure condition (Liu and Liptak, 2000). The water flows into the confined aquifer through recharge zones and flows up from the ground through artesian wells. The latter type of aquifer, an unconfined aquifer, contains a layer that has no barrier on top of the groundwater and maintains a pressure equal to atmospheric pressure. Thus, it provides a free water surface, which is called a “groundwater table.” The water flows into this aquifer by seeping from the ground surface and discharges throughout springs, rivers, as well as water table wells.

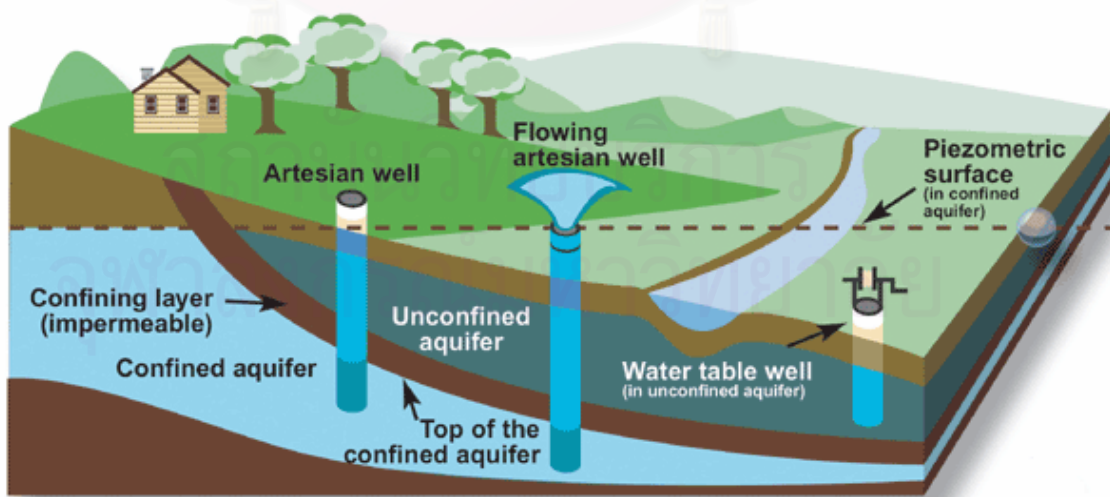


Fig. 2-2. Aquifer

(from Environment Canada, 2004)

As described above, groundwater is stored in aquifers. In an aquifer, groundwater can transmit and flow through from a higher level to a lower level of energy. It moves in response to gravity, pressure, and friction. The first two drive the water, and the latter restricts its motion (Pye, Patrick, and Quarles, 1983). The movement of groundwater can be described by Darcy's Law.

Darcy's Law

In 1856, Henri Darcy investigated the flow of water through beds of permeable sand and provided a model for groundwater movement. Darcy's Law generalizes the relationship of flow in porous media. The law was formulated based on the results of his experiments on the flow of water through beds of sand. Fig. 2-3 depicts the experiment while Eq. (2-1) and Eq. (2-2) state Darcy's equations.

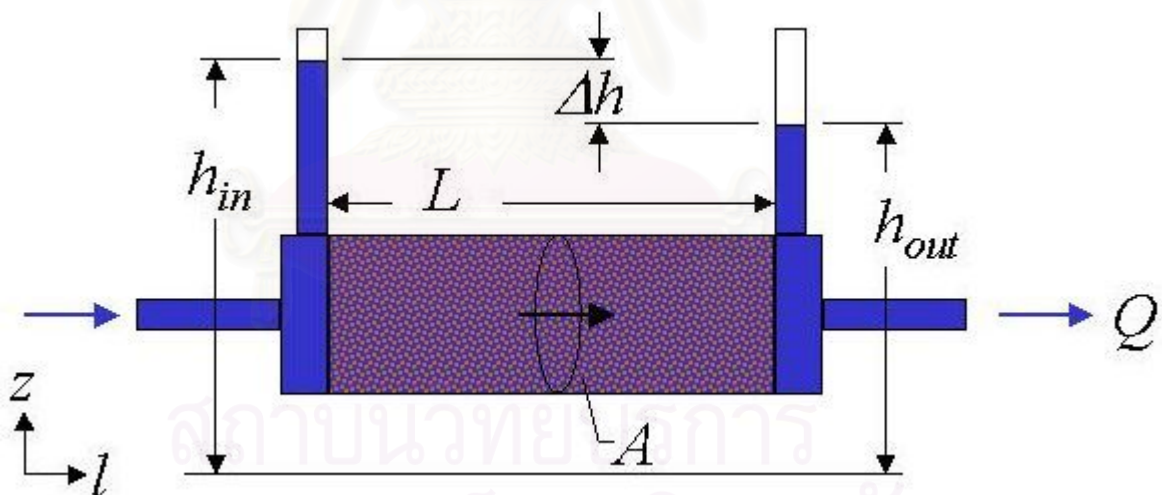


Fig. 2-3. Darcy's experiment
(from Brown, 1995)

$$Q = AK \frac{\Delta h}{L} \quad (2-1)$$

where Q = volumetric flow rate, [L^3T^{-1}]
 A = flow area perpendicular to L , [L^2]
 K = hydraulic conductivity, [LT^{-1}]
 L = flow path length, [L]
 h = hydraulic head, [L]
 Δ = denotes the change in h over the path L

$$Q = -AK \frac{dh}{dL} \quad (2-2)$$

when $L \rightarrow 0$

Moreover, the Q/A can be defined as Darcy's flux or Darcy's velocity (V). It is the volumetric flow per unit area, which can be stated as Eq. (2-3).

$$V = \frac{Q}{A} = -K \frac{dh}{dL} \quad (2-3)$$

Since groundwater always flows in the direction of the decreasing head or in other words from the higher energy to the lower energy, this is indicated by the negative sign. Likewise, this equation shows the relationship of the hydraulic gradient and the aquifer permeability that govern the velocity of groundwater. The hydraulic gradient is a derivative of the hydraulic head over the distance (dh/dL) where the measure of the permeability is the hydraulic conductivity (K). Note that this is an average discharge velocity through the entire cross section of the column, not the actual flow velocity. The actual flow is limited to the pore space only and is called seepage velocity (v_s) as shown in Eq. (2-4).

$$v_s = \frac{V}{n_e} \quad (2-4)$$

where $n_e =$ effective porosity (%)

Hydraulic Head

Hydraulic head is a specific measurement of water pressure or total energy per specific weight (the weight per unit volume of the material) at a specific point of interest. It is expressed in units of length, which can be described mathematically as shown in Eq. (2-5).

$$h = \frac{p}{\gamma} + z \quad (2-5)$$

$$\gamma = \rho g \quad (2-6)$$

where $p =$ water pressure, $[\text{ML}^{-1}\text{T}^{-2}]$
 $\gamma =$ water specific weight, $[\text{ML}^{-2}\text{T}^{-2}]$
 $z =$ elevation, $[\text{L}]$
 $g =$ acceleration of gravity, $[\text{LT}^{-2}]$
 $\rho =$ water density, $[\text{ML}^{-3}]$

The hydraulic head value can be used to determine a *hydraulic gradient* between two or more points over the length of the flow path.

$$\text{Hydraulic gradient } (i) = \frac{dh}{dL} = \frac{h_2 - h_1}{\text{Length}} \quad (2-7)$$

Hydraulic Conductivity

Hydraulic conductivity is a measure of the permeability of the porous media (Bedient, Rifai, and Newell, 1994). In other words, it is the property of soil or rock in

which water can transmit through. Therefore, the hydraulic conductivity value depends on the properties of the aquifer materials, which can vary over many orders of magnitude (Table 2-2). As a result of this, the Darcy's velocity can also vary over the same range.

Table 2-2. Saturated hydraulic conductivity (K) values found in nature

K (cm/s)	K (m/s)	Aquifer/ Permeability	Unconsolidated Sand & Gravel	Unconsolidated Clay & Organic
10^2	1	Good	Well Sorted Gravel	
10^1	10^{-1}			
1	10^{-2}		Well Sorted Sand or Sand & Gravel	
10^{-1}	10^{-3}			
10^{-2}	10^{-4}			
10^{-3}	10^{-5}	Poor	Very Fine Sand, Silt, Loess, Loam	Peat
10^{-4}	10^{-6}			Clay
10^{-5}	10^{-7}			
10^{-6}	10^{-8}			
10^{-7}	10^{-9}		None	Fat / Unweathered Clay
10^{-8}	10^{-10}			
10^{-9}	10^{-11}			
10^{-10}	10^{-12}			

(after Bear, 1972)

2.1.2 Contaminant Transport

Normally, groundwater is safer and more reliable for use than surface water. The reason is that surface water is more easily exposed to pollutants from many sources than groundwater is. Moreover, during the seepage of surface water through the ground to the aquifer, the ground material acts as a filter to purify the water. However, this does not mean that groundwater can not be contaminated. Once the contaminated substance is present in the groundwater, it is distributed far beyond the

site of contamination by the groundwater and may cause the aquifer to be unusable for decades. Additionally, a groundwater contaminant is difficult, and sometimes impossible to clean up. The remediation process is also an extremely costly operation. A groundwater contaminant can be defined as any physical, chemical, biological, or radiological substance or matter in groundwater. It can be introduced to the groundwater naturally or by human activities. However, the contamination from naturally occurring activities, such as the natural leaching of the soil and mixing with other groundwater sources with different chemistries, is usually small. Thus, human activities are the leading cause of groundwater contamination (Chester, 2000).

There are two categories of sources of groundwater contamination caused by human activity: point sources and non-point sources. Landfills, leaking underground storage tanks, leaking septic tanks, mining operations, and accidental spills are examples of point sources. Examples of non-point sources are agricultural operations and contaminated precipitation.

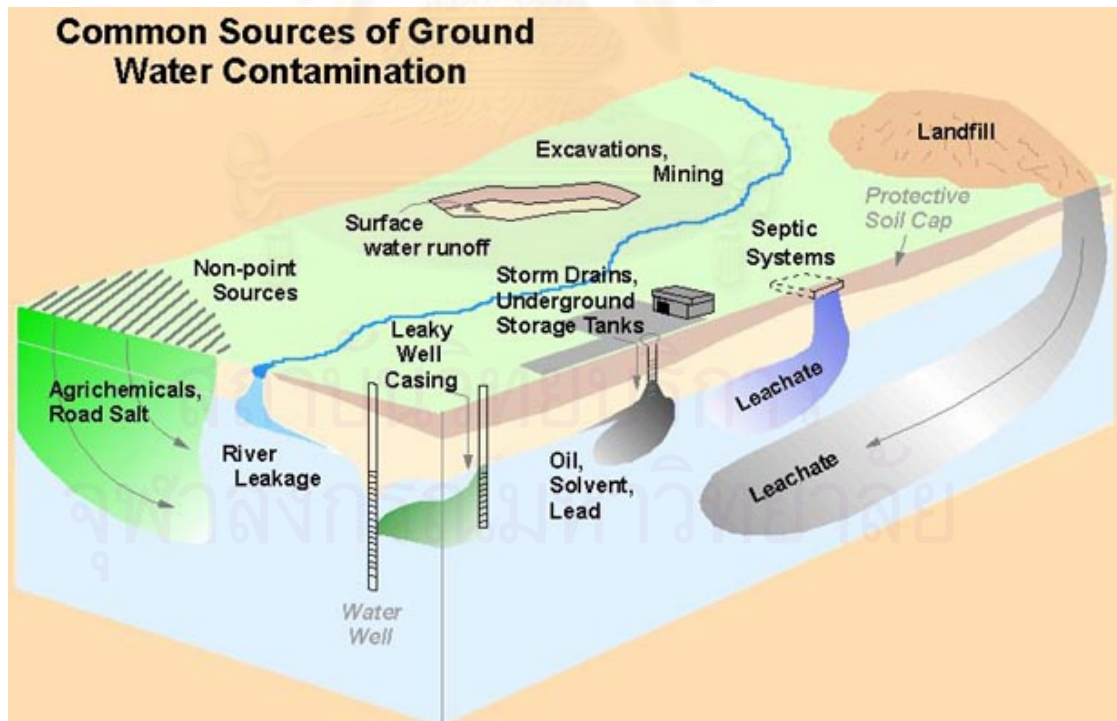


Fig. 2-4. Sources of groundwater contamination
(from Portneuf Valley Groundwater Guardian, 2001)

As stated above, when a contaminant substance is introduced into groundwater, it spreads and moves with the groundwater. There are three processes that influence the movement of a contaminant which are advection, caused by the flow of groundwater; hydrodynamic dispersion, caused by mechanical mixing and molecular dispersion; and retardation, caused by adsorption to soil particles.

Note that the transport of a contaminant in groundwater appears in three dimensions (x, y, and z). Therefore, while the equations provided in this chapter may be used to calculate the parameters in all three dimensions, only the x-dimension of the contaminant transport is shown here to simplify the explanation of the equations.

Advection

Advection is one of the processes of contaminant movement. In this process, the flowing groundwater causes the contaminant to move into porous media at the seepage velocity (Bedient et al., 1994). Therefore, when only advection is considered, a contaminant moves with the groundwater at the same rate as water due to the groundwater hydraulic gradient and no decrease in the concentration is observed. Then, the contaminant velocity equation can be written as shown in Eq. (2-8), which is the same as seepage velocity according to the Darcy's law. The mass flux equation can be written as seen in Eq. (2-9).

$$v_x = -\frac{K}{n_e} \frac{\partial h}{\partial x} \quad (2-8)$$

$$F_x = v_x n_e C \quad (2-9)$$

- where
- K = hydraulic conductivity, [LT⁻¹]
 - x = flow path length in x-axis direction, [L]
 - h = hydraulic head, [L]
 - n_e = effective porosity (%)
 - F_x = mass flux, [ML⁻²T⁻¹]
 - C = contaminant concentration, [ML⁻³]
 - v_x = seepage velocity in the direction of x, [LT⁻¹].

Hydrodynamic Dispersion

Hydrodynamic dispersion is the result of two processes: molecular diffusion and mechanical mixing. Freeze and Cherry (1979) defined hydrodynamic dispersion as the process in which solutes spread out and are diluted in contrast to simple advection alone.

Molecular diffusion is a molecular mass-transport process in which solutes move from areas of higher concentrations to areas of lower concentrations. The molecular diffusion process provides the term D_d , which is involved in calculating mass flux.

$$D_d = \text{molecular diffusion coefficient, [L}^2\text{T}^{-1}] \quad (2-10)$$

Mechanical mixing dispersion is a mixing process caused by velocity variations in the porous media. It can occur both in the longitudinal direction of the flow as well as in the transverse direction. According to Bachmat and Bear (1964), the mechanical mixing component of dispersion can be assumed to be proportional to the seepage velocity as follows:

$$\begin{aligned} D_L &= \alpha_L v \\ D_T &= \alpha_T v \end{aligned} \quad (2-11)$$

where D_L, D_T = mechanical mixing component of dispersion in
longitudinal and transversal direction, [L²T⁻¹]
 α_L, α_T = longitudinal and transversal dispersivity, [L]
 v = seepage velocity, [LT⁻¹]

Finally, the hydrodynamic dispersion coefficient can be written as follows (Bear, 1972, 1979):

$$D_x = D_L + \frac{D_d}{n_e}$$

$$D_y = D_T + \frac{D_d}{n_e}$$
(2-12)

where D_x = the hydrodynamic dispersion in the x direction, $[L^2T^{-1}]$
 D_y = the hydrodynamic dispersion in the y direction, $[L^2T^{-1}]$
 n_e = effective porosity (%)

Mass flux caused by dispersion can be expressed as the following equation:

$$F_x = -n_e D_x \frac{\partial C}{\partial x}$$
(2-13)

The sum of the mass flux influenced by both advection and hydrodynamic dispersion can be written as the following equation:

$$F_x = v_x n_e C - n_e D_x \frac{\partial C}{\partial x}$$
(2-14)

Retardation

In this process, the migration of contaminants in groundwater is obstructed by the adsorption mechanism. The retardation coefficient can be calculated based on the distribution or adsorption coefficients of the contaminant and the characteristics of the porous medium.

$$R = 1 + K_d \left(\frac{\rho_d}{n} \right) \quad (2-15)$$

where K_d = distribution or adsorption coefficient, [L^3M^{-1}]
 ρ_d = bulk density of the aquifer, [ML^{-3}]
 n = total porosity of soil [%]

The velocity of the contaminant in groundwater can be calculated as follows:

$$V_c = \frac{v}{R} \quad (2-16)$$

where V_c = mean chemical velocity, [L/T]
 v = seepage velocity, [L/T]
 R = retardation factor, note that if $R = 1$, $V_c = v$

Governing Equation and Boundary Conditions

The governing equation for one-dimensional chemical transport in groundwater with advection, dispersion, and retardation is (Van Genuchten and Alves, 1982) as follows:

$$R \frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - v \frac{\partial C}{\partial x} \quad (2-17)$$

where R = retardation factor
 D = dispersion coefficient
 v = seepage velocity [L/T]

Boundary and initial conditions:

$$C(x, t = 0) = 0$$

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

$$C(x = 0, t > T) = 0$$

$$C(x = 0, 0 < t \leq T) = C_0$$

From the governing equation and boundary conditions shown above, a one-dimensional steady-state flow equation can be derived with the condition described below as shown in Eq. (2-18) for a continuous source and Eq. (2-19) for an instantaneous source (Bedient and Wayne, 1992; Van Genuchten and Alves, 1982). However, in a real situation, a contaminant flows in three directions: x, y, and z. A three-dimensional steady-state flow in the x-direction injected instantaneously can be written as shown in Eq. (2-20) (Baetsle 1969; Bear, 1972).

$$C(x,t) = \frac{C_0}{2} \left[\operatorname{erfc} \left(\frac{x - V_c t}{2\sqrt{Dt}} \right) + \exp \left(\frac{V_c x}{D} \right) \operatorname{erfc} \left(\frac{x + V_c t}{2\sqrt{Dt}} \right) \right] \quad (2-18)$$

where
$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - \left(\frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du \right)$$

$$C(x,t) = \frac{M}{(4\pi D_x t)^{1/2}} \exp \left[-\frac{(x - V_c t)^2}{4D_x t} \right] \quad (2-19)$$

$$C(x, y, z, t) = \frac{M}{8(\pi t)^{3/2} (D_x D_y D_z)^{1/2}} \exp \left[-\frac{(x - V_c t)^2}{4D_x t} - \frac{y^2}{4D_y t} - \frac{z^2}{4D_z t} \right] \quad (2-20)$$

where

- M = chemical mass, [M] = $C_0 V_0$
- C_0 = injected concentration at $x=0$, [M/L³]
- V_0 = the original volume, [L³]

Note that the concentration of contamination decreases as the plume moves away from the source and disperses in other directions.

Fig. 2-5 shows an example of a typical breakthrough curve plotted between C/C_0 and time (days) at a point of interest, 60 meters from the source. C_0 is the initial concentration at the source while C is the concentration at a time measured at 60 meters from the source. As shown in the graph, the concentration is zero at the beginning of time = 0 days. After that, the relative concentration increases due to the dispersion in the flow direction and retardation of the soil particle. Finally, the relative

concentration reaches 1 and will continue with this number in the case of continued injection.

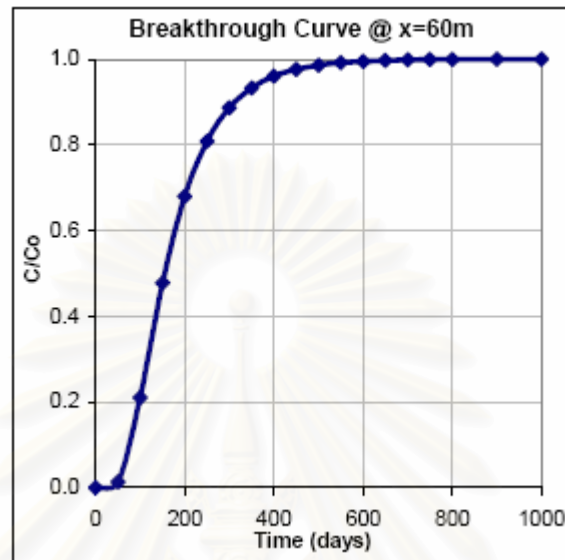


Fig. 2-5. Breakthrough curve of C/C_0 and time (days) at a point of interest (from *Civil, Construction & Environmental Engineering, Oregon State University, 2006*)

2.2 Artificial Neural Network

2.2.1 Background

In recent years, the interest in artificial neural networks (ANNs) has grown rapidly. Various applications of ANNs have become popular. They are being used in corporate financial analysis, currency price prediction, speech and facial recognition programs, stock trading advisory systems, and more.

In 1943 (McCulloch and Pitts, 1943), the ANN was proposed by McCulloch and Pitts as a new way of modeling brain functions. According to Gurney (1997), a neural network is an interconnected assembly of simple processing elements, units, or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

The inspiration for the development of ANNs came from the desire to produce artificial systems capable of intelligent computations similar to those that the human brain routinely performs. One of the most powerful features of neural networks is their ability to learn and generalize from a set of training data. They adapt the strengths/weights of the connections between neurons so that the final output activations are correct. The ANN learns to develop a solution system by studying the examples of a large number of inputs associated with a resulting set of outputs.

The key to successfully training an ANN is choosing the right network architecture and training algorithm. In this study, a feedforward backpropagation neural network will be used to predict the concentration of the contaminant as a function of time for the given locations and model parameters.

2.2.2 Types of Artificial Neural Networks (ANNs)

Here, the types of neural networks are summarized. There are many kinds of ANNs by now; however, nobody knows exactly how many as a result of the new inventory or modification of old ones.

The main kinds of learning algorithms:

- *Supervised learning*

In supervised learning, the target values or the desired outputs are known and are given to the ANN during training so that the ANN can adjust its weights to try to match its outputs to the target values. After the training phase, the ANN is tested by giving it only input values and seeing how close the outputs given from the ANN are to the target values.

- *Unsupervised learning*

In unsupervised learning, the correct results are not introduced to the ANN during the training phase. Unsupervised ANNs usually perform some kind of data compression, such as dimensionality reduction or clustering.

The major kinds of network topology:

- *Feedforward*

In a feedforward ANN, the connections between units do not form cycles. This kind of network topology usually generates a response to an input quickly. Moreover, it is able to be trained using a wide variety of efficient conventional numerical methods.

- *Feedback*

In a feedback or recurrent ANN, there are cycles in the connections. Each time an input is presented to some feedback ANN, the ANN must iterate for a potentially long time before it produces a response. Feedback ANNs are usually more difficult to train than feedforward ANNs.

The major kinds of accepted data are categorical and quantitative:

- *Categorical variables* take only a finite number of possible values, and there are usually several or more cases falling into each category. Categorical variables may have symbolic values that must be encoded into numbers before being given to the network. Both supervised learning with categorical target values and unsupervised learning with categorical outputs are called “classification.”
- *Quantitative variables* are numerical measurements of some attribute, such as length in meters. The measurements must be made in such a way that at least some of the arithmetic relations among the measurements reflect analogous relations among the attributes of the objects that are measured. Supervised learning with quantitative target values is called “regressing.”

The neural network used in this study can be categorized as a feedforward neural network trained by a supervised learning algorithm. Quantitative data are also used in this study.

2.2.3 Feedforward Neural Network Architecture

A Single-Layer Neural Network

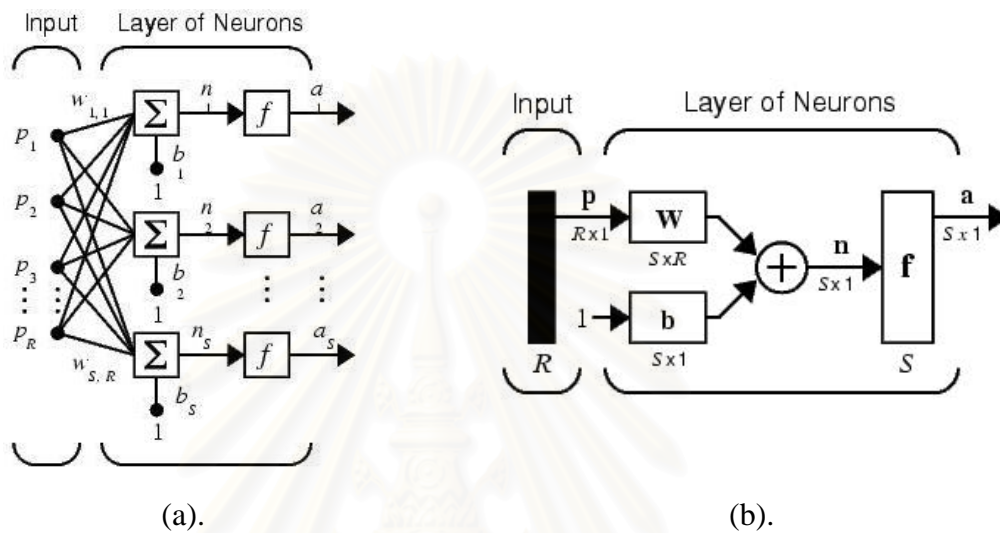


Fig. 2-6. (a). A single-layer network (b). A single-layer network using abbreviated notation.

(from MATLAB, 2005)

Fig. 2-6 depicts a single-layer network with R input elements in input vector and S neurons in the layer. Each element of the input vector \mathbf{p} [Eq. (2-21)] is transmitted through a connection that multiplies its strength by the scalar weight w . Since the number of the input elements is greater than 1, the weights become a matrix \mathbf{W} [Eq. (2-23)], which consists of the weights of each connection. Then, the i^{th} neuron gathers its weighted inputs and bias to form its own scalar output n_i [Eq. (2-24)], which becomes an input to the transfer function. Finally, the transfer function f produces the scalar output a_i [Eq.(2-26)]. Each node is connected to all the nodes in the adjacent layers and the signal fed at the input layer flows forward layer-by-layer to the output layer (Gurney, 1997). Note that the number of neurons in the layer (S) does not have to be equal to the number of input elements (R).

Input vector with R element:

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad (2-21)$$

Input matrix of N vector with R element:

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N} \\ \vdots & \vdots & \cdots & \vdots \\ p_{R,1} & p_{R,2} & \cdots & p_{R,N} \end{bmatrix} \quad (2-22)$$

Weight matrix:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \cdots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \quad (2-23)$$

Output of neuron:

$$n_i = w_{i,1}p_1 + w_{i,2}p_2 + \cdots + w_{i,R}p_R + b_i \quad (2-24)$$

$$\mathbf{n} = \mathbf{Wp} + \mathbf{b} \quad (2-25)$$

Output of transfer function:

$$a_i = f(n_i) \quad (2-26)$$

$$\mathbf{a} = \mathbf{f}(\mathbf{n}) \quad (2-27)$$

where R = input elements

S = number of neurons

N = number of patterns

$$1 \leq i \leq S$$

To explain the notation in Fig. 2-6 (b), \mathbf{p} is an R length input vector or element. Note that inputs can be more than one vector, which is denoted as \mathbf{P} . \mathbf{P} is an input matrix consisting of N input vectors or patterns [Eq. (2-22)]. \mathbf{W} is an $S \times R$ matrix as shown in Eq. (2-23). The row indices on the elements of matrix \mathbf{W} indicate

the destination neuron of the weight while the column indices indicate the input element for that weight. Thus, the indices in $w_{2,1}$ denote that the strength of the signal from the first input element to the second neuron. The values \mathbf{a} , \mathbf{b} and \mathbf{n} are respective output vectors from the transfer function, bias vector, and output vector from the neuron with S length, see Eq. (2-25) and (2-27).

In conclusion, when an input vector \mathbf{p} [Eq. (2-21)] from an input matrix \mathbf{P} [Eq. (2-22)] is introduced to a layer of neurons, the layer of neurons provide the output vector \mathbf{a} [Eq. (2-27)]. A layer of neurons includes the weights matrix \mathbf{W} [Eq. (2-23)], the multiplication operation, the bias vector \mathbf{b} , the summer \mathbf{n} [Eq. (2-25)], and the transfer functions \mathbf{f} .

A Multiple-Layer Neural Network

A network can have several layers. The layer that produces the network output is called the output layer. The output layer generates the output vector and neurons are assigned according to the numbers of output or target elements. All other layers are called “hidden layers.” The hidden layers always exist before the output layer. The hidden neurons are assigned depending on the study. Each layer has a weight matrix \mathbf{W} , a bias vector \mathbf{b} , and an output vector \mathbf{a} . Note that the outputs of each intermediate layer are the inputs to the following layer. For more understanding, the following diagrams (Fig. 2-7) illustrate the feedforward neural network with multiple layers of neurons.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

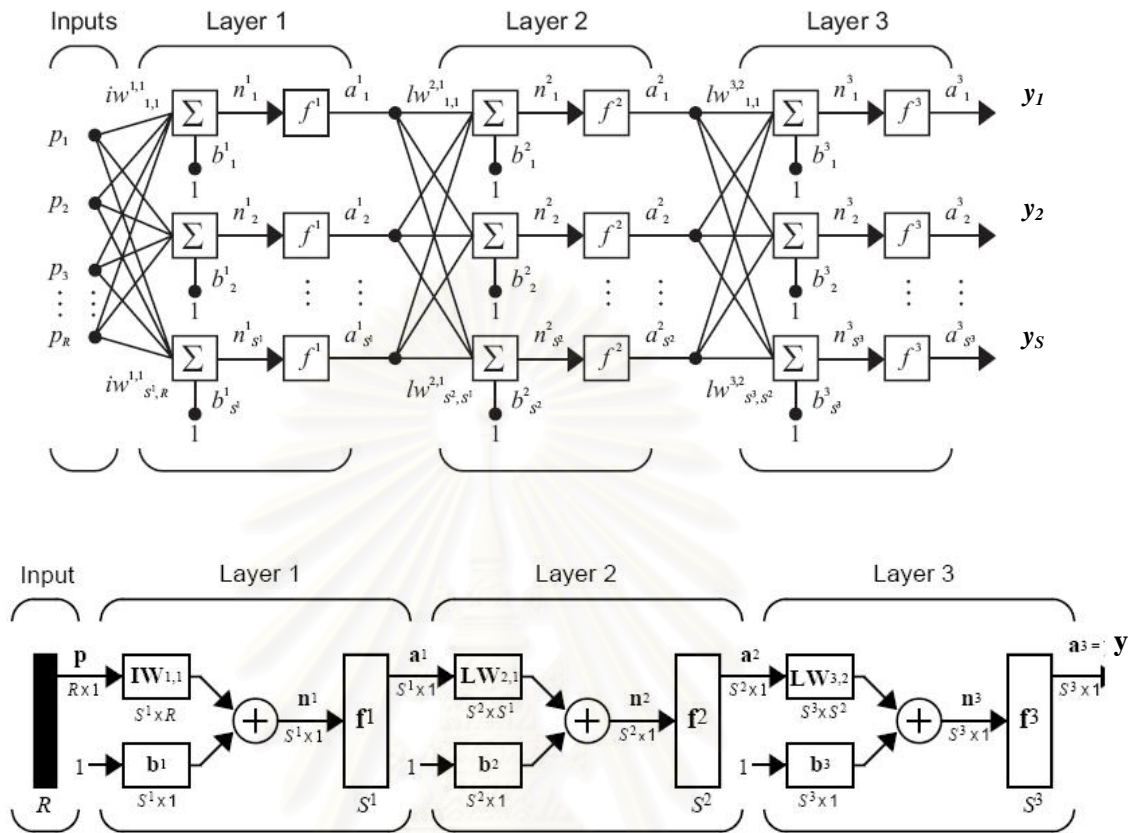


Fig. 2-7. Multiple-Layer Neural Network
(from MATLAB, 2005)

The network shown above consists of an output layer and 2 hidden layers. The first hidden layer receives the input vector of R elements and each element is connected to each neuron in the first hidden layer through the weight matrix $\mathbf{IW}^{1,1}$. S indicates the number of neurons. Each neuron in the first hidden layer (n^1_i) gathers the summation and sends it to a transfer function (f^1). Then the output vector (\mathbf{a}^1) is provided from the neuron layer, which becomes the input vector to the following layer. And the output of the network (in the figure, it is assumed to be the third layer) is labeled as \mathbf{y} . The superscript to the variable indicates its association with that layer number. The weight matrix connected to inputs is called input weight ($\mathbf{IW}^{1,1}$) and the weight matrices coming from hidden layer outputs are called layer weights ($\mathbf{LW}^{2,1}$ or $\mathbf{LW}^{3,2}$). As the weight matrix is the relation between two layers, the superscript to the weight matrix variable indicates the destination and the source, respectively. For

example, $\mathbf{LW}^{2,1}$ denotes the weights matrix that connect the output from the first hidden layer as a source of input to the second hidden layer. The output from each layer can be explained in following equations.

$$\mathbf{a}^1 = \mathbf{f}^1 (\mathbf{IW}^{1,1} \mathbf{p}^1 + \mathbf{b}^1) \quad (2-28)$$

$$\mathbf{a}^2 = \mathbf{f}^2 (\mathbf{LW}^{2,1} \mathbf{a}^1 + \mathbf{b}^2) \quad (2-29)$$

$$\mathbf{a}^3 = \mathbf{f}^3 (\mathbf{LW}^{3,2} \mathbf{a}^2 + \mathbf{b}^3) \quad (2-30)$$

Multiple-layer networks are quite powerful. For instance, a network of 2 hidden layers, where the first hidden layer is sigmoid and the second hidden layer is linear, can be trained to approximate any function arbitrarily well. This kind of 2-hidden-layer network is used extensively in backpropagation (MATLAB, 2005).

Transfer Function

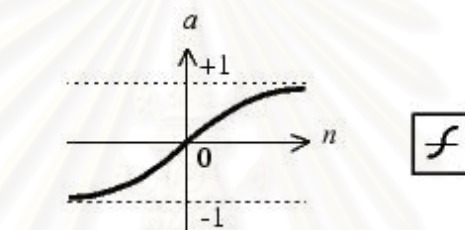
Many transfer functions can be associated with neurons. Nonlinear [Eq. (2-31), (2-32)] and linear [Eq. (2-33)] functions are commonly selected to be used as transfer functions. Transfer functions for the hidden neurons are needed to introduce nonlinearity into the network to make multiple-layer networks powerful. The transfer function must be differentiable, and it helps if the function is bounded. The sigmoid functions such as hyperbolic tangent and logistic function are the most common choices. Functions such as the hyperbolic tangent that produce both positive and negative values tend to yield faster training than functions that produce only positive values, such as logistic, because of better numerical conditioning. For the output neurons, the transfer function should be suited to the distribution of the target values. When the last layer of a multiple-layer network has sigmoid neurons, a linear transfer function is preferable at the output neuron. If the output layer of a multiple-layer network has sigmoid neurons, then the outputs of the network are limited to a small range. If linear output neurons are used, the network outputs can take on any values. Multiple layers of neurons with nonlinear transfer functions allow the network to

learn nonlinear and linear relationships between input and output vectors. The algorithms and graphs are shown below.

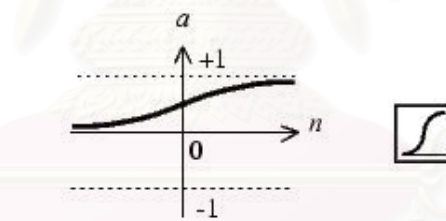
Hyperbolic tangent:
$$a = f(n) = \frac{2}{(1 + e^{(-2)n}) - 1} \quad (2-31)$$

Logistic:
$$a = f(n) = \frac{1}{1 + e^{-n}} \quad (2-32)$$

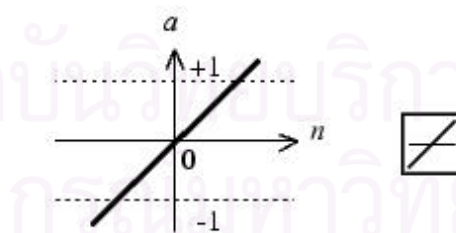
Linear:
$$a = f(n) = n \quad (2-33)$$



(a) $a = \text{tansig}(n)$



(b) $a = \text{logsig}(n)$



(c) $a = \text{purelin}(n)$

Fig. 2-8. Three examples of the transfer function: (a) hyperbolic tangent, (b) logistic, and (c) linear
(from MATLAB, 2005)

2.2.4 Feedforward Backpropagation Neural Network (FBNN)

A FBNN is a supervised learning technique used for training artificial neural networks. Since this technique was introduced, FBNN has undergone many modifications to overcome its limitations. In general, FBNN is a feedforward network trained by backpropagation (Werbos 1974, 1994). The FBNN, a gradient descent algorithm (Rumelhart, McClelland, and the PDP research group, 1986), consists of two passes: the forward pass and the backward pass. In the first pass, the input vectors and the corresponding target vectors are used to train the network. The later pass involves with the process of modifying weights to obtain a minimum error between the output from the network and the target related to the input. The error that was found in the former iteration or epoch is used to train each of the hidden neurons according to the gradient descent. After that the weights are updated. These processes involve performing computations backwards through the network (Fig. 2-9), hence, the name backpropagation (Gurney, 1997).

Technically, the input vectors and the corresponding target vectors are used to train a network until it can approximate a function.

Standard backpropagation is a gradient descent algorithm. The gradient is computed for nonlinear multiple-layer networks, in which the network weights are moved along the negative of the gradient of the performance function. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient, Quasi-Newton, and Levenberg-Marquardt. The optimization technique used in this study is going to be explained in Section 2.2.5.

Properly trained feedforward backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. During training, the weights and biases of the network are iteratively adjusted to minimize the network output errors. The algorithms use the gradient of the performance function to determine how to adjust the weights to maximize performance.

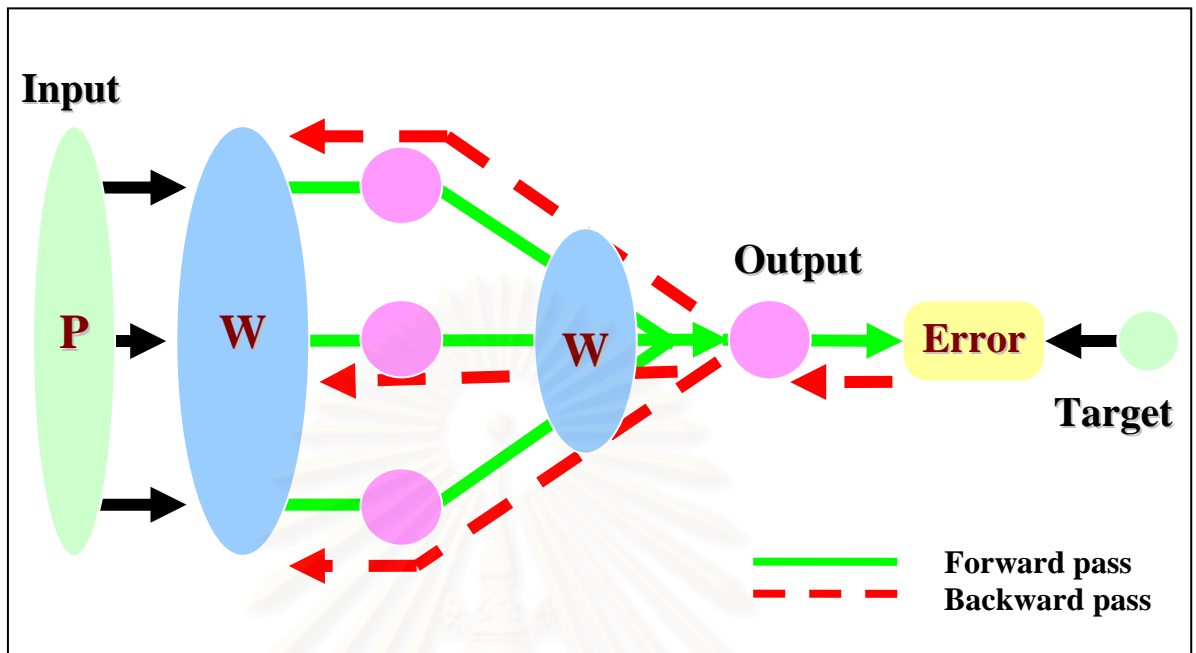


Fig. 2-9. The structure of a feedforward backpropagation neural network

Gradient Descent on an Error Algorithm

This approach is based on defining a measure of the difference between the actual network output and the target vector. This difference is then treated as an error (E) to be minimized by adjusting the weights (w), see Eq. (2-34), (2-35), and (2-36). Thus, the objective is to find the minimum sum of errors over the training set where this sum of errors is considered to be a function of the weights of the network (Fig. 2-10).

$$e^p = (t^p - y^p)^2 \quad (2-34)$$

$$E = \frac{1}{N} \sum_{p=1}^N e^p \quad (2-35)$$

$$\Delta w_i = -\alpha \frac{\partial E}{\partial w_i} \quad (2-36)$$

where e^p = error of training pattern p
 t^p = target of training patter p

- y^p = output of pattern p
 E = total mean square error
 N = the number of pattern
 w_i = weight
 α = learning rate

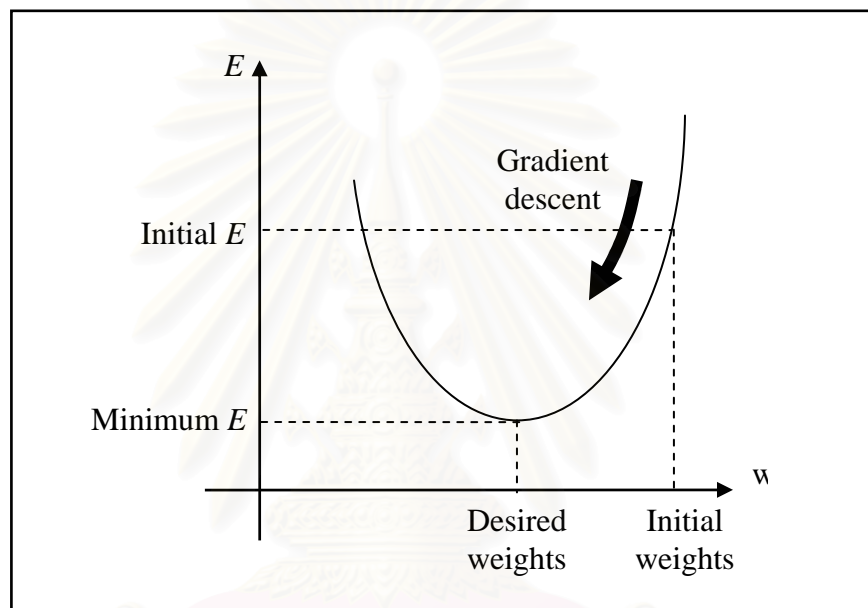


Fig. 2-10. Gradient descent for a network
(from Gurney, 1997)

2.2.5 Bayesian Regularization: Improving Generalization

One of the problems that occur during network training is called “overfitting.” When this type of problem occurs, the error on the training set is driven to a very small value, but when new data is presented to the network, the error is large. The network has memorized the training and overfit the data examples but it has not learned to generalize well enough to new situations. One method for improving the network’s generalization is to use a network that is just large enough to provide an adequate fit. The size of the network is depended on the number of hidden layers and their neuron (MATLAB, 2005). The larger the network, the more complex the

function the network creates. On the other hand, the smaller the network, the lower the ability of the network to overfit the data. Thus, it is difficult to know how large a network should be for a specific application.

Regularization is one technique for improving generalization and avoiding overfitting. This method modifies the performance function as shown in Eq. (2-37), (2-38), and (2-39). The mean of squares of the network errors on the training set is selected to be an example for explanation. The performance function can be written as shown in Eq. (2-37). To improve generalization, the performance function is modified by adding a term that consists of the mean of the sum of squares of the network weights and biases Eq. (2-38) and the performance ratio.

$$F = mse = \frac{1}{N} \sum_{p=1}^N (e^p)^2 = \frac{1}{N} \sum_{p=1}^N (t^p - y^p)^2 \quad (2-37)$$

$$msereg = \gamma mse + (1 - \gamma) msw \quad (2-38)$$

$$msw = \frac{1}{n} \sum_{j=1}^n w_j^2 \quad (2-39)$$

where N = number of pattern in input matrix
 n = number of neuron in layer

γ is the performance ratio which has to be optimized. However, the point is, it is difficult to determine the optimum value for the performance ratio parameter. The Bayesian framework of MacKay (1992) has solved this problem. In this framework, it is assumed that the weights and biases of the network are random variables with specified distributions, which the performance parameters are related to.

Note that this Bayesian algorithm works best when the network inputs and targets are scaled to the range of -1 to +1.

2.3 Related Research

Ranjithan, Eheart, and Garrett (1993) used an ANN to simulate a pumping index for a hydraulic conductivity realization to remediate groundwater at a contaminated site. A two-layer feedforward backpropagation network with a sigmoid transfer function ANN was trained and tested. The appropriate number of hidden neurons, the sensitivity of the ANN to the number of hidden neurons, and training patterns were studied. The result showed that by increasing the number of hidden neurons the performance of the ANN in training increased, while the performance of ANN in testing was stable at an optimal number of hidden neurons.

Roger and Dowla (1994) simulated a regulatory index for a multiple pumping realization containing multiple plumes at a contaminated site by using an ANN. The same technique used by the former researchers was used in this study. This research suggested that the number of training patterns should be greater than or equal to the number of weights to improve ANN performance.

Singh, Datta, and Jain (2004) proposed the methodology of using an ANN to identify unknown groundwater pollution sources. A feedforward backpropagation ANN was developed and trained. The data set inputted into the network was the set of source flux data and the resulting concentration measurement data at specified observation locations in the aquifer. Two different scenarios were considered. The first had two potential sources and four observation wells while the other was more complex with multiple potential sources and a larger number of observation wells. The results showed that ANN can be used to solve the complex problem of unknown groundwater pollution source identification.

Morshed and Kaluarachchi (1998) focused their research on improving the current understanding of ANN applications in groundwater flow and contaminant transport (GFCT). They studied the ANN concepts related to architecture, sampling, training, and multiple-function approximations. The backpropagation algorithm and genetic algorithm were compared. The results showed that the genetic algorithm performed less robustly than the backpropagation algorithm. The flow and transport of a trichloroethylene (TCE) dissolved in water through a partially saturated porous medium was studied. Data were generated for different scenarios (e.g., the

breakthrough curve with linear adsorption and with nonlinear adsorption) using the computer code, HYDRUS.

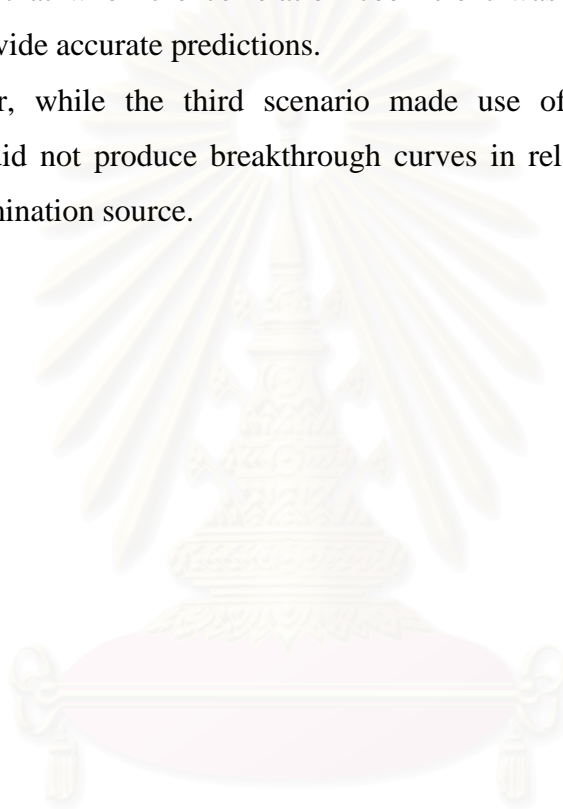
The first scenario was a simulation of the breakthrough curve with linear adsorption. In this scenario, the only input was time and single-layer networks with 3, 15, and 30 were used to simulate the concentrations corresponding to the time. Moreover, the sigmoid and the hyperbolic tangent function were used as the transfer functions. The results showed that training with a hyperbolic tangent transfer function provided a better performance than the sigmoid function and the 3-neuron network provided a better performance compared to the 15-neuron and the 30-neuron networks.

The second scenario was a simulation of a breakthrough curve with nonlinear adsorption. Here, the applicability of ANN for predicting a concentration was studied when the mass transport was complicated due to non-linear solid phase adsorption. Thus, time and the linear Freundlich isotherm (n) were the input elements. The concentration was simulated with $n = \{0.5, 1, \text{ and } 2\}$. A single-layer network with 5 neurons was used while the hyperbolic tangent was used as the transfer function. The results showed that ANN prediction was poor. However, ANN performance increased when the number of patterns in the training data as well as the number of neurons in a hidden layer increased.

The third scenario was the use of an ANN to simulate the breakthrough curve parameters of breakthrough time, t_b ; time to reach maximum concentration level, t_{MCL} ; time to maximum concentration, t_{max} ; and maximum concentration, C_{max} . Thus, there were four output elements. The ANN was trained by using the flow and the transport parameters as inputs. In general, there are seven input elements which are the hydraulic conductivity, grain size distribution index, water flux, dispersivity, dissolved phase, Freundlich coefficient, and Freundlich exponent. The former three are the flow parameters while the later five are the transport parameters. The application of the ANN was divided into the effect of the flow parameters, the effect of the transport parameters, and the effect of both the flow and transport parameters. In order to study these three types of effects, the flow and transport parameters were varied in the domains of [15, 60], [0.025, 0.100], [5, 20], [0.8, 3.2], [0.0005, 0.0020], [0.15, 0.60], and [0.5, 1.0], respectively. However, when the effect of flow parameters

was of concern, the transport parameters were constant. Similarly, when the effect of transport parameters was of concern, the flow parameters were constant. A hidden layer with 7 neurons was used to study the effect of the flow parameters, while 11 neurons were used to study the effect of the transport parameters. For the effect of both the flow and transport parameters, a hidden layer with 15 neurons was used. The results showed that when the correlation coefficient was between 0.98 and 1, the ANN could provide accurate predictions.

However, while the third scenario made use of the flow and transport parameters, it did not produce breakthrough curves in relation to various distances from the contamination source.



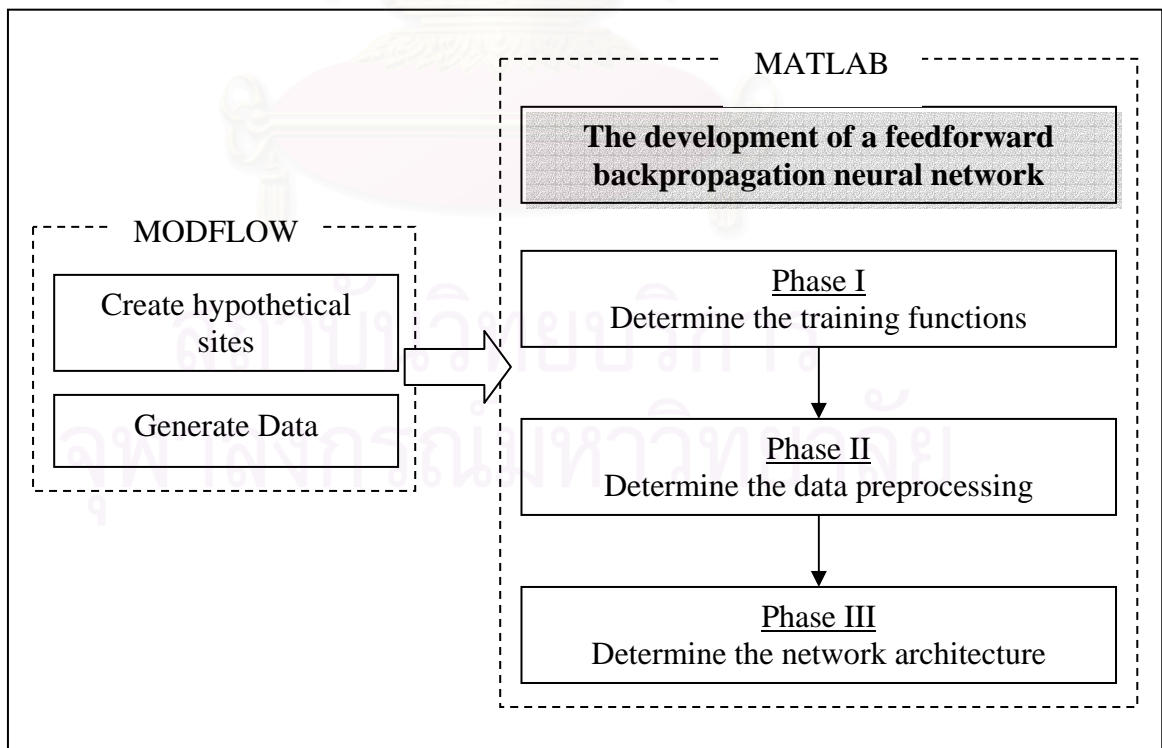
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER III

METHODOLOGY

Before going step by step through the methodology, an overall picture of this approach might be useful. At the beginning, MODFLOW was used to create the data set. Then, the data set was used to train and test the ANN's ability to predict the contaminant's transportation in groundwater. This study is not trying to replace MODFLOW altogether, but trying to provide a more rapid option when less time or some unknown model parameters are present. In addition, using MODFLOW simulations together with ANN prediction trained by data collected from real sites might provide great results.

3.1 The Methodology Framework



3.2 The Creation of the Hypothetical Site Using MODFLOW

The data was generated using the three-dimensional finite-difference groundwater flow model, Visual MODFLOW version 2.8.2. This model is normally used to simulate systems for water supply, containment remediation, and mine dewatering. It is most appropriate in those situations where a relatively precise understanding of the flow system is needed to make a decision. To use MODFLOW, files must be prepared that contain hydraulic parameters (hydraulic conductivity, transmissivity, specific yield, etc.) and boundary conditions (locations of impermeable boundaries and constant heads).

The simulations were performed to generate the breakthrough curve (i.e. the concentration of contaminant as a function of time). Many steps and model parameters were involved in order to simulate the one hypothetical site. These steps are described in the following sections.

Step 1: Create and define a flow model.

In this first step, the model dimensions and topology were defined as shown in Table 3-1, Table 3-1.Fig. 3-1, and Table 3-1.Fig. 3-2.

Table 3-1. Model Dimensions

Model Dimensions	Axis	Meters
Length (top to bottom)	Y	2000
Width (left to right)	X	1000
Depth	Z	10
Slope	Z:Y	10:2000

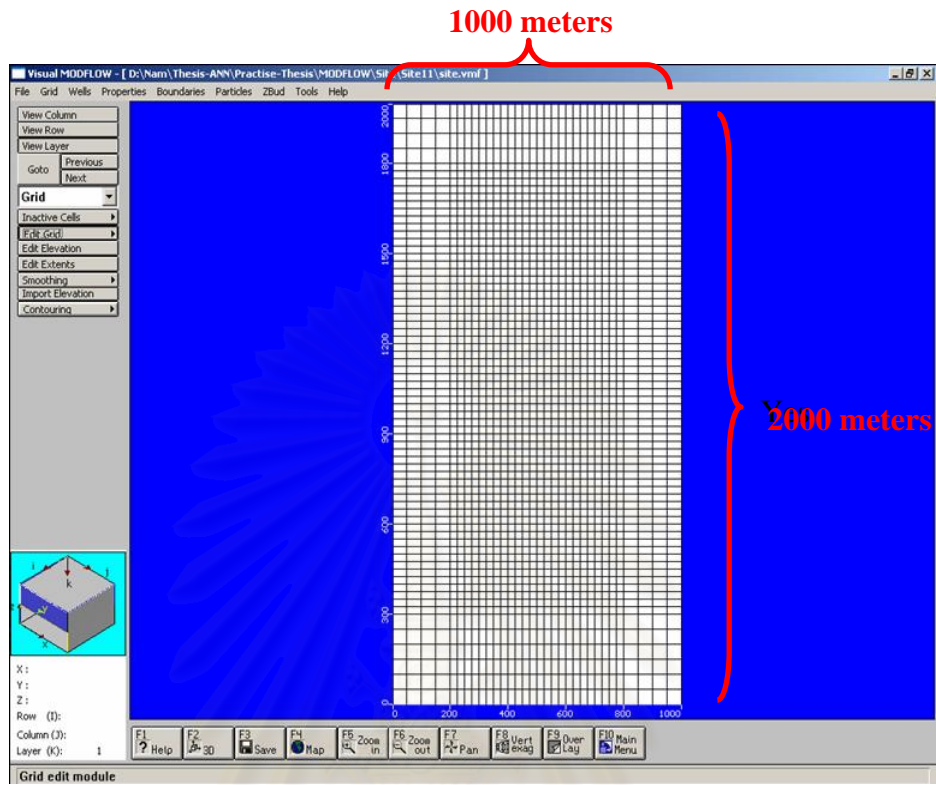


Fig. 3-1. Top view of the hypothetical site

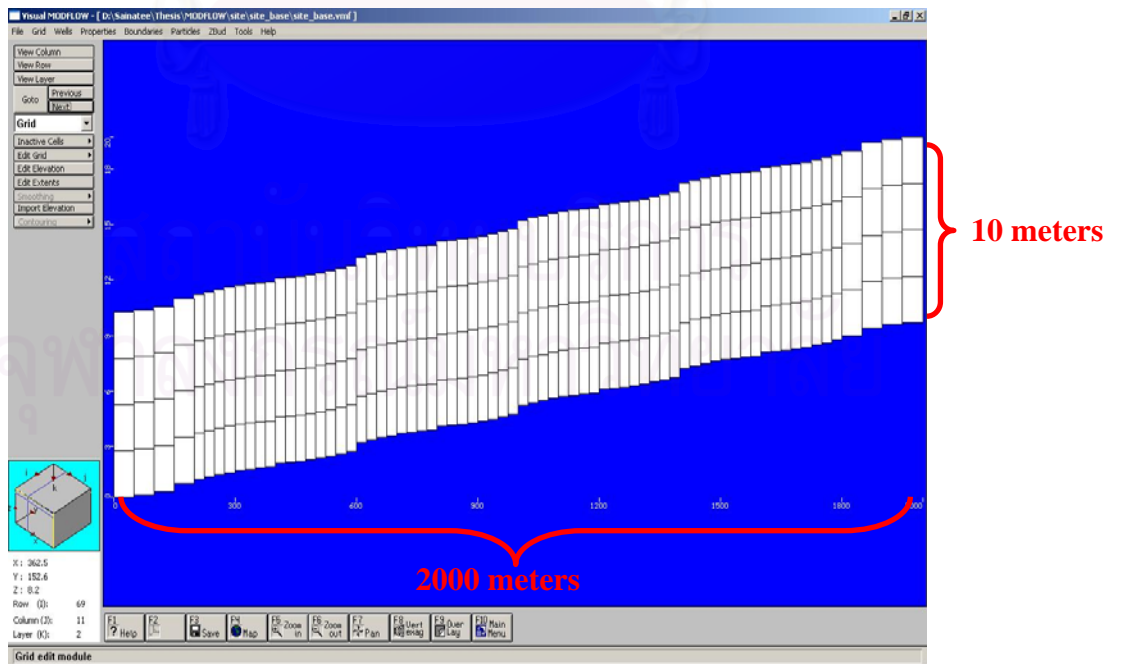


Fig. 3-2. Side view of the hypothetical site

Step 2: Assign model properties: hydraulic conductivity and dispersivity.

Next, the model properties were assigned. In order to create a set of data for training the ANN, the hydraulic conductivity was varied into 5 values and the longitudinal dispersivity was varied into 3 values. The reason for choosing these values is described in the next sections. The 5 values of hydraulic conductivities were varied as shown in Table 3-2. Using the assumption of horizontal isotropy, the hydraulic conductivity in the 3 axes (x, y, and z) were set at the same value. This hypothetical site consisted of 1 type of soil from the top to bottom layer. The 3 values of longitudinal dispersivity are shown in Table 3-3. The dispersivity variables of the other directions were automatically assigned by the MODFLOW program. The summary of these values are shown in Table 3-4. Other model properties were set as default values given by MODFLOW. The model parameters or properties and their values are shown in Table 3-5.

Table 3-2. The varied values of hydraulic conductivity (m/s)

C1	0.0001
C2	0.0002
C3	0.0003
C4	0.0007
C5	0.001

Table 3-3. The varied values of longitudinal dispersivity (m)

D1	1
D2	50
D3	500

Table 3-4. The summary of the dispersivity variables

Longitudinal Dispersivity	same as in Table 3-3
Horizontal to Longitudinal Ratio	0.1
Vertical to Longitudinal Ratio	0.01
Molecular Diffusion Coefficient	0.0

Table 3-5. The default property values given by MODFLOW

Specific Storage (1/m)	0.0001
Specific Yield	0.2
Effective Porosity	0.15
Total Porosity	0.15

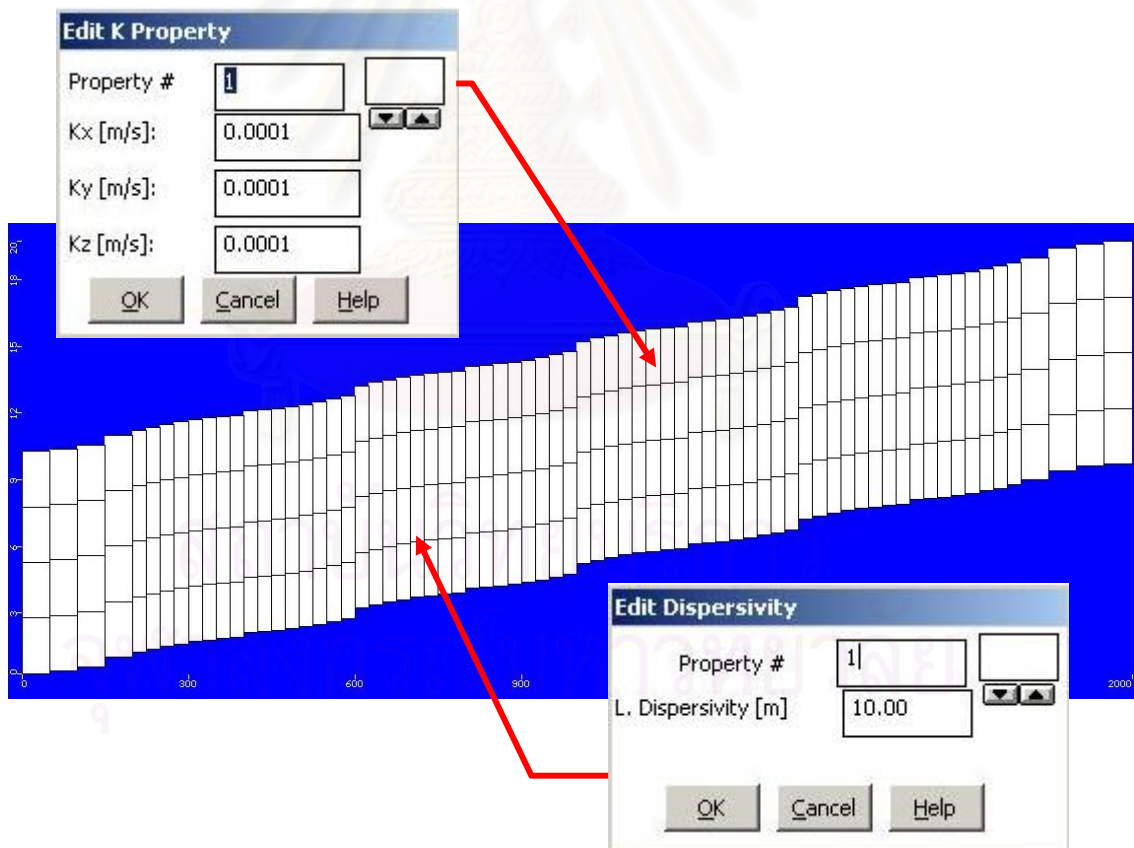


Fig. 3-3. Assigning hydraulic conductivity and longitudinal dispersivity

Step 3: Assign model boundary conditions: constant head and recharge rate.

In this step, the model boundary conditions were assigned, beginning with the recharge boundary. Two recharge boundaries (i.e. recharge flux at the entire top layer of model and recharge flux at the source of contamination) were included in this hypothetical site. The recharge flux at the entire top layer of model was assigned at 100 mm/year for 7300 days (20 years). This was assumed to be the precipitation to the area. The recharge flux at the source of contamination was assigned to be varied for 3 values to create a data set for training the ANN. Those 3 values are shown in Table 3-6. The reason for choosing these 3 values is described in the next section.

Table 3-6. The varied recharge rate at the source of contamination (mm/year)

R1	25
R2	500
R3	2500

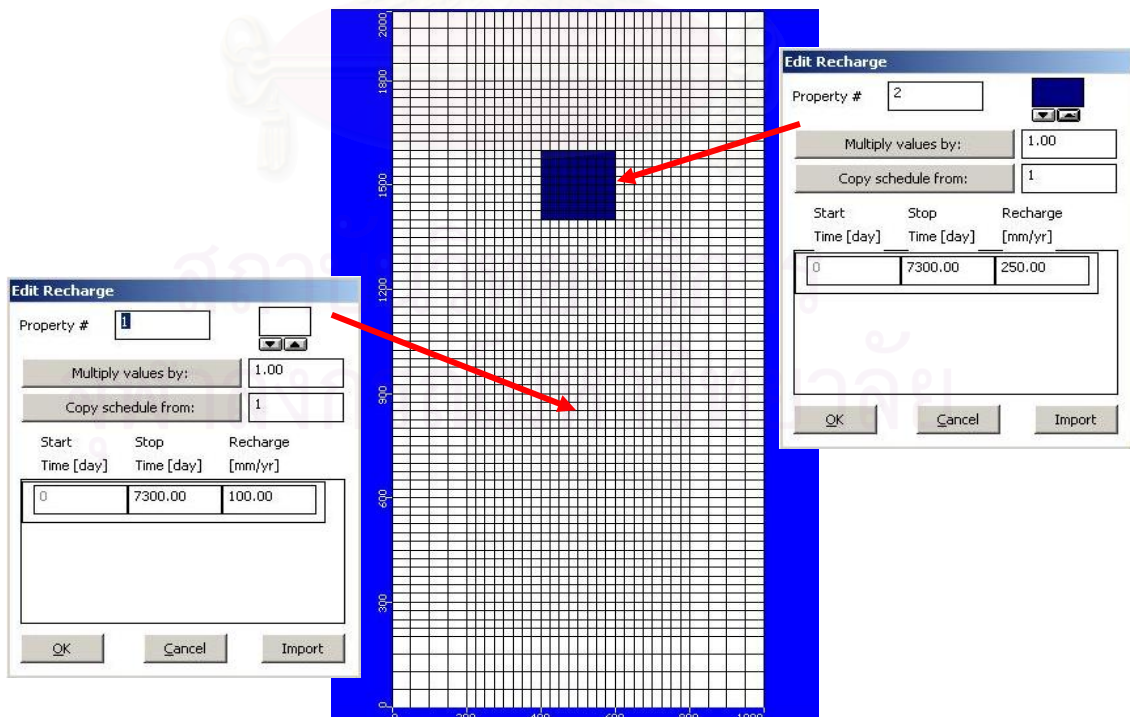


Fig. 3-4. Assigning the recharge boundaries

Next, the constant head boundary conditions were assigned to the aquifer at the north and the south boundaries. The constant head along the southern boundary of the model was fixed at 18 meter whereas the constant head along the northern boundary of the model was varied from 21 to 28 meters (as shown in Table 3-7) in order to vary the constant head difference between those two boundaries. All 4 layers had the same constant head boundary as the top layer had.

Table 3-7. The varied values of constant head difference (m)

	The constant head at the northern boundary (m)	The constant head at the southern boundary (m)	The constant head difference (m)
H1	21	18	3
H2	23	18	5
H3	25	18	7
H4	28	18	10

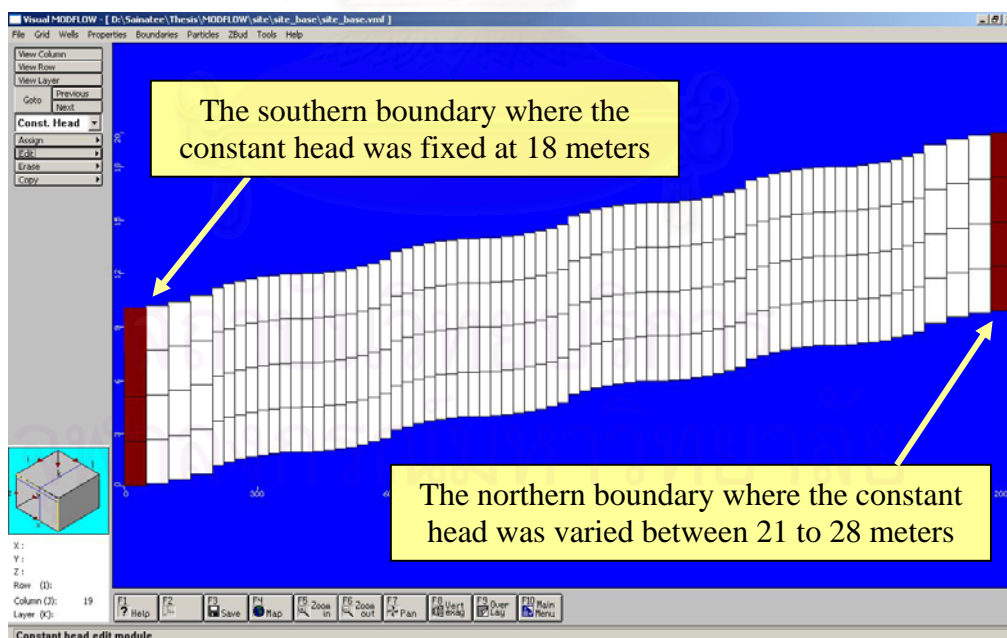


Fig. 3-5. Assigning the constant head boundaries (cross section of the hypothetical site)

Step 4: Assign model transport parameters and chemical reaction parameters.

This step was necessary to complete a simplified transport model. The transport engine and the initial conditions for the contaminant transport scenario were setup. All the transport and chemical reaction parameters were set as given in the Visual MODFLOW Tutorial (MODFLOW, 1999). In this study, jet fuel was the contaminant. The information is shown in Table 3-8.

Table 3-8. The model transport and chemical reaction parameters

Transport Engine	MT3DMS
Sorption (or dual-domain transfer)	Linear Isotherm (equilibrium-controlled)
Reactions	No Kinetic reactions
Number of species (chemical)	1
Distribution Coefficient [1/(mg/L)]	1.58E-7
Mobility	YES
Initial Concentration (mg/L)	0

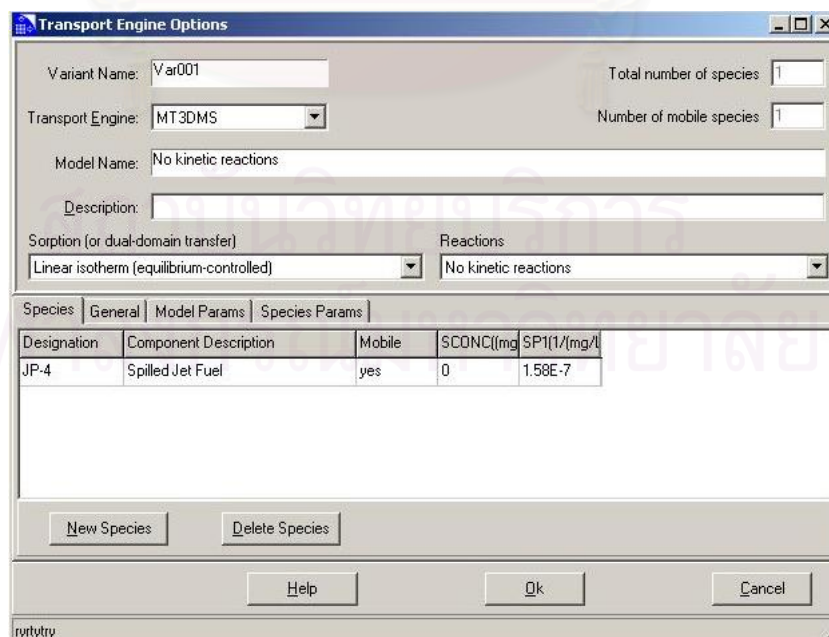


Fig. 3-6. Setup of assigning the model transport parameters

Step 5: Designate the contaminant source and concentration.

In this step, the recharge concentration was designated at the source of contamination. The concentration of this particular chemical was given at 5000 mg/L for 7300 days (20 years).

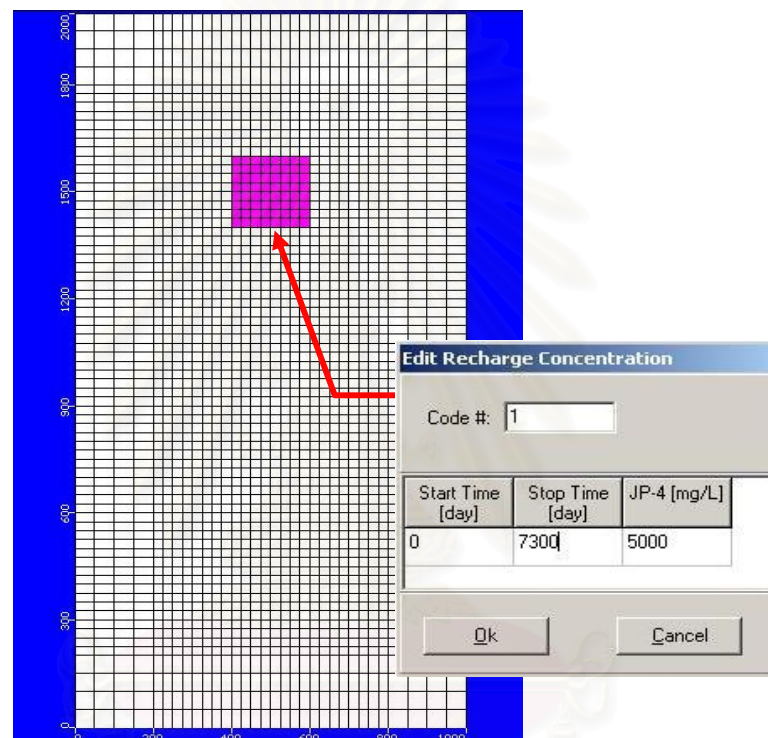


Fig. 3-7. Assigning the concentration of the contaminant at the source of contamination

Step 6: Add observation wells.

In order to simulate the breakthrough curve (i.e. the concentration of the contaminant versus time), one or more observation wells have to be added to monitor the concentration at the specific locations. The locations of observation wells depend on where the concentration is needed to be monitored. In this research, 5 observation wells were added. All the simulated concentrations at those observation wells were used for training the ANN. The locations of the 5 observation wells are shown in

Table 3-9 and Fig. 3-8. In order to obtain the average contaminant concentration from the bottom to the top of the wells, the screen depth of those 5 observation wells was fixed.

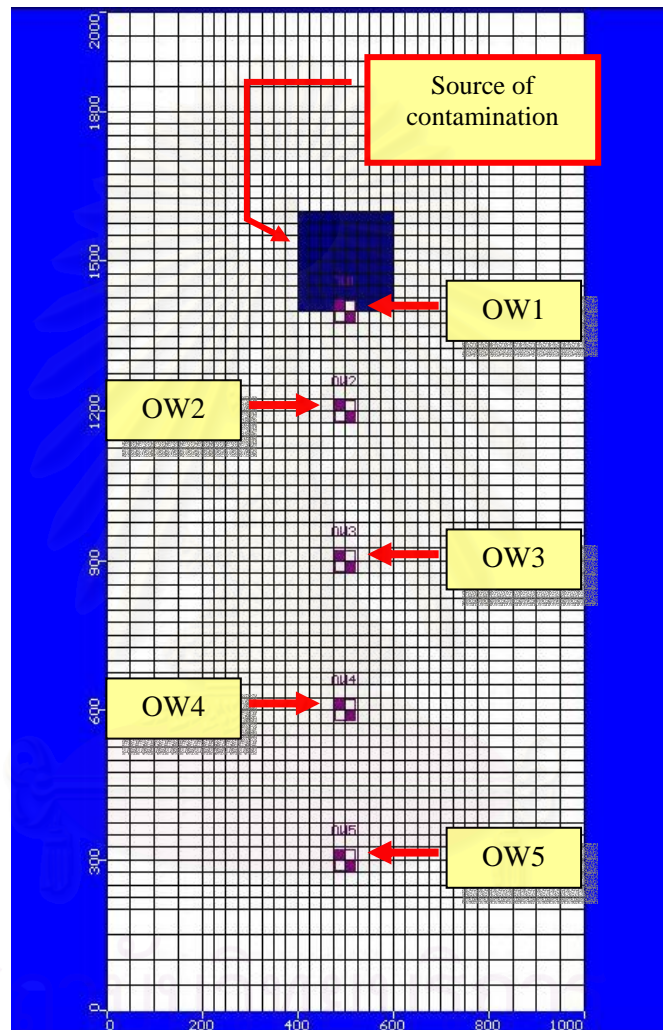


Fig. 3-8. The locations of the observation wells

Table 3-9. The locations of the observation wells

Location on model	X coordinate	Y coordinate	Distance from the center of the source of contamination. (m)
The center of the source of contamination	500	1500	-
OW1	500	1400	100
OW2	500	1200	300
OW3	500	900	600
OW4	500	600	900
OW5	500	300	1200

Step 7: Run MODFLOW: Flow simulation and MT3DMS simulations.

After assigning all the model parameters, the next step is to run the model. “Steady State Flow” was selected as the type of flow for this hypothetical site. The run options for flow simulations and transport simulations were set the same as in the Visual MODFLOW Tutorial (MODFLOW, 1999). In addition, the solver options used to calculate the flow solution were selected as the WHS solver, the default solver developed by Waterloo Hydrogeologic Inc. It is the fastest and most stable MODFLOW solver available (Fig. 3-9). The upstream finite difference advection method was chosen to simulate the contaminant transport and the generalized conjugate gradient solver (GCG) was applied to find an implicit rather than explicit solution (Fig. 3-10).

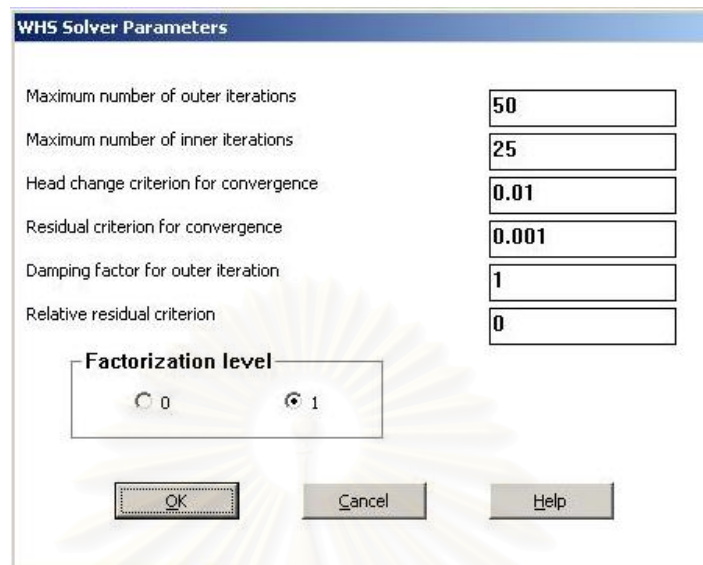


Fig. 3-9. Setting the WHS solver parameters

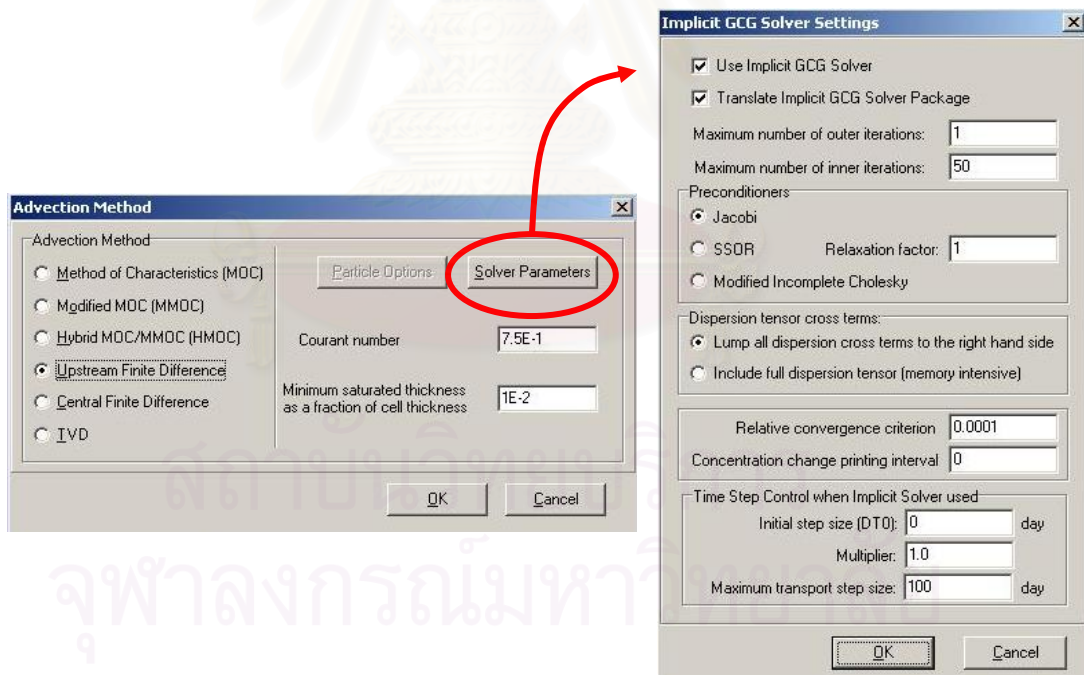


Fig. 3-10. Setting the transport simulation options

3.3 The Determination of the Inputs for the ANNs

At the beginning, the site_base was created. All the model parameters were set the same as in Section 3.1, except for some parameters that were varied. The model parameters assigned to the site_base are summarized in Table 3-10. Many sites were generated by varying the value of one parameter while the other parameters remained fixed. Finally, the parameters were determined for the ANN.

To determine which values of hydraulic conductivity should be selected for input into the ANN model, every model parameter was fixed except for hydraulic conductivity. Then, 6 sites were created and the breakthrough curves of those 6 sites and site_base were compared as illustrated in Fig. 3-110. These values of hydraulic conductivity used for creating the 6 sites represent the aquifer with good permeability. This procedure was also used for longitudinal dispersivity, constant head difference and recharge flux at the source of contamination. Hence, 5, 7, and 5 sites for determining the value of longitudinal dispersivity, constant head difference, and recharge rate, respectively, were created and the breakthrough curves were compared as shown in Fig. 3-120, Fig. 3-13, and Fig. 3-14. The values of longitudinal dispersivity, constant head difference, and recharge rate at the source of contamination were chosen by trial and error until the site can not be created by MODFLOW. The monitored concentrations at the second observation well were used for plotting breakthrough curves. Table 3-11 summarizes the sites that were created in this step and their values of hydraulic conductivity, longitudinal dispersivity, constant head difference, and the recharge rate at the source of contamination.

After considering the plots of the breakthrough curves in Fig. 3-11, Fig. 3-12, Fig. 3-13, and Fig. 3-14, 5 hydraulic conductivities, 3 longitudinal dispersivities, 4 constant head difference, and 3 recharge fluxes at the source of the contamination (mark with * in the figures) were selected to create the data set for training the ANNs. Those values were determined by selecting the highest, the lowest, and the values in between in the graphs.

Table 3-10. The parameters of the site_base

Site dimensions (meters)	1000 x 2000 x 10
Slope	10:2000
Layers	4
Type of soil	1
*Hydraulic conductivity in three direction (m/s)	0.0001
Specific Storage (1/m)	0.0001
Specific Yield	0.2
Effective Porosity	0.15
Total Porosity	0.15
Recharge rate through top layer (mm/year)	100
*Recharge rate at the source of contamination (mm/year)	250
*Constant head at the north boundary (m)	22
Constant head at the south boundary (m)	18
*Constant head difference (m)	6
Concentration of contaminant (mg/l)	5000
*Longitudinal Dispersivity (m)	10
Horizontal to Longitudinal Dispersivity ratio	0.1
Vertical to Longitudinal Dispersivity ratio	0.01
Molecular Diffusion Coefficient	0
Sorption (or dual-domain transfer)	Linear Isotherm (equilibrium-controlled)
Reactions	No Kinetic reactions
Number of species (chemical)	1
Distribution Coefficient [1/(mg/L)]	1.58E-7
Mobility	YES
Initial Concentration (mg/L)	0

Note: * the parameters with values that were varied

Table 3-11. The values of the 4 parameters at each site

Site name	Conductivity (m/s)	Longitudinal Dispersion (m)	Constant Head Difference (m)	Recharge Rate (mm/year)
site_base	0.0001	10	4	250
site_cd1	0.0002	10	4	250
site_cd2	0.0003	10	4	250
site_cd3	0.0005	10	4	250
site_cd4	0.0007	10	4	250
site_cd5	0.0008	10	4	250
site_cd6	0.001	10	4	250
site_dp1	0.0001	0.1	4	250
site_dp2	0.0001	1	4	250
site_dp3	0.0001	50	4	250
site_dp4	0.0001	100	4	250
site_dp5	0.0001	500	4	250
site_ch1	0.0001	10	3	250
site_ch2	0.0001	10	5	250
site_ch3	0.0001	10	6	250
site_ch4	0.0001	10	7	250
site_ch5	0.0001	10	8	250
site_ch6	0.0001	10	9	250
site_ch7	0.0001	10	10	250
site_re1	0.0001	10	4	25
site_re2	0.0001	10	4	100
site_re3	0.0001	10	4	500
site_re4	0.0001	10	4	1000
site_re5	0.0001	10	4	2500

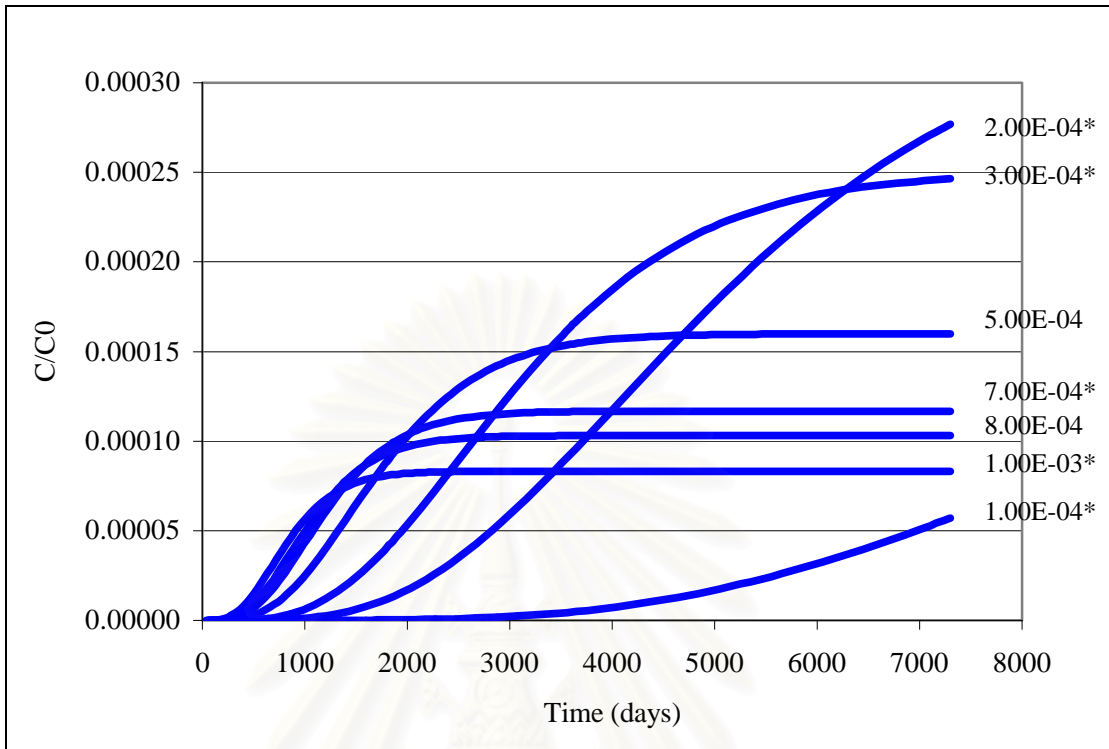


Fig. 3-11. The comparison between 7 breakthrough curves from 7 values of hydraulic conductivity

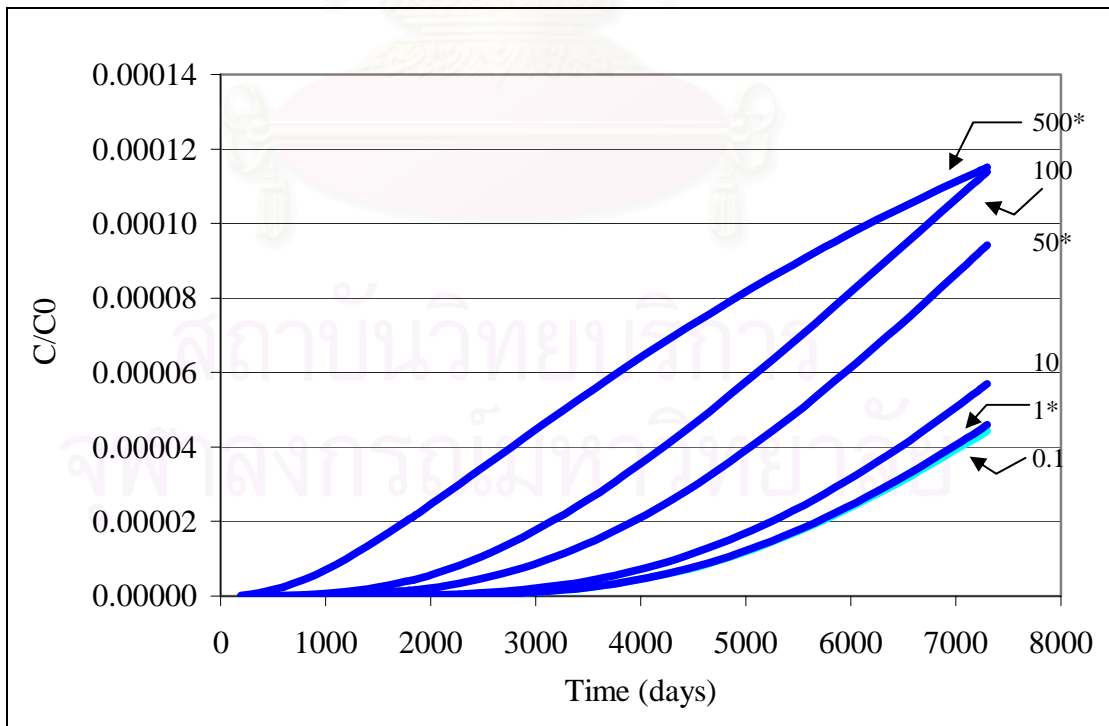


Fig. 3-12. The comparison between 6 breakthrough curves from 6 values of longitudinal dispersivity

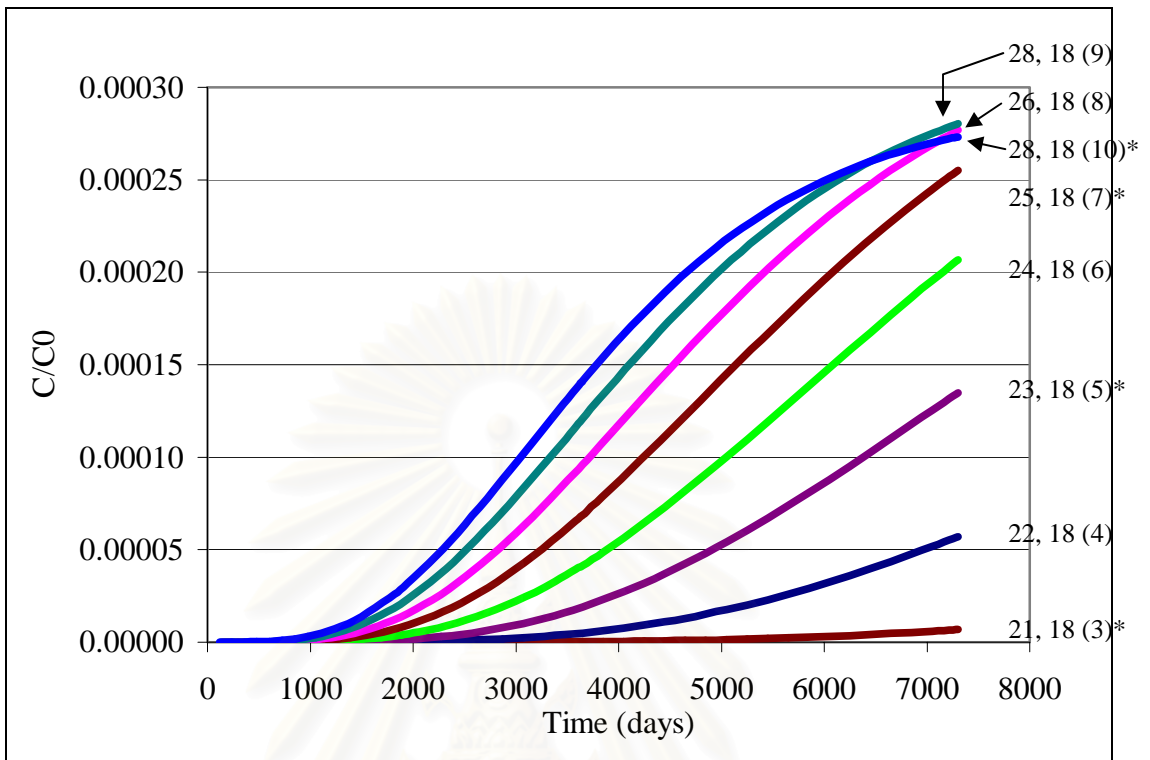


Fig. 3-13. The comparison between 8 breakthrough curves from 8 values of constant head difference

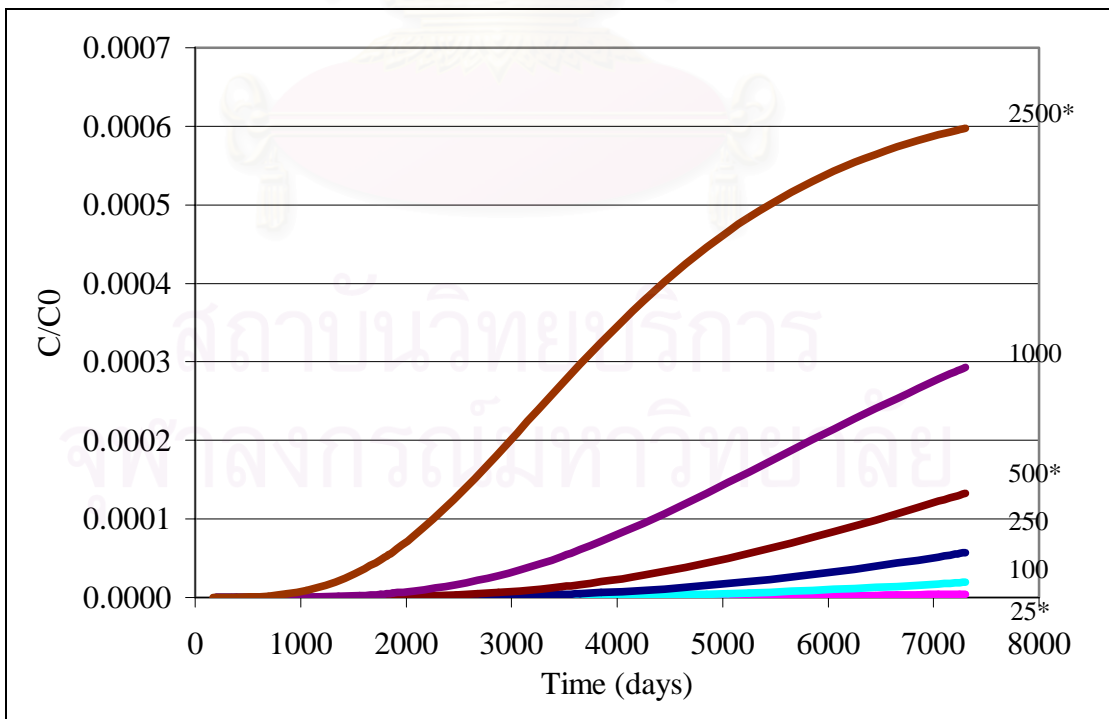


Fig. 3-14. The comparison between 6 breakthrough curves from 6 values of recharge flux at the source of the contamination

3.4 The Generation of the Data

After the determination of model input parameters and their values, the breakthrough curves of the combinations of the 4 parameters were generated. Table 3-12 summarizes the values of each parameter that was used to generate the breakthrough curve. To enhance the understanding of how the parameters were combined to create one hypothetical site, it can be graphically depicted as shown in Fig. 3-15. Finally, 180 sites were created as a result of the combination of the 4 parameters (i.e. hydraulic conductivity, longitudinal dispersivity, constant head difference, and recharge rate at the source of the contamination). Each site contained the concentrations of the contaminant at 5 observation wells corresponding to time at the specific distance from the source of contamination.

Table 3-12. The summary of the variation of the 4 parameters

Parameters	Values	
Hydraulic Conductivity (m/s)	C1	0.0001
	C2	0.0002
	C3	0.0003
	C4	0.0007
	C5	0.001
Longitudinal Dispersivity (m)	D1	1
	D2	50
	D3	500
Constant Head Difference (m)	H1	3
	H2	5
	H3	7
	H4	10
Recharge Rate at the Source of Contamination (mm/year)	R1	25
	R2	500
	R3	2500

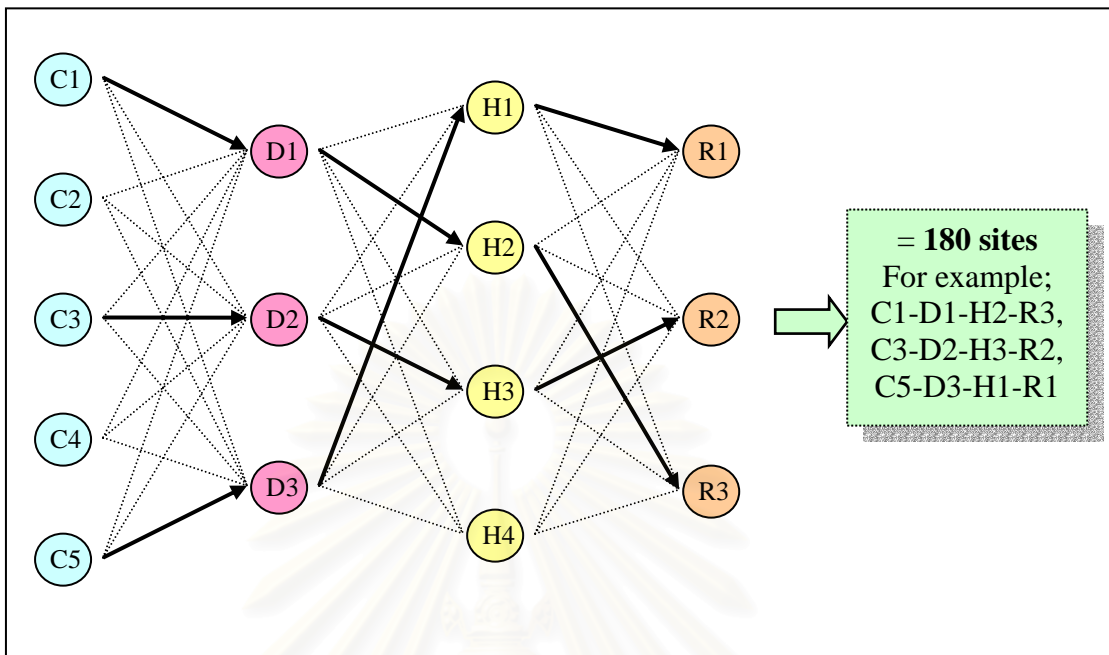


Fig. 3-15. The combination of the 4 parameters to create one site

3.5 The Preparation of the Data Set

The outputs (i.e. the breakthrough curve) of 180 sites from the MODFLOW simulation are given in the *.obs file. This file contains the numbers of time steps and the related concentration of the contaminant to each time step for each of the observation wells. The results are presented in column format. Before introducing the 180 sites to the ANN, the training and the testing data set were prepared. The steps of data preparation are described in this section. From this point, every step was implemented by writing a computer code (MATLAB, 2005).

Step 1: Translate the *.obs files given from MODFLOW into *.m files used in MATLAB (Organize the data to be available for training ANN).

In the first step, the concentrations (C) given from MODFLOW were divided by 5000 (C_0), the initial concentration at the source of contamination. The C/C_0 (i.e. the ratio of the measured concentration (C) at the specific location and the initial

testing data set. The sites that were in the training and the testing data sets were randomly selected. One important thing had to be considered: the amount of each value in the same parameter should be comparable. The method of random selection is described in a flow chart (Fig. 3-17).

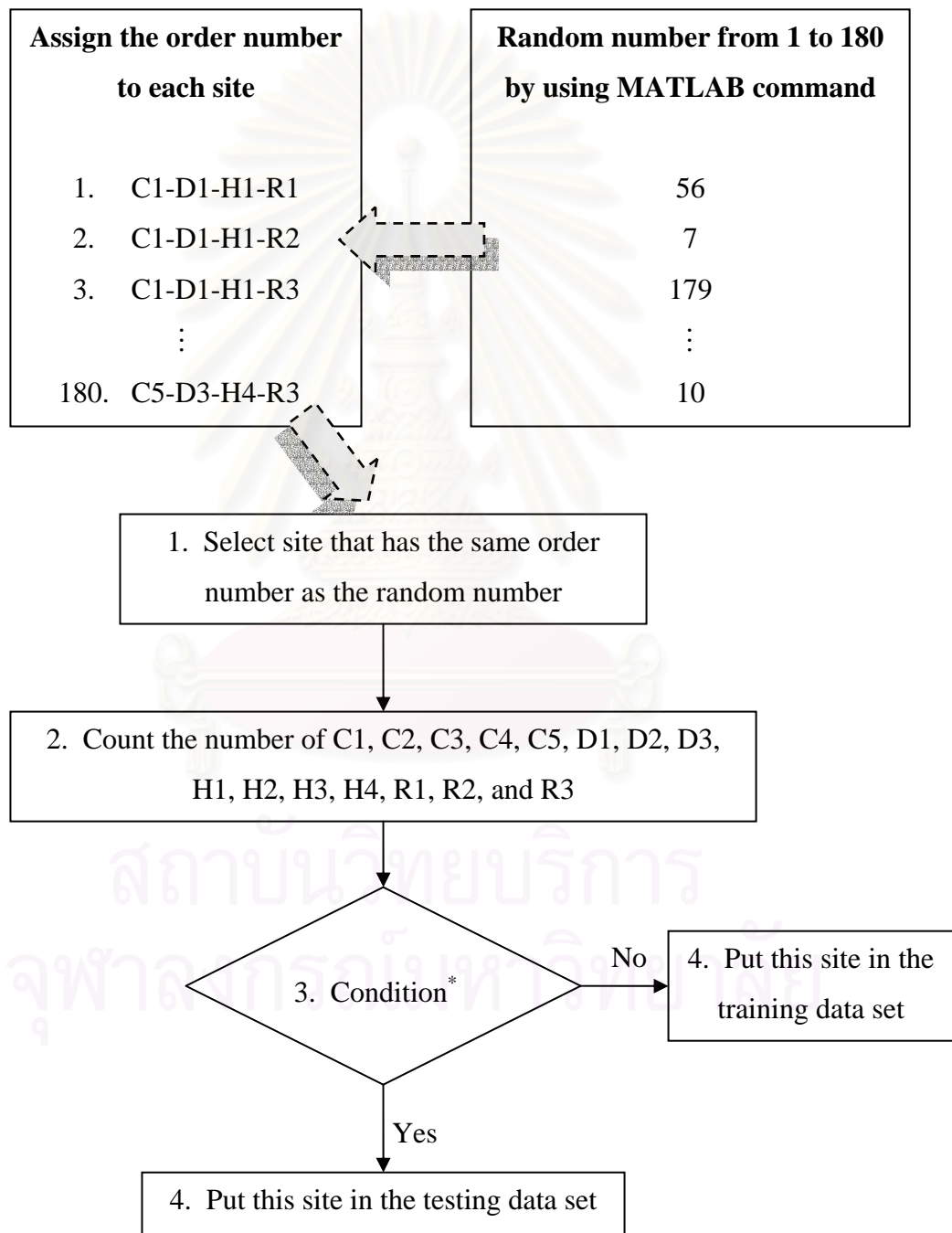


Fig. 3-17. Steps of creating the data sets

The condition was if the number of C1s, C2s, C3s, C4s, or C5s was greater than 24; or the number of D1s, D2s, or D3s was greater than 40; or the number of H1s, H2s, H3s, or H4s was greater than 30; or the number of R1s, R2s, or R3s was greater than 40 then the selected site was put in the testing data set. But if the condition was not met, then the selected site was put in the training data set. Steps 1 through 4 were repeated until all 180 sites were selected to create the data set.

Finally, the training data set included 118 sites while the testing data set included 62 sites (see Appendix A). The numbers of sites in the training and testing data sets are shown in Table 3-13.

Table 3-13. The numbers of sites that contain each value of the 4 model parameters used in training and testing data sets

The model parameter	Number of sites in training data set (sites)	Number of site in testing data set (sites)
C1	24	12
C2	24	12
C3	23	13
C4	23	13
C5	24	13
D1	38	22
D2	40	20
D3	40	20
H1	30	15
H2	30	15
H3	29	16
H4	29	16
R1	39	21
R2	40	20
R3	39	21

Next, the data from the 118 sites in the training data set were combined and likewise the data from the other 62 sites in the testing data set were combined. The method of combination is illustrated in Fig. 3-180. From Fig. 3-16, it can be seen that one site contained 5 columns representing the C/C_0 values at the 5 observation wells. One of the 5 columns of C/C_0 was picked and joined together with the 6th – 10th columns. Then, the data matrix became 6 columns. The 7th column was the distance of the observation well from the source (D), which corresponded to the C/C_0 of the selected observation well. Finally, the data matrix contained 7 columns with numerous rows. Each row indicated a particular pattern. These steps were repeated until the combination of 118 and 62 sites had finished.

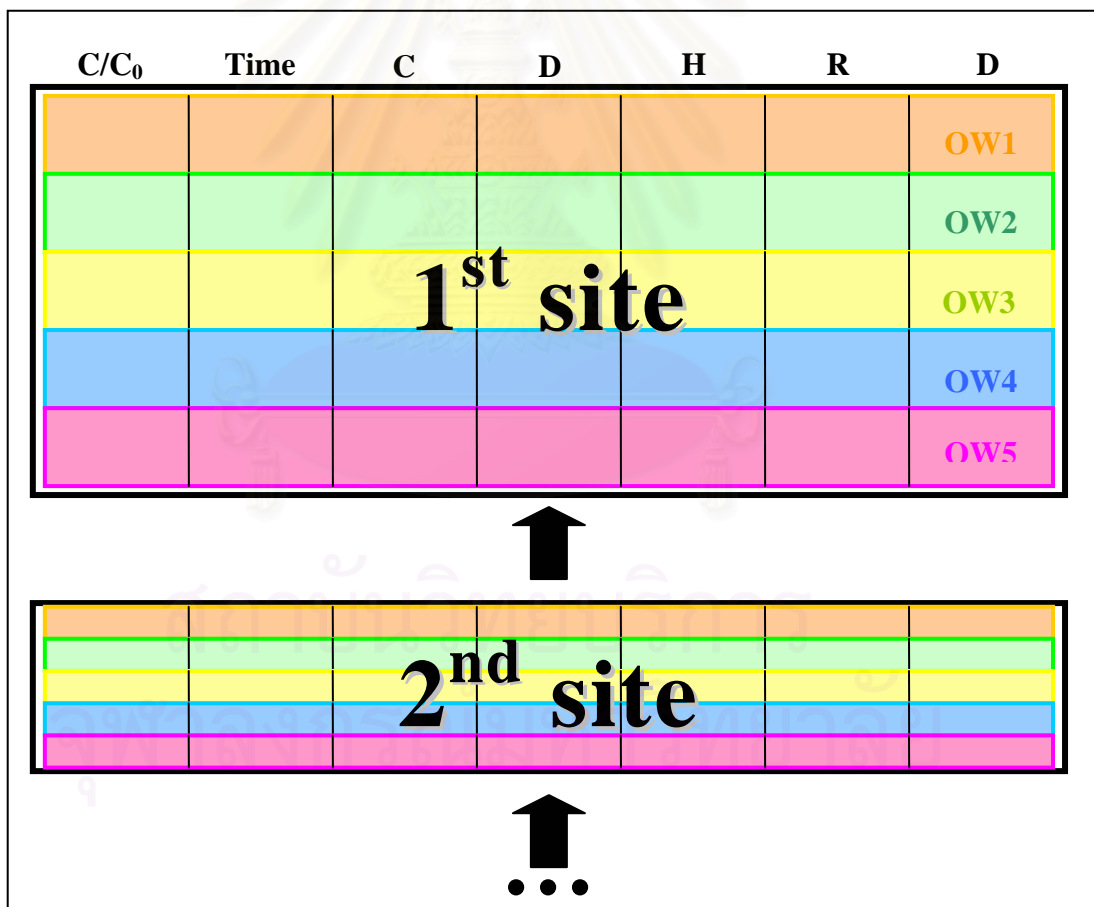


Fig. 3-18. The combination of the data

Step 3: Flip matrixes.

As a result of the combination of the 118 and 62 sites and because one site contains a matrix of 7 columns with many numbers of rows, the training and the testing data sets produced matrixes with size of 88,235 x 7 and 44,550 x 7, respectively. However, to introduce the data sets to the ANN, the matrixes had to be flipped (Fig 3-19). Thus, one column was for one pattern (not one row for one pattern). Another word, one input pattern is a vector of seven elements. Then, the training and the testing data sets finally produced the matrixes of 7 x 88,235 and 7 x 44,550, respectively.

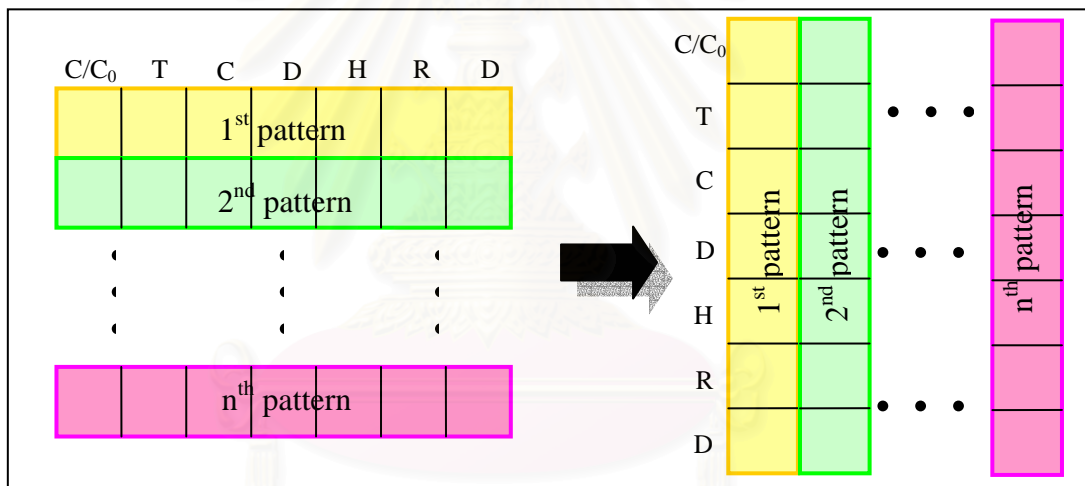


Fig. 3-19. How the data sets were flipped

Step 4: Assign input and target groups of the training data set.

In this step, the training data set was divided into 2 groups, input and target. The testing data set was divided in the same way. The target group was C/C₀. The input group consisted of 6 elements: time, hydraulic conductivity, longitudinal dispersivity, constant head difference, the recharge rate at the source of contamination, and distance.

Step 5: Normalize data.

This step was added in Phase II: Numerical Experiment No. 2 and No. 3 (see Section 3.8). In order to improve the performance of the network, the training data set was normalized as follows (MATLAB, 2005). The testing data set was normalized in a similar manner.

1. The input matrixes of the training and the testing data sets were rescaled into log scale.

$$[P_i] = \log[P] \quad (3-1)$$

where $[P]$ = input matrix

2. The input matrix in log scale (P_i) and the target vector of the training data were normalized in order to have the data fall in the range of -1 to +1 by using the “**premnmx**” computer code. The **premnmx** preprocessed the network training set by normalizing the inputs and targets so that they fell in the interval [-1, 1].

$[PN, \text{minp}, \text{maxp}, TN, \text{mint}, \text{maxt}] = \text{premnmx}(P, T)$

where

- P = matrix of input vectors (in log scale)
- T = matrix of target vectors (not in log scale)
- PN = matrix of normalized input vectors
- minp = vector containing minimums for each P
- maxp = vector containing maximums for each P
- TN = matrix of normalized target vectors
- mint = vector containing minimums for each T
- maxt = vector containing maximums for each T

The algorithm can be described as seen below.

$$p_n = \left[2 \times \frac{(p - \min p)}{(\max p - \min p)} \right] - 1 \quad (3-2)$$

$$t_n = \left[2 \times \frac{(t - \min t)}{(\max t - \min t)} \right] - 1 \quad (3-3)$$

3. The testing data set was normalized in the same way as the training data set. This step used the computer code “**trammx**” which transformed the network input set using minimum and maximum values that had been previously computed by **premmx**. The algorithm can be described using the same equations as in Eq. (3-2) and Eq. (3-3).

```
pn_test = trammx(p_test,minp,maxp)
```

where **p_test** = matrix of input vectors of testing data (in log scale)

pn_test = matrix of normalized input vectors

minp = vector containing original minimums for each input of training data

maxp = vector containing original maximums for each input of training data

3.6 The Development of the ANN and Network Training

At this stage, the training data set has been prepared for training the network. The input vector included 6 elements, while the target consisted of 1 element. And the data were normalized.

The ANN was implemented using a feedforward backpropagation neural network (FBNN). The FBNN was developed using MATLAB, a commercially available computer programming software program. The ANN training was

performed using the default values given by MATLAB except for the performance goal and the maximum number of epochs. Both were decided at 1E-6 and 500, respectively. The performance goal is defined as the sum square error of the output from the ANN and the desired target. The epoch is the iteration. The hyperbolic tangent was chosen as the transfer function in the hidden layers whereas the linear transfer function was used in the output layer. Next, the MATLAB code is explained.

Step 1: Create the network.

In this step, the N-layer feedforward backpropagation neural network ('net') was created using "newff." The syntax is shown below.

```
net = newff(PR, [S1 S2 . . . SN], {TF1 TF2 . . . TFN}, BTF, BLF, PF)
```

where PR = matrix of minimum and maximum values for each input elements
 S_i = Size of i^{th} layer, for N layers
 TF_i = Transfer function of i^{th} layer, default = 'tansig'
 BTF = Backpropagation network training function, default = 'traingdx'
 BLF = Backpropagation weight/bias learning function, default = 'learngdm'
 PF = Performance function, default = 'mse'

An example of the syntax used in this research is explained below. This syntax created a 1-hidden-layer FBNN. The hidden layer consisted of 6 neurons whereas the output layer consisted of 1 neuron. The transfer functions were a hyperbolic tangent for the hidden layer and a linear transfer function for the output layer. The training function was 'trainlm.' The weight/bias learning function and performance function were used as defaults.

```
net = newff(minmax(pn),[6 1],{'tansig','purelin'},'trainlm')
```

where p_n = input vector from Eq. (3-2)
 $\text{minmax}(p_n)$ = matrix of minimum and maximum value for each element of input vector (p_n)
 tansig = hyperbolic tangent transfer function
 purelin = linear transfer function

Step 2: Assign the training parameters.

The performance goal and the maximum number of epochs were assigned by the following code.

```
net.trainParam.epochs = 500  
net.trainParam.goal = 1e-6
```

Step 3: Train the network.

After creating network, the network 'net' was trained according to the network training function and parameters. The "train" code takes the network inputs and targets and returns the new network, training record, network outputs and network errors. The syntax is shown below.

```
[net,tr,Y,E]= train(net,pn,tn)
```

where p_n = input vector from Eq. (3-2)
 t_n = target vector from Eq. (3-3)
 t_r = training record (epoch and performance)
 Y = network outputs
 E = network errors

3.7 The Testing of Network

After completing the training process, the ability of the network to predict the C/C_0 from the unseen input was tested by using the testing data set. The mean square error was calculated. The following syntax explains how the network was tested. The “**sim**” code simulates the output by taking the input of the testing data set using the trained network ‘net.’

```
a_sim = sim(net,pn_test)
```

Because the input had been normalized before it was introduced to train the ANN, the output from the network was in normalized form. Thus, it needed to be unnormalized to convert it back to the real value. The computer code used for this task was “**postmnmx**.” It postprocessed the network output after prediction using **sim** and converted the data back into unnormalized units. **postmnmx** is always used when the output is predicted by using the normalized input.

```
a = postmnmx(a_sim, mint, maxt);
```

where a = unnormalized output vectors
a_sim = output vector given from network simulation
mint = vector containing original minimums for each target
of training data
maxt = vector containing original maximums for each target
of training data

The algorithm of the **postmnmx** is described below.

$$a = [0.5 \times (a_sim + 1) \times (\max t - \min t)] + \min t \quad (3-4)$$

3.8 The Network Performance Criteria

In this study, 2 performance criterions were used to determine the accuracy and the ability of the network's prediction. These criterions were calculated using the normalized target vector of the testing data set and the output from network simulations. The algorithms and computer code are described below.

1. Mean Square Error (MSE)

$$\begin{aligned} e &= t_{n_test} - a_{sim} \\ perf &= \mathbf{mse}(e) \end{aligned}$$

where t_{n_test} = normalized target vector of the testing data set
 a_{sim} = output vector of network before unnormalized
 e = the error vector of each t_{n_test} and a

$$MSE = \frac{\sum_{i=1}^{i=n} e^2}{n} \quad (3-5)$$

2. Correlation Coefficient (R)

The correlation coefficient gives the quality of a least squares fitting to the original data. By using the computer code given from MATLAB, "**corrcoef**," the correlation coefficient is calculated by inputting the column vector of the target of the testing data set and the vector of output from the network. The syntax and algorithm (Spiegel, 1992) can be defined as follows:

$$R = \mathbf{corrcoef}(t_{n_test}, a)$$

where t_test = target vector of testing data set
 a = unnormalized output vector

$$R = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (3-6)$$

where x = target vector of testing data set (t_test)
 y = unnormalized output vector (a)
 n = number of data point

3.9 The Numerical Experiments

In order to find the network configuration that provides the best prediction of C/C_0 , the experiment was divided into 3 phases in order to obtain three specific objectives. The performances were compared by considering the performance criteria and the plots. The results are shown in the next chapter. Next, the objective and the method of each phase are explained. Table 3-16 concludes the methodology used to determine the network configuration.

Phase I: Determination of the training function.

The objective of this phase was to find which training function was going to work well in this study. Thus, the 'site_*' data set, which included all the sites in Table 3-11 except for 'site_base,' was used to create the training data set, while 'site_base' was used as the testing data set. 2-hidden-layer FBNN, containing one layer with 6 and the other with 3 neurons, were created. As described in Section 3.6 Step 1, the training function was chosen to create the FBNN. 12 training functions (Table 3-14) were used and their performances were compared.

Table 3-14. Training function and its description

Training Function	Description
traingd	Basic gradient descent. Slow response, can be used in incremental mode training
traingdm	Gradient descent with momentum. Generally faster than traingd. Can be used in incremental mode training.
traingdx	Adaptive learning rate. Faster training than traingd, but can only be used in batch mode training.
trainrpf	Resilient backpropagation. Simple batch mode training algorithm with fast convergence and minimal storage requirements.
traincgf	Fletcher-Reeves conjugate gradient algorithm. Has smallest storage requirements of the conjugate gradient algorithms
traincgq	Polak-Ribière conjugate gradient algorithm. Slightly larger storage requirements than traincgf. Faster convergence on some problems
traincgb	Powell-Beale conjugate gradient algorithm. Slightly larger storage requirements than traincgq. Generally faster convergence
trainscg	Scaled conjugate gradient algorithm. The only conjugate gradient algorithm that requires no line search. A very good general purpose training algorithm
trainbfg	BFGS quasi-Newton method. Requires storage of approximate Hessian matrix and has more computation in each iteration than conjugate gradient algorithms, but usually converges in fewer iterations
trainoss	One step secant method. Compromise between conjugate gradient methods and quasi-Newton methods
trainlm	Levenberg-Marquardt algorithm. Fastest training algorithm for networks of moderate size. Has memory reduction feature for use when the training set is large

trainbr	Bayesian regularization. Modification of the Levenberg-Marquardt training algorithm to produce networks that generalizes well. Reduces the difficulty of determining the optimum network architecture
---------	--

(from MATLAB, 2005)

Phase II: Determination of the method for the preprocessing training data set.

The objective of this phase was to identify the best preprocessing method for the data set. A common method is to make all the elements lay around zero. In this phase, 3 numerical experiments were conducted. Data that were preprocessed were used in the second and the third experiments, but not in the first. In the third experiment, only the input elements (not the target elements) were rescaled into log scale before being normalized. The target elements can not be rescaled into log scale because of some minus values. The results from the networks with and without preprocessing data were compared. The data set was created from the combination of the 4 model parameters (i.e. hydraulic conductivity, longitudinal dispersivity, constant head difference, and the recharge rate at the source of the contamination). The 2-hidden-layer FBNN, containing one layer with 6 and the other with 3 neurons, was trained using the training function decided upon in Phase I. The preprocessing methods have already described in Section 3.5 Step 5.

Phase III: Determination of the network architecture.

The objective of this phase was to find the network architecture that provided the most accurate of C/C_0 prediction. 1- and 2-hidden-layer FBNNs with various numbers of neurons were trained (Table 3-15) and the results were compared. The training and the testing data sets were created from the combination of the 4 model parameters. In this phase, the training and the testing data sets were normalized by the best method determined from Phase II. Moreover, in order to avoid premature

saturation (according to Maier and Dandy, 1996), the initial weights should be randomly distributed between -0.1 and +0.1. The results of the network with and without rescaling weight were compared. Since weights and biases were randomly selected by the program, the networks were created and trained 10 times in order to have a variety of initial weights. The network performance with the lowest mean square error and the correlation coefficient closet to 1 was selected as a representative of that network architecture.

Table 3-15. The network architecture

The number of hidden layers	The number of neurons at each hidden layer	Total parameters (including biases)
1	[6]	49
	[12]	97
	[24]	193
	[36]	289
	[48]	385
	[60]	481
2	[6 3]	67
	[10 5]	131
	[16 8]	257
	[20 10]	361
	[24 12]	481

Table 3-16. The summary of the phases and the numerical experiments

Phase/ Objective	Numerical Experiment	Description
<p style="text-align: center;"><u>Phase I</u></p> <p>To determine the training function</p>	1	<ul style="list-style-type: none"> • The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 hidden neurons, were trained using the training functions described in Table 3-14. • The training and the testing data sets were created from ‘site_*’ (all the sites in Table 3-11 except for ‘site_base’) and ‘site_base’, respectively.
<p style="text-align: center;"><u>Phase II</u></p> <p>To determine the method of data preprocessing</p>	1	<ul style="list-style-type: none"> • The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 hidden neurons, were trained using the training function that provided the best performance as determined from Phase I. • The training and the testing data sets were created from the combination of the 4 model parameters. • No preprocessing data
	2	<ul style="list-style-type: none"> • The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 hidden neurons, were trained using the training function which provided the best performance as determined from Phase I. • The training and the testing data sets were created from the combination of the 4 model parameters. • The data were normalized in the range of -1 to +1.
	3	<ul style="list-style-type: none"> • The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 hidden neurons, were trained using the training function which provided the best performance as determined from Phase I. • The training and the testing data sets were created from the combination of the 4 model parameters.

		<ul style="list-style-type: none"> The input elements (not the target elements) were rescaled into log scale. Then, the data were normalized in range of -1 to +1.
<p><u>Phase III</u></p> <p>To determine the network architecture and determine whether the weights should be normalized</p>	1	<ul style="list-style-type: none"> FBNNs were trained using the training function that provided the best performance as determined from Phase I. The training and the testing data sets were created from the combination of the 4 model parameters. The data preprocessing method was chosen in phase II. The 1- and 2-hidden-layer FBNNs were trained with various numbers of neurons as shown in Table 3-15.
	2	<ul style="list-style-type: none"> FBNNs were trained using the training function which provided the best performance as determined from phase I. The training and the testing data sets were created from the combination of the 4 model parameters. The data preprocessing method was chosen in phase II. The 1- and 2-hidden-layer FBNNs were trained with various numbers of neurons as shown in Table 3-15. The initial weights were normalized in range of -0.1 to +0.1.

3.10 The Flowchart of Methodology

The methodology can be graphically depicted as shown in Fig. 3-200. The flowchart shows the procedure for training one network. This procedure was followed for every experiment.

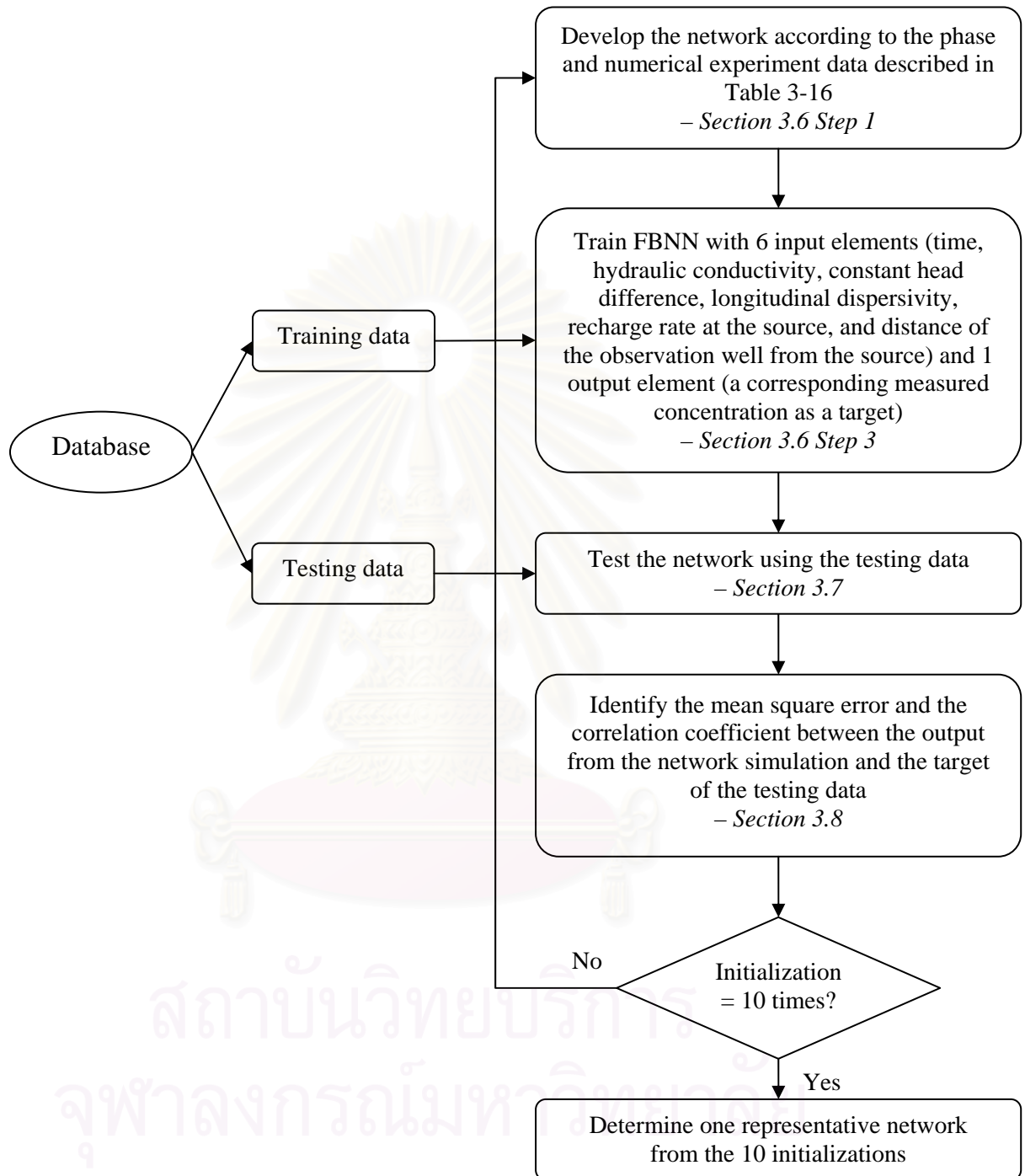


Fig. 3-20. Flow chart of the methodology

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Phase I: Determine the Training Function

After training the feedforward backpropagation neural network (FBNN) with 12 training functions (as shown in Table 3-14), the training functions 'trainlm' and 'trainbr' were found to provided the best prediction of C/C_0 . The mean square error values of the testing data given from the network when 'trainlm' and 'trainbr' were used were lower than those of the networks trained by the other training functions. In addition, the correlation coefficient of the networks trained by 'trainlm' and 'trainbr' were closer to 1 when compared with the results from the other networks. All the results from the networks trained by the 12 training functions are shown in Appendix B (Table B-1).

When comparing the results of the 2 training functions, 'trainlm' and 'trainbr', the mean square error and the correlation coefficient of the testing data given were not significantly different. Because of this, one network using 'trainlm' and one network using 'trainbr' as a training function were created and trained again. This time, the data set was created from the combination of the 4 model parameters (i.e. hydraulic conductivity, longitudinal dispersivity, constant head different, and recharge rate at the source of the contamination) as described in Section 3.4 instead of using the data from 'site_*' (all the sites in Table 3-11). The results provided in Appendix B (Table B-2) indicate that using 'trainbr' as a training function produced lower mean square error value and correlation coefficient closer to 1. Moreover, training with the 'trainbr' function has been found to improve generalization. One approach to this process is the Bayesian framework of David MacKay (MacKay, 1992). In this framework, the weights and biases of the network are assumed to be random variables with specified distributions.

Based on the results from this phase and the advantage of its ability to help with generalization, 'trainbr' was selected as the training function for next training phase in this study.

4.2 Phase II: Determine the Data Preprocessing Method

Three numerical experiments were conducted. In all 3 experiments, 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, using the training function 'trainbr' were used. The data set was created from the combination of the 4 model parameters. Each of the 3 experiments utilized a different preprocessing data method. The first experiment used the data without preprocessing. The second experiment used the data that had been preprocessed by normalizing the entire data elements in range of -1 to +1. The data in the last experiment was preprocessed in the same way as in the second experiment but only the input elements (not the target elements) of the training and testing data were rescaled into log scale before normalized in range of -1 to +1.

The results, in the forms of the mean square error and the correlation coefficient of the testing data, from the 3 experiments are shown in Table 4-1, Table 4-2, and Table 4-3, respectively. Fig. 4-1 (a, b, and c) illustrates the graphical output of the 3 experiments. The network predictions were plotted versus the desired targets as dots. The 6 solid lines indicate the boundaries of 5%, 10% and 20% error from each desired point. The 3 figures are the networks that provided the best prediction of each experiment. The best network of each experiment has been highlighted in gray in the tables.

The plots and the results shown in the tables below demonstrate that a higher degree of network prediction accuracy was achieved when data was preprocessed before being introduced to train the network. Moreover, the network prediction was much better when the input elements of the training data set were rescaled into log scale before normalized in range of -1 to +1.

Thus in the next phase, the data were normalized in the range of -1 to +1 after the input elements were rescaled into log scale before being introduced to the network.

Table 4-1. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 1. The data used in this experiment were not preprocessed.

ith Initialization	MSE of testing data set	Correlation Coefficient
1	1.02E-08	8.55E-01
2	2.17E-09	9.69E-01
3	4.99E-09	9.28E-01
4	2.98E-09	9.57E-01
5	3.48E-09	9.49E-01
6	3.64E-09	9.48E-01
7	4.19E-09	9.40E-01
8	1.83E-09	9.74E-01
9	2.10E-09	9.70E-01
10	1.95E-09	9.72E-01

Table 4-2. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 2. The data used in this experiment were preprocessed by normalization in the range of -1 to +1.

ith Initialization	MSE of testing data set	Correlation Coefficient
1	1.44E-09	9.80E-01
2	1.45E-09	9.79E-01
3	1.56E-09	9.78E-01
4	5.80E-09	9.23E-01
5	1.44E-09	9.80E-01
6	1.56E-09	9.78E-01
7	1.67E-09	9.76E-01
8	1.57E-09	9.77E-01
9	1.33E-09	9.81E-01
10	1.76E-09	9.75E-01

Table 4-3. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase II: Numerical Experiment No. 3. The data used in this experiment were preprocessed by rescaling the input elements into log scale before normalizing the data in the range of -1 to +1.

ith Initialization	MSE of testing data set	Correlation Coefficient
1	1.45E-10	9.98E-01
2	1.42E-10	9.98E-01
3	2.06E-10	9.97E-01
4	1.78E-10	9.97E-01
5	1.81E-10	9.97E-01
6	3.34E-10	9.95E-01
7	3.09E-10	9.96E-01
8	3.12E-10	9.96E-01
9	2.76E-10	9.96E-01
10	1.36E-10	9.98E-01

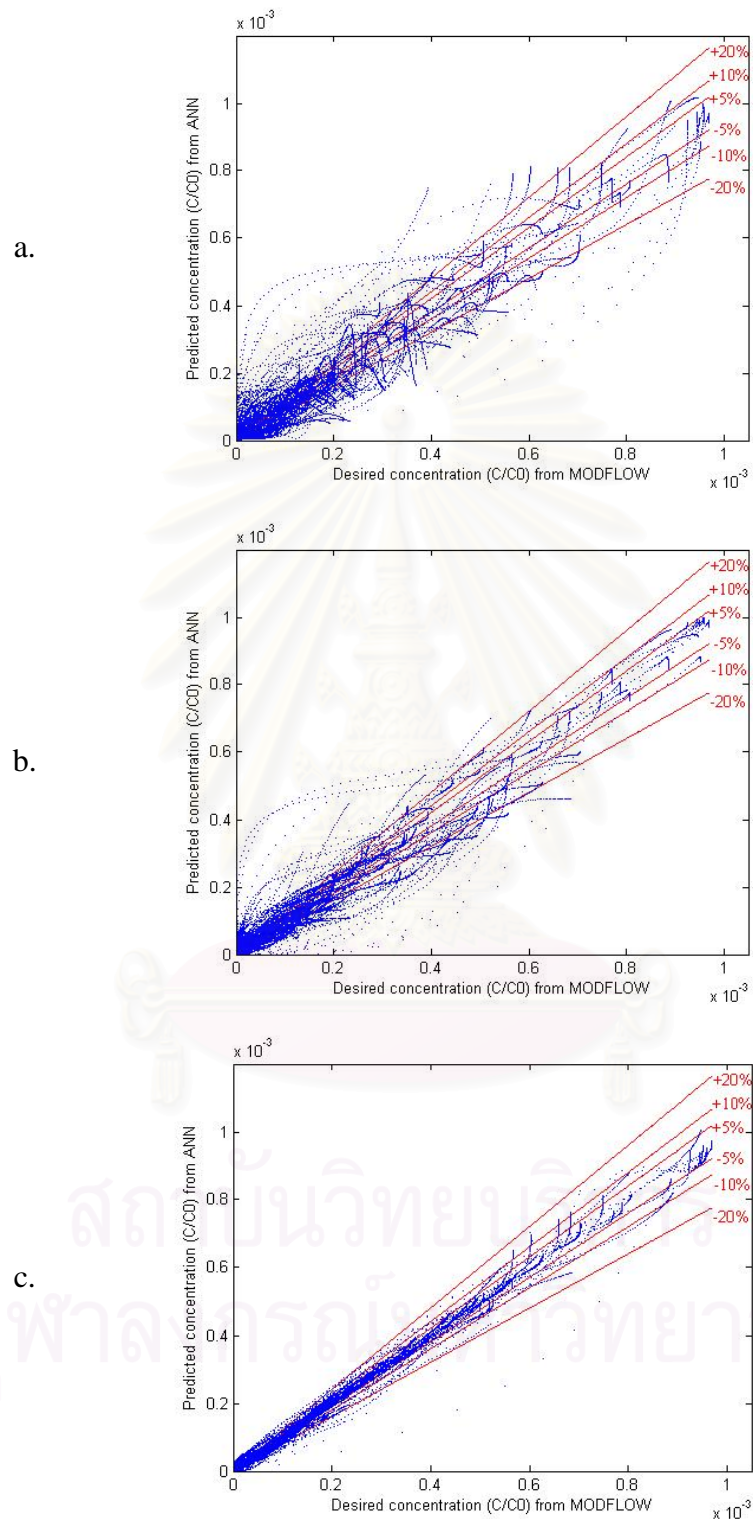


Fig. 4-1. The plot between the desired C/C_0 versus the predicted C/C_0 simulated from the ANN. (a. Experiment No. 1, b. Experiment No. 2, and c. Experiment No. 3)

4.3 Phase III: Determine the Network Architecture

This phase's aim was to determine the network architecture that provided the best C/C_0 prediction. Ten network architectures were trained (Table 3-15) using training function 'trainbr' (as a result from Phase I). The data set was created from the combination of the 4 model parameters. The preprocessing of the data was done by rescaling the input elements into log scale. Then, the data were normalized in the range of -1 to +1. This preprocessing method was selected as it provided the best network predictions in Phase II.

In this phase, 2 numerical experiments were conducted. According to the suggestion of Maier and Dandy (1996), the initial weights were normalized between -0.1 and +0.1. Thus, Experiment No. 1 was done without normalizing the initial weights, while the initial weights were normalized between -0.1 and +0.1 in Experiment No. 2.

Table 4-4 and Table 4-5 show the mean square error values and the correlation coefficients from Experiment No. 1 and No. 2, respectively. The results shown in the tables are the representatives of the 10 network architectures. The representatives were selected based on the best prediction compared with 10 initializations within the same network architecture. However, the entire results of this phase are shown in Appendix B. In Appendix B, the results of the 1- and 2-hidden-layer FBNNs from Experiment No. 1 are shown in Table B-3 and Table B-4, respectively. Table B-5 and Table B-6 show the results from Experiment No. 2.

Fig. 4-2 and Fig. 4-3 illustrate the graphical output of the 1- and 2-hidden-layer FBNNs, respectively. The network predictions were plotted versus the targets as dots. The 6 solid lines indicate the boundaries of 5%, 10% and 20% errors from each desired point. The figures are the networks that provided the best prediction within the same network architecture. The mean square errors and the correlation coefficients of those networks are shown in Table 4-4 and Table 4-5. The percentages of the number of data points in each of the percent error boundaries are described in Table 4-6 and Table 4-7.

When comparing the results of the same number of hidden layers, the results shown in the tables below and illustrated in the figures demonstrate that by increasing the number of neurons, in other words by increasing the number of network parameters, the networks provided lower mean square errors in the testing data set as well as closer to 1 correlation coefficients. However, the 2-hidden-layer FBNN provided more accurate predictions even when the number of the network parameters was smaller. For example, the 2-hidden-layer FBNN, containing one layer with 10 and the other with 5 neurons (131 parameters), predicted the C/C_0 more accurately than the 1-hidden-layer FBNN with 24 neurons (193 parameters).

In addition, normalizing the initial weight in range of -0.1 to +0.1 before training network (Experiment No. 2), the C/C_0 prediction did not present a much better result than when the network that was trained without normalizing the initial weight (Experiment No. 1).

However, even if the prediction might be more accurate when the number of layers and parameters are higher, the experiment in this phase was stopped at the 2-hidden-layer FBNN, containing one layer with 24 and the other with 12 neurons, due to the satisfaction of reaching the optimum of 3 criterion which are low mean square error, close to 1 correlation coefficient, and high percentage of number of data point in 5% error boundary.

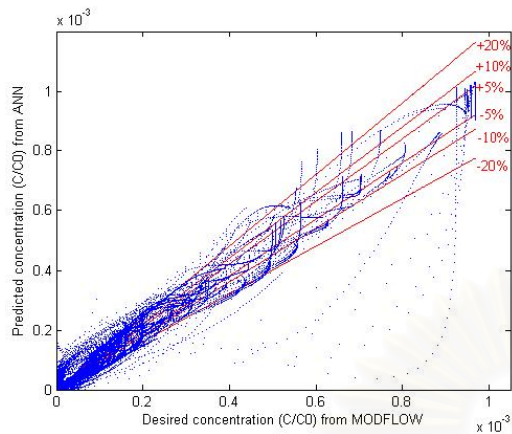
Because of the reason above and the results shown in the tables and figures, the 2-hidden-layer FBNN, containing one layer with 24 and the other with 12 neurons, without rescaling the initial weight in range of -0.1 to +0.1 can be defined as the best network as it provided the most accurate prediction in this research.

Table 4-4. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase III: Numerical Experiment No. 1. The initial weights were not normalized.

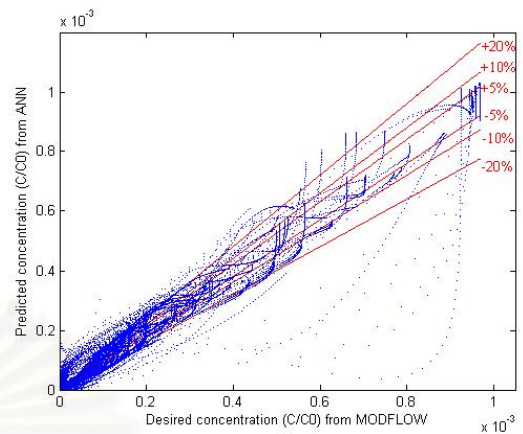
No. of neurons at each hidden layer	No. of parameters	MSE of testing data	Correlation Coefficient
<i>Network with 1 hidden layer</i>			
[6]	49	1.39E-09	0.98006
[12]	97	6.37E-10	0.99094
[24]	193	2.01E-10	0.99716
[36]	289	1.06E-10	0.99850
[48]	385	6.72E-11	0.99905
[60]	481	4.91E-11	0.99931
<i>Network with 2 hidden layers</i>			
[6 3]	67	1.36E-10	0.99807
[10 5]	131	4.57E-11	0.99935
[16 8]	257	1.31E-11	0.99981
[20 10]	361	1.24E-11	0.99982
[24 12]	481	6.29E-12	0.99991

Table 4-5. The mean square error and the correlation coefficient between the target of the testing data and the network prediction of Phase III: Numerical Experiment No. 2. The initial weights were normalized.

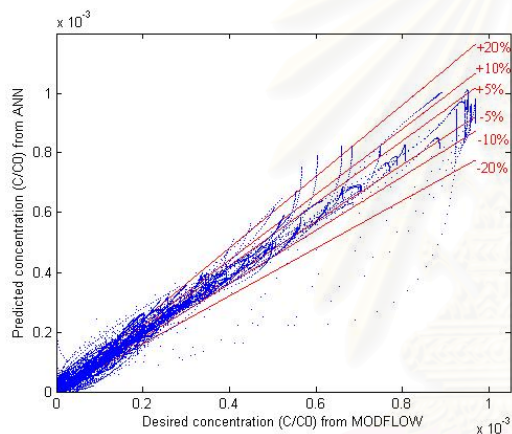
No. of neurons at each hidden layer	No. of parameters	MSE of testing data	Correlation Coefficient
<i>Network with 1 hidden layer</i>			
[6]	49	1.39E-09	0.98006
[12]	97	5.83E-10	0.99168
[24]	193	1.85E-10	0.99737
[36]	289	1.16E-10	0.99836
[48]	385	7.44E-11	0.99895
[60]	481	5.93E-11	0.99916
<i>Network with 2 hidden layers</i>			
[6 3]	67	1.24E-10	0.99825
[10 5]	131	3.81E-11	0.99946
[16 8]	257	1.04E-11	0.99985
[20 10]	361	7.58E-12	0.99989
[24 12]	481	8.49E-12	0.99988



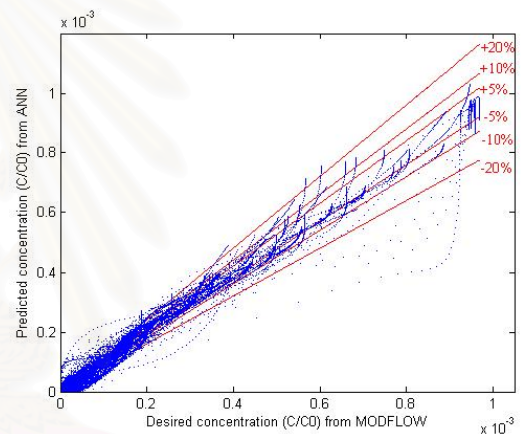
(a). [6]



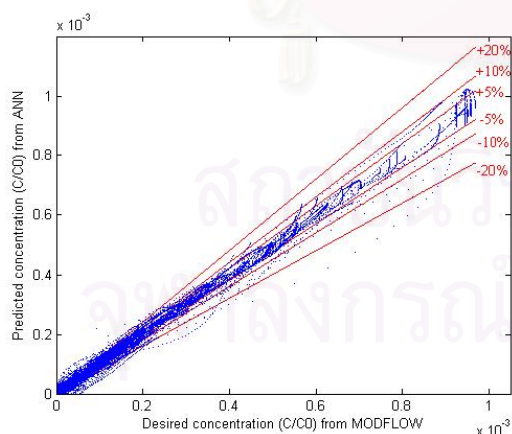
(b). [6]



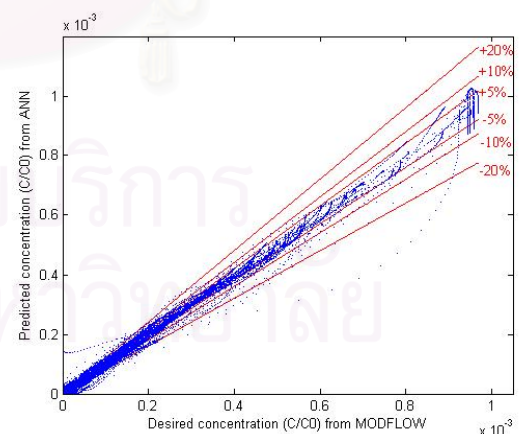
(a). [12]



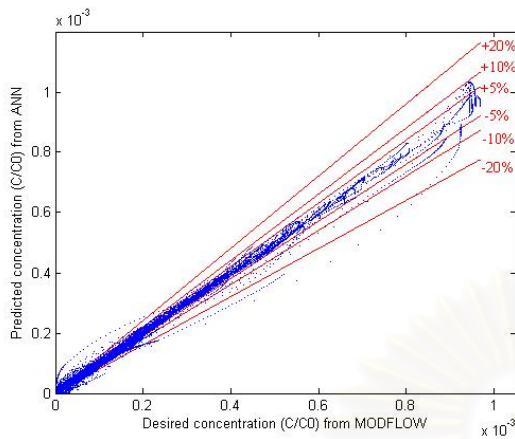
(b). [12]



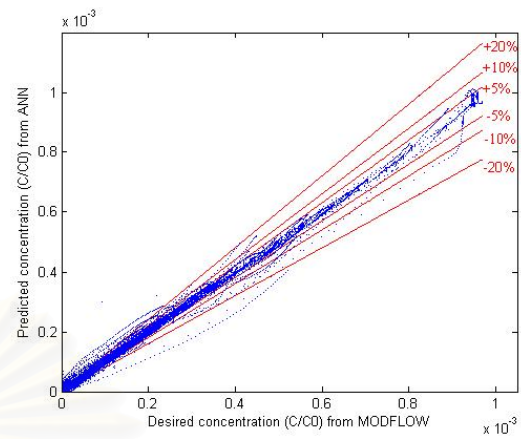
(a). [24]



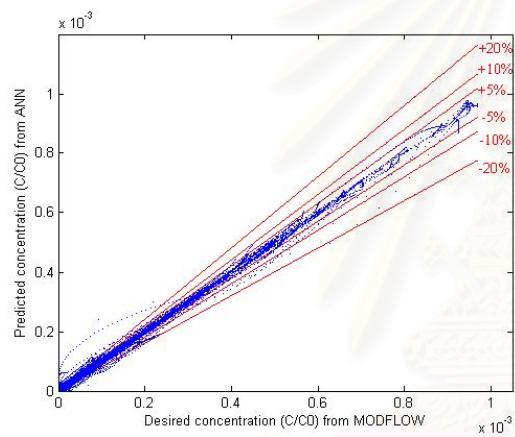
(b). [24]



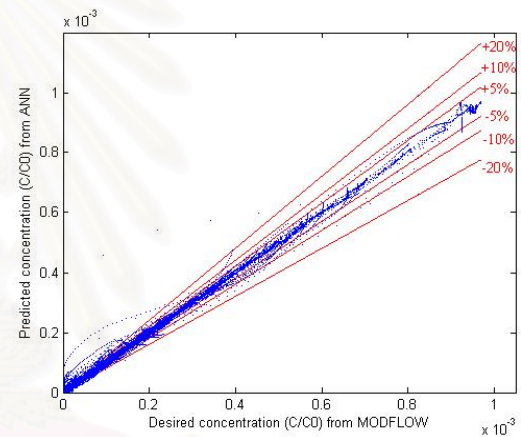
(a). [36]



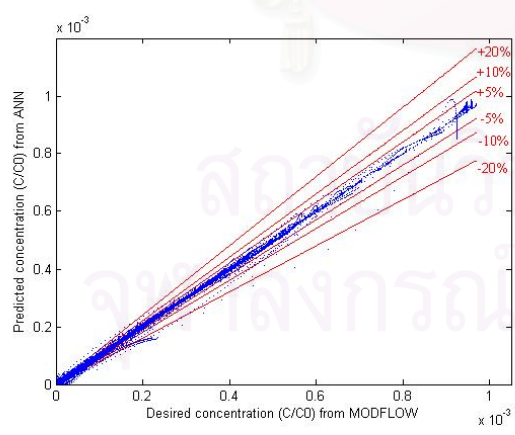
(b). [36]



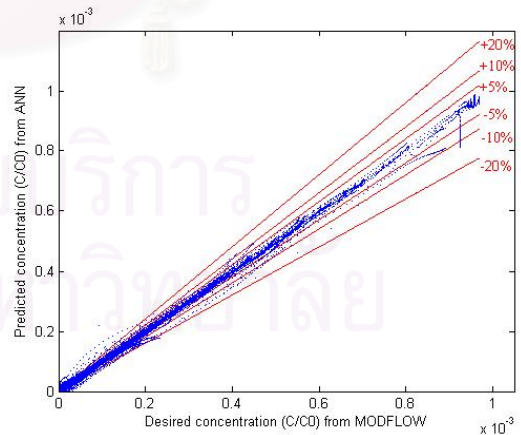
(a). [48]



(b). [48]



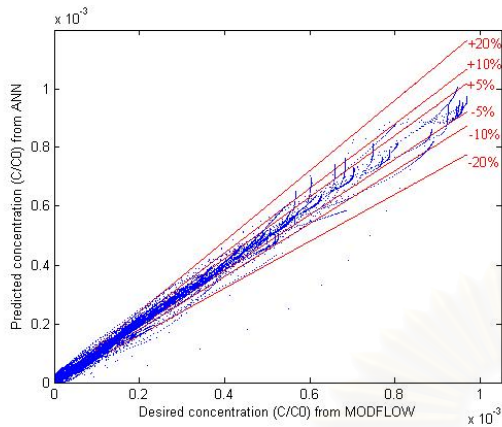
(a). [60]



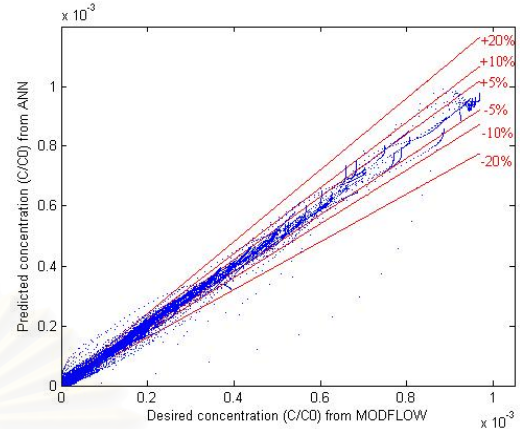
(b). [60]

Fig. 4-2. The plot of the desired C/C_0 versus the predicted C/C_0 simulated from the 1-hidden-layer FBNNs. The number in brackets indicate the number of neurons.

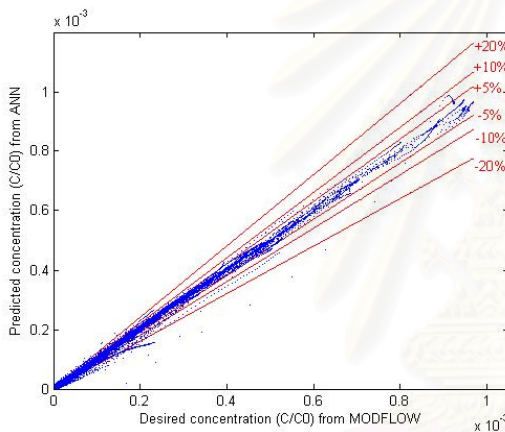
(a). Experiment No. 1 and (b). Experiment No. 2



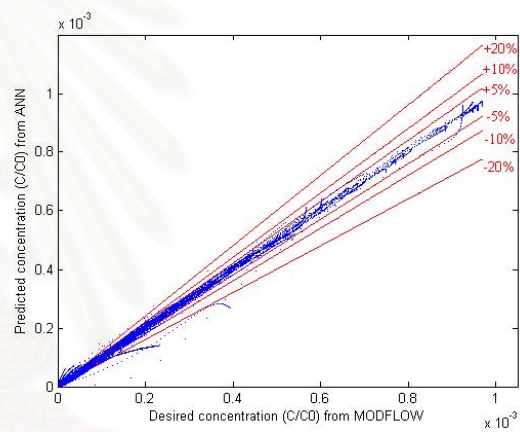
(a). [6 3]



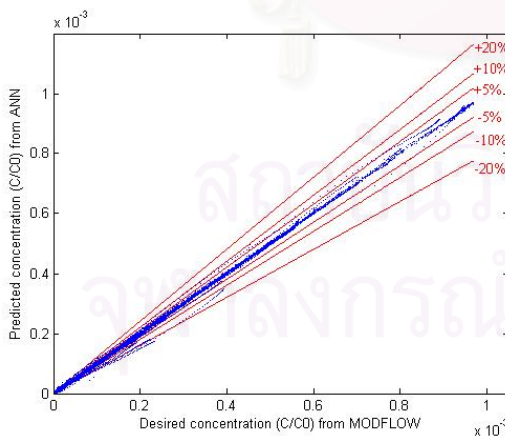
(b). [6 3]



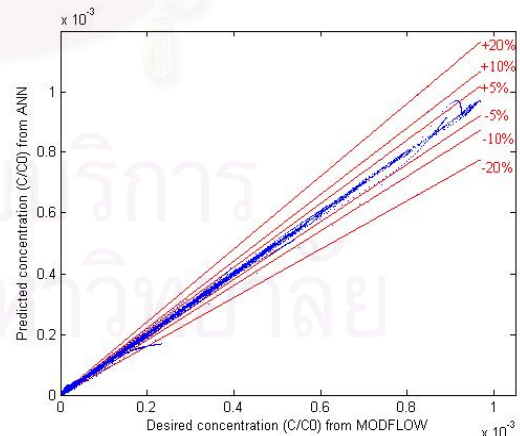
(a). [10 5]



(b). [10 5]



(a). [16 8]



(b). [16 8]

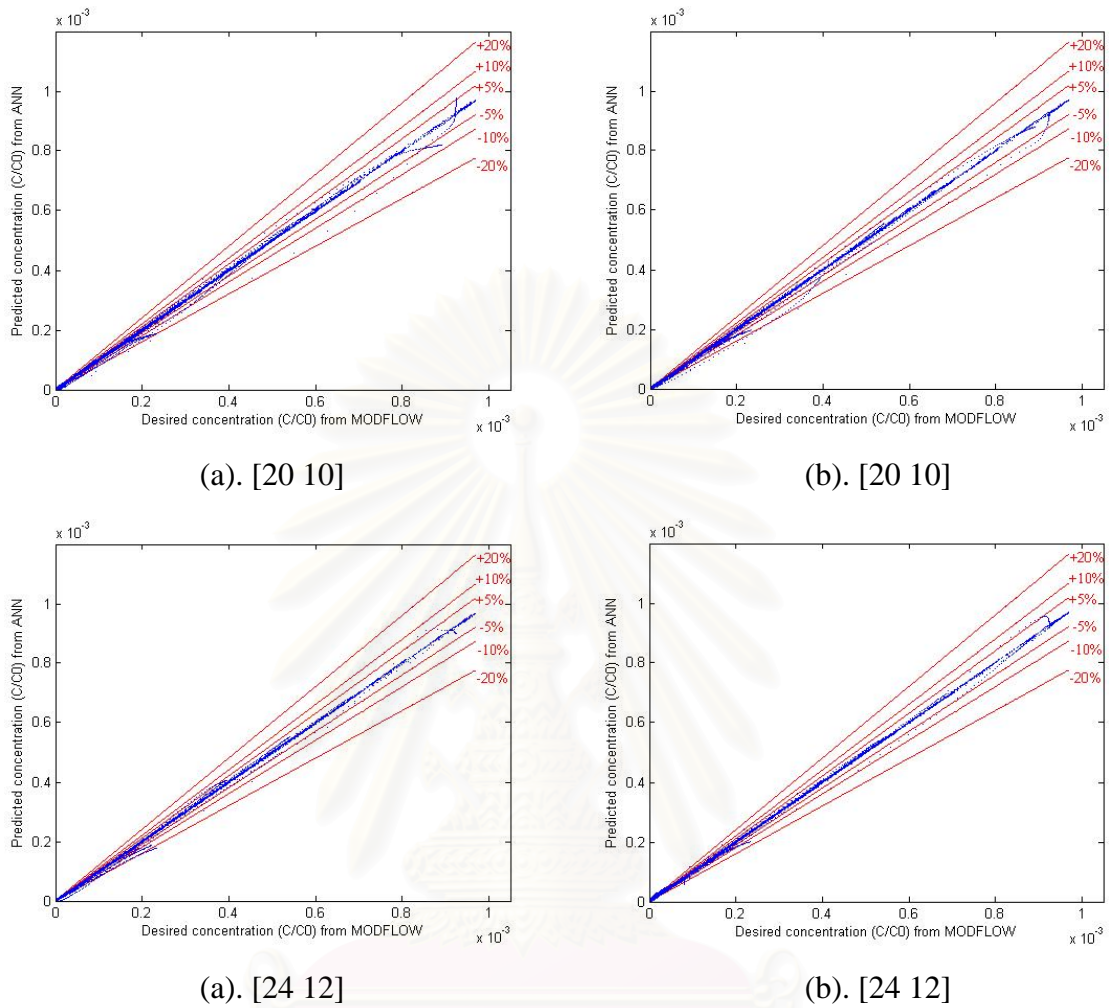


Fig. 4-3. The plot of the desired C/C_0 versus the predicted C/C_0 simulated from the 2-hidden-layer FBNNs. The numbers in brackets are the number of neurons in the first and second layers, respectively.

(a). Experiment No. 1 and (b). Experiment No. 2

Table 4-6. The percentage of the number of data points in the error boundaries
(Experiment No. 1)

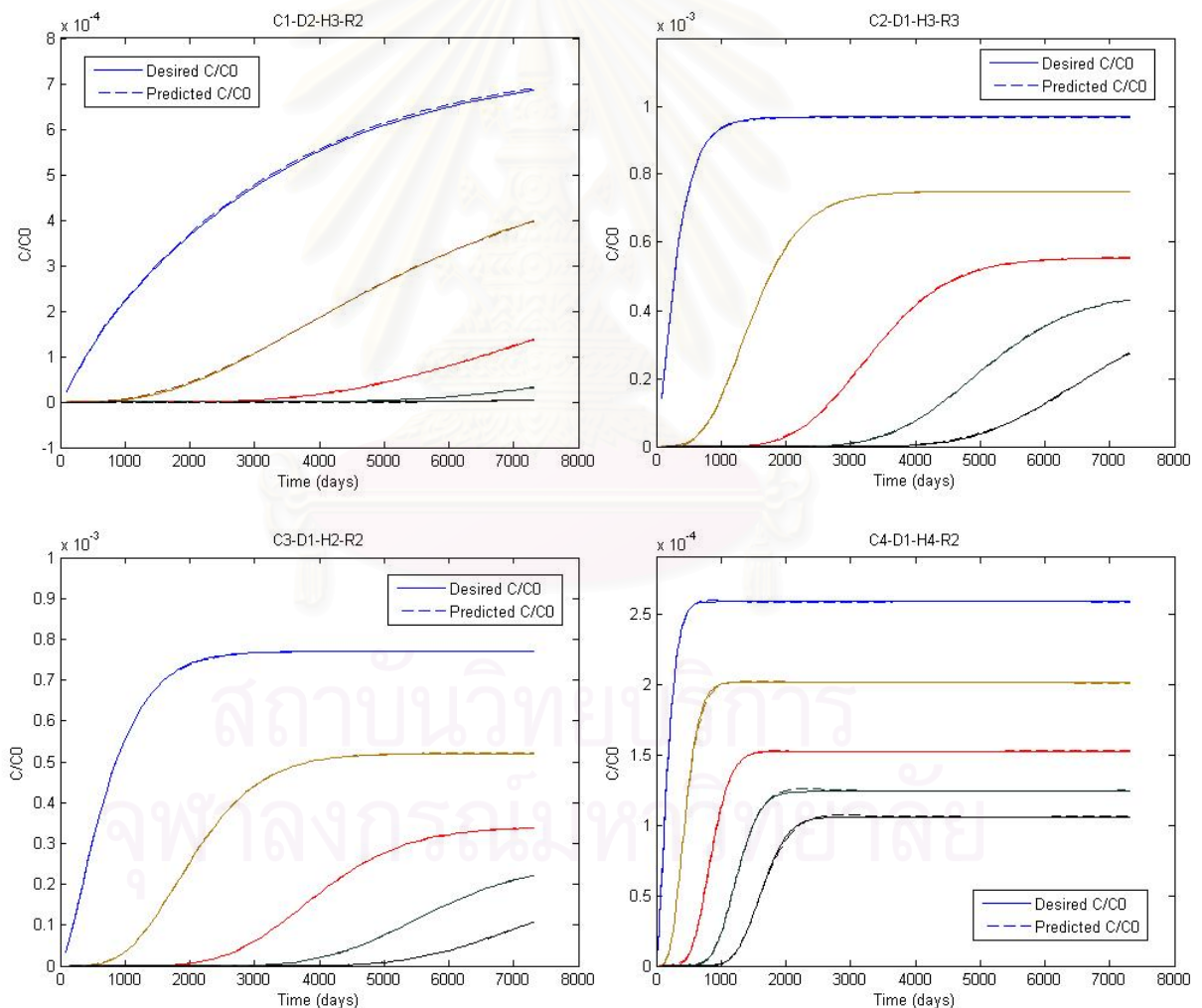
No. of neurons at each hidden layer	No. of parameters	The percent of the number of data points in the error boundaries			
		$\pm 5\%$	$\pm 10\%$	$\pm 20\%$	$> \pm 20\%$
<i>Network with 1 hidden layer</i>					
[6]	49	13.37	11.20	12.73	62.69
[12]	97	20.22	12.19	13.66	53.93
[24]	193	27.39	14.44	11.88	46.28
[36]	289	38.92	10.98	8.18	41.92
[48]	385	42.47	10.74	7.08	39.71
[60]	481	46.29	7.84	6.71	39.16
<i>Network with 2 hidden layers</i>					
[6 3]	67	35.23	13.84	11.27	39.66
[10 5]	131	49.35	11.97	7.84	30.84
[16 8]	257	59.35	8.13	7.94	24.58
[20 10]	361	61.54	9.70	6.64	22.12
[24 12]	481	69.40	8.19	4.44	17.97

Table 4-7. The percentage of the number of data points in the error boundaries
(Experiment No. 2)

No. of neurons at each hidden layer	No. of parameters	The percent of the number of data points in the error boundaries			
		$\pm 5\%$	$\pm 10\%$	$\pm 20\%$	$> \pm 20\%$
<i>Network with 1 hidden layer</i>					
[6]	49	13.37	11.20	12.73	62.69
[12]	97	19.50	12.38	11.63	56.48
[24]	193	30.81	15.04	11.18	42.97
[36]	289	37.57	11.71	9.59	41.13
[48]	385	40.70	11.19	8.36	39.76
[60]	481	44.68	8.73	8.57	38.02
<i>Network with 2 hidden layers</i>					
[6 3]	67	38.31	15.73	8.66	37.31
[10 5]	131	50.42	10.99	8.83	29.76
[16 8]	257	61.08	9.72	8.22	20.98
[20 10]	361	64.76	9.59	6.95	18.70
[24 12]	481	68.77	8.13	6.75	16.34

4.4 Illustration of Breakthrough Curve

In this section, some of the breakthrough curves of the 62 sites in the testing data set are presented. Each figure represents each site, and the graphs were plotted for the 5 observation wells. The solid line is the desired C/C_0 given from MODFLOW, whereas the dashed line is the C/C_0 predicted from the 2-hidden-layer FBNN, containing one layer with 24 and the other with 12 neurons, trained by using Bayesian Regularization.



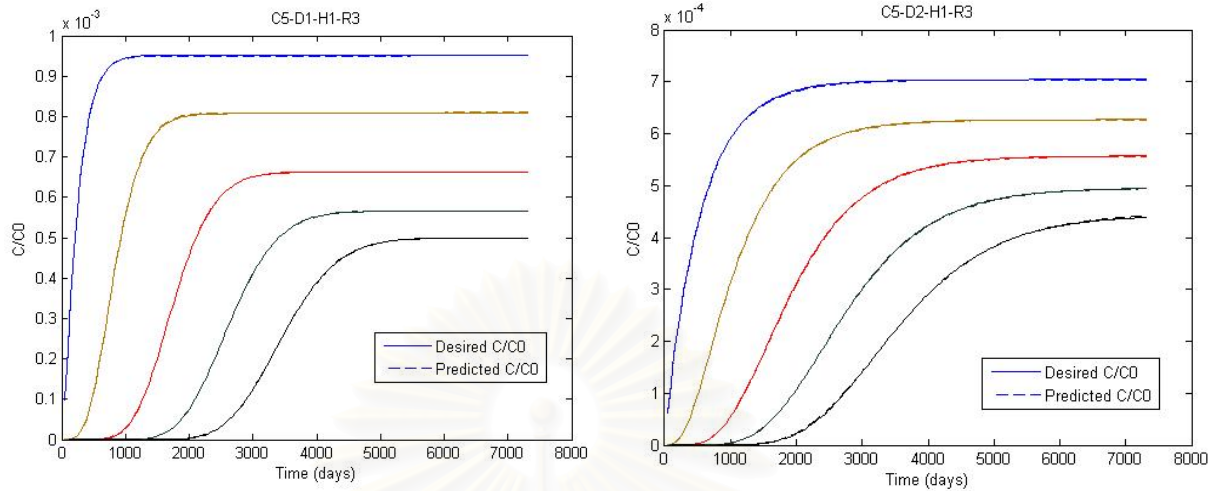
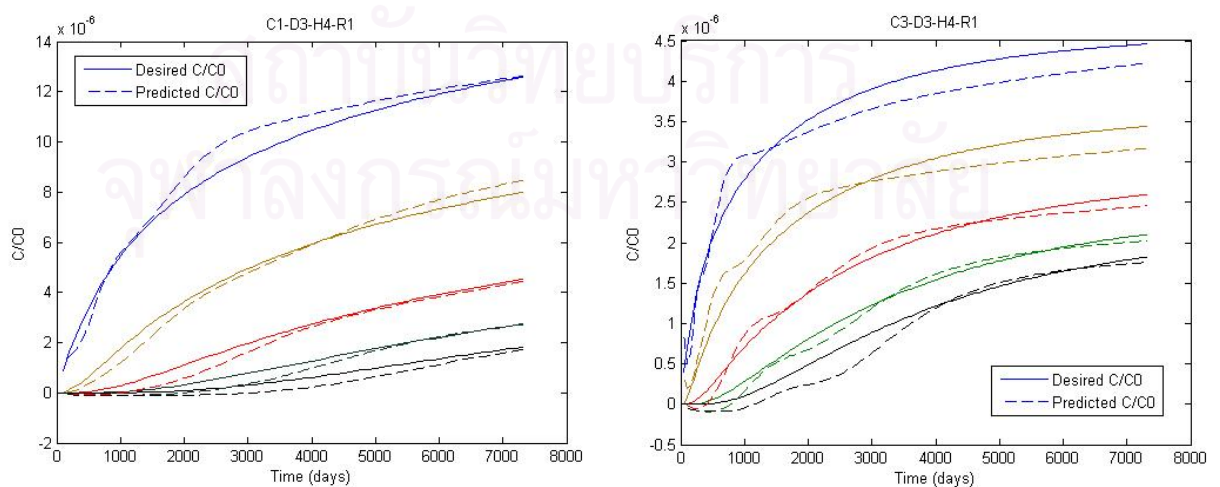


Fig. 4-4. Breakthrough curves of the 5 observation wells of each site

The figures above illustrate some of the sites that had the best predictions of C/C_0 from the ANN. However, the predictions of the sites whose maximum C/C_0 were low were not fairly accurate as shown below. The cause of this decreased accuracy of prediction at low C/C_0 is not clear yet. In fact, the C/C_0 of those sites were found to be mostly under 10^{-5} and the concentration at a particular time (C) might be even lower under 10^{-5} if the initial concentration (C_0) involves with a very low number. Because of this reason, the less accurate prediction at low concentration is perhaps insignificant.



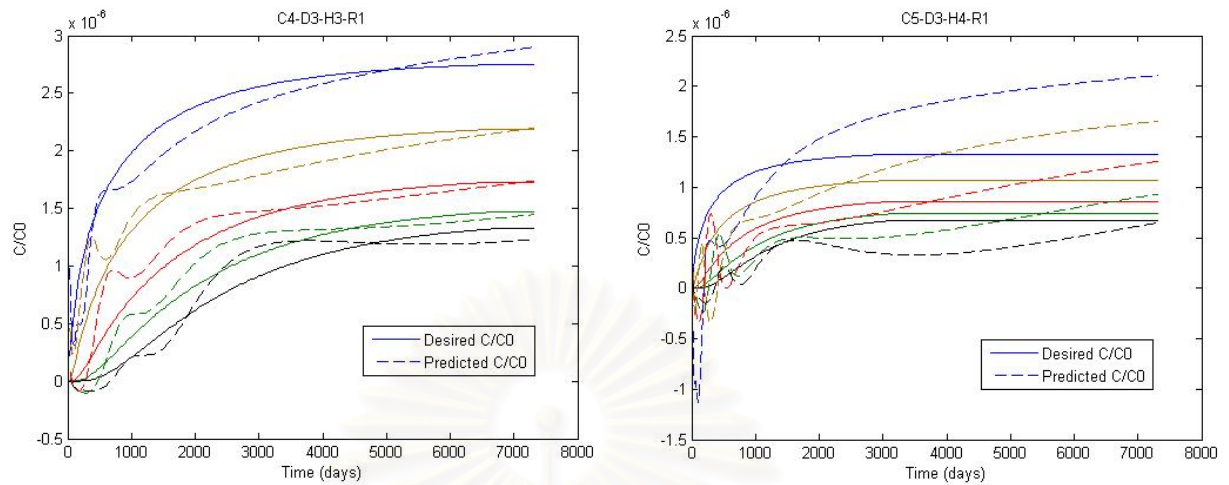


Fig. 4-5. Some examples of the less accurate predictions

4.5 Practical Application of a Developed Neural Network

After identifying the best neural network which provided the most accurate prediction of contamination migration (Phase III), a Graphic User Interface (GUI) was developed by using the MATLAB computer program. It is practical for a user to use. Moreover, the result can be obtained immediately within a few second. Within this application, only 5 contaminant migration parameters need to be entered. The 5 parameters are the hydraulic conductivity, longitudinal dispersivity, constant head difference, recharge rate at the source of the contamination, and distance from the source of the contamination. The best network determined from Phase III is used to predict the C/C_0 from the 5 input parameters. Then, the breakthrough curve of C/C_0 versus time is plotted. Moreover, the C/C_0 at a particular time can be calculated by giving the time parameter (see Fig. 4-6).

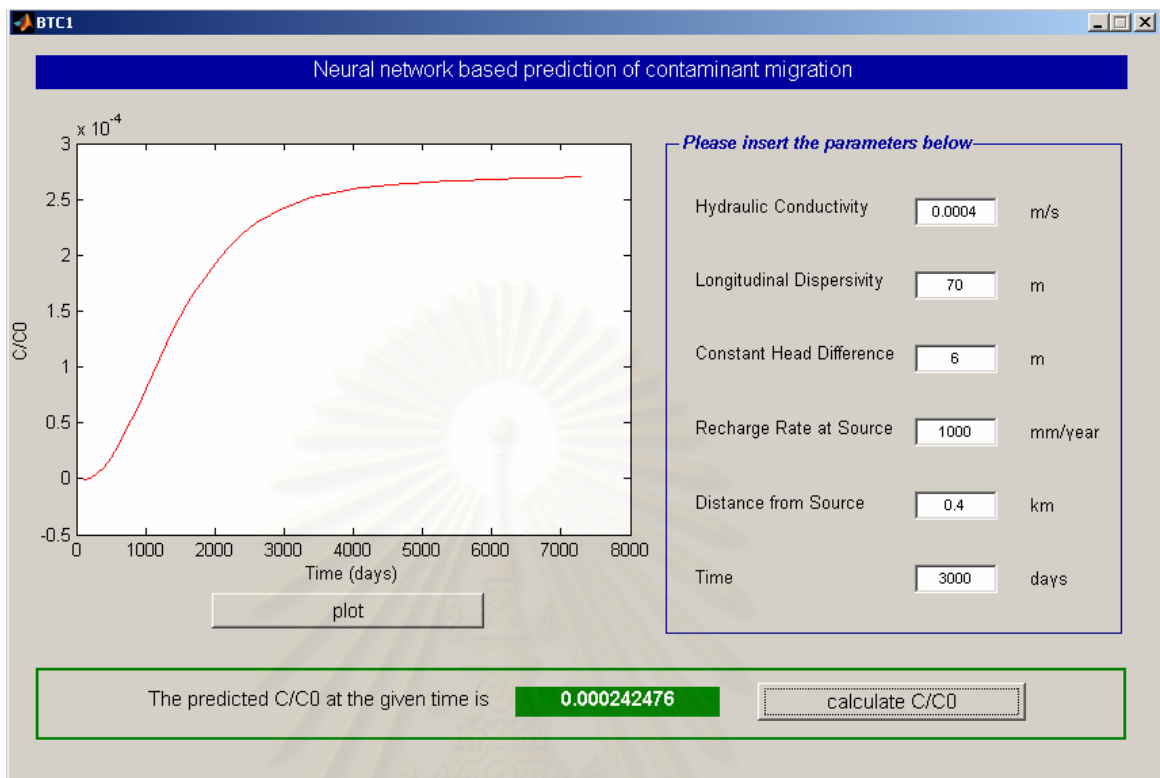


Fig. 4-6. Graphic User Interface

4.6 Efficiency of ANN applications in different scenarios

The 2-hidden-layer FBNN, containing one layer with 24 and the other with 12 neurons, was applied to predict the concentrations in different scenarios. It was found that when the values of the entered parameters were out of the range that was used to train the network, the network produced less accurate results. Moreover, when there was more than 1 type of soil, the soil profile changed and the predictions were less accurate. These are the limitations of applying this approach to scenarios that are different from the scenarios used to train the network.

CHAPTER V

SUMMARY AND SUGGESTION

5.1 Summary

This research studied the feasibility of using an artificial neural network to predict a breakthrough curve of concentration as a function of time without actually running the complicated 3-dimensional numerical simulations. Within this approach, only a set of data is need for training the ANN. There were no fundamental equations of groundwater flow and contaminant transport involved in this work. The network is trained and it learns from training input data and their corresponding targets. The database of spatial and temporal variations of typical contaminant concentrations for typical subsurface conditions was created by using the Visual MODFLOW program. The database contained 180 sites, which were created from the combination of 4 parameters: hydraulic conductivity, longitudinal dispersivity, hydraulic head difference, and recharge rate at the source of the contamination. Then, the database was divided into the training and the testing data sets. By random selection (as described in Section 3.4), the training data set consisted of 118 sites, whereas the testing data set consisted of 62 sites. A feedforward backpropagation neural network was used in this study.

Three phases were conducted. In Phase I, the 12 training functions provided by MATLAB were used to train the network. The results showed that training the network using the Bayesian Regularization technique ('trainbr') as a training function provided more accurate C/C_0 predictions with lower mean square errors and closer to 1 correlation coefficients in the testing data set. Moreover, this technique helped to avoid overfitting and improved generalization.

Phase II was conducted to determine the most appropriate method for preprocessing the data. Three numerical experiments were conducted. In the Numerical Experiment No. 1, the network was trained using the training data without

preprocessing. In the next 2 numerical experiments, the data was preprocessed by normalization in range of -1 to +1. The difference between Experiment No. 2 and No. 3 is that in Experiment No. 3, only the input elements (not the target elements) were rescaled into log scale before normalization. The target elements can not be rescaled into log scale because it contained minus values. The results of the comparison between the 3 numerical experiments demonstrated that the network trained using the preprocessed data provided lower mean square errors and closer to 1 correlation coefficients than the network trained using data without preprocessing. However, the prediction of C/C_0 was more accurate when rescaling the input elements into log scale and normalizing the data in the range of -1 to +1.

In the last phase, the 1- and 2-hidden-layer FBNNs with various numbers of neurons were trained. Obviously, the 2-hidden-layer FBNNs predicted the concentrations more accurately than the single-layer FBNNs even when the numbers of parameters are equal. The means square error values of the testing data set decreased and the correlation coefficients were closer to 1 when the number of network parameters increased. Finally, a 2-hidden-layer FBNNs, containing one layer with 24 and the other with 12 neurons, provided the best prediction of the contaminant concentrations. However, the predictions of the sites whose maximum C/C_0 were under 10^{-5} were not fairly accurate. The cause of this decreased accuracy of prediction at low C/C_0 was not determined.

After identifying the network which provided the best C/C_0 prediction (the result from Phase III), a Graphic User Interface (GUI) was developed. It is practical for users to use. Only the values of the 5 model parameters (i.e. hydraulic conductivity, longitudinal dispersivity, constant head difference, recharge rate at the source of the contamination, and distance from the source) need to be known in order for the program to plot the breakthrough curve of C/C_0 versus time.

5.2 Suggestions

The use of the ANN to predict the contaminant migration in this study was deemed feasible. However, this was just the first step. The database for training the ANN was created from a non-complexed hypothetical site. Moreover, only

hydrodynamic process (i.e. groundwater flow and transport parameters) were concerned without physicochemical and microbiological processes such as chemical reaction, adsorption, biodegradation, etc. In a real situation, the physical, chemical, or profile of soil might be a lot more complex. There is not only one type or one layer of soil as in the hypothetical site using in this study, but multiple, as well as chemical phenomena might occur. Before applying the ANN in a real situation, further study is necessary. The next step might need to add more input elements to ANN. For example, the input elements might include the depth and the hydraulic conductivity at that depth when applying ANN with sites containing more than 1 type of soil in different layers as well as some physiochemical or microbiological parameters. However, it is quite possible for it to be successful in the next step. In order to have more complex and realistic study site, the database can be from both data collected from the field together with data generated from MODFLOW or another commercial program used for predicting groundwater contaminant migration.

If the next step of its use with a more realistic phenomena and contaminant conditions is successful, this ANN decision making tool can be deemed a reliable screening tool for estimating contaminant migration. It is rapid since less time is needed and easy to use since a few parameters are required to create the concentration as a function of time. It can also be applied to predict concentration of contaminant for other similar sites or areas without necessity to know the boundary or some model conditions for creating a new site as the traditional groundwater models always require. Moreover, on environmental management point of view, this approach is useful for identifying critical sites and prioritizing the clean-up and further investigative efforts when the contaminations in groundwater occur.

REFERENCES

- Bachmat, Y. and Bear, J. (1964). The general equations of hydrodynamic dispersion in the homogeneous isotropic porous medium. J. Geophys. Res 69(12) : 2561-2567.
- Bear, J. (1972). Dynamics of fluids in porous media. New York : American Elsevier.
- Bear, J. (1979). Hydraulics of groundwater. New York : McGraw-Hill.
- Beatsle, L. H. (1969). Migration of radionuclides in porous media. Progress in nuclear energy series XII, Health physics, 707-730. Elmsford, NY : Pergamon Press.
- Bedient, P. B., Rifai, H. S., and Newell, C. J. (1994). Groundwater contamination: Transport and remediation. Englewood Cliffs, NJ : Prentice-Hall.
- Bedient, P. B. and Wayne, C. H. (1992). Hydrology and floodplain analysis. 2nd ed. Massachusetts : Addison-Wesley Publishing Company.
- Bouwer, H. (1978). Groundwater hydrology. New York: McGraw-Hill.
- Brown. G. (1995). Darcy's law basics and more [online]. Available from: <http://biosystems.okstate.edu/Darcy/LaLoi/basics.htm> [cited 10 October 2006].
- Chester, D. R. (2000). Groundwater contamination. vol. 1: Contamination, source & hydrology. Lancaster, PA : Technomic Pub.
- Civil, Construction & Environmental Engineering, Oregon State University. (2006). Contaminant resources [online]. Available from: <http://ccee.oregonstate.edu/enve/igw/sources/index.php> [cited 10 October 2006].
- Edwards, K. (2000). 1-D Step injection: advection, disp, retardation [online]. Available from: <http://www.lmnoeng.com/Groundwater/transportStep.htm> [cited 10 October 2006].
- Edwards, K. (2000). 3-D Slug injection: advection, dispersion [online]. Available from: <http://www.lmnoeng.com/Groundwater/transportPulse.htm> [cited 10 October 2006].
- Environment Canada. (2004). Groundwater – Nature's hidden treasure [online]. Available from: http://www.ec.gc.ca/WATER/en/info/pubs/FS/e_FSA5.htm [cited 10 October 2006].
- Freeze, R. A. and Cherry, J. A. (1979). Groundwater. Englewood Cliffs, NJ : Prentice Hall.
- Geophysics research forum (U.S.), Geophysics study committee. (1984). Groundwater contamination. Commission on physical sciences, mathematics, and resources, National research council. Washington, D.C. : National Academy Press.
- Gurney, K. (1997). An introduction to neural network. London : UCL Press.
- Haykin, S. (1994). Neural network: A comprehensive foundation. New York : IEEE Press
- Illinois State Geological Survey. (2004). The hydrologic cycle [online]. Available from: <http://www.isgs.uiuc.edu/isgshome/hydrocycle.htm> [cited 10 October 2006].

- Liu, D. H. F. and Liptak, B. G. (2000). Groundwater and surface water pollution. New York : CRC Press.
- MacKay, D. J. C. (1992). Bayesian interpolation. Neural computation 4(3) : 415-447.
- Maier, R. H. and Dandy, C. G. (1996). The use of artificial neural networks for the prediction of water quality parameters. Water resource research 32(4) : 1013-1022.
- MATLAB, Version 7.1.0.246 (R14). (2005). Neural network toolbox: User manual. The math works.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas imminent in nervous activity. Bulletin of mathematical biophysics 5 : 115-133.
- Morshed, J. and Kaluarachchi, J. J. (1998). Application of artificial neural network and genetic algorithm in flow and transport simulations. Advances in water resources 22(2) : 145-158.
- Pye, V. I., Patrick, R., and Quarles, J. (1983). Groundwater contamination in the United States. Philadelphia, PA : University of Pennsylvania. Press.
- Portneuf Valley Groundwater Guardian. (2001). Aquifer protection [online]. Available from:
http://www.idahogeology.org/Services/Hydrogeology/PortneufGroundWaterGuardian/my_drinking_water/aquifer_protect/protect.html [cited 10 October 2006].
- Ranjithan, S., Eheart, J. W., and Garrett, J. H. (1993). Neural network-based screening for groundwater reclamation under uncertainty. Water resources research 29(3) : 563-574.
- Roger, L. L. and Dowla, F. U. (1994). Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water resources research 30(20) : 457-481.
- Rumelhart, D. E., McClelland, J. L., and the PDP research group. (1986). Parallel distributed processing. vol. 1: Explorations in the microstructure of cognition. Cambridge, MA : MIT Press.
- Singh, R. M., Datta, B., and Jain, A. (2004). Identification of unknown groundwater pollution sources using artificial neural networks. Journal of water resources planning and management 130(6) : 506-514.
- Spiegel, M. R. (1992). Correlation Theory. Theory and problems of probability and statistic. 2nd ed., 294-323. New York : McGraw-Hill.
- Van Genuchten, M. Th. and Alves, W. J. (1982). Analytical solutions of the one-dimensional convective-dispersive solute transport equation. Washington, D.C : United States Department of Agriculture.
- MODFLOW, Version 2.8.2. (1999). Visual MODFLOW Tutorial. Waterloo Hydrogeologic Inc.
- Werbos, P.J. (1974/1994). The Roots of Backpropagation. New York: John Wiley & Sons.



APPENDICES

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

APPENDIX A

Table A-1: The list of the sites in the training data set

C1-D1-H1-R1	C2-D1-H1-R2	C3-D1-H1-R1	C4-D1-H1-R1	C5-D1-H1-R1
C1-D1-H1-R2	C2-D1-H2-R1	C3-D1-H2-R3	C4-D1-H1-R2	C5-D1-H1-R2
C1-D1-H2-R2	C2-D1-H2-R2	C3-D1-H3-R1	C4-D1-H1-R3	C5-D1-H2-R1
C1-D1-H2-R3	C2-D1-H3-R1	C3-D1-H3-R2	C4-D1-H2-R1	C5-D1-H2-R3
C1-D1-H3-R2	C2-D1-H4-R2	C3-D1-H3-R3	C4-D1-H2-R2	C5-D1-H3-R1
C1-D1-H4-R2	C2-D1-H4-R3	C3-D1-H4-R2	C4-D1-H2-R3	C5-D1-H3-R2
C1-D1-H4-R3	C2-D2-H1-R1	C3-D1-H4-R3	C4-D1-H3-R1	C5-D1-H3-R3
C1-D2-H1-R1	C2-D2-H1-R2	C3-D2-H1-R1	C4-D1-H3-R3	C5-D1-H4-R1
C1-D2-H1-R2	C2-D2-H1-R3	C3-D2-H1-R2	C4-D1-H4-R3	C5-D1-H4-R3
C1-D2-H2-R1	C2-D2-H2-R2	C3-D2-H2-R1	C4-D2-H1-R1	C5-D2-H2-R3
C1-D2-H2-R3	C2-D2-H3-R1	C3-D2-H2-R3	C4-D2-H1-R2	C5-D2-H3-R1
C1-D2-H3-R1	C2-D2-H3-R2	C3-D2-H3-R1	C4-D2-H1-R3	C5-D2-H3-R3
C1-D2-H3-R3	C2-D2-H3-R3	C3-D2-H3-R3	C4-D2-H2-R1	C5-D2-H4-R1
C1-D2-H4-R1	C2-D2-H4-R1	C3-D2-H4-R1	C4-D2-H2-R2	C5-D2-H4-R2
C1-D2-H4-R2	C2-D2-H4-R3	C3-D2-H4-R2	C4-D2-H2-R3	C5-D2-H4-R3
C1-D2-H4-R3	C2-D3-H1-R1	C3-D3-H1-R1	C4-D2-H4-R2	C5-D3-H1-R1
C1-D3-H1-R1	C2-D3-H1-R2	C3-D3-H1-R2	C4-D2-H4-R3	C5-D3-H1-R2
C1-D3-H1-R2	C2-D3-H2-R1	C3-D3-H1-R3	C4-D3-H1-R2	C5-D3-H2-R1
C1-D3-H1-R3	C2-D3-H2-R3	C3-D3-H2-R1	C4-D3-H2-R2	C5-D3-H2-R2
C1-D3-H2-R1	C2-D3-H3-R1	C3-D3-H2-R3	C4-D3-H2-R3	C5-D3-H2-R3
C1-D3-H2-R2	C2-D3-H3-R2	C3-D3-H3-R2	C4-D3-H3-R2	C5-D3-H3-R1
C1-D3-H3-R2	C2-D3-H3-R3	C3-D3-H3-R3	C4-D3-H4-R1	C5-D3-H3-R2
C1-D3-H3-R3	C2-D3-H4-R2	C3-D3-H4-R1	C4-D3-H4-R3	C5-D3-H4-R2
C1-D3-H4-R2	C2-D3-H4-R3			C5-D3-H4-R3

Table A-2: The list of the sites in the testing data set

C1-D1-H1-R3	C2-D1-H1-R1	C3-D1-H1-R2	C4-D1-H3-R2	C5-D1-H1-R3
C1-D1-H2-R1	C2-D1-H1-R3	C3-D1-H1-R3	C4-D1-H4-R1	C5-D1-H2-R2
C1-D1-H3-R1	C2-D1-H2-R3	C3-D1-H2-R1	C4-D1-H4-R2	C5-D1-H4-R2
C1-D1-H3-R3	C2-D1-H3-R2	C3-D1-H2-R2	C4-D2-H3-R1	C5-D2-H1-R1
C1-D1-H4-R1	C2-D1-H3-R3	C3-D1-H4-R1	C4-D2-H3-R2	C5-D2-H1-R2
C1-D2-H1-R3	C2-D1-H4-R1	C3-D2-H1-R3	C4-D2-H3-R3	C5-D2-H1-R3
C1-D2-H2-R2	C2-D2-H2-R1	C3-D2-H2-R2	C4-D2-H4-R1	C5-D2-H2-R1
C1-D2-H3-R2	C2-D2-H2-R3	C3-D2-H3-R2	C4-D3-H1-R1	C5-D2-H2-R2
C1-D3-H2-R3	C2-D2-H4-R2	C3-D2-H4-R3	C4-D3-H1-R3	C5-D2-H3-R2
C1-D3-H3-R1	C2-D3-H1-R3	C3-D3-H2-R2	C4-D3-H2-R1	C5-D3-H1-R3
C1-D3-H4-R1	C2-D3-H2-R2	C3-D3-H3-R1	C4-D3-H3-R1	C5-D3-H3-R3
C1-D3-H4-R3	C2-D3-H4-R1	C3-D3-H4-R2	C4-D3-H3-R3	C5-D3-H4-R1
		C3-D3-H4-R3	C4-D3-H4-R2	

APPENDIX B

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from ‘site_*’ (all the sites in Table 3-11 except for ‘site_base’) and the testing data set was created from ‘site_base.’

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
traingd	1	7.30E-07	NaN
	2	8.13E-10	NaN
	3	2.62E-06	NaN
	4	5.95E-08	NaN
	5	2.86E-07	NaN
	6	5.14E-10	NaN
	7	7.67E-09	NaN
	8	7.36E-08	NaN
	9	1.69E-09	-4.71E-02
	10	1.98E-07	NaN
traingdm	1	9.13E-07	NaN
	2	2.81E-06	NaN
	3	7.53E-06	NaN
	4	1.78E-05	1.00E-17
	5	6.73E-07	NaN
	6	2.87E-06	NaN
	7	2.53E-06	NaN
	8	1.12E-06	-5.53E-02
	9	5.21E-04	5.53E-02
	10	9.22E-07	NaN

Note: NaN = Not-a-Number

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from ‘site_*’ (all the sites in Table 3-11 except for ‘site_base’) and the testing data set was created from ‘site_base.’ (*continued*)

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
traingdx	1	9.14E-10	3.72E-02
	2	8.98E-10	-3.00E-02
	3	7.92E-08	NaN
	4	9.04E-10	NaN
	5	3.58E-01	4.41E-02
	6	9.10E-10	NaN
	7	3.46E-06	-3.08E-02
	8	4.74E-03	3.32E-02
	9	4.00E-02	-3.92E-02
	10	9.01E-10	NaN
trainrp	1	9.06E-10	NaN
	2	8.94E-10	NaN
	3	8.67E-10	NaN
	4	8.92E-10	NaN
	5	9.05E-10	NaN
	6	9.02E-10	NaN
	7	8.92E-10	NaN
	8	9.01E-10	NaN
	9	8.89E-10	NaN
	10	8.94E-10	4.90E-02

Note: NaN = Not-a-Number

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from 'site_*' (all the sites in Table 3-11 except for 'site_base') and the testing data set was created from 'site_base.' (*continued*)

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
traincgf	1	8.62E-10	4.16E-02
	2	2.61E-07	1.07E-03
	3	1.44E-09	-3.73E-02
	4	3.54E-02	4.03E-01
	5	7.09E-10	6.32E-02
	6	3.18E-02	5.70E-01
	7	1.01E-09	5.70E-02
	8	4.53E-01	1.70E-02
	9	2.67E-08	5.34E-02
	10	7.55E-02	9.03E-02
traincgp	1	8.69E-09	1.27E-02
	2	7.13E-03	-1.01E-01
	3	8.40E-02	4.81E-01
	4	5.22E-02	-1.41E-01
	5	8.80E-10	4.97E-02
	6	4.53E-02	2.11E-03
	7	3.33E-02	5.06E-01
	8	9.11E-10	4.72E-02
	9	5.93E-10	-5.42E-02
	10	9.59E-10	-4.73E-02

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from 'site_*' (all the sites in Table 3-11 except for 'site_base') and the testing data set was created from 'site_base.' (*continued*)

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
traincgb	1	1.23E-09	-4.87E-02
	2	1.18E-02	3.01E-01
	3	4.85E-10	4.28E-02
	4	1.15E-09	4.57E-02
	5	1.73E-08	-5.12E-02
	6	7.79E-10	3.33E-02
	7	8.22E-10	-5.02E-02
	8	9.00E-10	-4.31E-02
	9	6.68E-09	-5.32E-02
	10	1.51E-09	5.40E-02
trainscg	1	6.72E-04	-9.09E-02
	2	3.55E-03	-2.10E-01
	3	1.36E-09	2.81E-02
	4	1.53E-03	-1.28E-01
	5	1.13E-02	-8.76E-01
	6	1.22E-09	5.84E-02
	7	1.03E-09	6.70E-02
	8	1.01E-09	-4.97E-02
	9	9.28E-10	5.67E-02
	10	9.63E-10	4.20E-02

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from 'site_*' (all the sites in Table 3-11 except for 'site_base') and the testing data set was created from 'site_base.' (*continued*)

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
trainbfg	1	3.56E-08	-7.51E-01
	2	9.64E-10	5.50E-02
	3	8.81E-10	-3.94E-02
	4	8.86E-10	-4.13E-02
	5	9.08E-10	-5.50E-02
	6	9.12E-10	4.58E-02
	7	9.01E-10	NaN
	8	9.42E-10	-1.07E-02
	9	8.78E-10	-5.05E-02
	10	8.96E-10	4.01E-02
trainoss	1	9.00E-10	NaN
	2	9.00E-10	4.86E-02
	3	8.99E-10	-3.87E-02
	4	9.00E-10	NaN
	5	8.86E-10	-3.76E-02
	6	9.01E-10	-3.41E-02
	7	9.01E-10	NaN
	8	8.96E-10	-4.88E-02
	9	9.11E-10	3.36E-02
	10	9.00E-10	5.10E-02

Note: NaN = Not-a-Number

Table B-1: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained with different types of training functions. The training data set was created from 'site_*' (all the sites in Table 3-11 except for 'site_base') and the testing data set was created from 'site_base.' (*continued*)

Training Function	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
trainlm	1	6.02E-10	8.90E-01
	2	3.11E-11	9.74E-01
	3	5.27E-10	9.13E-01
	4	7.32E-10	8.84E-01
	5	4.10E-10	7.68E-01
	6	1.19E-09	2.53E-01
	7	3.39E-10	5.31E-01
	8	6.02E-10	6.22E-01
	9	4.14E-10	7.98E-01
	10	9.10E-10	8.58E-01
trainbr	1	1.30E-09	7.12E-01
	2	7.95E-10	6.41E-01
	3	7.40E-10	-5.45E-01
	4	1.79E-10	8.23E-01
	5	5.91E-10	8.40E-01
	6	2.06E-10	7.98E-01
	7	1.60E-09	8.46E-01
	8	4.97E-10	9.70E-01
	9	2.57E-10	9.42E-01
	10	4.61E-10	7.52E-01

Table B-2: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction from Phase I. The 2-hidden-layer FBNNs, containing one layer with 6 and the other with 3 neurons, were trained using training functions ‘trainlm’ and ‘trainbr’. The training and the testing data sets were created from the combination of the 4 model parameters.

Train Function	ith Initialization	MSE of testing data set	Correlation Coefficient
trainlm	1	2.40E-09	9.66E-01
	2	2.89E-09	9.58E-01
	3	3.91E-09	9.43E-01
	4	2.21E-09	9.69E-01
	5	2.14E-09	9.70E-01
	6	2.13E-09	9.69E-01
	7	2.54E-09	9.64E-01
	8	3.04E-09	9.56E-01
	9	2.45E-09	9.65E-01
	10	4.91E-09	9.30E-01
trainbr	1	1.02E-08	8.55E-01
	2	2.17E-09	9.69E-01
	3	4.99E-09	9.28E-01
	4	2.98E-09	9.57E-01
	5	3.48E-09	9.49E-01
	6	3.64E-09	9.48E-01
	7	4.19E-09	9.40E-01
	8	1.83E-09	9.74E-01
	9	2.10E-09	9.70E-01
	10	1.95E-09	9.72E-01

Table B-3: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the single-layer FBNNs with various numbers of neurons.

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
6	1	1.62E-09	0.97673
	2	2.10E-09	0.97120
	3	1.39E-09	0.98006
	4	2.24E-09	0.96769
	5	2.14E-09	0.96935
	6	2.39E-09	0.96576
	7	2.19E-09	0.96989
	8	1.74E-09	0.97500
	9	1.62E-09	0.97673
	10	5.19E-09	0.93491
12	1	9.98E-10	0.98575
	2	8.38E-10	0.98803
	3	1.45E-09	0.97941
	4	9.08E-10	0.98701
	5	8.60E-10	0.98773
	6	1.09E-09	0.98439
	7	7.95E-10	0.98868
	8	8.33E-10	0.98822
	9	1.13E-09	0.98395
	10	6.37E-10	0.99094

Table B-3: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the single-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
24	1	2.64E-10	0.99625
	2	2.42E-10	0.99655
	3	3.16E-10	0.99552
	4	2.36E-10	0.99669
	5	2.84E-10	0.99595
	6	2.06E-10	0.99708
	7	2.29E-10	0.99678
	8	3.26E-10	0.99536
	9	2.01E-10	0.99716
	10	3.57E-10	0.99491
36	1	1.39E-10	0.99803
	2	1.06E-10	0.99850
	3	1.12E-10	0.99840
	4	1.30E-10	0.99815
	5	1.23E-10	0.99827
	6	1.27E-10	0.99820
	7	1.29E-10	0.99818
	8	1.76E-10	0.99750
	9	1.28E-10	0.99818
	10	2.12E-10	0.99705

Table B-3: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the single-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
48	1	8.73E-11	0.99876
	2	9.54E-11	0.99865
	3	7.49E-11	0.99894
	4	7.88E-11	0.99888
	5	7.75E-11	0.99890
	6	7.49E-11	0.99894
	7	1.01E-10	0.99856
	8	7.97E-11	0.99887
	9	6.72E-11	0.99905
	10	7.77E-11	0.99890
60	1	8.57E-11	0.99879
	2	9.23E-11	0.99870
	3	5.13E-11	0.99927
	4	5.71E-11	0.99919
	5	8.19E-11	0.99884
	6	7.19E-11	0.99898
	7	7.30E-11	0.99896
	8	6.83E-11	0.99903
	9	8.18E-11	0.99884
	10	4.91E-11	0.99931

Table B-4: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the 2-hidden-layer FBNNs with various numbers of neurons.

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
6 and 3	1	1.45E-10	9.98E-01
	2	1.42E-10	9.98E-01
	3	2.06E-10	9.97E-01
	4	1.78E-10	9.97E-01
	5	1.81E-10	9.97E-01
	6	3.34E-10	9.95E-01
	7	3.09E-10	9.96E-01
	8	3.12E-10	9.96E-01
	9	2.76E-10	9.96E-01
	10	1.36E-10	9.98E-01
10 and 5	1	4.68E-11	0.99934
	2	2.16E-10	0.99693
	3	4.85E-11	0.99931
	4	4.57E-11	0.99935
	5	2.85E-10	0.99599
	6	2.71E-10	0.99616
	7	6.56E-11	0.99907
	8	7.53E-11	0.99893
	9	8.92E-11	0.99875
	10	1.22E-09	0.98416

Table B-4: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the 2-hidden-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
16 and 8	1	3.47E-11	0.99951
	2	4.80E-11	0.99932
	3	5.93E-11	0.99917
	4	2.08E-11	0.99971
	5	3.57E-11	0.99950
	6	1.31E-11	0.99981
	7	3.24E-11	0.99954
	8	2.87E-11	0.99960
	9	1.82E-11	0.99974
	10	1.59E-11	0.99977
20 and 10	1	3.61E-11	0.99949
	2	7.97E-11	0.99887
	3	2.58E-11	0.99964
	4	2.42E-11	0.99966
	5	2.73E-11	0.99962
	6	1.74E-11	0.99976
	7	1.66E-11	0.99976
	8	4.40E-11	0.99937
	9	1.43E-11	0.99980
	10	1.24E-11	0.99982

Table B-4: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 1). The results are from the 2-hidden-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
24 and 12	1	3.11E-11	0.99957
	2	1.09E-11	0.99985
	3	6.29E-12	0.99991
	4	7.18E-12	0.99990
	5	8.02E-11	0.99886
	6	1.02E-11	0.99986
	7	8.39E-12	0.99988
	8	6.50E-11	0.99908
	9	1.06E-11	0.99985
	10	3.57E-11	0.99950

Table B-5: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the single-layer FBNNs with various numbers of neurons.

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
6	1	1.92E-09	0.97287
	2	1.60E-09	0.97706
	3	2.14E-09	0.96929
	4	3.15E-09	0.95441
	5	1.39E-09	0.98006
	6	2.03E-09	0.97089
	7	2.05E-09	0.97156
	8	1.83E-09	0.97409
	9	2.05E-09	0.97156
	10	1.71E-09	0.97580
12	1	6.07E-10	0.99134
	2	1.07E-09	0.98492
	3	9.06E-10	0.98710
	4	1.30E-09	0.98150
	5	7.84E-10	0.98889
	6	7.58E-10	0.98940
	7	7.92E-10	0.98870
	8	5.83E-10	0.99168
	9	8.23E-10	0.98826
	10	6.37E-10	0.99092

Table B-5: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the single-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
24	1	3.07E-10	0.99566
	2	2.60E-10	0.99630
	3	3.48E-10	0.99506
	4	2.85E-10	0.99595
	5	2.84E-10	0.99596
	6	4.14E-09	0.94228
	7	2.55E-10	0.99636
	8	1.85E-10	0.99737
	9	3.41E-10	0.99515
	10	3.71E-10	0.99474
36	1	1.39E-10	0.99803
	2	1.71E-10	0.99757
	3	1.50E-10	0.99787
	4	1.79E-10	0.99751
	5	1.90E-10	0.99729
	6	1.79E-10	0.99750
	7	1.16E-10	0.99836
	8	1.97E-10	0.99722
	9	1.92E-10	0.99728
	10	1.70E-10	0.99759

Table B-5: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the single-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
48	1	9.79E-11	0.99861
	2	7.44E-11	0.99895
	3	1.04E-10	0.99853
	4	2.63E-10	0.99628
	5	1.05E-10	0.99851
	6	8.94E-11	0.99873
	7	1.50E-10	0.99789
	8	1.27E-10	0.99820
	9	1.08E-10	0.99846
	10	2.68E-10	0.99622
60	1	8.63E-11	0.99878
	2	3.35E-10	0.99543
	3	1.66E-10	0.99765
	4	4.54E-10	0.99362
	5	1.29E-10	0.99817
	6	9.93E-11	0.99859
	7	1.24E-10	0.99823
	8	1.41E-10	0.99801
	9	1.01E-10	0.99857
	10	5.93E-11	0.99916

Table B-6: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the 2-hidden-layer FBNNs with various numbers of neurons.

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
6 and 3	1	2.95E-10	9.96E-01
	2	1.70E-10	9.98E-01
	3	1.73E-10	9.98E-01
	4	3.21E-10	9.95E-01
	5	1.24E-10	9.98E-01
	6	1.47E-10	9.98E-01
	7	3.31E-10	9.95E-01
	8	1.60E-10	9.98E-01
	9	1.24E-10	9.98E-01
	10	1.97E-10	9.97E-01
10 and 5	1	3.96E-11	0.99944
	2	1.34E-10	0.99810
	3	4.66E-11	0.99934
	4	3.80E-11	0.99946
	5	7.24E-11	0.99898
	6	4.33E-11	0.99941
	7	4.35E-11	0.99940
	8	9.06E-11	0.99871
	9	7.86E-11	0.99888
	10	3.81E-11	0.99946

Table B-6: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the 2-hidden-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
16 and 8	1	1.34E-11	0.99981
	2	3.16E-11	0.99955
	3	2.04E-10	0.99714
	4	1.04E-11	0.99985
	5	3.67E-11	0.99948
	6	1.66E-11	0.99976
	7	4.42E-11	0.99938
	8	6.42E-11	0.99909
	9	2.57E-11	0.99964
	10	4.50E-11	0.99936
20 and 10	1	9.82E-12	0.99986
	2	1.17E-09	0.98337
	3	2.47E-11	0.99965
	4	3.30E-11	0.99953
	5	7.58E-12	0.99989
	6	1.77E-11	0.99975
	7	3.48E-09	0.95895
	8	2.80E-11	0.99962
	9	1.57E-11	0.99978
	10	8.40E-11	0.99881

Table B-6: The mean square error and the correlation coefficient between the target of the testing data set and the network prediction (Phase III: Numerical Experiment No. 2). The results are from the 2-hidden-layer FBNNs with various numbers of neurons. (*continued*)

No. of Neurons	i^{th} Initialization	MSE of testing data set	Correlation Coefficient
24 and 12	1	9.08E-12	0.99987
	2	1.12E-11	0.99984
	3	2.72E-11	0.99962
	4	1.53E-09	0.97837
	5	1.31E-11	0.99982
	6	8.49E-12	0.99988
	7	2.91E-10	0.99591
	8	2.02E-11	0.99973
	9	1.29E-11	0.99982
	10	3.83E-10	0.99456

BIOGRAPHY

Name: Miss Sainatee Champirat

Date of Birth: 24 July 1981

Place of Birth: Bangkok, Thailand

Education:

- March, 2007 Master of Science
Environmental and Hazardous Waste Management –
International program, Chulalongkorn University
- March, 2005 Bachelor of Science
Environmental Science and Technology
Faculty of Environment and Resource Studies, Mahidol
University

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย