INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

DISSERTATION

SOLVING THE INVERSE PROBLEM IN GROUNDWATER FLOW BY ITERATIVE INVERSION OF A NEURAL NETWORK

Submitted by

Abdalla Mohamed Taha Shigidi

Department of Civil Engineering

In partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

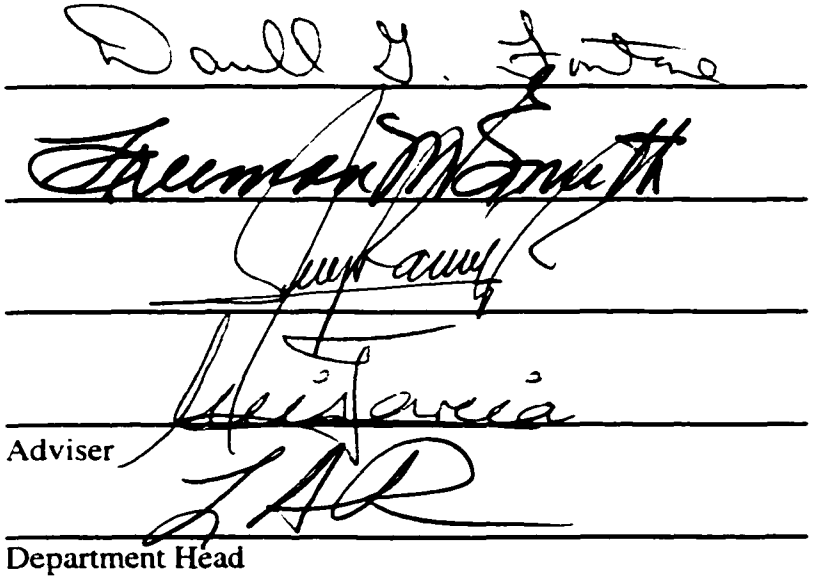
Summer 2000

COLORADO STATE UNIVERSITY

July 5, 2000

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY ABDALLA MOHAMED TAHA SHIGIDI ENTITLED SOLVING THE INVERSE PROBLEM IN GROUNDWATER FLOW BY ITERATIVE INVERSION OF A NEURAL NETWORK BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on graduate work



Adviser

Department Head

ABSTRACT

Solving the Inverse Problem in Groundwater Flow by Iterative Inversion of a Neural Network

A new methodology for solving the inverse problem in groundwater hydrology is developed and applied to a synthetic case study. An innovative aspect of the methodology is the use of a data driven approximation of the groundwater flow equation to calibrate a numerical model for a steady state groundwater system. An Artificial Neural Network (ANN) was successfully trained to produce the resulting hydraulic map when a complete transmissivity field is prescribed. The trained network was then iteratively inverted to match the prior information on transmissivity as well as piezometric head measurements. The hydraulic head maps resulting from the transmissivity field produced by the inverted ANN, are in good agreement with the hydraulic head maps produced from the original synthetic transmissivity field. The study shows that there is no unique solution to the inverse problem, and that an ensemble of solutions that honor the transmissivity measurements at their locations, and closely match the measured hydraulic head values can be obtained. Further more, the study shows that each of these non-unique solutions can be used to obtain accurate predictions.

Abdalla Mohamed Taha Shigidi
Department of Civil Engineering
Colorado State University
Fort Collins, CO 80523
Summer, 2000

ACKNOWLEDGEMENTS

There is a common misconception that a Ph.D. dissertation is the achievement of the student alone. Having gone through the process of finishing and writing a dissertation myself let me emphasize that this is not true. Throughout the process of choosing the topic of the dissertation, formulating a solution to the chosen problem, implementing the devised solution, and writing the manuscript, a Ph.D. student encounters those who encourage, inspire, and outright assist him/her. There is always, however, an individual who contributes the most to the student's effort, and without whom producing the final work would have been very difficult or outright impossible. In my case this role was fulfilled by my adviser **Dr. Luis Alfredo Garcia.**

I can not find the proper words to thank Dr. Garcia for his assistance and guidance during the course of my research, and in the writing of my dissertation. I find myself in awe of his dedication to his students and his commitment to excellence. I am glad to have worked under his supervision and to know him as friend.

I would also like to express my deep gratitude and appreciation to the members of my graduate committee **Dr. Jorge Ramirez, Dr. Darrell Fontane** and **Dr. Freeman Smith.** Doctors Ramirez and Fontane for their valuable comments and suggestions during the course of the research, and their invaluable comments on the manuscript. Dr. Smith for accepting to join my graduate committee at a very critical time, for

accommodating my timetable despite his busy schedule, and for his invaluable suggestions and comments on the manuscript.

I would also like to thank **Dr. Mohammed Siddiqui** for serving on my graduate committee. His absence from the final phase of this work was only instigated by his compelling family commitments.

To my beloved parents to whom I most indebted

To my dear wife Hiba who stood by my side

To the joy of my life ... my adorable daughters Aya and Nayat

Last but not least, to my brothers ElShafie, Mazin, Ihab and Ghassan

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Groundwater is an important source of water supply throughout the whole world, it accounts for more than 98% of the fresh water available for humans. Although a small percentage of this resource is being utilized, its use in irrigation, industries, municipalities, and rural homes continues to increase. According to the USGS, groundwater accounts for 22.4% of the total freshwater used in the United States during 1995 [Solley et al., 1995]. In the irrigation sector, groundwater accounted for more than 35% of the total amount used. In order to empower mankind to effectively and efficiently use the available groundwater resources, models that would enable engineers to predict the behavior of groundwater systems to various stresses and scenarios have been developed and employed. The most commonly used models are mathematical models, which express the behavior of groundwater systems in terms of physical equations. During the last three decades mathematical modeling techniques have increasingly proved their value in furthering the understanding of groundwater systems and, hence, in improving the evaluation, development, and management of groundwater resources, as well as the control of groundwater problems. It is now common practice to use simulation and mathematical models to analyze groundwater systems.

Almost all groundwater flow and transport models solve the relevant partial differential equations, using finite difference or finite element approximation methods. The physical layout of the area to be modeled is replaced with a discretized model domain referred to as a grid consisting of cells, blocks, or elements depending on whether finite difference or finite element methods are used. Knowledge of the aquifer parameters over the entire flow domain is essential to be able to run a model. As such, the parameters of each cell or block in the grid are given discrete values. These discrete values are referred to as model parameters, and the process of describing the model structure by a set of discrete values is called parameterization.

1.2 Defining the Inverse Problem

For a groundwater model to yield accurate predictions or inferences, the model parameters should accurately represent the real aquifer system. While some aquifer parameters can be measured in the field, such measurements are usually scarce and prone to error [Yeh 1986]. Furthermore, many parameters can not be measured directly, they are estimated indirectly by interpreting the results of a test (pump test, tracer test, etc) performed in situ or on a sample. Such estimates may be called "local values" of parameters since they reflect properties of the aquifer in the sample or in the vicinity of the well used in the test. The parameter field measurements are thus performed on a scale different from that required for modeling purposes, and the measured parameters are both conceptually and numerically different from their model counterparts.

The groundwater model outputs come in the form of hydraulic head and/or concentration data. These outputs can be corroborated by field measurements. Since hydraulic head and concentration measurements (usually referred to as soft data) are

limited to a finite number of observation wells and are usually not accurate, it is not surprising that when a model is constructed with parameters derived directly from field data, there is seldom good agreement between computed and measured soft data. To obtain such an agreement the model parameters must be calibrated against the observed soft data so as to obtain an acceptable fit between computed and observed data. Model calibration can also be used to fill in the missing model parameters.

Traditionally model calibration is done through straightforward trial-and-error [Strecker 1986]. In this approach, different sets of parameter values are selected for the model until the fit between the model outputs and field observations is judged satisfactory. The trial-and-error calibration process is easy to implement, but it is labor intensive (therefore expensive), frustrating (therefore often left incomplete), and subjective (therefore biased and can lead to non-unique results, the quality of which is difficult to evaluate).

During the past three decades there have been significant efforts to automate the model calibration process. The problem of estimating hydraulic parameters from observations and prior information by using a numerical or mathematical model became known as "parameter identification" or "inverse problem" [Hyun 1998]. In groundwater inverse problems the estimated parameters are usually hydrogeologic properties and the measured dependent variables are quantities such as piezometric head, solute concentration, or temperature.

Solving the inverse problem in essence is a task of identifying a model that would reverse a complex forward relation. If the forward relation between the measurement vector **z** and the unknown parameter vector **p** can be expressed as:

$$\mathbf{z} = \mathbf{F}(\mathbf{p}) + \boldsymbol{\varepsilon}$$

F is the forward operator that maps **p** to **z** where **ε** is the measurement error vector.

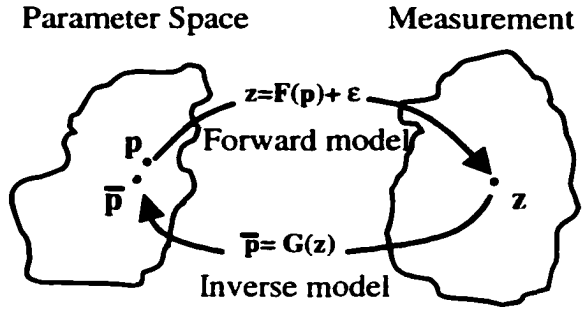


Figure 1.1 Graphical representation of the inverse problem

The basic goal of inverse estimation is to identify a generalized inverse operator **G** that maps the measurement vector to an estimate of **p.**

$$\bar{\mathbf{p}} = \mathbf{G}(\mathbf{z}) = \mathbf{G}[\mathbf{F}(\mathbf{p}) + \boldsymbol{\varepsilon}]$$

The objective of the inverse model or operator is to obtain optimal estimates of unknown parameters which are consistent with prior measurements of flow and transport parameters, and which best reproduce through the flow and transport equations, the measured heads, flow rates, and/or concentrations.

Use will be made in this research of the ability of artificial neural networks for mapping complex non-linear relations to approximate the groundwater flow equation. The neural network model can then be inverted through an iterative process to solve the inverse problem. An artificial neural network (ANN) is a system loosely modeled on the human brain. In recent years, it had been successfully used to map non-linear relations that are not well understood [Goh 1995, Buenfeld et al., 1998]

1.3 Research Objectives

The objective of this research is to develop a new methodology based on neural networks to solve the steady state groundwater inverse problem. The philosophy of the proposed methodology is to approximate the groundwater flow equation with an Artificial Neural Network which can latter be inverted to solve the inverse problem. The scope of this research is to investigate the plausibility of successfully training an Artificial Neural Network to provide a data driven approximation of the groundwater flow equation, and to demonstrate that such a network can be inverted through an iterative process to provide an inverse model for a hypothetical two-dimensional aquifer system. The research will then evaluate the ability of the inverse solutions to honor prior information, and their capacity to produce accurate predictions when used in groundwater flow models. Another goal of this research is to study other aspects of the inverse problem, namely the uniqueness of the solutions, and their stability, as well as their effect on attaining accurate predictions.

This dissertation is divided into seven chapters. The first chapter contains an introduction to the inverse problem and the research objectives. The second chapter contains a review of some of the literature in solving the inverse problem, as well as some of the uses of artificial neural networks in the field of engineering. The third chapter features the methodology that will be used to solve the inverse problem and the tools that will be utilized to achieve the research objectives. The fourth chapter details how the proposed methodology was implemented to solve the inverse problem for a hypothetical case study. In the fifth chapter the obtained results are scrutinized, and the overall performance of the methodology is evaluated. It is in this chapter where the

aspects of solution uniqueness and stability are studied and evaluated. The final conclusions and the recommendations for future research are given in chapters six and seven respectively.

CHAPTER 2

LITERATURE REVIEW

2.1 The Inverse Problem in Groundwater Flow

2.1.1 Introduction

In the inverse groundwater flow problem, the hydraulic head measurements are provided and the hydraulic conductivity or transmissivity values are to be obtained, while in the contaminant transport inverse problem, hydraulic head and concentration measurements are provided and the dispersivities and hydraulic conductivity values are sought. The general consensus among groundwater modelers is that the inverse problem may, at times result in meaningless solutions [Carrera and Neuman 1986b]. There are even those who argue that the inverse problem is hopelessly ill posed and as such, intrinsically unsolvable [Carrera and Neuman 1986b]. This pessimistic view aside, it has been established that a well-posed inverse problem can in practice yield an acceptable solution [McLauglin 1996].

During the past three decades, extensive research has been conducted to study the problem of parameter estimation for groundwater flow models. The various approaches proposed for solving the inverse problem for groundwater flow can be classified into three categories; the direct approach, the indirect approach, and the geostatistical approach [Peck et al 1988].

2.1.2 The Direct Approach

If the hydraulic heads and their spatial and temporal derivatives are known over the entire region, and if the measurement and mass balance errors are negligible, the flow equation becomes a first order partial differential equation in terms of the unknown aquifer parameters. (e.g. hydraulic conductivity) [Yeh 1986]. Using numerical methods, the linear partial differential equation obtained can be reduced to a linear systems of equations, which can be solved directly for the unknown aquifer parameters, and hence the name the "direct approach". All the solutions obtained by the direct approach are unstable. They exhibit large changes in the computed aquifer parameters when small errors are introduced in the hydraulic head values [McLauglin 1996].

Stallman [1956] presented one of the earliest applications of the direct inverse technique. By assuming that the transmissivity varies as a linear or quadratic function of space, the author used the finite difference approximation of the groundwater flow equation to solve for the transmissivity values given the exact hydraulic head values at all the nodes. The technique was found to be unstable when small errors are added to the measured heads, and it resulted in some estimated transmissivity values being less than zero.

Nelson [1960] devised a technique for calculating the transmissivity, provided a twice-differentiable function describing the steady state hydraulic heads throughout the aquifer exist. Realizing that this is an insurmountable task, Nelson [1961] proposed a graphical inverse technique. The graphical technique requires the graphical construction of a flow net for the aquifer in question. Using Darcy's law as a basis, the constructed flow net can be used to estimate the transmissivity distribution of the aquifer.

The graphical method was applied to a number of aquifers around the world [Peck et al 1988]. The technique however unrealistically assumed that the governing flow model was exact and that the mass balance errors were negligible. This gave rise to the work of Kleinecke [1971] who attempted solving the inverse problem using the direct approach, while minimizing the mass balance errors.

Kleinecke [1971] was the first to formulate the transient inverse problem as a linear programming problem. The author noted three obvious objectives for minimizing the mass balance errors.

1) Minimize the maximum absolute error value in the mass balance equation.
2) Minimize the sum of the absolute error values in the mass balance equation.
3) Minimize the sum over all nodes of the maximum absolute error values at each node for every time step.

It was Kleinecke's judgment that the third criterion guaranteed uniformity across the whole domain and eliminated the issue of particular nodes being neglected. This approach was applied to the Chino-Riverside basin in Southern California. The results however were unsatisfactory. Of the parameters to be estimated, only one-third of the storage coefficients, and one-half of the transmissivites were found to be non-zero. Additionally, the parameter estimates, which were obtained, were widely oscillating.

Neuman [1973] noted that the uncertainty associated with the inverse problem makes an approach based on the minimization of a single error function yield unsatisfactory results. He used parametric linear programming with the goal of optimizing the predictive capability of the model, after defining a multiple objective decision process. The decision process seeks to minimize the absolute value of mass

balance errors while allowing a subjective deviation of the parameter estimates from their prior measurements.

Hefez [1975] formulated the inverse problem as a quadratic programming problem with a quadratic objective function and linear constraints. The author also compared the quadratic programming formulation to the linear programming formulation proposed by Kleinecke [1971]. Even though the estimated parameters were found to be very sensitive to errors in the hydraulic head values, overall the quadratic programming approach was deemed better.

Most of the inverse research suggests that there are major difficulties associated with using the direct inverse methods [Mclauglin 1996]. The estimated parameters particularly the transmissivity is very sensitive to small errors in the hydraulic head data. Such errors inherently accompany any scheme that utilizes interpolation methods to approximate hydraulic head variations in the entire domain process.

2.1.3 The Indirect Approach

The philosophy of the indirect approach is quite different from the direct approach. The indirect approach seeks to minimize the difference between observed and calculated hydraulic heads at specified observation points. This formulation nullifies the need to obtain the hydraulic head values over the entire flow region, and does not require differentiation of the measured data. Numerous techniques have been developed to solve the groundwater inverse problem using the indirect approach, and even though they share the same performance index, they differ in the search method they employ to find the parameter values that minimize the difference between observed and calculated hydraulic

heads. Many of the formulations that employ the indirect approach have focused on formulating the inverse problem as a mathematical programming problem.

The first attempt to use an indirect approach to solve the inverse problem seems to be that of Jacquard and Jain [1965]. The authors used non-linear optimization to obtain parameter estimates, which minimize the sum of the squared hydraulic head residuals. The methodology was used to estimate the permeability distribution for a hypothetical petroleum reservoir. The methodology yielded good results in the vicinity of wells, and inaccurate results away from them.

Coats et al. [1968] and Slater and Durrer [1970] used linear optimization techniques to estimate permeability for hypothetical petroleum reservoirs. Because of the inherent non-linearity of the indirect inverse problem, these linear methods were found to be useful only in a very limited number of cases.

Yeh [1975] used an iterative quadratic programming technique for estimating transmissivity and storage coefficient for a hypothetical aquifer. The technique resulted in a good match between the observed and calculated hydraulic head values, but it was not capable of incorporating the prior knowledge about aquifer properties.

Cooley [1977] adapted the steady state two-dimensional inverse problem as a classical non-linear regression problem. The criterion adopted to define the optimal set of unknown hydraulic parameters was to minimize the weighted sum of squared differences between calculated and observed hydraulic heads. The methodology was applied [1977] to estimate the hydrologic parameters of two aquifers. It resulted in an excellent fit between calculated and observed heads, but it was not capable of incorporating prior

information. The methodology resulted in non-unique solutions, and was unstable when the number of estimated parameters was increased.

Some attempts were made to develop a parameter estimation technique that is enshrined in a statistical framework. These attempts were aimed at quantifying the uncertainty that is common to all aspects of inverse modeling. Neuman [1979] casted the work of Cooley [1977] in a statistical framework. He added a linearized error analysis that enable the estimation of the covariance of the transmissivity estimates as well as the square error of the estimates of the hydraulic head. The additional feature of emphasizing the prior measurements that are believed to be of higher reliability was also included. The author did not address the influence of adding significant errors to the measured head values on the stability of the technique. Cooley [1982] incorporated prior information into the weighted least square method. The technique he used differed from that of Neuman [1979] in that it incorporated information about more than one parameter, and that it allows for the incorporation of prior information with known reliability as well as for the incorporation of subjective prior data whose reliability is unknown.

Gavalas [1976] was the first to formulate the inverse problem in a Bayesian framework. The underlying philosophy of the Bayesian estimate is the assumption that the unknown parameters are random variables, with a certain probability distribution. The inverse problem is formulated as a minimization problem with a penalty function incorporating prior geological information added to the objective function. The author used this methodology to estimate the porosity and permeability for a hypothetical one-dimensional system. Shah et al. [1978] used the same hypothetical system studied by Gavalas [1976] to compare the Bayesian estimation with the weighted least squares

estimation without prior information. The Bayesian estimates were found to be superior when the correct parameter probability distribution function was known, and inferior to the least squares estimation otherwise.

Wilson et al. [1978] used the Kalman filter to solve the inverse problem. The procedure however falls in the class of Bayesian estimators, since it optimally combines a priori parameter values with the information contained in the observations to obtain the a posteriori parameter estimates. It is worth mentioning that the Kalman filter is most useful in applications where the number of unknowns is relatively small, and as such the authors were able to apply it only to small hypothetical problems.

Perhaps the most influential work in solving the groundwater inverse problem, is that of Carrera and Neuman [1986a,b,c]. They assumed that the parameters to be estimated in an inverse problem are deterministic, but unknown. They then used the maximum likelihood approach to find the parameter estimates that maximize the probability of obtaining the hydraulic head observations. The methodology is capable of incorporating prior information, and applicable to both the transient and steady state conditions. The methodology was tested using small synthetic cases, and applied to field data. The methodology does not ensure that the model will reproduce the hydraulic head observations. The authors are noted however for their excellent discussion of the uniqueness and stability of the inverse models, and providing practical guidelines for enhancing these properties.

2.1.4 The Geostatistical Approach

Geostatistics can be and have been used to characterize the spatial variability and correlation of a regionalized variable. Once the geostatistical structure of the

transmissivity field has been determined from the available data, kriging can be used to fill in the missing data. Kriging however does not consider the hydraulic head when estimating transmissivity. This means that parameter estimates obtained by kriging alone do not constitute a solution to the inverse problem, because it does not incorporate the information about transmissivity that the hydraulic head measurements contain.

Kitanidis and Vomvoris [1983] proposed a geostatistical approach to solving the inverse problem that accounts for the statistical relationship between transmissivity and hydraulic head. Once the spatial variability of the log transmissivity and hydraulic heads are determined, the maximum likelihood estimation is used to obtain the unknown statistical parameters of the assumed log transmissivity model. Cokriging can then be used to predict the spatial distribution of transmissivity conditioned on values of transmissivity and hydraulic heads. The approach of the authors emphasizes the statistical characterization of the log-transmissivity field, and assumes a linear relationship between transmissivity and hydraulic head. The formulation was applied to hypothetical one-dimensional aquifers, and was extended by Hoeksema and Kitanidis [1984] to estimate two-dimensional transmissivity fields. They tested the adapted methodology on hypothetical aquifers, and applied it to the Jordan aquifer in Iowa.

The geostatistical approach was also used by Hoeksema and Kitanidis [1985] to analyze the spatial correlation structure of transmissivity, hydraulic conductivity, and storage coefficient for thirty-one regional aquifers in the U.S. The method however does not ensure that the model will be able to reproduce measured hydraulic head data, and it amounts to estimating the stochastic structure underlying the transmissivity field.

La Venue [1992] coupled an adjoint sensitivity technique with a kriging algorithm to calibrate a flow model. The methodology directly identifies the regions where modification of the model's kriged transmissivity or boundary pressure values will directly improve the overall fit between measured and model-calculated heads at selected wells. At the locations identified as most sensitive to transmissivity changes, synthetic transmissivity values referred to as pilot points, are added to the transmissivity data base and used as input for kriging the transmissivity field.

2.1.5 Recent Contributions

With a growing consensus that the direct approach yields unstable inverse models, most of the recent work has focused on the indirect and geostatistical approaches. The majority of the work done during the last decade focused on improving the methodologies mentioned earlier and applying them to complicated hypothetical problems, and practical problems.

Ferraresi et al [1996] used the Kalman filter formulation to solve the inverse problem for two Libyan aquifers. The authors used the net recharge values instead of the hydraulic head values as a measurement space over which to perform the minimization of the difference between observed and calculated soft data. The method has the advantage that unlike the hydraulic heads, the net recharge values are linearly related to the groundwater flow model parameters. The hydraulic head values however were assumed to be perfectly measured.

Honjo [1999] improved the Bayesian formulation by introducing the Akaike's Bayesian Information Criterion. This criterion adapts the modern information and communication theory to predict the prior probability distribution of the unknown

parameters to be used in the Bayesian formulation. The technique eliminates the subjectivity involved in selecting a prior distribution that has always been associated with the applications of the Bayesian method, and make it applicable to real world problems. The authors demonstrated this by applying the methodology to a case study from Japan.

Hill [1998] used a nonlinear regression groundwater flow inverse model to solve for the parameters of a complicated three-dimensional synthetic aquifer in steady state. Aquifer heterogeneity, aerial recharge, and groundwater interaction with a lake and a stream characterize the test case. The exercise was conducted to demonstrate that non-linear regression could be used to solve complicated groundwater flow inverse problems. The test produced effective though non-unique calibrated models capable of accurately predicting the true system response. The test however was not extended to obtain the inverse solution when random errors are added to the hydraulic head values.

Weiss [1998] utilized both hydraulic head and tracer concentration data and used non-linear regression to solve the groundwater flow inverse problem. The concentration data was introduced with the primary intent of improving estimates of the parameters of a groundwater flow model.

During the 1990s and due to the tremendous increase in computing power, the trend of casting the inverse problem in a stochastic framework increased dramatically. RamoRao [1995] implemented the automated pilot point approach to generate an ensemble of transmissivity fields considered to be equally likely, each of which is in agreement with the measured transmissivity and pressure data. It consists of generating a selected number of conditionally simulated transmissivity fields, and then calibrating each of the fields to match the measured steady state or transient heads by using a least

squares criterion. Kriging coupled with adjoint sensitivity analysis is employed for the optimal location of pilot points, and gradient search methods are used to derive their optimal transmissivities. The methodology was not tested with a hypothetical case whose parameters are known, and against which future predictions could be compared. Rather, the methodology was applied to field problems [RamoRao et al. 1995, La Venue et al. 1995] in which future predictions obtained by the estimated transmissivity fields stand unchallenged.

It is safe to assume that the Linearized Cokriging developed by Kitanidis and Vomvoris [1983], and the Pilot Point method are not the only two available geostatistical approaches to solve the inverse problem in groundwater flow. Zimmerman [1998] described and compared seven geostatistical approaches including the above mentioned two. The ultimate objective in the study was to determine which of the seven geostatistical inverse techniques is better suited for making probabilistic forecasts of the potential transport of solutes in an aquifer where uncertainty in hydrogeologic properties are significant.

All the geostatistical methods used in the study seek to obtain the semivariogram that describe the structure of the sought transmissivity field and implement one of two approaches:

- Use only the transmissvity data to select the semivariogram and then condition the resulting transmissivity field to honor the hydraulic head data (e.g. Pilot Point Method).
- Use both the transmissivity and the hydraulic head data simultaneously to develop the semivariogram of the transmissivity field (e.g. Linearized Cokriging).

The various geostatistical approaches differ only in the methodology they utilize to implement either of the approaches listed above.

Attention was recently paid to other aspects of the inverse problem. Weiss [1998] developed procedures to identify the model parameters for which prior information may best stabilize the parameter set, and the ones for which prior information lead to the smallest possible errors in the final set of parameter estimates. These two criteria were referred to as the efficient and responsible use of prior information. The developed procedures established guidelines that screens the available parameters to identify those parameters for which prior information will be most efficient and responsible in determining the final values of the parameter estimates.

Sun [1998] studied the issue of model structures, and proposed a stepwise regression method to determine simultaneously both model structure and model parameters. The proposed methodology advocates starting with a homogenous model structure and step by step gradually increasing the complexity of the model structure. At each level of complexity, the fitting residuals and the model structure error are calculated to determine if a more complex structure is needed. Hyun [1998] suggested four methods to identify the optimum model structure to be used in solving the inverse problems. The study was inconclusive in ranking the performance of the four methods.

2.2 Neural Networks

2.2.1 Introduction

There is a general consensus that the human brain and nervous system, are the most sophisticated computer and information-processing device ever to exist. Some characteristics of the nervous system which label it the ultimate information processing

device, is its ability to learn and generalize from examples, to produce meaningful solutions to problems even when input data contain errors or are incomplete, to adapt solutions over time to compensate for changing circumstances, to process information rapidly, and to transfer readily between solution techniques.

Human beings have always been astute observers of nature. Indeed, almost all of humanity's scientific achievements have been attained by emulating nature and manipulating its laws. That is why it is no surprise that mathematicians, physicists, and computer scientists have exhibited a keen and growing interest in developing a class of computing devices that operate in a manner analogous to that of the biological nervous systems. These devices are known as Artificial Neural Networks.

A typical biological nerve cell (neuron) is composed of a cell *body*, a tubular *axon*, and a multitude of hair-like *dendrites*. Natural Neural Networks (NNNs) consist of tens of billions of densely interconnected neurons. Neurons essentially behave as microprocessors. Each neuron receives the combined output of many other neurons through the dendrites. If this signal is strong enough, the neuron is activated and produces an output, which is transmitted through the axons. The axon splits up and connects to the dendrites of many other neurons via the *synapses*. The strength of the synapses is modified as the brain learns.

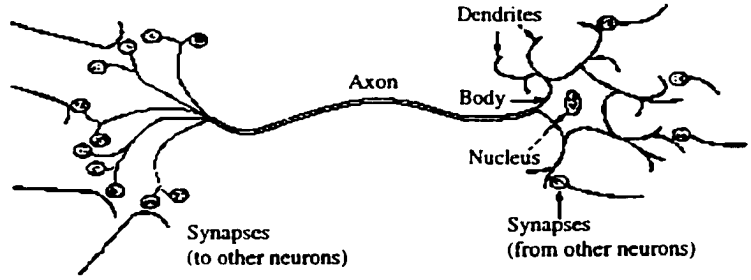


Figure 2.1 A biological neuron

Artificial Neural Networks (ANNs) models the biological neuron. They consist of layered nodes (cells) which are the processing elements and links (axons) which interconnect these elements. The processing elements and the interconnections between them represent the neural network architecture. Determining the architecture of a neural network for a specific problem entails defining the number of layers, the number of nodes in each layer and the interconnection scheme between the nodes.

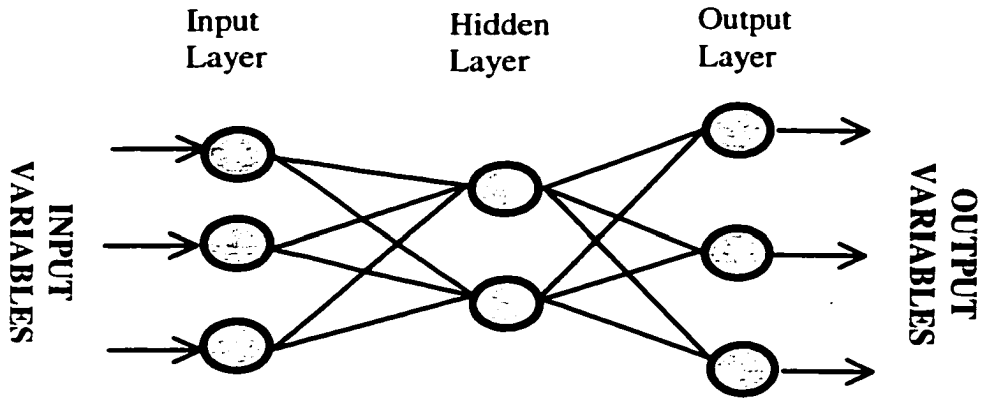


Figure 2.2 Typical two-layer feed forward artificial neural network

A single node is insufficient for many practical problems, and networks with a large number of nodes are frequently used. Each node output is assigned a weight and is carried by a link as input to another node. Each node receives the combined weighted output of many other nodes and performs a mathematical operation the product of which becomes the node's output. The node functions are simple, and the overall functions performed by the neurons are trivial. However, working collectively in the form of a network, neurons are capable of solving complicated problems, and approximating complex nonlinear relations.

A major concern in the development of a neural network is determining an appropriate set of weights that make it perform the desired function. There are many ways that this can be done; the most popular class of these algorithms is based on supervised learning. Typically, supervised training starts with a network comprising an arbitrary number of hidden neurons, a fixed topology of connections, and randomly selected values for the weights. The network is then presented with a set of training patterns, each comprising an example of the problem to be solved (the inputs) and its corresponding solution (the targeted outputs). Each example is input into the network, and the resultant output is compared to the targeted solution providing a measure of total error in the network for the set of training patterns. The weights are then adjusted by small amounts as prescribed by some rule so that on the next occasion the example problems are presented to the network, the error is reduced, and the output is closer to the targeted outputs. Typically the process is repeated many times until the network is able to reproduce, to within a specified tolerance, the corresponding solutions to each of the example problems. Upon processing many examples, it is anticipated that the network is able to generalize what it has practiced. In essence it is able to learn to provide accurate solutions to problems not used during training.

It is the ability of neural networks to provide a computational mechanism capable of acquiring and representing complex unknown interrelationships that has been most attractive to engineers and scientists. That is why the use of neural networks in various engineering fields has dramatically increased in recent years.

2.2.2 Neural Networks Applications

In their two-paper series, Flood and Kartman [1994a,b] provided insight into the usage and potential applications of artificial neural networks within the field of civil engineering. Whether evaluating the bending moment for a cantilever or selecting the optimum cable tensions for guyed communication towers, the authors gave examples of civil engineering problems that were successfully solved by artificial neural networks. In the subsequent section, a number of applications in which artificial neural networks were utilized in solving engineering problems will be described to demonstrate the capability and versatility of the technique.

Karunanithi et al. [1994] used neural network algorithms to estimate flows at an ungaged site on the Huron River, Michigan. Flows in streams and rivers are complex processes that are influenced by many factors such as watershed topography, vegetation cover, soil types, channel characteristics, groundwater aquifers, precipitation distribution, snow melt, urban activities, and so on. In this study daily stream flow records for the ungaged site were extended by the power model as well as neural networks. Comparison of the results of the power model and neural networks to the available station records suggests that the predictions of the neural network models closely follow the observed daily stream flows, and that they were more accurate than the power model predictions.

Kuligowski and Barros [1998] used a back propagation neural network to estimate missing rainfall data from nearby gages. The results of the neural network algorithm compared favorably to other techniques such as arithmetic and distance-weighted averages of the values from nearby gages, and also to linear optimization methods such as regression.

Sureerattanan and Phein [1997] used the back propagation neural network algorithm to forecast the daily stream flow of the Mac Klong River Basin in Thailand. The input to the network consisted of rainfall discharge and the past stream flow for the station under consideration. The output being a forecast of the stream flow. The pattern of the neural network forecast closely followed the observed data for the five stations considered with an accuracy of more than 90%. The results demonstrate that neural networks are capable of simulating highly non-linear relations without the need to understand the intricacies of that relation.

Smith and Eli [1995] used a back propagation neural network algorithm to obtain the runoff for a river basin given the rainfall patterns. The network was trained and tested on a simple synthetic watershed consisting of a 5x5 square grid configuration. The resulting neural network was capable of predicting the discharge peak and time of peak of the resulting runoff hydrographs when given the rainfall pattern. The network was thus able to simulate a highly non-linear, time varying, spatially distributed process.

Hsu et al. [1995] used an artificial neural network to model the rainfall-runoff relationship of the medium-size Leaf River Basin near Collins, Mississippi. The neural network model was compared to the linear ARMAX (Autoregressive Moving Average with Exogenous inputs) time series approach and the conceptual SAC-SMA (Sacramento Soil Moisture Accounting) model. The artificial neural network model was capable of effectively modeling the rainfall-runoff relationship, and its results were superior to both the ARMAX and the SAC-SMA models.

Goh [1995] used neural networks to estimate the relative density of soil samples from cone penetration test (CPT) results. The relation between the two parameters is

highly non-linear, and can only be described empirically. The back propagation algorithm used was successful in modeling the non-linear relationship, and it provided better results than the other standard empirical relations did. The trained network was used to synthesize charts that reliably relate the cone penetration test results to the relative density of the soil sample.

One of the complex problems that are solved by civil engineers is the determination of reservoir operation for the efficient management of available water. The problem is complicated by the involvement of random hydrological events. Raman and Chandramouli [1996] used a neural network model to determine the reservoir operating policies for the Aliyar Dam in India. The input for the neural network consisted of initial storages, inflows, and demand, while optimal releases represented the output. The operation policy was also determined by using a multiple linear regression model, a stochastic dynamic programming model, and a standard operating policy. By comparing the performance of the four different policy making methods for three years, the authors concluded that the operating policies generated by the neural network model, gives better performance than the three methods for the reservoir system under consideration.

Maier and Dandy [1996] used a neural network algorithm to forecast the salinity in the river Murray in Australia 14 days in advance. The transport of salinity downstream is dependent on flow and upstream salinities. The flow in the river is complicated by the presence of storages, locks, weirs, and barrages. The back propagation algorithm was used with upstream salinities, water level at locks and weirs and upstream flow at some gauging stations used as input for the neural network. The inputs were measured at one-day lags with each measurement used as an independent input for the neural network.

Measurements of salinity at the point of interest for the days before the forecast date were also used as inputs. The only output of the network was the salinity at Murray Bridge at the date of interest. The trained neural network was able to predict the salinity at Murray Bridge 14 days in advance for all of 1991 with an accuracy of more than 90 %.

Hwang and Chan [1990] introduced an iterative technique to constrain the solution of an artificial neural network engaged in non-linear mapping. In this technique, the neural network input is the sought solution. The iterative procedure is applicable to an already trained neural network. It starts with an initial solution (network input) and gradually moves toward the desired solution. An initial solution vector $\mathbf{A}^0$ is guessed and the trained network without satisfying the imposed constraints obtains its output response. The solution (network input) is then passed to a network constraint mechanism and results in a solution $\mathbf{A}^1$ with the constraints better satisfied. $\mathbf{A}^1$ becomes the new guess and is promptly used as input for the neural network. The procedure iterates several times until convergence is reached.

Davis et al. [1993] used the iterative neural network technique to estimate snow parameters such as snow density, snow temperature and snow depth from remote sensing measurements. The iterative neural network algorithm was used to reconstruct a synthetic terrain of snow parameters from their corresponding measurements. The reconstructed parameter contours were found to be in good agreement with the original synthetic parameter contours. The authors also retrieved the snow parameters from their corresponding measurements by using an explicit neural network mapping. The authors concluded that for the problem at hand, the iterative technique was superior to the explicit technique.

These are some examples where neural networks have been successfully employed to solve a variety of engineering problems. Neural networks are now being increasingly applied in many branches of science. What makes neural networks attractive is their ability to map complex relations without the need to understand the laws that govern that relation.

CHAPTER 3

THEORY AND METHODOLOGY

The methodology developed in this research is comprised of two major tasks. The first task is to successfully train an Artificial Neural Network (ANN) to approximate the relation between any possible transmissivity field of the aquifer being modeled, and the hydraulic head values as described by a groundwater flow model. This in effect will produce a data driven model capable of mapping the relation between any realization of transmissivity of a specific groundwater aquifer and the resulting hydraulic head values as computed by the groundwater flow equations. The second task is to invert this model to solve the inverse problem so as to produce a transmissivity field that will honor the known transmissivity values and reproduce the known hydraulic head values when used in a groundwater flow model.

In this chapter the methodology introduced to solve the inverse problem is outlined, the computational tools used to implement the solution are explained, and the software used are described. It is important to note that the study was conducted on a hypothetical two-dimensional confined aquifer under steady state conditions. The flow chart shown in Figure 3.1 presents the steps implemented in this research. These steps constitute the general methodology used and it can be summarized as follows:

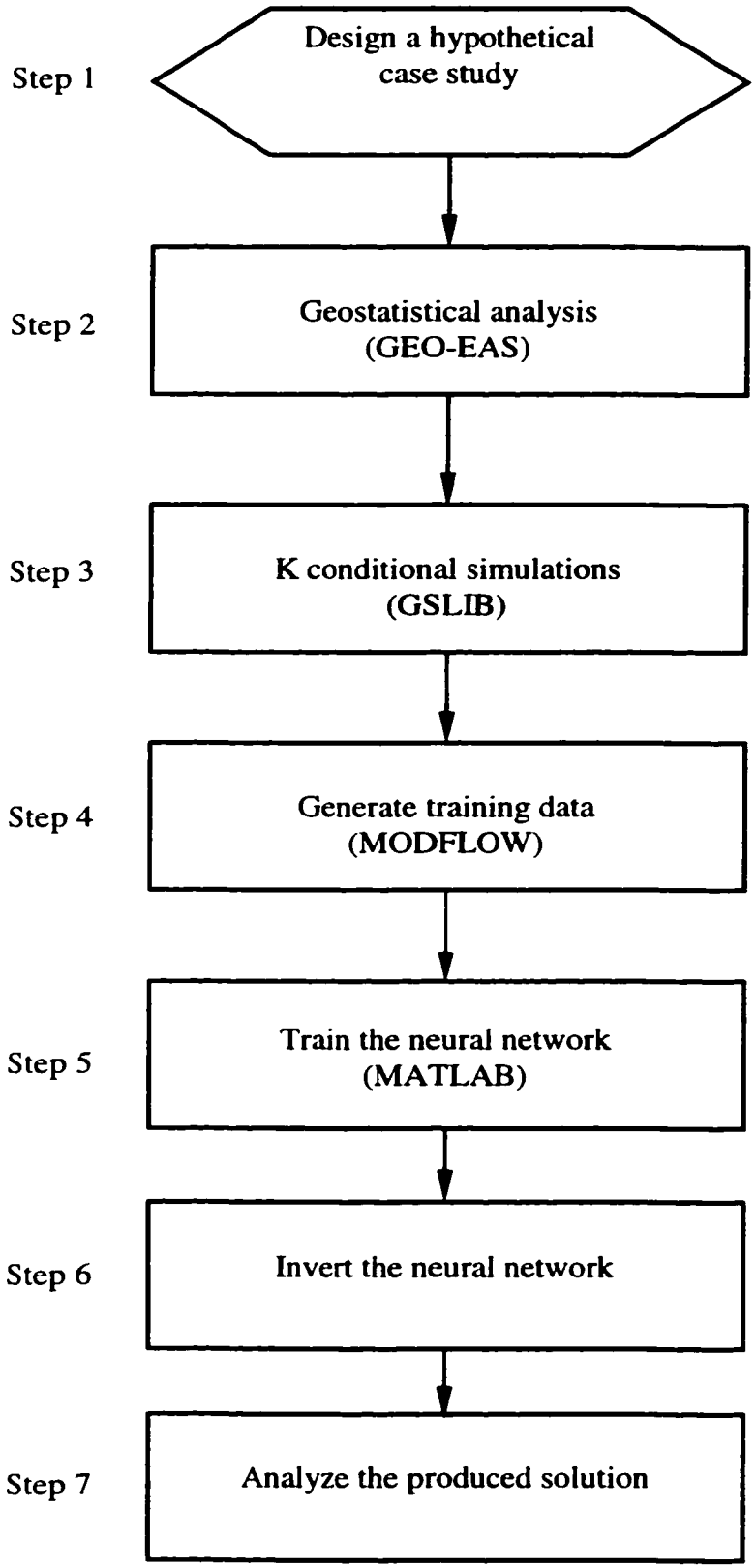


Figure 3.1 Flow chart for solving the inverse problem

1. Design a hypothetical case study on which the methodology will be implemented. All the parameters needed to model the problem are to be defined (i.e. hydraulic conductivity values, boundary conditions, aquifer type and thickness, stresses, and starting hydraulic head values). The parameters that are to be used to solve the inverse problem (named hereafter field values) are to be selected in this step.
2. Use the field hydraulic conductivity values to obtain the geostatistical distribution of the hydraulic conductivity field. The software GEO-EAS will be used to this end.
3. Generate a series of alternate hydraulic conductivity fields that will account for the field data and their spatial correlation. The Sequential Gaussian Simulation Method (SGS) will be used for this purpose.
4. Run a groundwater flow model to generate the resulting hydraulic head distribution for each simulated hydraulic conductivity field. This will generate input and output sets that can be used to train a neural network. The well-known model MODFLOW will be used for this purpose.
5. Once the training sets are obtained, a neural network will be identified and trained to map the relation between the transmissivity fields and the corresponding hydraulic head field. The commercial package MATLAB will be used to conduct the training process.
6. After the neural network has been trained, the hydraulic head and transmissivity values will be used to invert the network and estimate a transmissivity field that would honor the field data.

7. The proposed inverse methodology will then be investigated to evaluate its stability, and the uniqueness of its solutions. The accuracy of the results produced by the methodology as well as the effect of parameterization will be studied.

3.1 Groundwater Flow Models

The governing equation that describes a steady state two-dimensional flow in a heterogeneous isotropic saturated aquifer is given by

$$\frac{\partial}{\partial x}(K_{xx} b \frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(K_{yy} b \frac{\partial h}{\partial y}) = q \quad \cdots\cdots\cdots\cdots (3.1)$$

Where;

K_{xx} = hydraulic conductivity in the x-direction (L/T)

K_{yy} = hydraulic conductivity in the y-direction (L/T)

b = aquifer thickness (L)

q = recharge to the aquifer (L/T)

(x,y) = cartesian coordinate system (L)

Finite difference groundwater models can provide accurate solutions for this equation. In these models, each partial derivative that appears in the equation is replaced by a finite difference approximation. When these differences are evaluated at each point, the result is a set of simultaneous equations that can be solved either directly or by various iterative procedures. MODFLOW [Harbaugh and Mac Donald 1996] is used in this research to model the hydraulic head values. It is a modular three-dimensional finite difference flow model developed by the U.S. Geological Survey. The model has been modified so that it can read the transmissivity field from an external file outside the standard input data files, as well as to create an output file that contains the corresponding hydraulic head fields.

3.2 Geostatistics

Geostatistics is concerned with the study of regionalized variables (variables that vary or fluctuate in space and/or time). Geostatistics offers a collection of deterministic and statistical tools aimed at understanding and modeling spatial variability. Geostatistical methods namely kriging have been extensively used in the field of groundwater to characterize and interpolate unsampled hydraulic conductivity values from available field data.

Kriging, the Best Linear Unbiased Estimator (BLUE) is used to obtain an optimal estimate of the variable Z(x) at an unsampled location x, based on sampled values $Z(x_1)$, $Z(x_2)$, $Z(x_3)$,.....,$Z(x_n)$. Perhaps the most important property of kriging is that it restores the exact values of sampled variables. Equation 3.1 shows how an estimate $Z_k(x)$ of Z(x) can be obtained from a linear combination of measured values.

$$Z_k(x) = \sum_{i=1}^{n} \omega_i \cdot Z(x_i) \quad \cdots\cdots\cdots\cdots\cdots (3.2)$$

Where ω_i are estimated weights.

The weights are determined so as to minimize the estimation variance and produce an unbiased estimate. To achieve this a covariance model for the data at hand is needed. The covariance has been traditionally obtained from semi-variograms which are used to model the spatial variability of the regionalized variable in question.

The semi-variogram is defined as the variance of the increment [Z(x+h)-Z(x)], where h is the lag distance ($h=x_i-x_{i+1}$). Mathematically speaking, the semi-variogram is defined as follows:

$$\gamma(h)=\frac{1}{2}E([Z(x+h)-Z(x)]^2) \quad \cdots\cdots\cdots\cdots\cdots (3.3)$$

It can be retrieved from experimental data by the following equation:

$$\gamma(h)=\frac{1}{2N(h)}\sum_{i=1}^{N(h)}[Z(x_i+h)-Z(x_i)]^2 \quad \cdots\cdots\cdots\cdots(3.4)$$

Where N(h) is the number of sample pairs separated by the lag (h).

To know the value of the semi-variogram at any distance h, a model fitting process is used to approximate the experimental semi-variogram by a continuous function. The main feature of the semi-variogram model is that it increases steadily until it reaches an asymptotic value (called the sill) around which it stabilizes. The sill corresponds to the variance of the measured data, and the distance at which it is reached is called the range. Any two points that are farther apart than the range do not bear any correlation to each other.

To analyze the spatial data of any aquifer, one would follow the following four main steps:

1. Analyze the measured data using its statistical parameters including the mean, the median, the standard deviation and the probability distribution. The mean is assumed to be constant in the absence of any drift.
2. Define the spatial correlation structure of the data by constructing an experimental semi-variogram.
3. Fit a theoretical model to the experimental variogram.
4. Use kriging to estimate the values of the sought groundwater parameters at unsampled locations.

The public domain geostatistical package GEO-EAS was used in this study to develop experimental semi-variograms and to fit theoretical models to them. This widely used package was developed by the U.S. Environmental Protection Agency (EPA).

3.3 Sequential Gausssian Simulation

To generate the hydraulic conductivity fields needed to train the intended neural network, a random field generator capable of building alternative, equally probable, conditional realizations are needed. Since the variable of concern (hydraulic conductivity) is known to posses a lognormal distribution [Freeze and Cherry 1979], a stochastic approach capable of simulating normal distributions is needed. The Sequential Gaussian Simulation (SGS) method was chosen to perform this task. It is perhaps the most straightforward algorithm for generating conditional realizations of a multivariate Gaussian field. The algorithm utilizes a semi-variogram to characterize the available data and simple kriging to sequentially simulate the unsampled data. A brief description of the steps undertaken by the SGS algorithm is given herein. These steps are also represented in Figure 3.2.

1. Transform the available hydraulic conductivity data into y-data with a standard normal distribution, and define the semi-variogram model for the y-data.
2. Define a random path that visits each cell of the grid once. (The path need not be regular).
3. Use kriging with the defined semi-variogram to determine the mean and the variance of the variable at the first unsampled point.
4. Draw a simulated value $y(x_1)$ from the parameters obtained in step 3.
5. Add the simulated value $y(x_1)$ to the data set. This newly generated point will be added to the data set of sampled points, and will be used for all subsequent generations.
6. Proceed to the next cell and repeat steps 3 through 5 until all the cells are simulated.

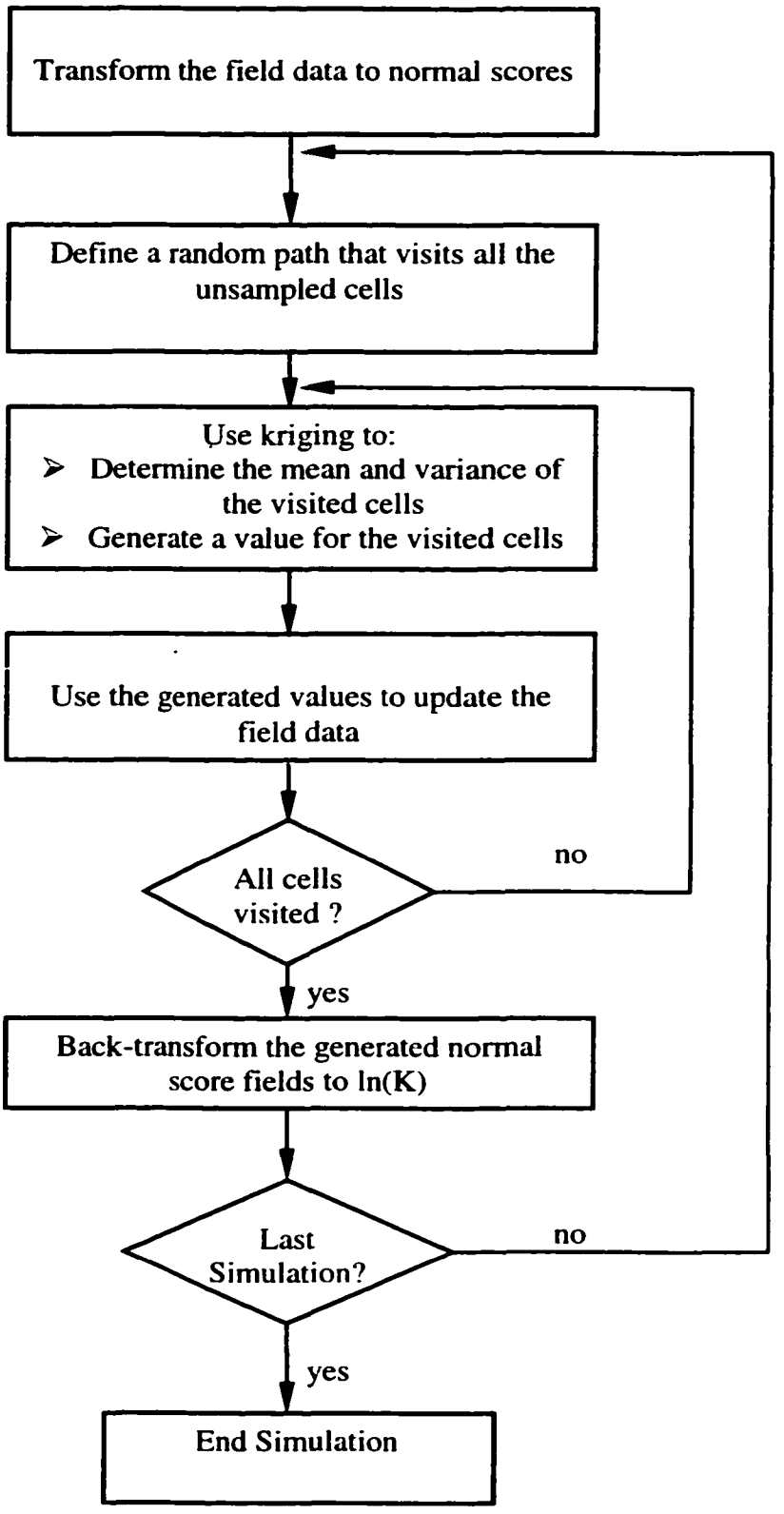


Figure 3.2 Sequential Gaussian Simulation algorithm

7. Back transform the simulated normal values (y-data) into simulated hydraulic conductivity values.
8. If (L) multiple realizations are desired, steps 2 through 7 of the algorithm are repeated L times with a different path for each realization.

GSLIB is the name of a geostatistical software library developed at Stanford. It served as the basis for some of the most commonly used public-domain geostatistical software such as GEO-EAS [Englund and Sparks 1991] and ISIM3D [Gomez-Hernandez and Srivastava 1990]. SGSIM is the name of the GSLIB routine that is used to perform the Sequential Gaussian Simulation. Like the rest of the GSLIB programs it is written in ANSI standard FORTRAN 77. The SGSIM routine is capable of generating both conditional and unconditional simulations.

3.4 Feedforward Neural Networks

Artificial Neural Networks (ANNs) are universal estimators of multivariate nonlinear mappings that are capable of learning and generalizing from examples (training data). The key to successfully training an Artificial Neural Network is choosing the right network architecture and training algorithm. A feedforward artificial neural network is used in this study to approximate the relation between hydraulic conductivity/Transmissivity values of the region in question, and the resulting hydraulic conductivity values. Feedforward networks are a subclass of layered networks in which there is no intra-layer connections, and whose main feature is that connections are allowed from a node in layer "i" only to nodes in layer i+1 (Fig. 3.3).

Feedforward neural networks are among the most common neural networks in use [Mehrotra et al. 1996], they were chosen to be used in this study due to the fact that they are simple, are easily trained, and can be readily inverted. The feedforward process from

which the name was derived involves presenting an input pattern to input layer neurons that pass the input values into the first hidden layer. Each of the hidden layer nodes (neurons) computes a weighted sum of the inputs, passes the sum through the transfer (activation) function, and presents the result to the next layer until the output layer is reached.

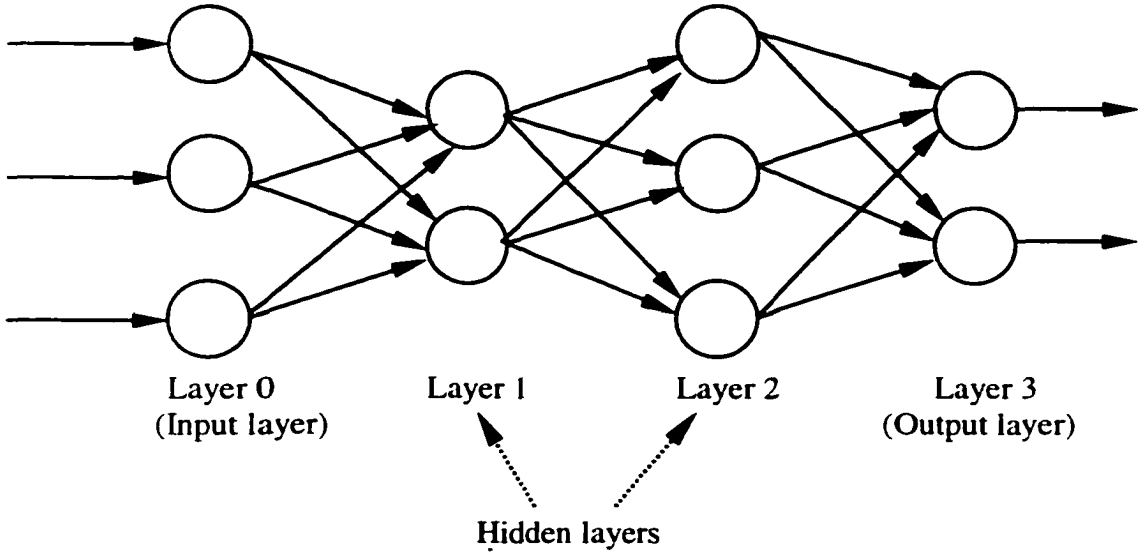


Figure 3.3 A feedforward 3-2-3-2 neural network

Determining the architecture of a neural network involves the determination of the number of layers in the network, as well as the number of nodes (neurons) in each layer. It also entails the determination of the type of the transfer (activation) function to be used in each layer. Transfer functions can be defined to have certain characteristics, however, the three most commonly used transfer functions are the tan-sigmoid (tansig), log-sigmoid (logsig), and the linear transfer functions. These functions are continuous and differentiable every where, their equations and features are given in Table 3.1.

Table 3.1 Transfer functions properties

Name	Equation	Features
Linear (lin)	lin(x) = m*x	output can have any value
Log-sigmoid (logsig)	logsig(x) = 1/(1+exp(-x))	0 ≤ output ≤ 1
Tan-sigmoid (tansig)	tansig(x) = tanh(x)	-1 ≤ output ≤ 1

Once the initial network architecture is defined, random weights are assigned to each link in the net, and the training process starts. A training algorithm that modifies the network weights to minimize the error between the desired and actual output of the network is needed. Once trained, the network weights are frozen and can be used to compute output values for new input samples. If the target error is not achieved, the network architecture has to be modified, and the training process restarted. The training algorithm should work irrespective of the weight values that preceded training.

In this study, the training process was performed by the commercial package MATLAB, which includes a number of training algorithms including the backpropagation training algorithm. This is a gradient descent algorithm that has been used successfully and extensively in training feedforward neural networks. The term backpropagation refers to the manner in which the gradient is computed for multilayer networks. A number of MATLAB scripts and FORTRAN codes were written to prepare the data for training and to create an interface between the various software packages that are used in this research. The flow chart shown in figure 3.4 depicts the steps that were carried out to train the artificial neural network.

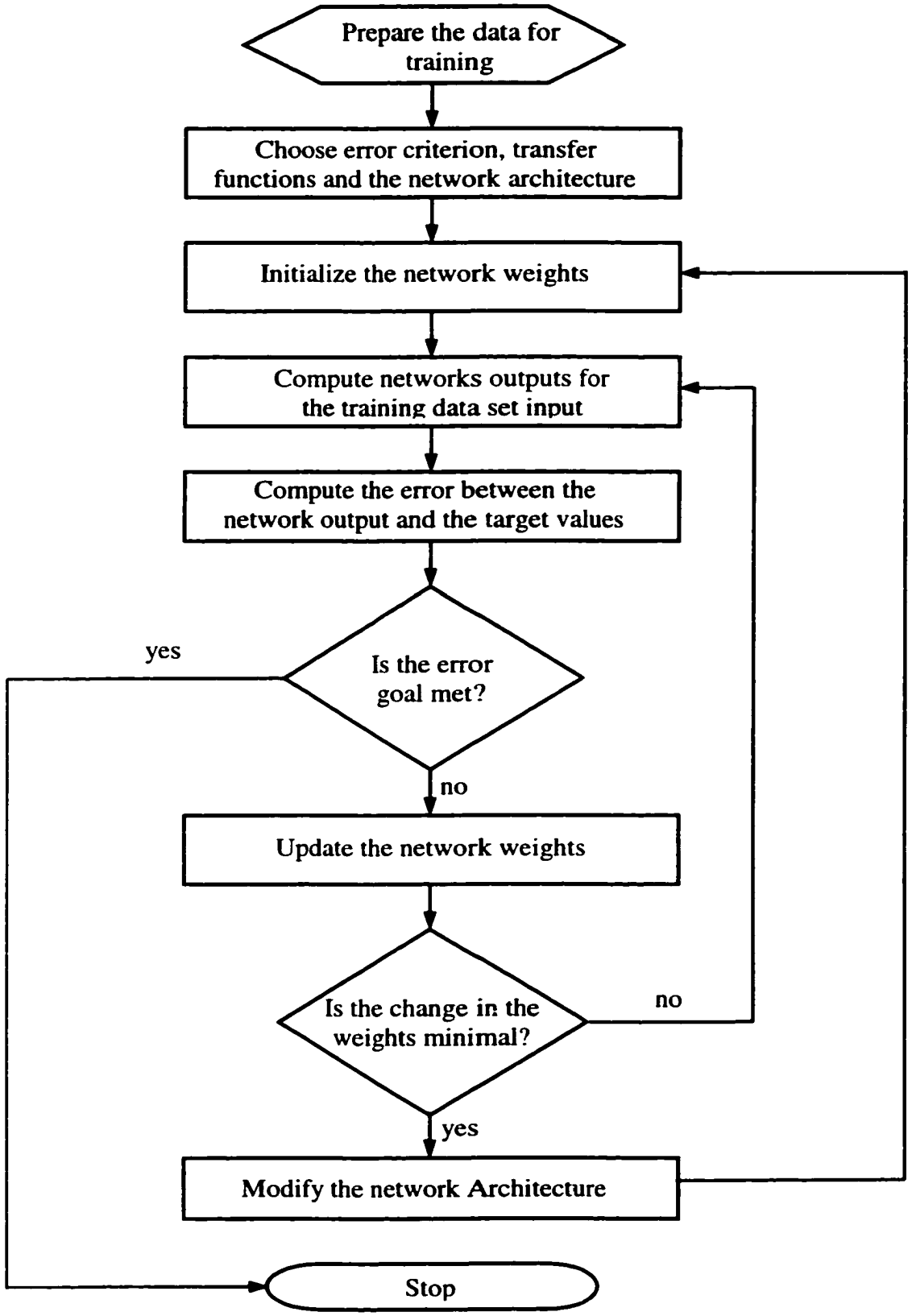


Figure 3.4 Network training flow chart

3.5 Network Inversion

One of the key elements to the implementation of the methodology presented in this research is the inversion of the trained Artificial Neural Network. The trained ANN is expected to be capable of producing the resulting hydraulic head values when a complete set of transmissivity values is introduced to it. The goal of the inversion algorithm is to use the trained ANN as a vehicle to calibrate an initial transmissivity guess, so that it honors the field transmissivity values and at the same time reproduces the field hydraulic head values (target) when used in a groundwater flow model.

The inversion algorithm starts with a transmissivity field generated from the field transmissivity values. This field dubbed initial guess is put through the trained ANN to compute the resulting hydraulic head values. The ANN output for each cell is compared to its target value, and the error "E" is computed according to a differentiable prescribed criterion. The error "E" is then back-propagated through the network to update the transmissivity guess. The process is repeated until it converges. Figure 3.5 show a schematic of the ANN inversion process.

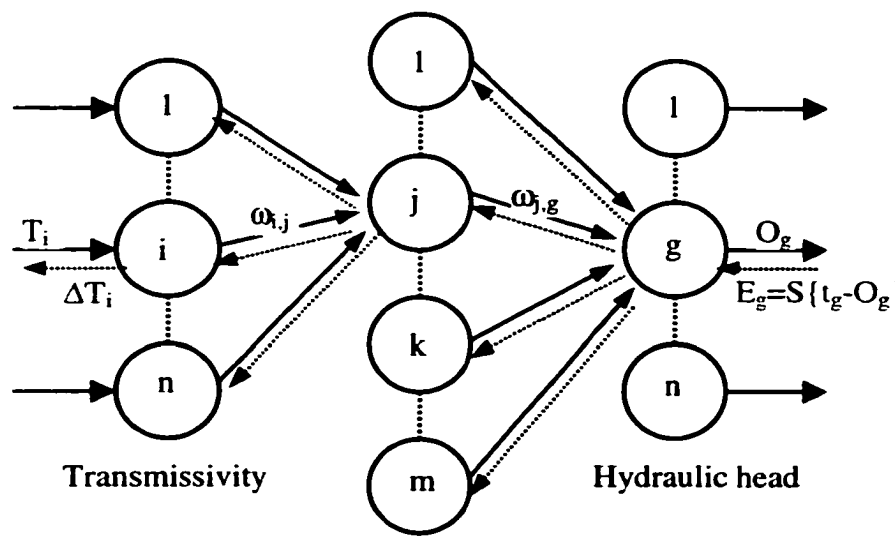


Figure 3.5 Schematic of network inversion process

The ANN inversion process can be cast mathematically as a minimization problem in which an input vector T^n is identified such that the network error "E" given by $E=S\{F(T^n)-t\}$ is minimized.

Where

$S\{x\}$ = a differentiable mathematical operation on x

$F(T^i)$ = ANN output for input vector T^i

t = hydraulic head target values

This optimization problem can be addressed by doing a gradient search in the input space. The gradient search starts with an initial input T^0 which is updated by an amount proportional to the partial derivative of the error "E" with respect to T

$$T^{i+1} = T^i - \frac{\partial E}{\partial T^i} \quad \text{.....(3.5)}$$

The applicability of this procedure hinges upon the fact that the partial derivative $\frac{\partial E}{\partial T}$ exist, and from this condition stems the need to have a differentiable error criterion, as well as differentiable transfer functions in the ANN to be inverted. For a two layer ANN similar to the one shown in figure 3.5, the network output "O = $F(T^n)$" is given by

$$O_g = Fn_2\{\sum_{j=1}^{m} \omega_{j,g} \cdot (Fn_1\{\sum_{i=1}^{n} \omega_{i,j} T_i\})\} \qquad for\ g = 1,.....n \quad \text{.....(3.6)}$$

Where; Fn_i = transfer function of the i^{th} layer.

Using the chain rule, the value of $\frac{\partial E}{\partial T}$, for this network can be computed as follows;

$$\frac{\partial E}{\partial T} = \sum_{g=1}^{n} \left\langle \frac{\partial E}{\partial O_g} \cdot \left[Fn'_2 \left\{ \sum_{j=1}^{m} \omega_{j,g} \cdot \left(Fn_1 \left\{ \sum_{i=1}^{n} \omega_{i,j} T_i \right\} \right) \right\} \cdot \omega_{j=1,\ldots m,g} \right] \cdot \left[Fn'_1 \left\{ \sum_{i=1}^{n} \omega_{i,j} T_i \right\} \cdot \omega_{i=1\ldots n,j} \right] \right\rangle \quad \text{...(3.7)}$$

Where Fn_i' = the derivative of the transfer function Fn_i.

The matlab script "INVERSE" (Appendix A) was written to invert a trained ANN. The flowchart of this script is shown in figure 3.6.

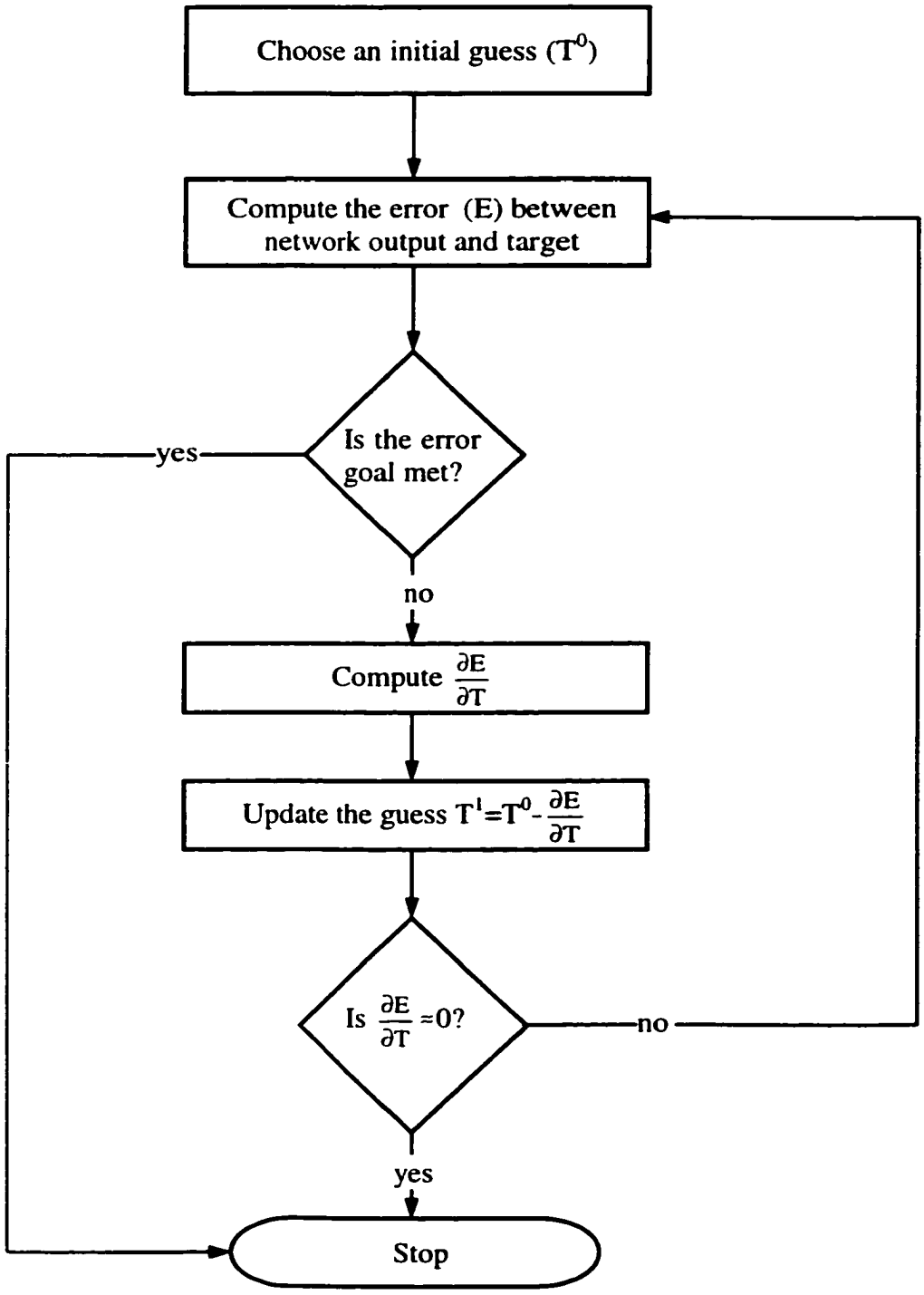


Figure 3.6 ANN inversion algorithm

CHAPTER 4

SOLVING THE INVERSE PROBLEM

4.1 Designing the Hypothetical Problem

It is a common practice among groundwater researchers to use synthetic case studies to evaluate inverse problem solving techniques. The main reason a synthetic case study is preferred to a real case study, is that the parameters that are sought by solving the inverse problem are known in a synthetic case, but can not be obtained if a real case is used. Hypothetical cases thus allow for a better evaluation of the accuracy of the technique used to solve the inverse problem.

In a typical study, the aquifer parameters (i.e. boundary conditions, aquifer thickness, hydraulic conductivity values, starting hydraulic head values, and aquifer stresses) are designated. These parameters are put in a groundwater model and the resulting hydraulic head values are obtained. The aquifer parameters that are to be estimated by the inverse technique are assumed to be unknown, and they are solved for by using the parameters and the hydraulic head values that were assumed to be known. The parameters generated by the inverse technique are compared to those of the original synthetic case to evaluate the inverse solution methodology used.

A synthetic case study was designed to test the proposed inverse methodology. Three main considerations were observed when designing the case study, these are:

1. The case study should be sufficiently simple to allow testing without excessive computer costs, and tedious interpretation of the results.
2. The case study should be sufficiently complex to help scrutinize the developed inverse technique.
3. The case study should provide for a realistic representation of aquifer parameter values and boundary conditions.

A square confined aquifer having an area of 2.25 Km^2 was chosen to perform the study. No-flow boundaries, a constant thickness of 15 meters, and a constant recharge rate of 10 cm/year characterize the region. The aquifer domain was discretized to a 15x15 uniformly spaced grid of 100 meter, and each cell within the grid was assigned a number (Figure 4.1). Seven cells were designated constant hydraulic head values. The specified head boundary formed by these cells may be conceptualized as a river running through the aquifer system. The hydraulic head values associated with this boundary are given in Appendix B. Elsewhere in the active part of the grid, a starting hydraulic head of 150 meters was assigned. Two pumping wells were assigned to cells number 116 and 123. The discharge rates of these two wells were 5,000 m^3/day and 10,000 m^3/day respectively. The aquifer was assumed to be in a sandy soil with hydraulic conductivity values ranging between 2.88 m/day and 75 m/day. With the maximum and minimum hydraulic conductivity values defined, and assuming a spherical model with a nugget value of 0.3 and a range of 450 m, an unconditional hydraulic conductivity field was generated by using Sequential Gaussian Simulation (SGS). The generated field assigned a unique hydraulic conductivity value to each of the 225 cells in the discretized aquifer domain (Appendix B). These values were used in MODFLOW to obtain the steady state hydraulic head values in the aquifer as a result of the applied stresses (Appendix B).

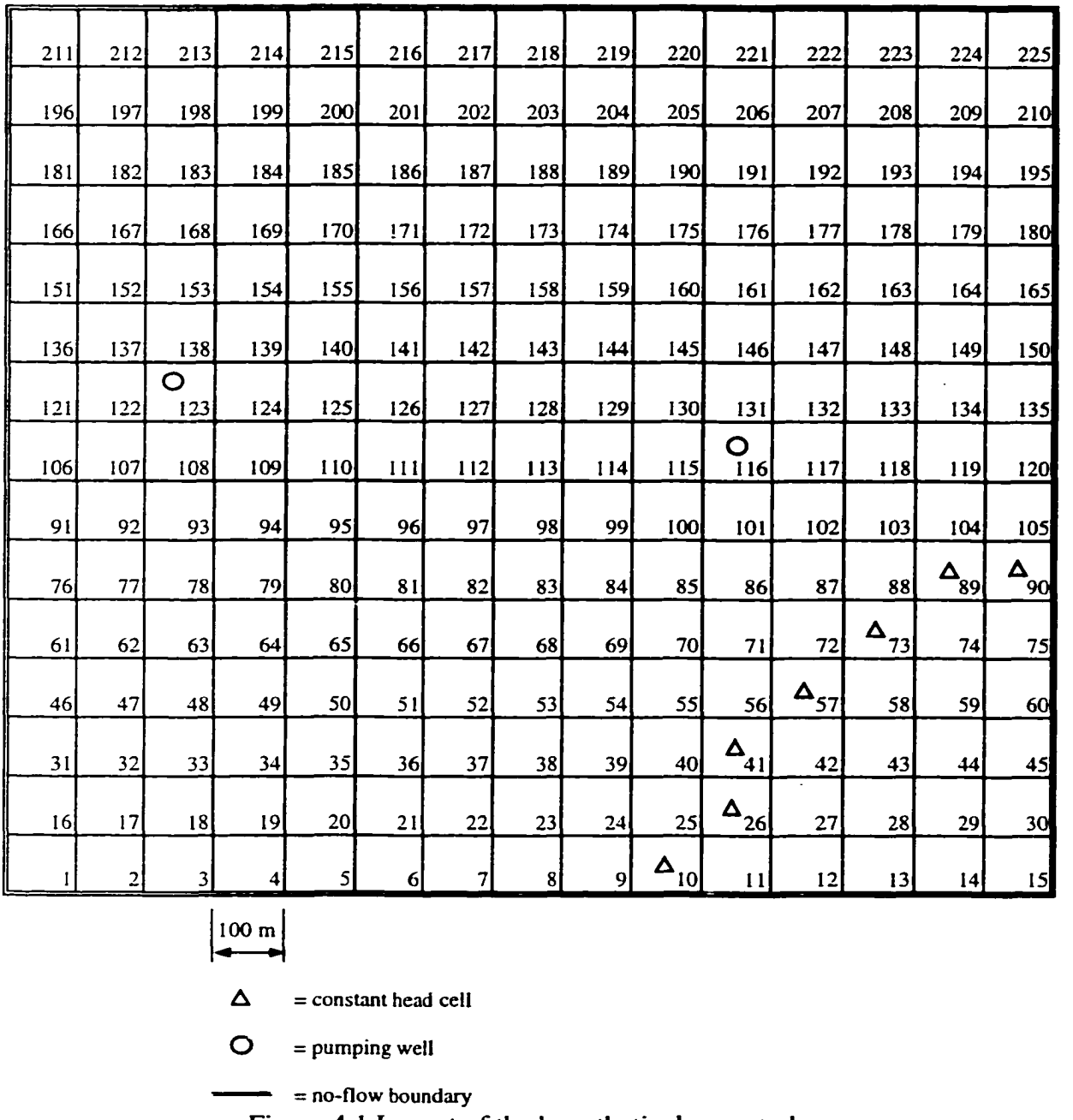


Figure 4.1 Layout of the hypothetical case study

Solving the inverse problem entails characterizing the hydraulic conductivity map of the region in question from limited hydraulic conductivity and hydraulic head values. In order to choose the cells in which the hydraulic conductivity and hydraulic head values were assumed to be known, 53 random numbers between 1 and 225 were generated. The first 29 generated numbers were adopted as the cell numbers where both the hydraulic

conductivity and the resulting hydraulic head values were known. The remaining 24 numbers were considered to be the numbers of the cells where only the resulting hydraulic head values were known. Figure 4.2 shows the location and numbers of these cells.

4.2 Generating the Training Data

Artificial Neural Networks (ANNs) are mapping functions that can be trained to map any non-linear complex relation. Sets of input data and their corresponding output vectors are needed to train the network. Once properly trained, the network provides a data driven model which is capable of giving reasonable answers when presented with input vectors that have never been encountered during the training process.

In order to train an artificial neural network to simulate an implicit relation like the one between transmissivity and hydraulic head as displayed in the groundwater flow equation, sets of different transmissivity values and the corresponding hydraulic head values have to be generated. This is done in this study by stochastically generating alternative equally probable hydraulic conductivity fields from the available 29 hydraulic conductivity field values, and then using the modified MODFLOW program to generate the corresponding hydraulic head fields.

A statistical analysis of the 29 available hydraulic conductivity (K) data shows that their values range from 6.67 m/day to 65.14 m/day with a mean value of 25.69 m/day, and a standard deviation of 17.94. The statistical analysis of the natural logarithm of available hydraulic conductivity data (ln(K)) reveals that the values of "ln(K)" range between 1.90 and 4.18 with a mean value (μ) of 3.01 and a standard deviation (σ) of 0.70.

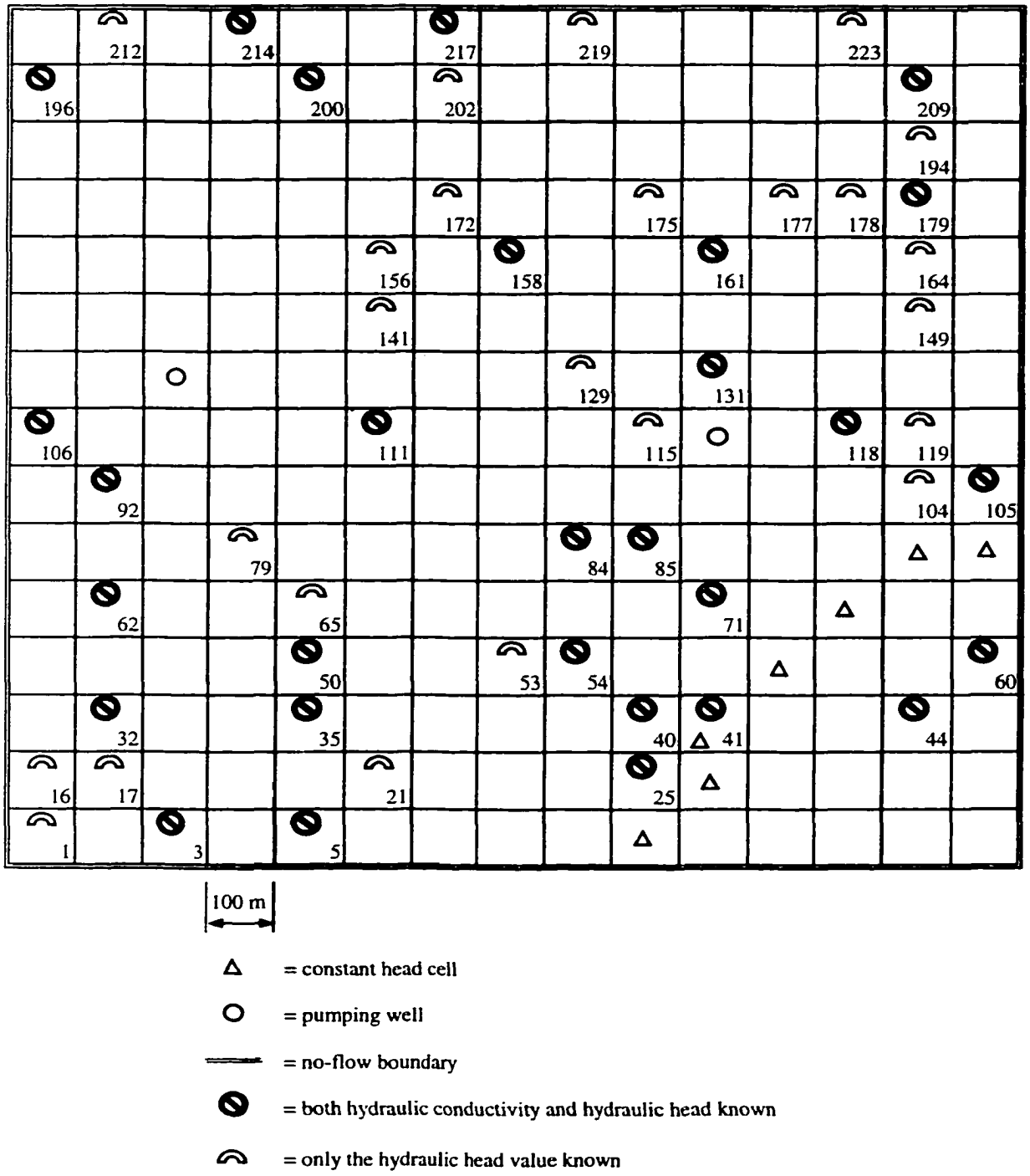


Figure 4.2 Location of known parameters

As is often the case with field data, the available hydraulic conductivity values are spatially clustered. To obtain a histogram and distribution that is representative of the entire model area, declustering weights are assigned to each of the available data points, whereby the values in areas with more data receive less weight than those in sparsely sampled areas. The GSLIB subroutine DECLUS was used to decluster the ln(K) data. The DECLUS subroutine is based on the cell size declustering method [Deutsch 1989], in which the weights are inversely proportional to the number of samples falling within the same area (Appendix C)

The Sequential Gaussian Simulation algorithm (SGSIM) work with the normal scores of the field data to perform the simulations in normal space, and then back-transform the simulated values. The GSLIB subroutine NSCORE was thus used to transform the 29 available declustered ln(K) data to a distribution of normal scores with a mean of 0.13 and a variance of 0.83 (Appendix C).

Defining the semi-variogram structure of the normal score is essential for performing the Sequential Gaussian Simulations. A geostatistical analysis of the normal scores of the field data (Appendix C) was performed to represent its correlation structure. A Gaussian model was fitted to the experimental variogram (Figure 4.3). The fitted model has a sill value of 0.90, a nugget value of 0.03, a spatial correlation range of 730 m, and is defined by the following equation:

$$\gamma(h) = 0.90[1 - \exp(-\left(\frac{3h}{730}\right)^2)] + 0.03 \quad \text{.....(4.1)}$$

Where;

γ = semi-variogram value

h = lag distance [L]

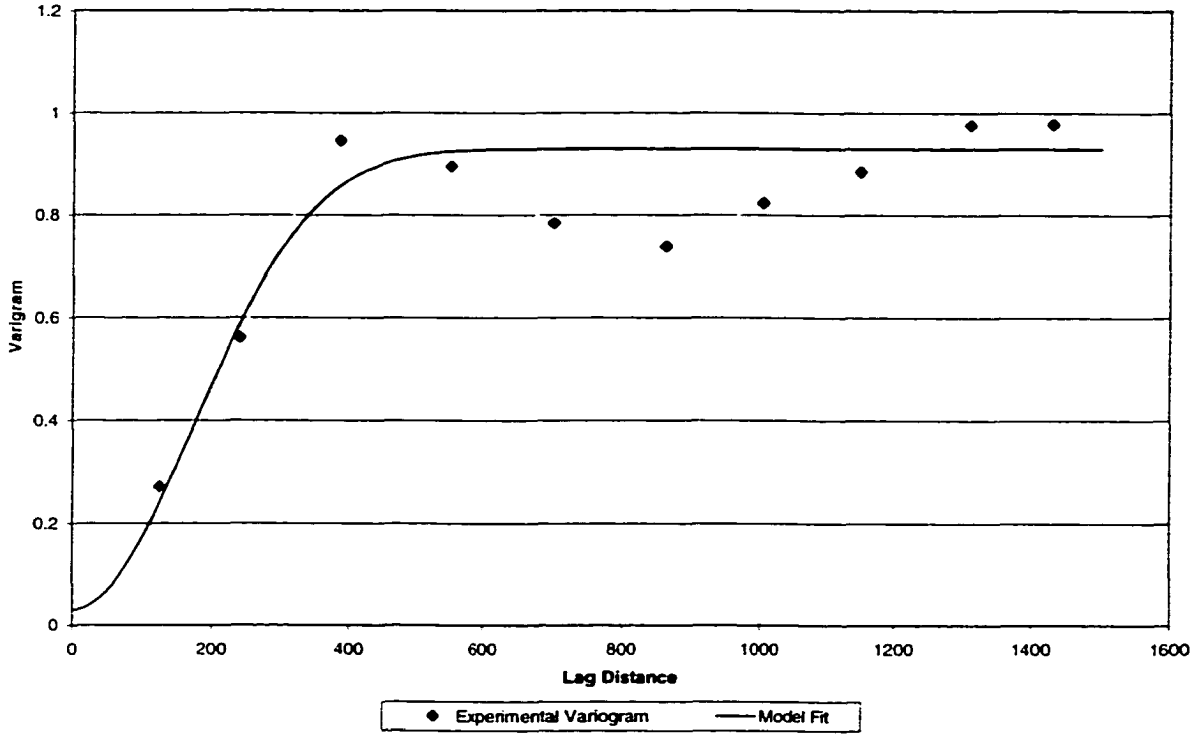


Figure 4.3 Semi-variogram of normal scores

The Sequential Gaussian Simulation algorithm SGSIM was used in conjunction with the developed semi-variogram to generate 1,500 conditional realizations of the hydraulic conductivity field over a grid of 15×15 cells covering the whole study with 225 one hundred meter square cells. The values of ln(K) generated by the random field generator SGS were trimmed to a maximum value of 4.76 and a minimum value of 1.26. These two limits correspond to the mean (3.01) plus or minus two and half standard deviations (2.5×0.7). This range covers about 99% of the data that can be sampled from the statistical distribution of ln(K) data, and includes both the minimum and maximum values measured in the field.

The large number of simulations generated (1,500) was chosen to cover a large range of possible hydraulic conductivity fields, and to produce enough data sets to train, validate, and test the intended artificial neural network. The generated hydraulic conductivity fields were then transformed to transmissivity fields by multiplying them by the aquifer thickness. MODFLOW was subsequently used to generate the hydraulic head fields that correspond to each of the 1500 transmissivity fields.

4.3 Data Preparation

The generated training data were arranged into two (225×1,500) matrices. One matrix (INPUT) contains the 1,500 generated transmissivity fields (i.e. 1,500 sets of input neurons), and the second matrix contains the corresponding 1,500 hydraulic head fields (i.e. 1,500 sets of output neurons). Each vector (column) of the matrix contains the transmissivity/hydraulic head values for the 225 cells of the hypothetical case study, and each cell is to be used as a network input or output neuron.

The values of the 1,500 input-output pairs were preprocessed to fall in the range between zero and one. For each cell, the maximum and minimum generated values were obtained, and the values were scaled to lie between 0.05 and 0.95 leaving some room for extrapolation. The importance of preprocessing the output of the neural network training data stems from the fact that the most commonly used transfer functions in artificial neural networks (logsig and tansig) generate outputs that range between 0 and 1, and −1 and 1 respectively. It was also discovered during the course of inverting the trained network, as will be explained latter, that it is essential to preprocess the input vectors so that they lie between 0.05 and 0.95.

The training data was divided into four subsets. The first subset is the training set, which consist of 1000 input-output pairs (patterns) and is used for computing the gradient and updating the network weights. Each pattern comprises a transmissivity field (the inputs), and its corresponding hydraulic head values (the targeted outputs). The second subset which consists of 200 input-output pairs is the validation set. The error on this set dubbed "validation error" is monitored during the training process. The validation error will normally decrease during the initial phase of the training as does the training set error. However, when the network begins to over-fit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights at the minimum of the validation error are returned (Figure 4.4). This method does not suggest using the validation set for training. Network weight changes are computed solely on the basis of the network's performance on the training set. With this stopping criterion however, final weights do depend on the validation set in an indirect manner.

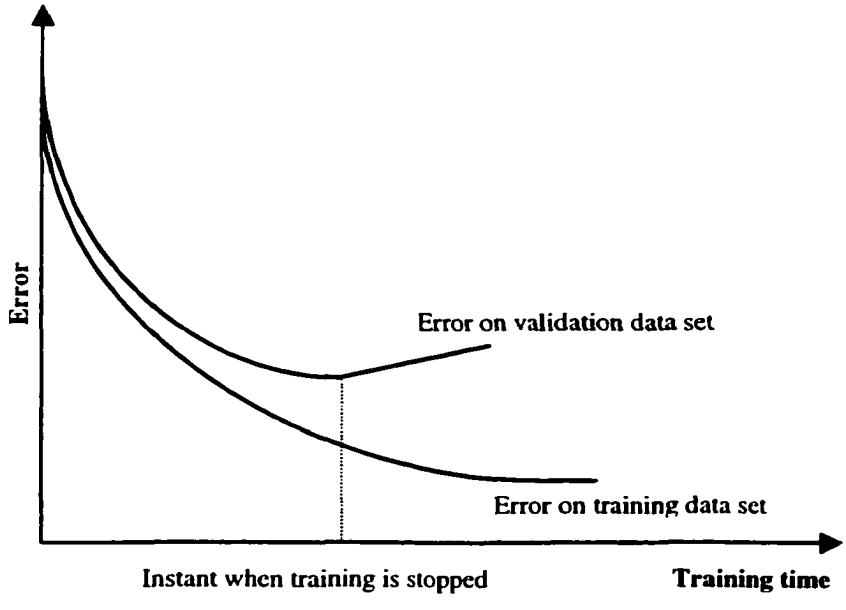


Figure 4.4 Change in error with training time, on training and validation sets

The third subset is the test set and it consists of 200 patterns. This subset is also not used during the training. Its error however is calculated and monitored during the training process to test the consistency of the network. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this may indicate a problem in the ability of the network to generalize for different data sets. This is usually an indication of poor division of the training data set.

The fourth subset is the analysis set and it consists of 500 patterns. This set is used in the post-training analysis of the trained network and is composed of the input-output pairs that were not used in the actual training process, which may include the validation and test subsets. The purpose of the post-training analysis is to investigate and quantify the ability of the trained network to track the output targets. This goal cannot be fully achieved by monitoring the errors on the training, validation and test sets. Such monitoring generally assesses the average performance of the network, without detailing and quantifying the ability of each of the network outputs to track their targets reasonably. The post-training analysis is achieved by performing a regression analysis between the neural network's response and the corresponding target. The regression returns three parameters, the slope, the y-intercept, and the correlation coefficient (R-value) between the network outputs and the targets. If the network outputs are exactly equal to the targets (perfect fit), the slope would be one, the y-intercept would be zero, and correlation coefficient would be one. Of the three parameters, the correlation coefficient and the slope are the most significant, for they provide a good measure of the network's ability to track and map the changes that occur in the targets when the inputs are changed. The post-training analysis is conducted for each neuron separately, thus

generating a set of regression and slope parameters equal to the total number of network outputs.

4.4 Determining the ANN Architecture

Determining the ANN architecture is for the most part a problem dependent trial and error process. The process entails determining the number of input neurons, the number of output neurons, the network's transfer (activation) functions, the error function used to monitor the network's performance, the number of hidden layers, as well as the number of neurons in the hidden layers. Some of these parameters namely the number of input and output neurons, are decided by the nature of the problem. For the problem at hand, the number of input and output neurons is equal to the number of cells to which the problem domain has been discretized. The input layer will consist of the 225 values comprising the transmissivity field, and the output layer will consist of the resulting 225 hydraulic head values computed from MODFLOW.

The error criterion provides a measure of the total error between the network's output and the training set targets. The error criterion does not play a direct role in determining the network's weights, and the only condition for choosing it is that it should be a differentiable function. For the problem at hand, the mean square error criterion was chosen such that the network's error is computed using equation 4.2.

$$E = \frac{1}{P} \cdot \frac{1}{N} \sum_{p=1}^{P} \sum_{g=1}^{N} (t_g - O_g)_p^2 \quad \cdots\cdots\cdots\cdots (4.2)$$

Where;

P = number of training patterns (=1000)

N = number of neurons in the output layer (=225)

t_g = target value for cell number "g"

O_g = ANN output for cell number "g"

From a theoretical point of view, increasing the number of hidden layers in an ANN enhances the ability of the network to model complex solution surfaces, and model the relation between the transmissivity and hydraulic head fields. Indeed it has been proven that a network with two hidden layers of neurons operating sigmoidal activation functions is capable of modeling any solution surface of practical interest [Lapedes and Farber 1988]. From a practical stand point, a network with a single hidden layer that has enough neurons is usually able to provide an acceptable model, and is less cumbersome to invert. It will be shown during the course of this research that an ANN with one hidden layer can be successfully used to model the relation between the transmissivity and hydraulic fields as described by the groundwater flow equation.

Since both the input and output values of the network were scaled to lie between 0 and 1 and bearing in mind the work of Lapedes and Farber [1988], a log-sigmoid transfer/activation function was used in the two layers (hidden and output layers) of the ANN.

Determining the appropriate network architecture has thus far been reduced to determining the number of neurons in the hidden layer. The goal is to find the minimum number of neurons in the hidden layer which would ensure that the neural network will provide a generalized model capable of mapping the relation between the data sets which were not used in the training process. It is evident from the literature [Mehrotra, et al. 1997] that a large number of hidden neurons can produce a neural network capable of memorizing the training data. Such a network will perform poorly on the validation and test data, while closely fitting the training data. In addition a large number of hidden

neurons will slow down the operation of the network, both during training and in the course of inverting it to solve the inverse problem. On the other hand, a network with too few hidden neurons will not have the resources to learn and model the relation of interest.

4.5 Training the ANN

4.5.1 Determining the Network's Weights

There is no direct and precise way for determining the most appropriate number of neurons to include in the hidden layers of a neural network. The training process starts with a small number of neurons that are gradually increased until there is no further significant increase in the network's performance. The training for the case study problem was started using five neurons in the hidden layer. The training algorithm randomly selected the initial weights, and the network training started for the training set which comprises 1000 training patterns. Each pattern was input into the network in turn, and the output was compared to the target solution thus providing a measure of the error in the network for the whole training set. The gradients of the error were calculated for each training pattern and were added for the whole training set. The weights were adjusted accordingly so that on the next occasion the training patterns were presented to the network, the error would be reduced, and the output would get closer to that required. Each cycle in which the training set was presented to the network and the weights were updated is called an epoch. The number of epochs in the training process was determined by the performance of the network on the validation set. As soon as the validation error started to rise or remained unchanged for five epochs in a row, the training process was terminated.

Figure 4.5 shows the training progress for the network with 5 neurons in the hidden layer. The graph shows that the errors of the validation and test sets closely followed the training set error during the whole training process. This indicates that the network was not powerful enough to memorize the training patterns. The network's ability to model the relation between the transmissivity and hydraulic head fields is yet to be scrutinized.

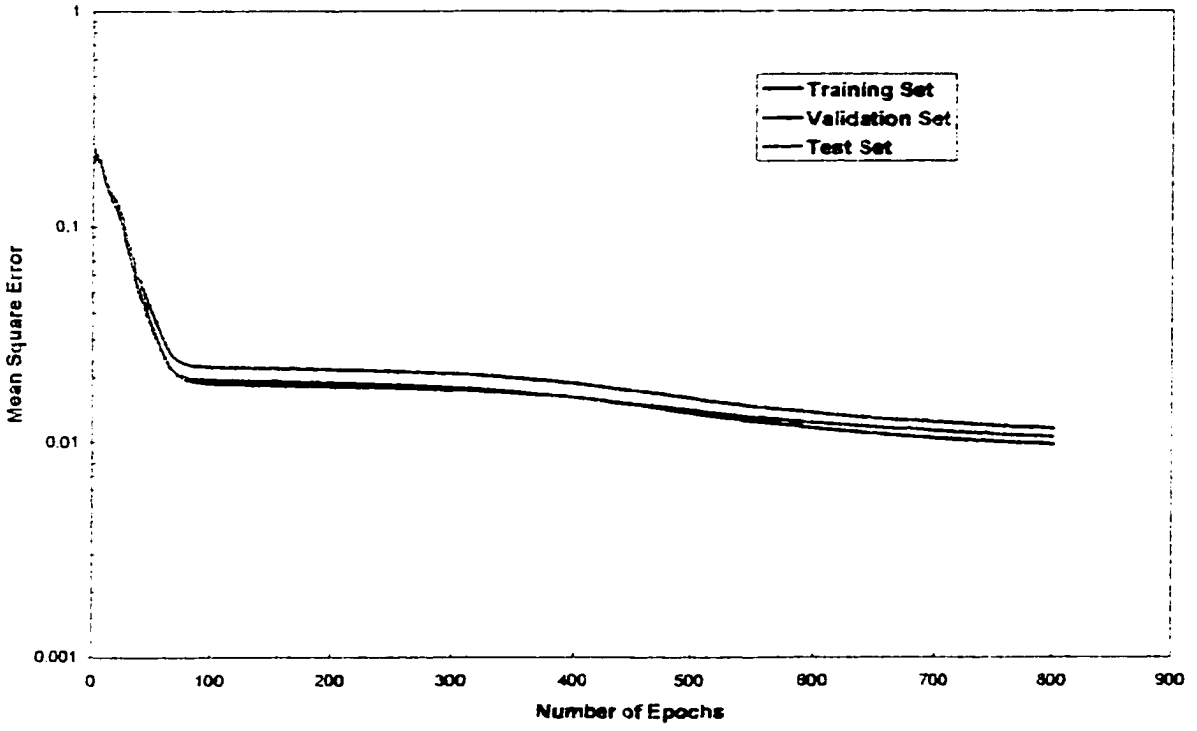


Figure 4.5 Error performance for the network with 5 hidden neurons

The network was also trained with 10, 25, 50, 75, 100, 150, 200, 225, 250, 300 and 325 neurons in the hidden layer. Figure 4.6 shows the training progress for the network with 300 hidden neurons. This plot shows that after 80 epochs the error performance on the validation and test sets departed from that of the training set. The

training continued however due to the fact that there was still an improvement in the validation set error. This network has the resources to memorize the training patterns, and it will always yield better results for the training patterns than for patterns that were not used in the training. The fact that the validation error followed the test error throughout the training process, is an indication of the ability of the network to generalize for different data sets.

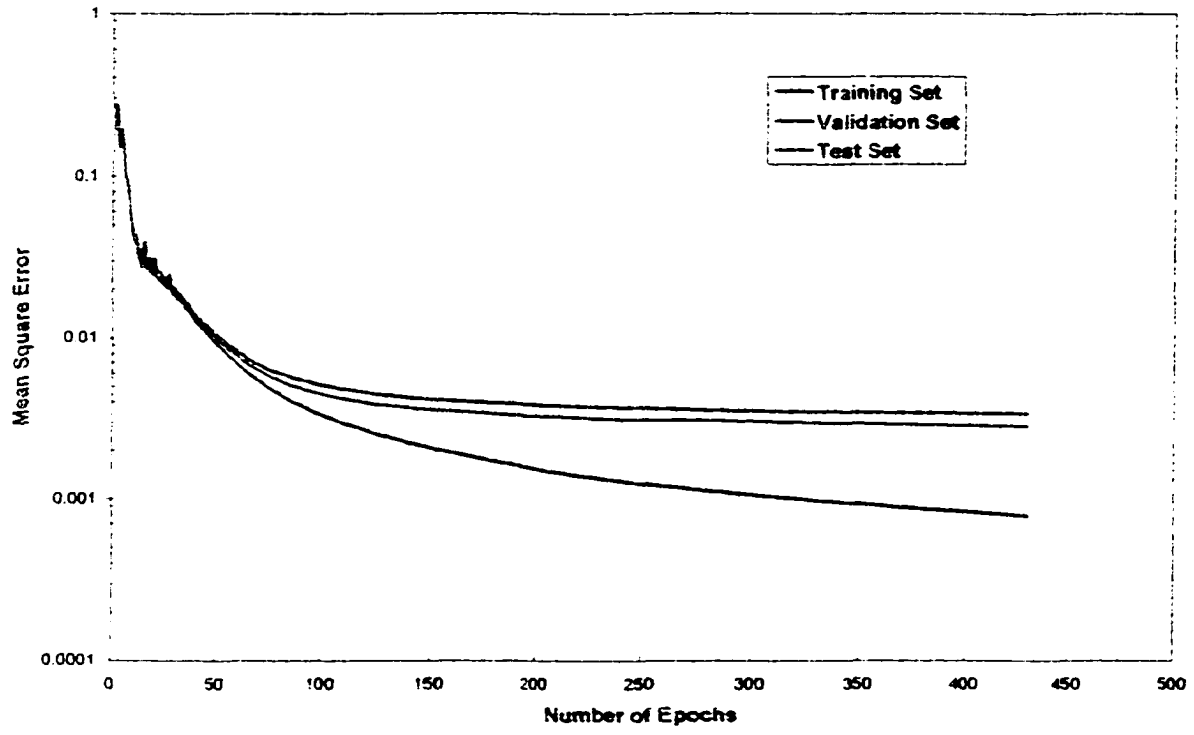


Figure 4.6 Error performance for the network with 300 hidden neurons

Even though the network's training error is a valuable tool in comparing two different network options, it provides little insight into the ability of the network to successfully map the desired relation. This is particularly true in this case due to the fact that the output value of each neuron had been independently scaled to lie between zero

and one. This makes it more difficult to interpret the actual magnitude of the network error. In order to effectively evaluate the twelve trained networks, post-training analysis was conducted for each one of them.

4.5.2 Post-training Analysis

Post-training analysis is instrumental in qualitatively and quantitatively evaluating the network's performance. The qualitative evaluation is achieved by visually comparing plots of the 500 analysis set targets for each neuron (cell) versus their corresponding network output. The quantitative evaluation is conducted by obtaining the correlation coefficients and slopes for each of these plots. Figure 4.7 depicts the post-training analysis results for network number one (featuring five hidden neurons) for four selected cells, namely cells number 10, 68, 163 and 200. With the exception of cell number 10, the graphs show that there was some correlation between the network's and MODFLOW outputs. Referring back to Figure 4.1, it can be seen that cell number 10 is one of seven designated constant head cells. Indeed cell number 10 has a constant hydraulic head value of 154.0 m. It can be seen (Figure 4.7) that the network estimates for this cell centered around the 154 m mark with a discrepancy of about ±0.5 m. The fact that the cell has a constant hydraulic head value lead to the unrealistic correlation coefficient of infinity shown in Figure 4.7. Not withstanding these results, the network was capable of producing a good (99.5% accurate) estimate for this cell and the six other constant head cells. Figures 4.8 and 4.9 show a summary of the post-training results for network number one for all the 218 variable head cells in the problem domain grid. The network was able to produce solutions that have an average correlation of 0.656 to the target MODFLOW values with a variance of 0.214. The average slope obtained from the post-

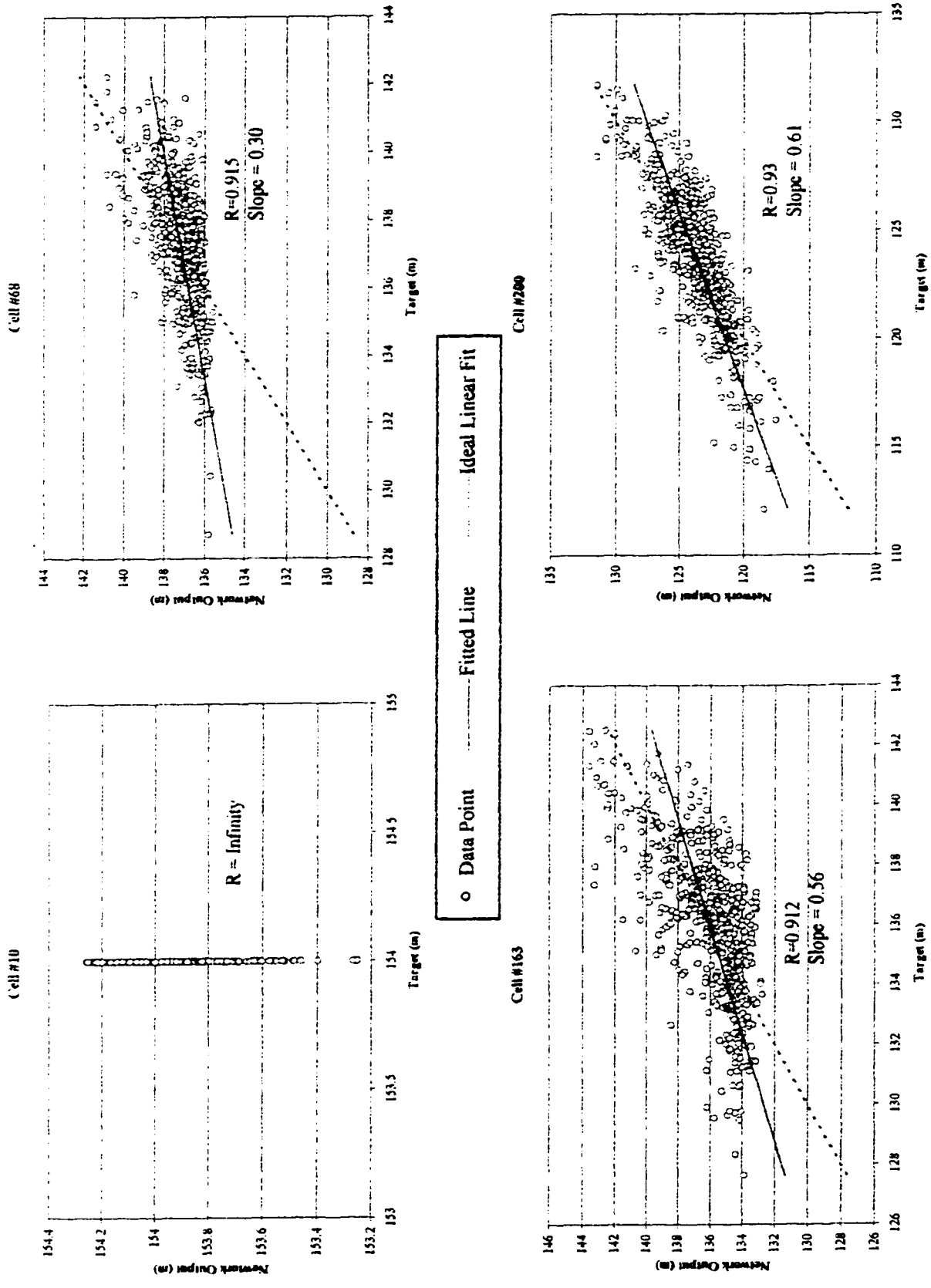


Figure 4.7 Sample of the post-training analysis results for network number one (featuring five hidden neurons)

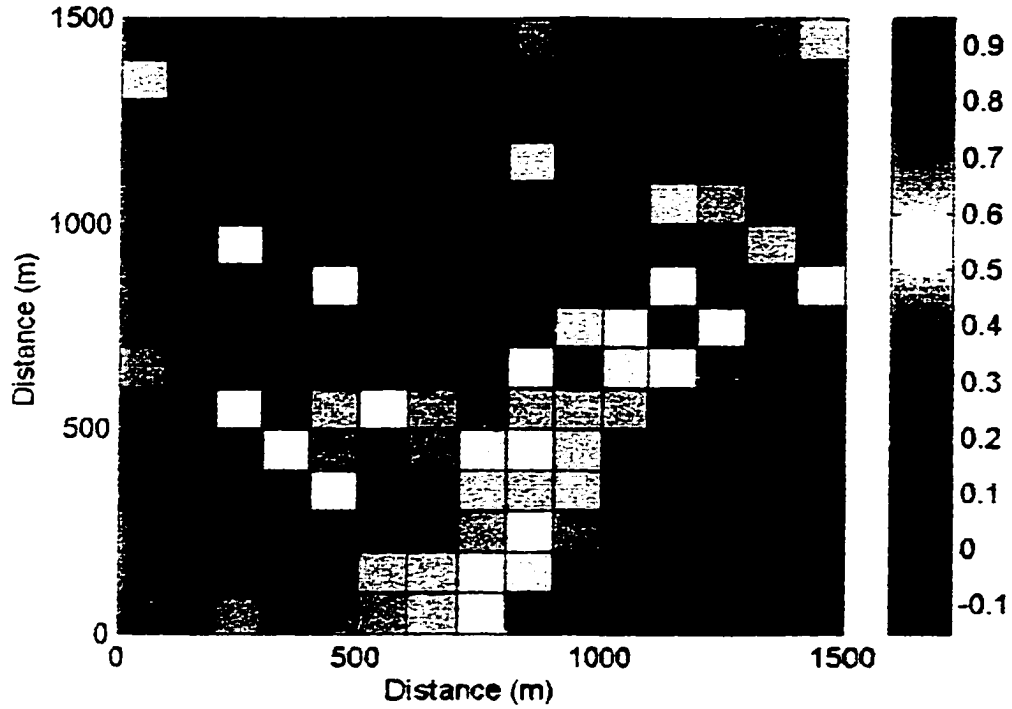


Figure 4.8 Correlation coefficient values from the post-training analysis of network number one (possessing five neurons in the hidden layer)

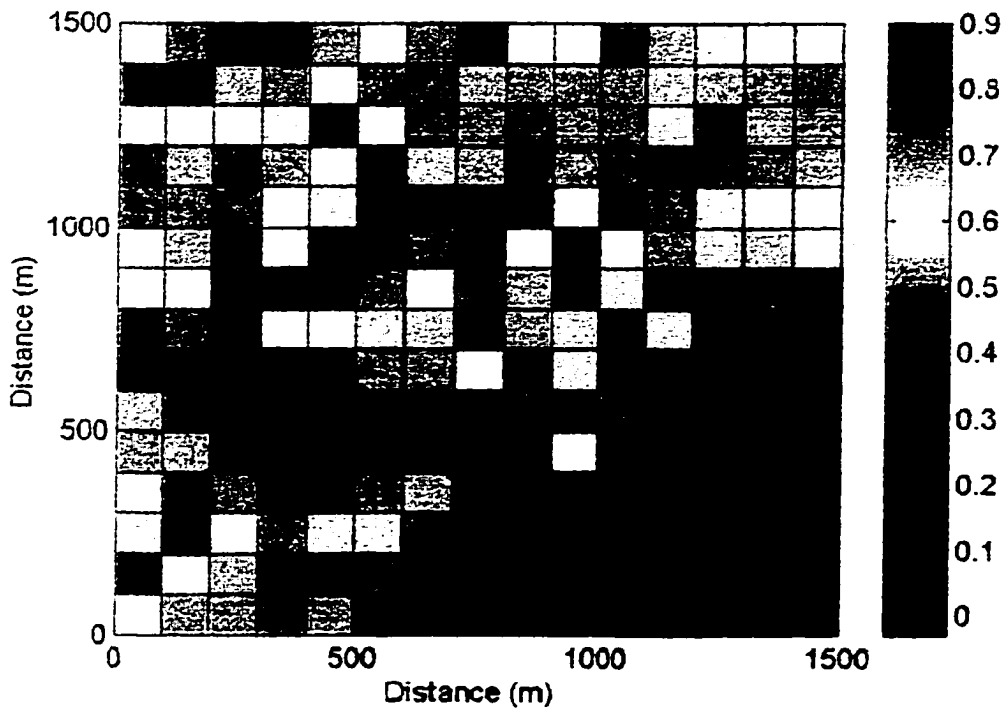


Figure 4.9 Slope values from the post-training analysis of network number one (possessing five neurons in the hidden layer)

training analysis was 0.501 with a variance of 0.211. These results indicate that network number one was not capable of modeling the relation between transmissivity and hydraulic head as intended, and that a better network with more neurons in the hidden layer is to be sought.

Figure 4.10 shows a sample of the post-training results for network number 11 (featuring 300 hidden neurons) for cells number 26, 68, 163, and 200. The plots show that there was a remarkable improvement from network number one, and that cell # 26 which is another constant head cell, was adequately estimated by this network. Table 4.2 summarizes the results obtained by this network for all seven constant head cells.

Table 4.1 Post-training analysis for constant head cells

Cell number	Target value (m)	Mean value from post-training analysis (m)	Standard deviation of post-training analysis estimates (m)
10	154.0	154.00	0.05
26	153.5	153.49	0.07
41	153.3	153.30	0.07
57	153.0	153.00	0.08
73	152.5	152.51	0.08
89	152.0	151.99	0.09
90	152.0	152.00	0.07

The summary of the performance of network number 11 can be seen in Figures 4.11 and 4.12 which feature the correlation coefficient and the slope values for all the 218 variable head cells. The average correlation coefficient was 0.914 with a variance of 0.015, while the mean slope was 0.905 with a variance of 0.029. From the plots it was observed that the worst correlation coefficient and slope values were at the two cells in which the pumping wells are located. These post-training analysis results thus indicate that network number 12 has the capability to reasonably track and model all of the 225

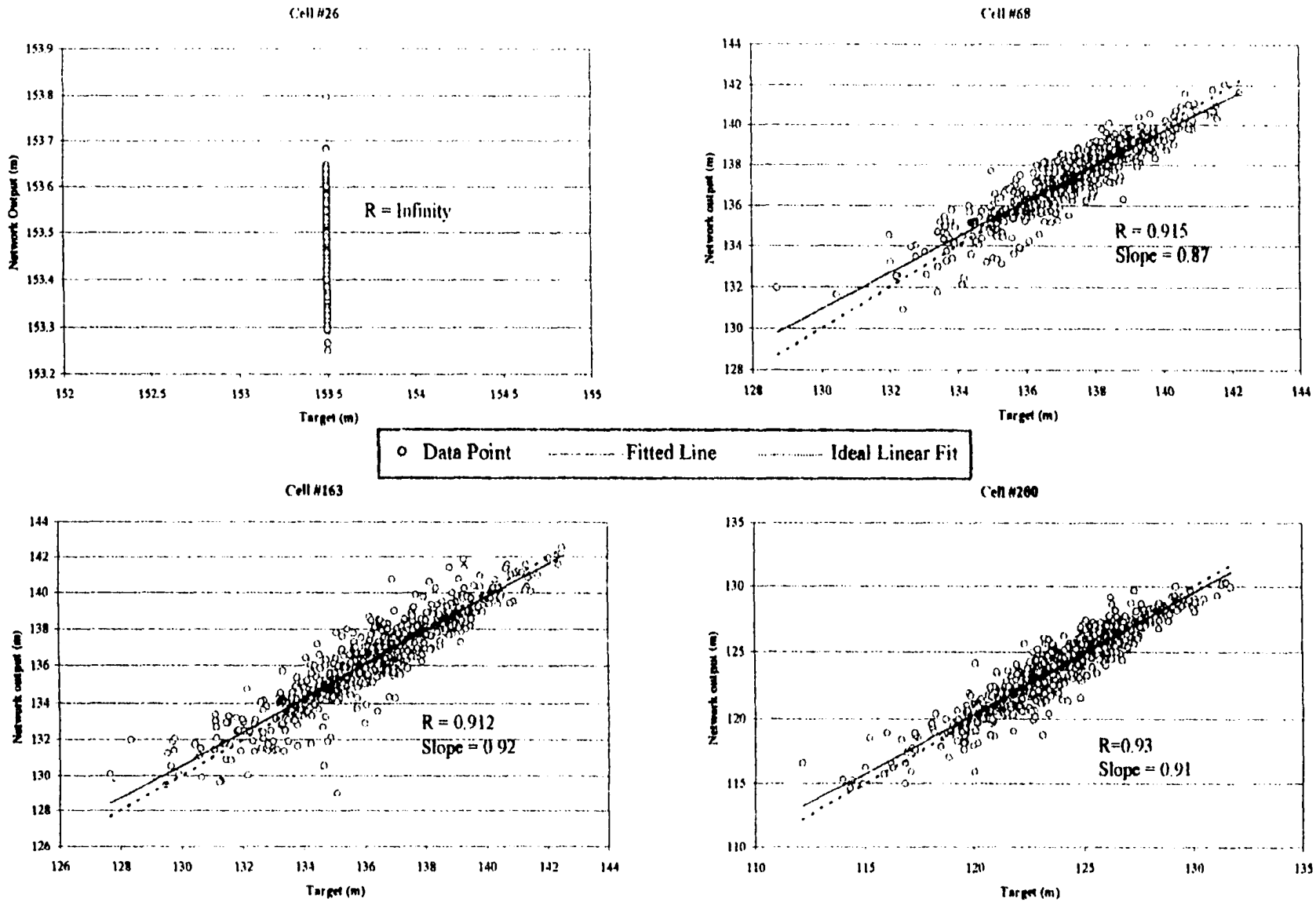


Figure 4.10 Sample of the post-training analysis results for network number 11 (featuring 300 hidden neurons)

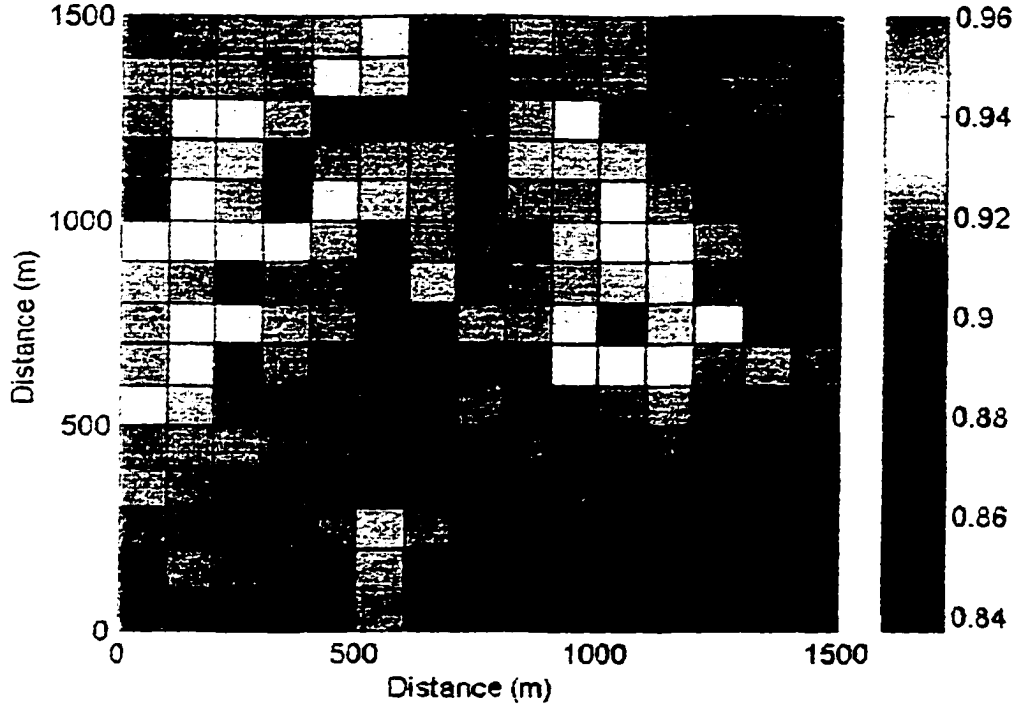


Figure 4.11 Correlation coefficient values from the post-training analysis of network number eleven (possessing 300 neurons in the hidden layer)

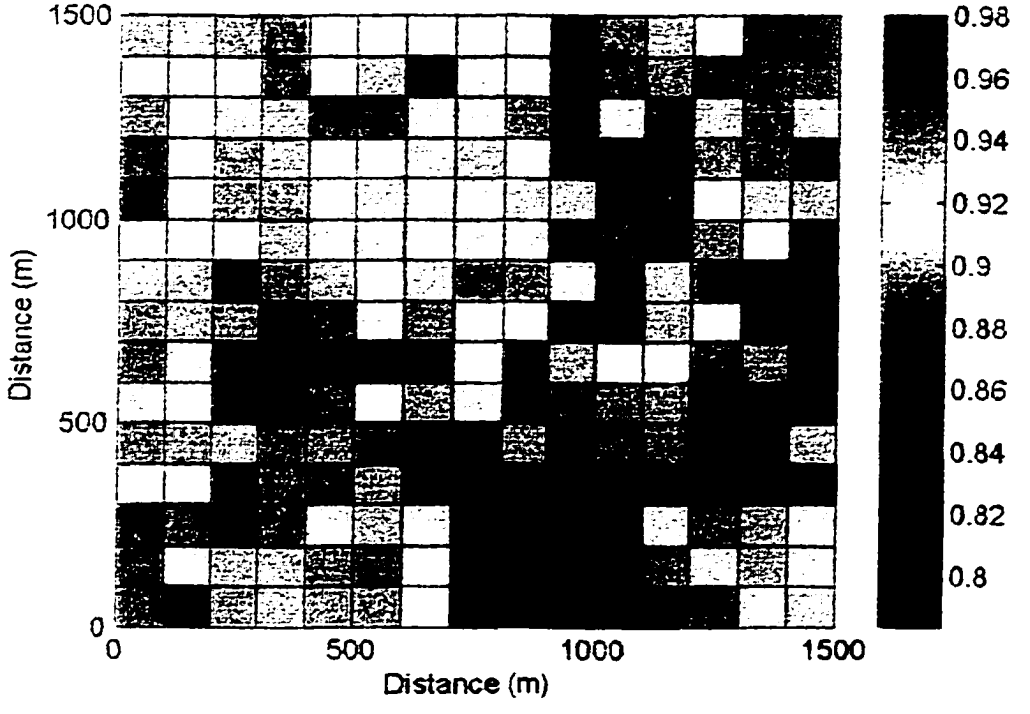


Figure 4.12 Slope values from the post-training analysis of network number eleven (possessing 300 neurons in the hidden layer)

cells in the problem domain. The question that remains to be answered is whether it is the best network to use in solving the inverse problem.

4.5.3 Effect of the Number of Hidden Neurons

In order to determine the best network capable of modeling the relation between the transmissivity and hydraulic head fields, a post-training analysis was conducted on all of the 12 trained networks. The results of this analysis were then used to conduct a sensitivity analysis of the average slope and correlation coefficient values generated by a network to the number of hidden neurons in that network. Figure 4.13 depicts the variation of the average post-training analysis results (correlation coefficient and slope) with the number of neurons in the hidden layer as obtained from all of the 12 trained networks.

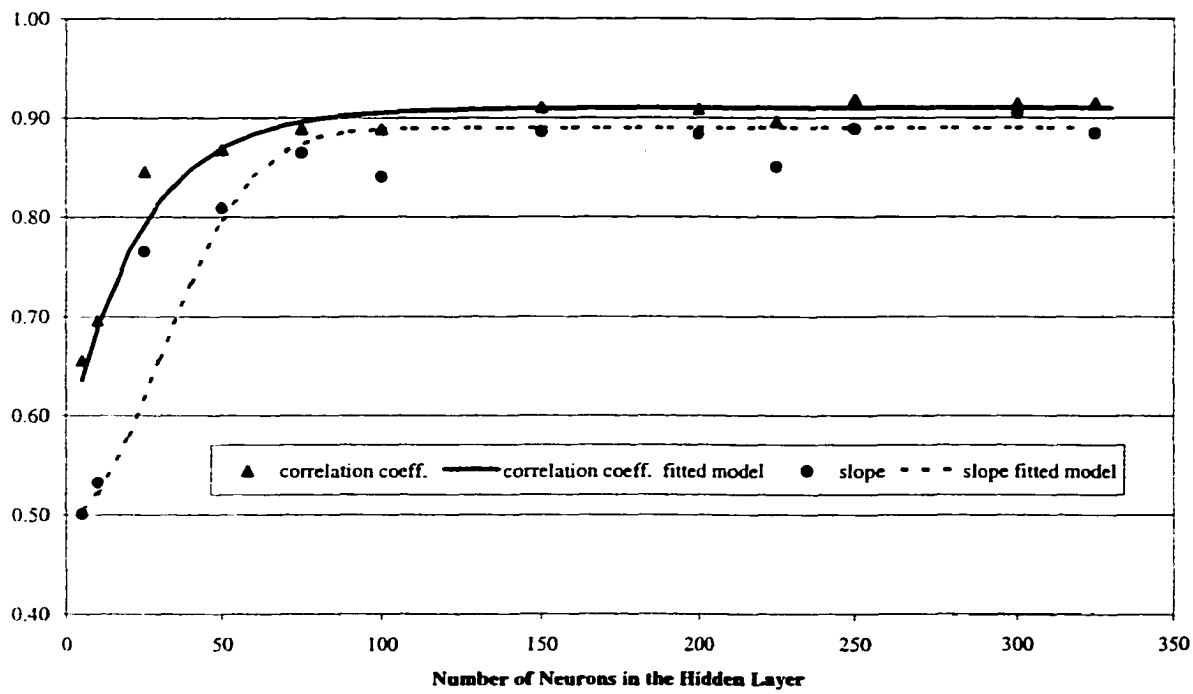


Figure 4.13 Variation of the slope and correlation coefficient values as a function of the number of neurons in the hidden layer

The graph shows that the average slope and correlation coefficients increase with the number of hidden neurons until a sill value beyond which the increase is minimal. The correlation coefficient increase follows an exponential model with a range of 70 hidden neurons beyond which the average correlation coefficient reaches an asymptotic value of about 0.91. The slope on the other hand follows a fitted Gaussian model with a range of 125 hidden neurons beyond which a sill value of 0.89 is reached.

From the results obtained thus far it is apparent that a network with 125 neurons is sufficient to model the relation of interest. Another parameter however that is to be considered, is the variance of the slope and correlation coefficient. It is important that the slope and correlation coefficient fields generated by the adopted network exhibit a small variance. A large variance indicates that not all the cells in the grid are modeled with the same accuracy. This could lead to some regions of the study area being poorly modeled by the ANN, thus adversely affecting the prospects of producing a viable inverse solution.

The variation of the variance of the slope and correlation coefficient with the number of neurons in the hidden layer for all the 12 trained networks is depicted in Figure 4.14. The plot shows that the variance of both of the slope and the correlation coefficient exhibit an exponential decay relation to the number of hidden neurons. This leads to the conclusion that the larger the number of hidden neurons, the better the ability of the network to model all the cells with the same accuracy. A point is reached however after which increasing the number of hidden neurons will not lead to any significant reduction in the variance values.

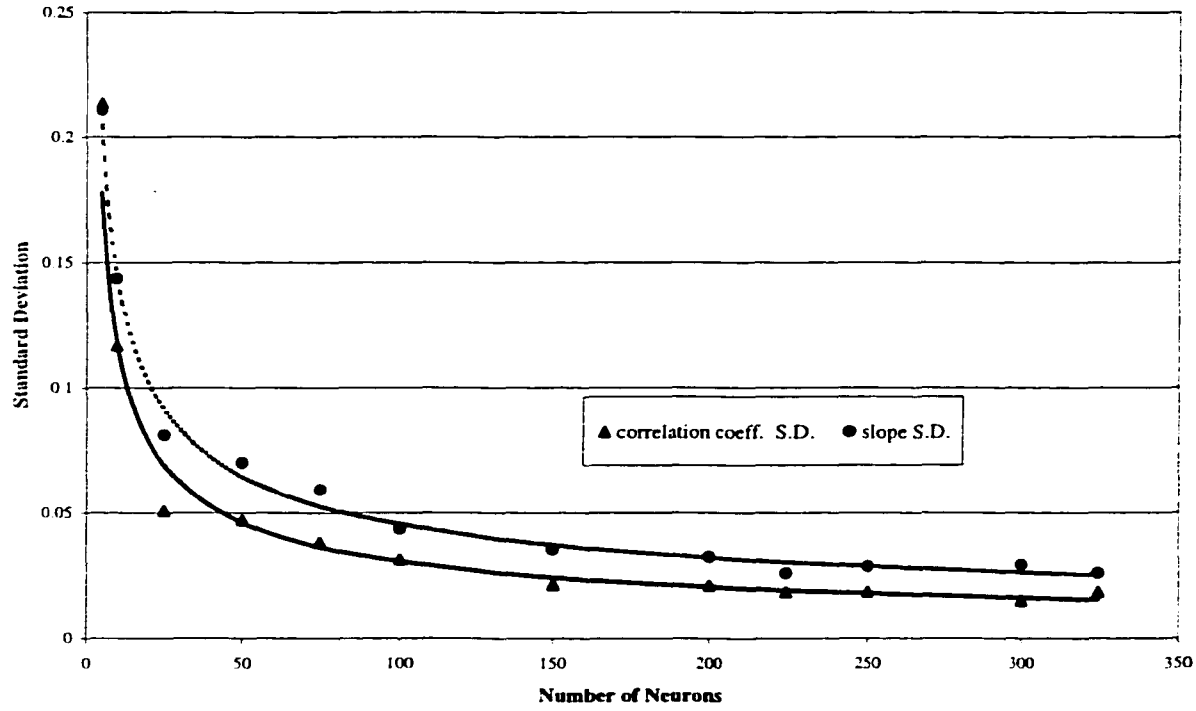


Figure 4.14 Variation of the correlation coefficient and slope standard deviation values as a function of the number of neurons in the hidden layer

It can be observed from the plotted data in Figures 4.13 and 4.14, that network number 11 (featuring 300 hidden neurons) produced the highest mean slope and correlation coefficient, and the lowest correlation coefficient variance. It was thus decided to use an ANN with 300 neurons in the hidden layer to solve the inverse problem in this case.

4.6 Effect of the Number of Training Patterns

One of the factors that can significantly influence a network's ability to learn and generalize is the number of patterns in the training network. Artificial neural networks are capable of mapping relations within the range of values comprising the training space as defined by the training set. The limited ability of artificial neural networks to extrapolate beyond the training surfaces (input and output) makes it necessary to expand the training

domain to encompass all the possible transmissivity and hydraulic head values for each cell. This can best be achieved by increasing the number of training patterns to include all the possible transmissivity fields. A large training set however can significantly increase the time and resources needed to train the network. The ability of the network to map all the possible solution surfaces can also be hampered by other factors such as the number of hidden neurons. If less than sufficient hidden neurons are used, the gains that can be made by adding more training patterns quickly decrease. Using insufficient training patterns on the other hand will generally lead to a poor network performance in a post-training analysis, and can thus be easily detected.

The choice of using 1000 training patterns in the training process of the ANN was reached through a trial and error process. In order to validate this choice, a sensitivity analysis of the slope and correlation coefficient of the number of training patterns was conducted for network number 11 (featuring 300 hidden neurons). The network was trained by using 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200 training patterns. Post-training analysis was conducted for all these 13 new networks by using the last 300 training patterns in the 1500 pattern training set (patterns 1201-1500).

The results of the sensitivity analysis are shown in Figures 4.15 and 4.16. Figure 4.15 shows the variation of the mean slope and correlation coefficient with the number of training patterns. The plot reveals that both parameters noticeably increase in value with an increase in the number of training patterns up to a certain point beyond which there is no significant increase when additional training patterns are used. Two exponentially increasing models were successfully used to fit the generated data points, and model the sensitivity of the average slope and correlation coefficient to the number of training

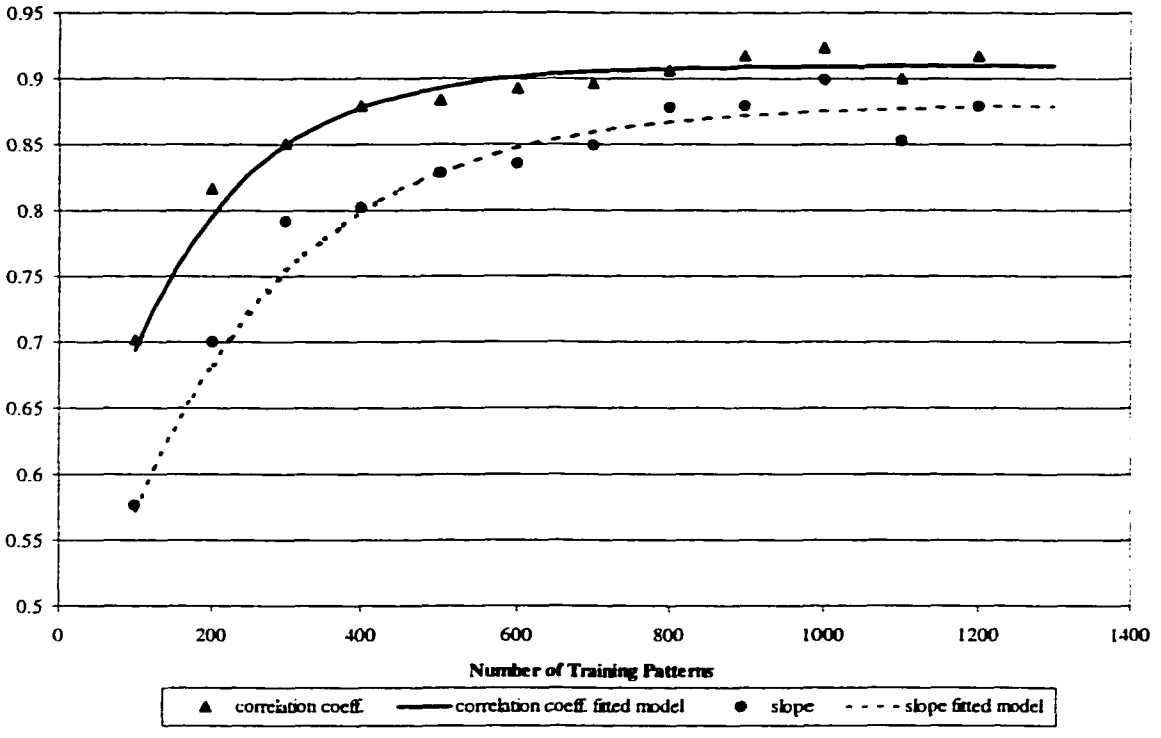


Figure 4.15 Variation of the correlation coefficient and slope mean values as a function of the number of training patterns

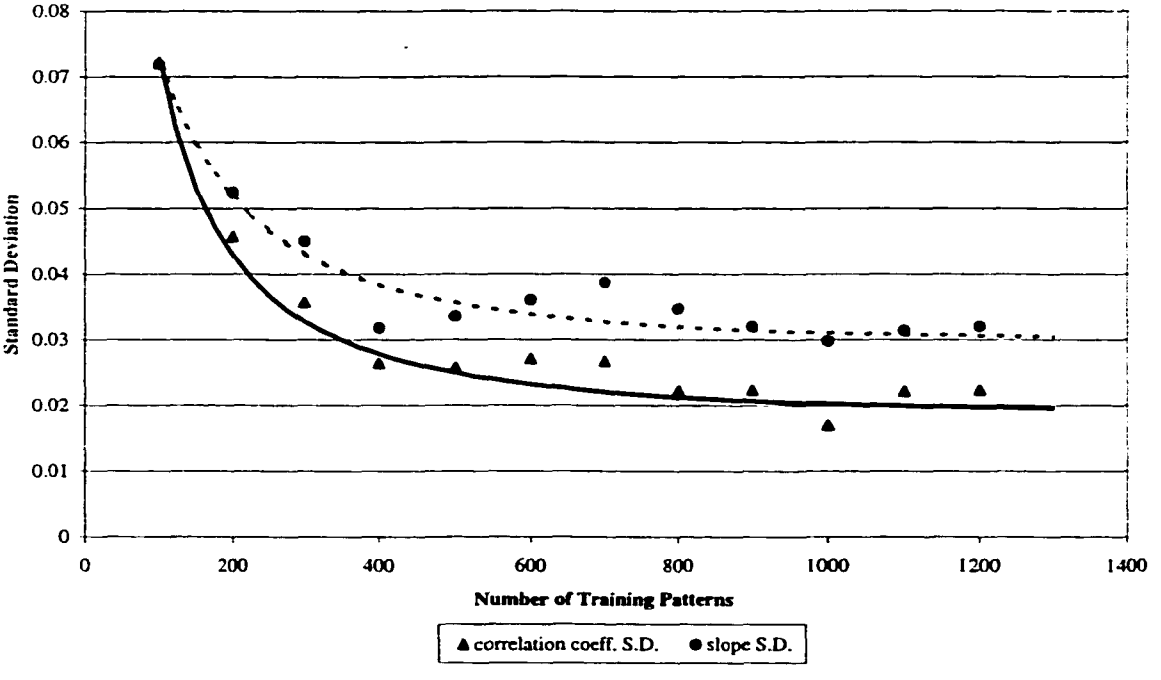


Figure 4.16 Variation of the correlation coefficients and slope standard deviation values as a function of the number of training patterns

patterns. The models reveal that using more than 800 training patterns to train the network will do little to improve the mean slope and correlation coefficients produced by the network.

Figure 4.16 depicts the sensitivity of the variance of both of the slope and correlation coefficients to the number of training patterns. The plots show that two exponentially decreasing models could be used to fit the generated data points. The fitted models reveal that little gain is attained by using more than 1000 training patterns to train the network.

4.7 Inverting the Trained Network

The last remaining step is to invert the trained network, so as to solve the inverse problem. Following the network inversion algorithm explained in chapter 3, one of the 1500-transmissivity realizations was used as an initial guess. Using one of these realizations will ensure that the transmissivity field measurements and the local structure of the transmissivity field represented in the semi-variogram are incorporated in the produced solution. The inversion algorithm (matlab script INVERSE) was written and conditioned such that the values of the 29 cells in which the transmissivity values were known were not updated in the inversion process. This ensured that the final solution would honor the transmissivity field measurements.

The inversion algorithm updated the transmissivity field until a field that minimized the error between the target hydraulic head values and the network output was reached. In order to monitor the progress of the inversion process, a criterion had to be defined. The mean percentage error (MPE) defined as the mean of the absolute error between the 53 target values (hydraulic head field measurements) and the corresponding

network outputs was chosen for this purpose, and was thus computed in each iteration. The inversion algorithm was conditioned to stop when the MPE value ceases to exhibit a significant reduction in value per iteration (i.e. converge to a minimum).

One of the problems that were faced during the first attempts to use the inversion algorithm was that some of the transmissivity values to which the algorithm converged included negative values. To avoid such unrealistic results, the network training process was repeated after preprocessing the training transmissivity values for each cell to lie between 0.05 and 0.95 as described earlier. By conditioning the inversion algorithm to retain all the produced transmissivity values between 0 and 1, the convergence of the algorithm to a realistic and physically meaningful solution was ensured. The curtailment of the transmissivity values to lie between 0 and 1 is also important to prevent extrapolation by the network beyond the input surface for which it had been trained. During each iteration the network computes an output from a transmissivity field (input) and then updates this field. If the cell values of the input field are far beyond the range of values for which the network had been trained, a large noise in the errors used to update the input transmissivity field might thus be induced by a poor network performance. Such propagation of error might lead to erroneous results and can inhibit the convergence of the network inversion process.

Figure 4.17 depicts the progress of the inversion process for the problem at hand. The graph shows that the chosen monitoring criterion (MPE), starts by decreasing significantly in the beginning of the process, and then exponentially slows down. A point is reached after which little reduction in the mean percentage error is attained per

iteration. This is the point when the inversion process is stopped, and the updated transmissivity field is returned as the solution to the inverse problem.

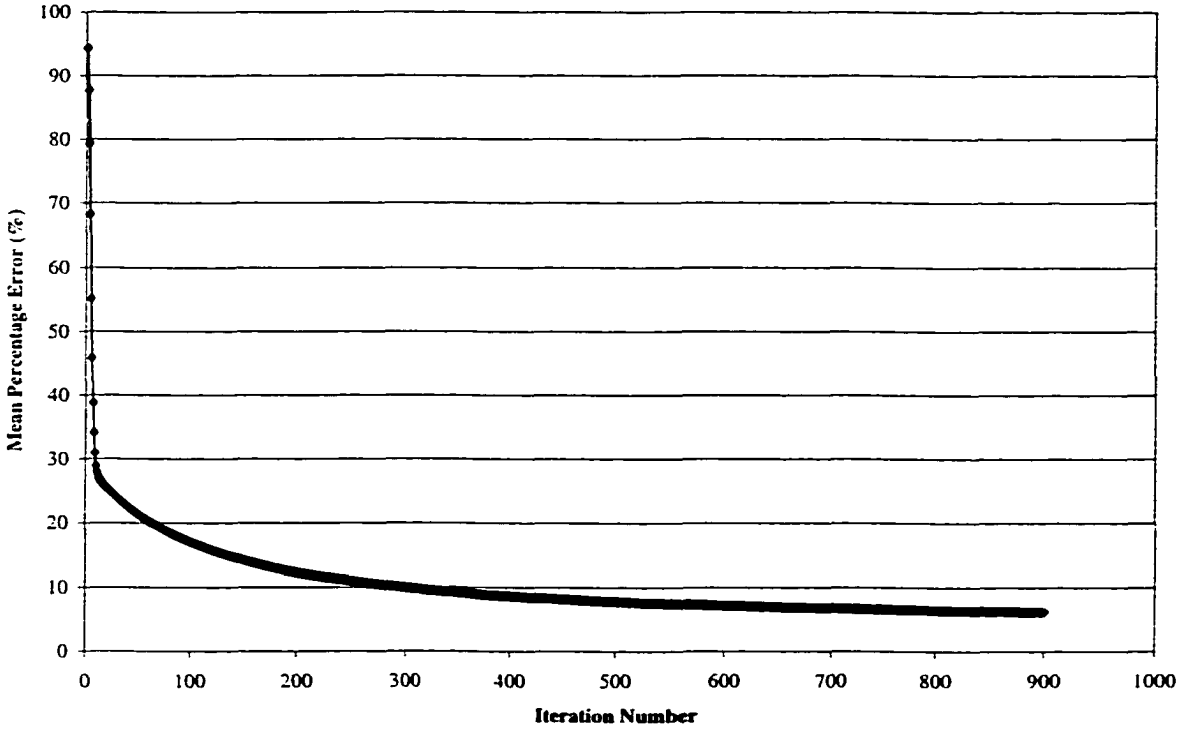


Figure 4.17 Network inversion progress

4.8 Scrutinizing the Inverse Solution

With an inverse solution at hand, the next step will be to compare the produced solution to the generated parameters of the hypothetical case study. Figure 4.18 depicts the two transmissivity field maps of the hypothetical case study and the one generated by solving the inverse problem. By comparing the two plots and the numerical values for each cell, it became clear that even though the produced transmissivity field exactly honored all the 29-transmissivity field data, it is a distinct field from that of the hypothetical case study. This conclusion was further validated by plotting the transmissivity value for each cell for the hypothetical case study (target) versus the corresponding value in the inverse solution transmissivity field. The plot shown in Figure

4.19 reveals that there is no correlation between the two transmissivity fields. The 29 field transmissivity values that were honored and exactly reproduced by the inverse solution are the points that lie on the "Perfect Correlation Line". A statistical analysis reveals that the transmissivity field obtained from the inverse solution and referred to hereafter as (Solution ***A***), has a mean of 407.5 m^2/day and a standard deviation of 301.2 m^2/day, as opposed to a mean of 393.3 m^2/day and a standard deviation of 264.6 m^2/day for the target transmissivity field.

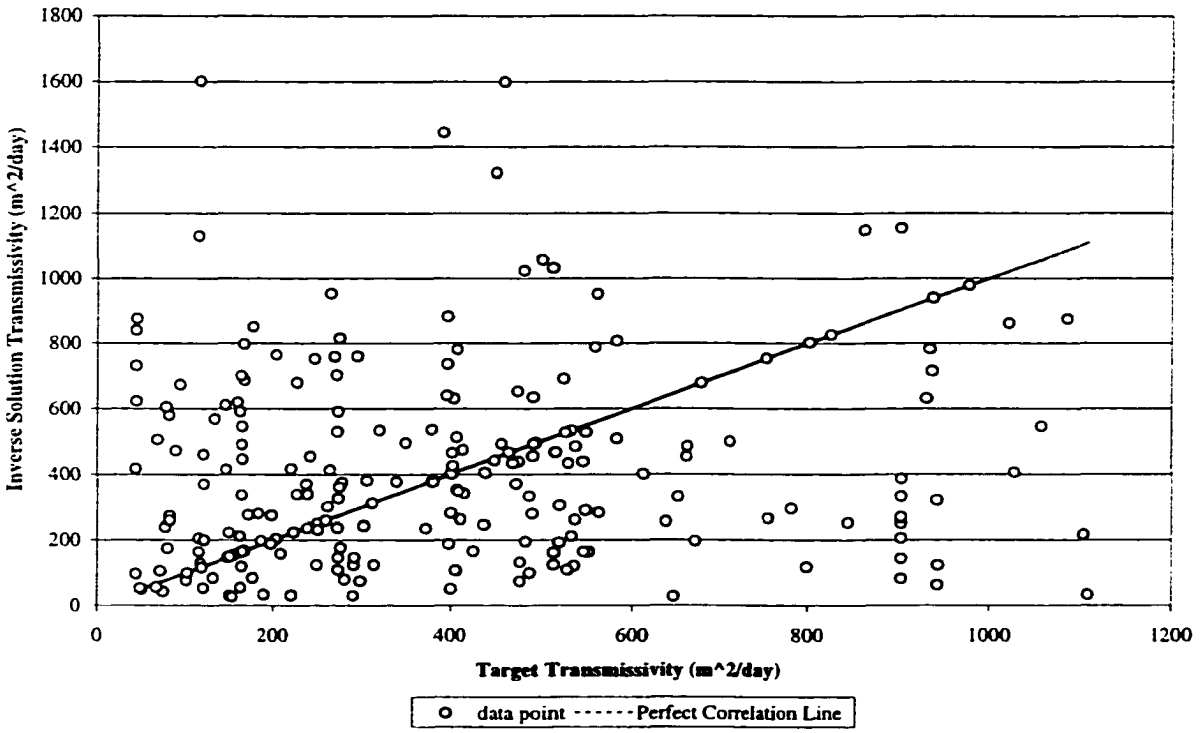


Figure 4.19 Comparison of target and solution ***A*** transmissivity fields

The two transmissivity fields were then used in MODFLOW to compute the resulting hydraulic head maps for each one under the case study stresses and boundary conditions. The results (Figure 4.20) reveal that in-spite of the fact that the inverse transmissivity field was totally different from the target transmissivity field, it was

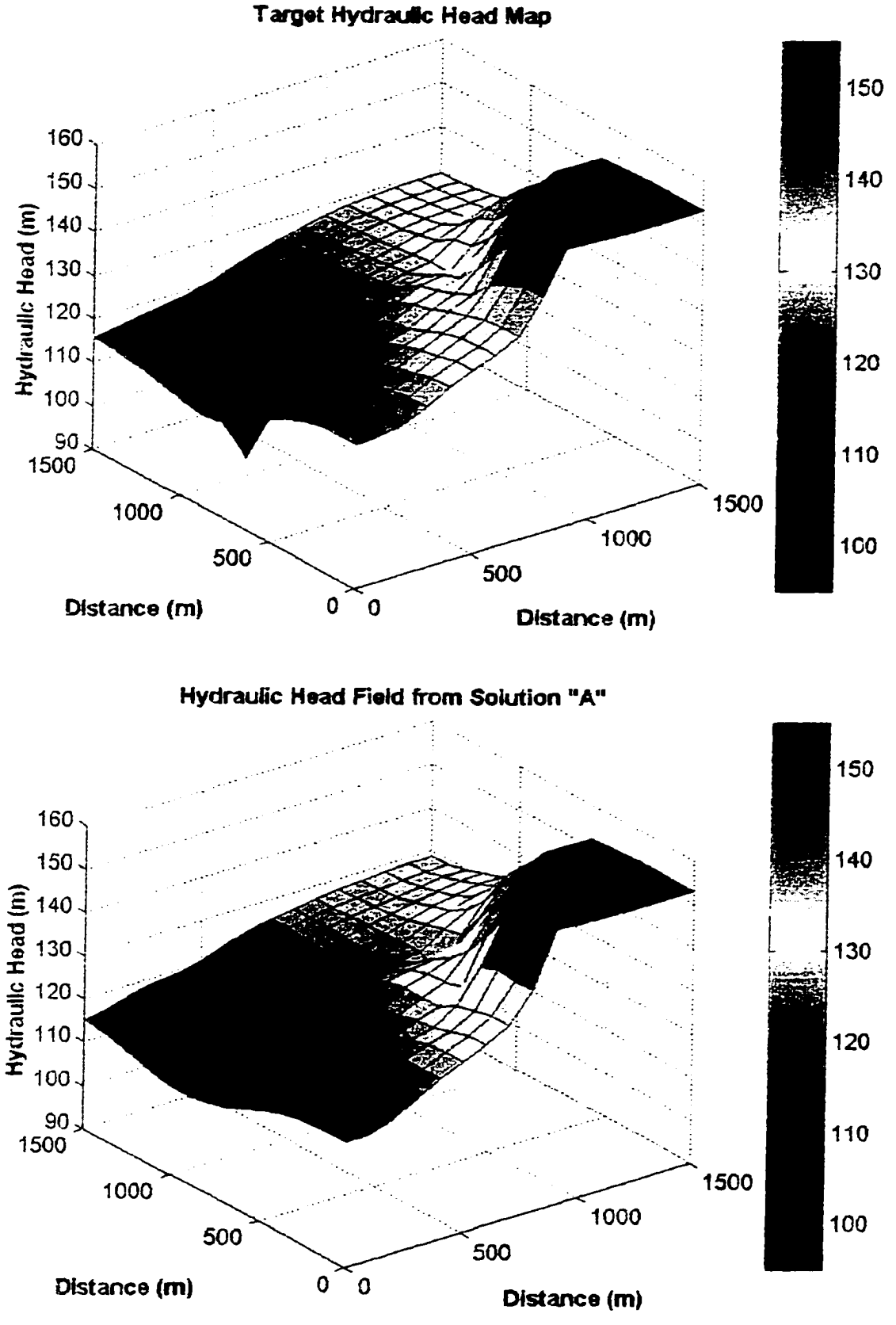


Figure 4.20 Hydraulic head maps from the hypothetical case study and inverse solution ***A***

capable of reproducing the main features of the target hydraulic head map. A plot of the target hydraulic head value for each cell versus the corresponding hydraulic head values from solution "***A***" (Figure 4.21), reveals that the two entities are strongly correlated, with a correlation coefficient (R) of 0.99 and a slope of 1.03. These values are very close to the desired (R=1 and slope=1) values which indicate congruency.

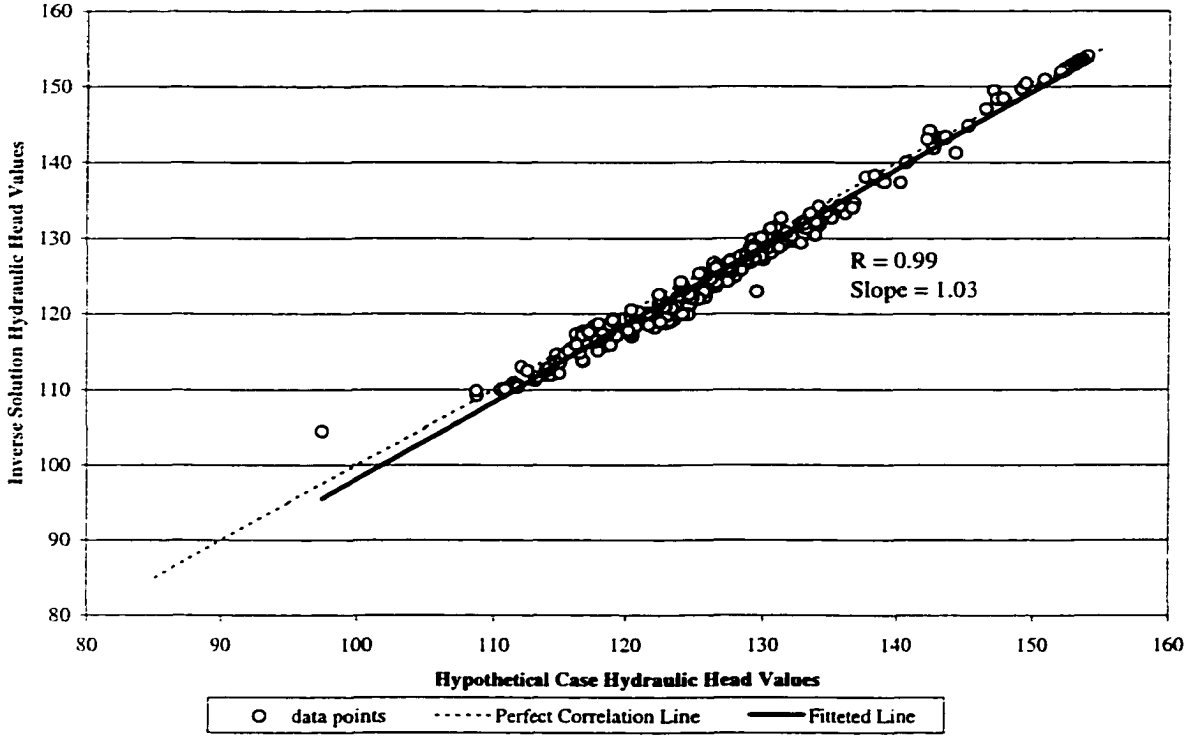


Figure 4.21 Comparison of the hydraulic head field generated by solution ***A*** to the target hydraulic head field

Another comparison between the target hydraulic head field and the one produced by the inverse solution was conducted by computing the percentage error between comparable cells in the two fields. The result was an error map for the whole field (Figure 4.22). The error map shows that with the exception of the location of the two pumping wells, the hydraulic head map produced by the inverse solution closely matched

the target hydraulic head map. The mean and standard deviation of the percentage error was computed to be 0.92 and 1.21 respectively. It is worth mentioning that during the post-training analysis of the trained neural network, the cells which hosted the pumping wells exhibited the worst correlation coefficient values, and that the hydraulic head value at these two points were presumed to be unknown when solving the inverse problem.

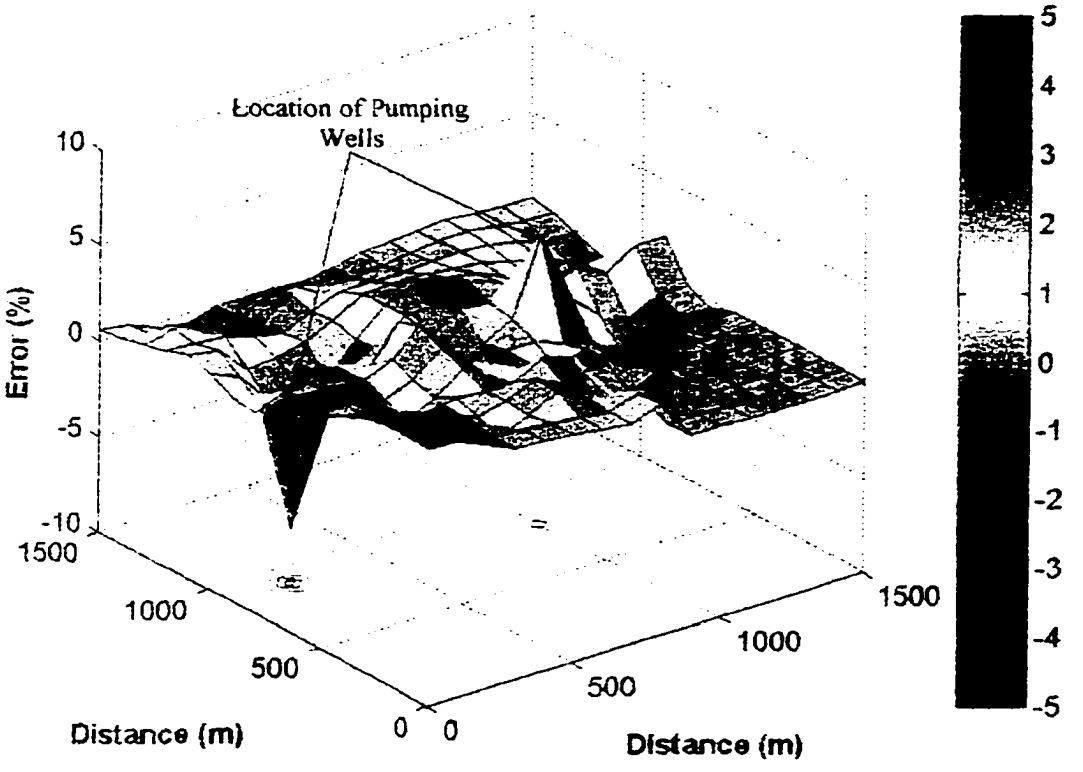


Figure 4.22 Hydraulic head error map for solution ***A***

CHAPTER 5

ANALYSIS AND DISCUSSION

5.1 Overview

It has been shown thus far that an artificial neural network can be successfully trained to provide a good approximation of the relation between the transmissivity and hydraulic head values as defined by the groundwater flow equation. Furthermore, it has been demonstrated that the trained network can be used in conjunction with a set of a-priori information (known field values), to produce a solution to the inverse problem.

In the methodology presented in this research, the trained artificial neural network (ANN) acts as the search engine or optimizer, which searches for a solution that will satisfy the designated constraints (i.e. a-priori information). The fact that the optimizer (trained network) provides a very good approximation of the groundwater flow equation, ensures that the generated inverse solution is always certain to reproduce the features and known values of the hydraulic head map when it is used in a groundwater flow model.

The inverse algorithm used starts with an initial transmissivity guess drawn from the field transmissivity data. This guess is updated during each iteration of the inversion process with the goal of minimizing the error between the neural network's output and the target hydraulic head values. In this process, the difference between the target and network's output for each cell lead to updating the transmissivity values of all the cells of

the initial guess. This means that rather than merely calibrating some of the input cells to match the hydraulic head in a few output cells, the network inversion process is that of calibrating a transmissivity field as a whole (all the input cells) so as to produce an output field (surface) which satisfies some predetermined constraints. The output surface constraints would be in the form of field hydraulic head measurements.

It is important to note that additional constraints can be easily added to control the manner in which the input (transmissivity) surface is updated. Values of cells in the input space can be preserved unaltered if their values are known, or they can be conditioned to fall within a certain range if their exact value is not known but the range in which it lie is. This is a very important feature that is lacking in almost all the currently available inverse problem solving techniques. The importance of this feature stems from the fact that hydrologists usually have an idea about the soil characteristics of the region they are interested in modeling. By simply knowing the soil type, one can identify the range of values in which the transmissivity values are most likely to lie within. One can thus easily incorporate this information in the form of constraints, and tailor the generated inverse solution to incorporate these types of field observations.

With the inverse problem in mind, perhaps the first thought one might get after being introduced to the ability of neural networks to map complex nonlinear relations, is to train a network that directly maps the inverse relation. Such a network will have the transmissivity field as an output while using the hydraulic head values as input. At first sight, this approach might appear to be promising and much simpler than the approach used in this research. It is tantamount however to an attempt to solve the inverse problem by using a direct approach. Such an approach has been shown to be unstable, where small

errors in the hydraulic head values lead to large errors in the estimated transmissivity values [McLaughlin 1996]. Another problem with such a direct approach is that it requires knowledge of the hydraulic head values for all the cells, a task that can be very expensive to achieve. One can argue that the missing hydraulic values can be obtained by an interpolation of the known values. This will yield inaccuracies in the interpolated hydraulic head values, and give rise to the problem of instability as had been demonstrated in the literature.

Another major problem with the direct approach is that is not capable of incorporating or honoring the known transmissivity values. This is in stark contrast to the methodology introduced in this research, where the local structure of the transmissivity field is incorporated in the form of the initial guess, the field transmissivity measurements are exactly honored, and the field observations in the form of soil characteristics are easily incorporated.

5.2 Refining the Inverse Solution

5.2.1 The Refinement Process

The hydraulic head error map (Figure 4.22) for solution **A** was obtained by comparing the hydraulic head values resulting from it, to the hydraulic head values of the hypothetical case study. The map has clearly shown that relatively high errors in estimating the hydraulic head in the well vicinity were obtained. In order to reduce these errors, the hydraulic head values at the location of the two pumping wells were assumed to be known, and were included in the set of a-priori information to solve the inverse problem and obtain a new solution (solution **A'**).

The hydraulic head error map resulting from solution ***A'*** (Figure 5.1) shows that the errors in the two cells in which the two pumping wells are located were reduced by up to 62% when compared to solution ***A***. A close examination of the error values in the remaining 223 cells shows that it has slightly worsened. The mean square of the hydraulic head errors for these 223 cells increased to 3.01 as opposed to 1.98 for solution ***A***.

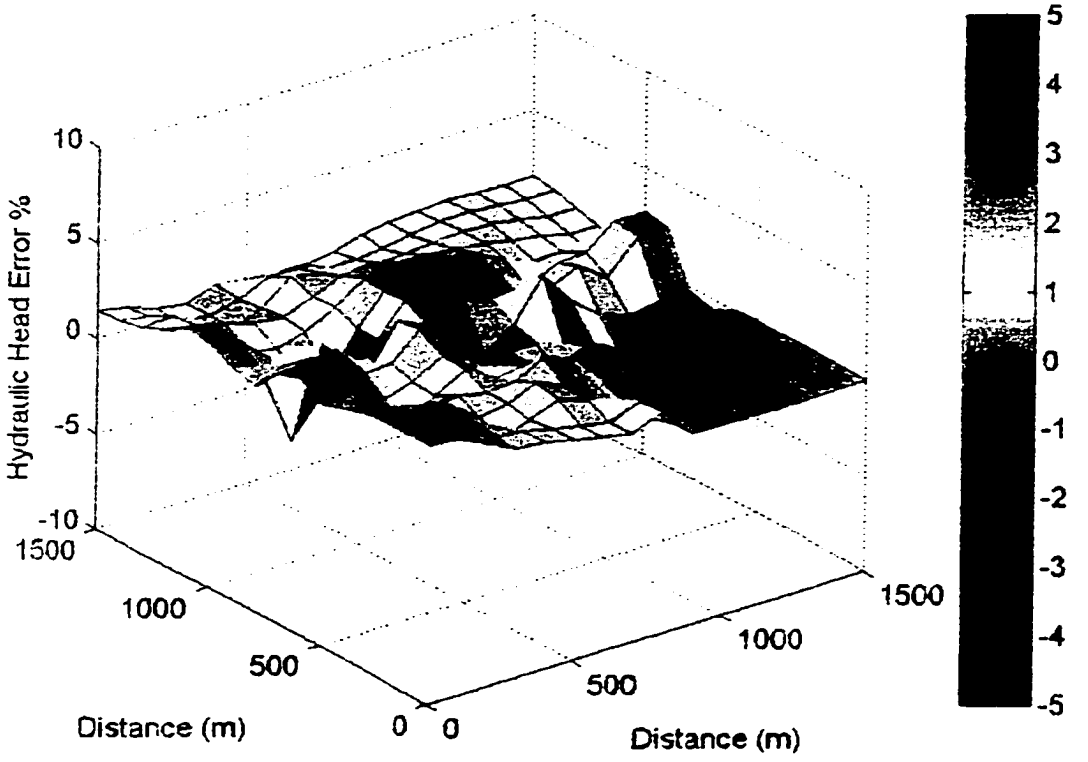


Figure 5.1 Hydraulic head error map from solution ***A'***

The worsening of the error values was attributed to the fact that the solution was driven at its final stages by the network output for cells 116 and 123 (the two cells in which the pumping wells are located), which were not adequately mapped by the neural network. This introduced additional errors in the difference between the target

and the network output for these two cells. With these additional errors dominating the calibration of the transmissivity surface towards the end of the inversion process, the error was propagated to the inverse solution, and hence to its calculation of the hydraulic head map. If these additional errors become relatively large, they could potentially lead to large errors in any calculations by the inverse solution.

It can thus be concluded that the post-training analysis of the neural network can give a good indication of the data points that should or should not be used to solve the inverse problem. The cells that are not adequately mapped by the neural network could simply be removed from the set of a-priori information. However, an alternative solution was devised and used in this research.

The adopted solution is based on dividing the available hydraulic head field data into two sets. One set (full set) contains all the data points including the hydraulic head value at the two inadequately mapped cells. The second set (partial set) contains only the well-mapped cells (i.e. it excludes the hydraulic head values at the location of the two pumping wells). The idea is to start the network inversion process with the "full set", while monitoring the rate of change of the error between the target values and the network output for the two sets (full and partial). Once the rate of change of the error of the "full set" becomes more than that of the "partial set", the partial set is used instead of the full set to complete the network inversion process. This technique will partially use the field data of the inadequately mapped cells as long as they are not driving the inversion process. As soon as these points start to dominate the process, they are dropped out leaving only the hydraulic head values of the well-mapped cells to complete the quest for the inverse solution.

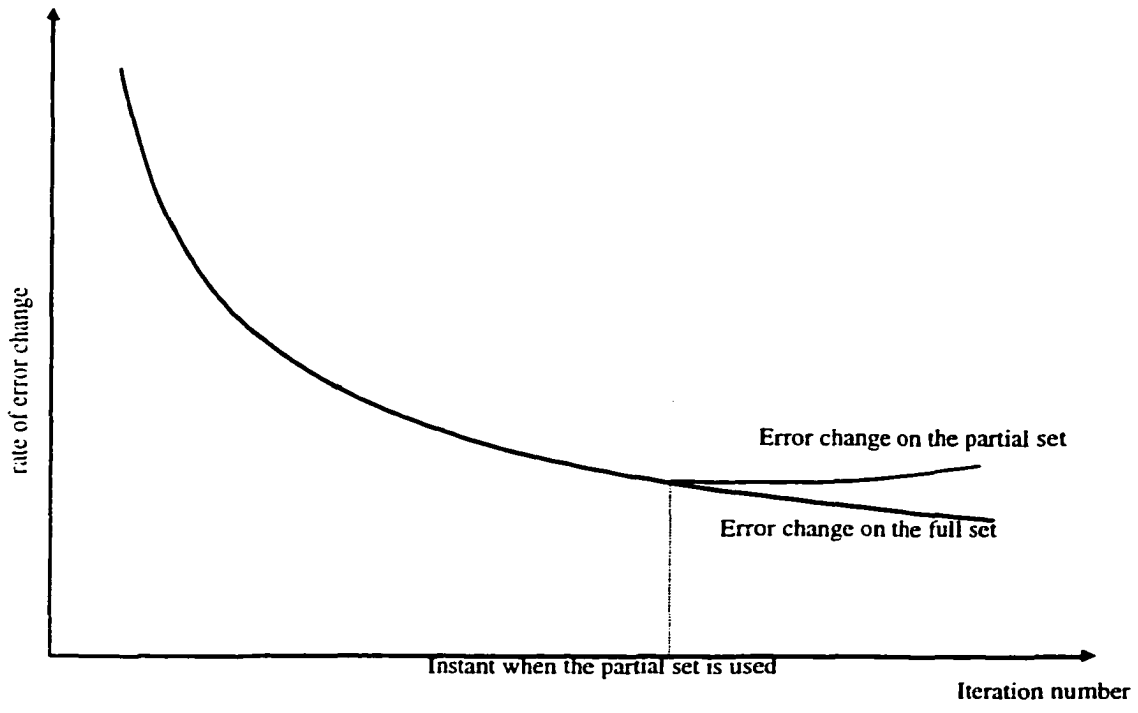


Figure 5.2 Process of solution refinement technique

The technique of partially using some of the data during the inversion process can easily be extended to incorporate information about the reliability of different hydraulic head field data in the inverse solution. The field data can be classified into two or more sets according to their reliability, and while the inversion process might be started with all the data points, the unreliable data could be discarded during the inversion process. The least reliable data would be discarded first while the most reliable data would be used to guide the inversion process to its completion.

Using the above mentioned technique the inverse solution was solved for, and the solution that was generated and which will be referred to as (Solution ***A***''), was used to compute the resulting hydraulic head error map (Figure 5.3). The plot shows that the hydraulic head error at the location of the two pumping wells remained virtually

unchanged when compared to the results of solution ***A***. The mean of the squared hydraulic head errors on the other hand showed a slight improvement for the whole grid as well as for the grid minus the two cells in which the pumping wells are located. A summary of the results is given in table 5.1.

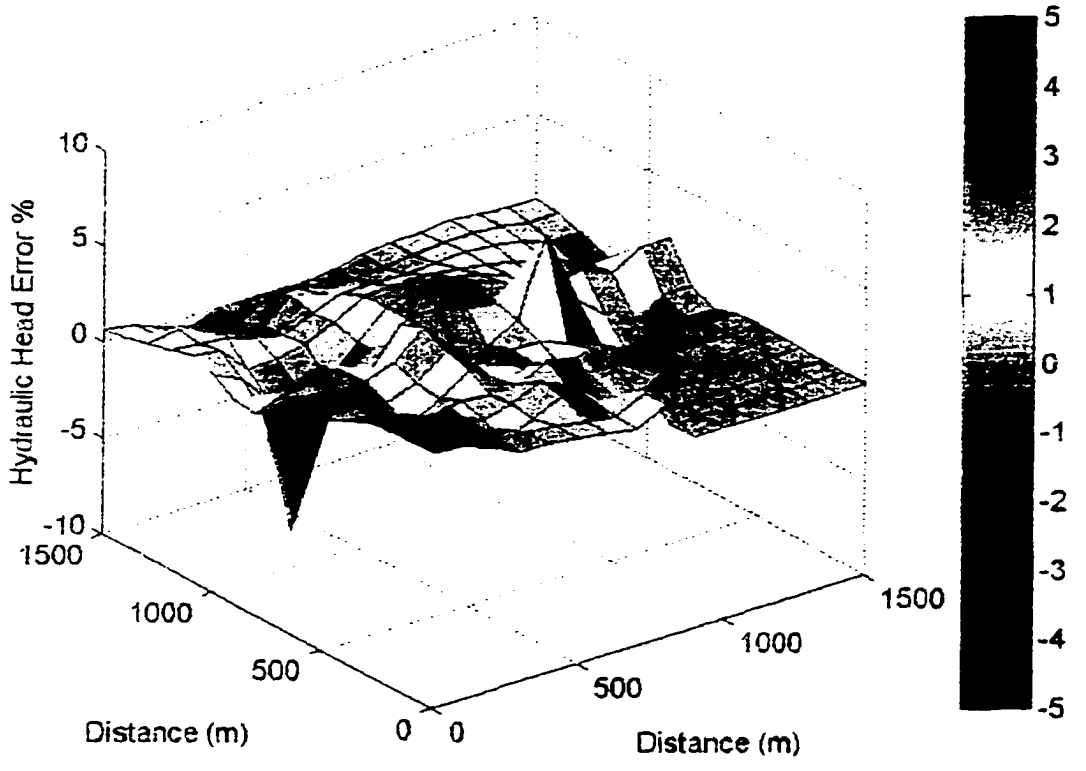


Figure 5.3 Hydraulic head error map from solution ***A''***

Table 5.1 Comparison of solutions ***A***, ***A'*** and ***A''***

Solution Designation	Hydraulic head error at cell #116	Hydraulic head error at cell #123	Mean squared hydraulic head error for all 225 cells	Mean squared hydraulic head error for 223 cells
A	5.12%	-7.17%	2.31	1.98
A'	2.95%	-2.68%	3.06	3.01
A''	5.22%	-7.09%	2.26	1.94

It can be easily observed from figure 5.3 that the hydraulic head errors at the two cells that contain the two pumping wells are still relatively higher than the mean hydraulic head error for the whole grid. These high errors were attributed to the inability of the neural network to accurately map the excessively large drawdowns that usually occur in the vicinity of pumping wells. These steep drawdowns act as singularities in an otherwise smooth surface. It is the limited ability of the neural network being used to map discontinuous surfaces, which leads to its relatively poor performance in mapping the two cells which host the pumping wells.

5.2.2 Comparison of Results

The question which presents itself at this point, is whether the transmissivity fields produced by solutions ***A*** and ***A"*** are alike or totally different. In order to answer this question, the transmissivity value for each cell from solution ***A*** was plotted versus the corresponding value from solution ***A"***. The result which is depicted in Figure 5.4 shows that the two transmissivity fields are almost identical, with the transmissivity value for each cell from solution ***A"*** being slightly different from the value of the corresponding cell in solution ***A***. The plot also shows that the 29 field transmissivity values were exactly honored in both solutions. Nevertheless, the combined effect of the small changes in the transmissivity values for each cell resulted in a slightly different hydraulic head error maps for the two solutions.

A visual comparison of the hydraulic head maps obtained from solutions ***A*** and ***A"***, with the target hydraulic head map (Figure 5.5), shows that the two inverse solutions were capable of mapping the target with good accuracy. This is in stark contrast to the

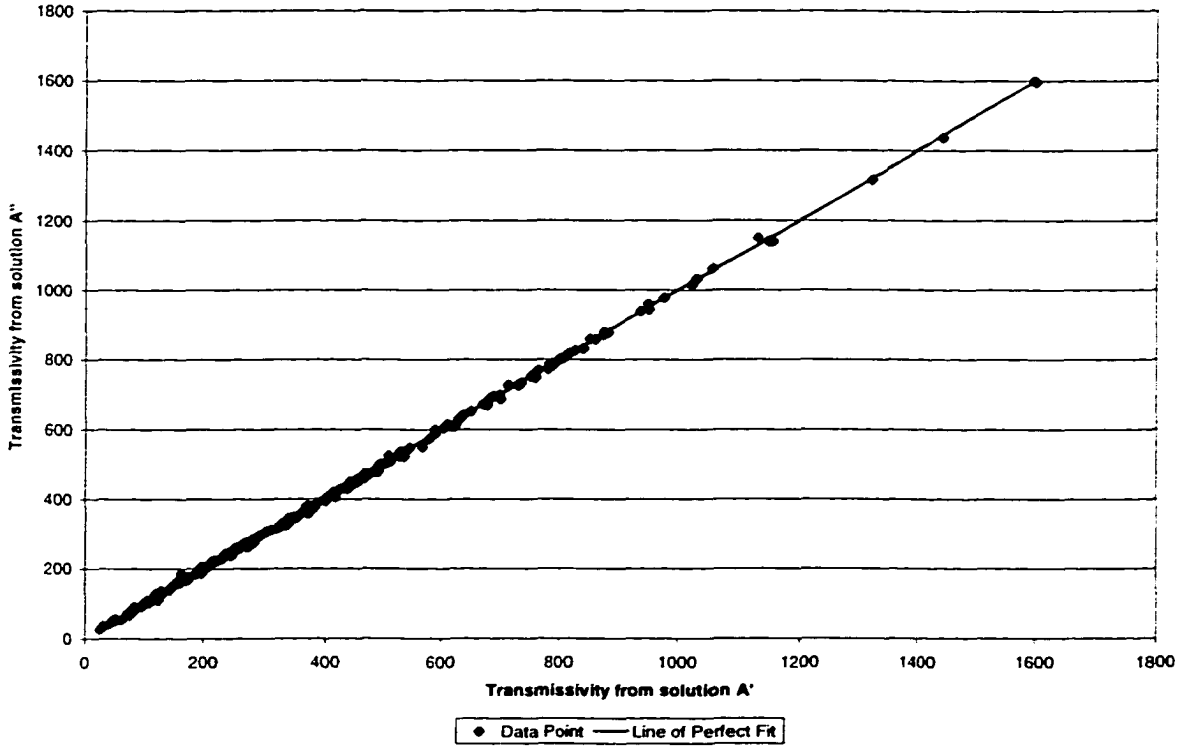


Figure 5.4 Correlation of solutions ***A'*** and ***A''***

hydraulic head map obtained by using a transmissivity field that is obtained by a simple krigging of the field transmissivity values (Figure 5.5). Even though simple krigging allows for the honoring of all the transmissivity field measurements, unlike the inverse solution result, it is incapable of matching the field measurements of the hydraulic head.

5.3 Introduction of a Stress Scenario

The main objective of any model calibration process is to produce model parameters that can be used to make reliable predictions of the behavior of the system being modeled when it is subjected to new hydrologic stresses. The success or failure of any model calibration process is thus directly linked to the calibrated model's ability to mimic the system's response when it is subjected to complex hydrologic stresses that are different from the ones under which the calibration process was conducted.

In order to fully evaluate the obtained inverse solutions, a new stress scenario including two transiently pumping wells located at cells number 185 and 163 was created. The rate of pumping for the two wells was designated at 8,000 m^3/day and 13,000 m^3/day respectively. The high pumping rates in effect induce a groundwater mining scenario in which large drawdowns are experienced in a short period of time. This will lead to a transient response that will be difficult of simulate without a well-calibrated model.

Using the transmissivity field of the hypothetical case study and the hydraulic head map it generated (calibration target) as an initial hydraulic head field, the response of the aquifer to the two new pumping wells was obtained by using the groundwater flow model MODFLOW. The responses of the system using the transmissivity fields and the hydraulic head maps obtained from solutions ***A***, ***A'***, and ***A''*** were also computed. It might be helpful at this point to recall that solutions ***A***, ***A'*** and ***A''*** were obtained under the following different conditions:

Solution ***A*** : The inverse solution was obtained without using the hydraulic head values at the location of the pumping wells.

Solution ***A'*** : The inverse solution was obtained by fully using the field hydraulic head at the location of the pumping wells.

Solution ***A''*** : The inverse solution was obtained by partially using the field hydraulic head values at the location of the pumping wells.

A sample of the results showing the hydraulic head map after pumping for three, seven and ten days are shown in Figures (5.6-5.8). The hydraulic head maps

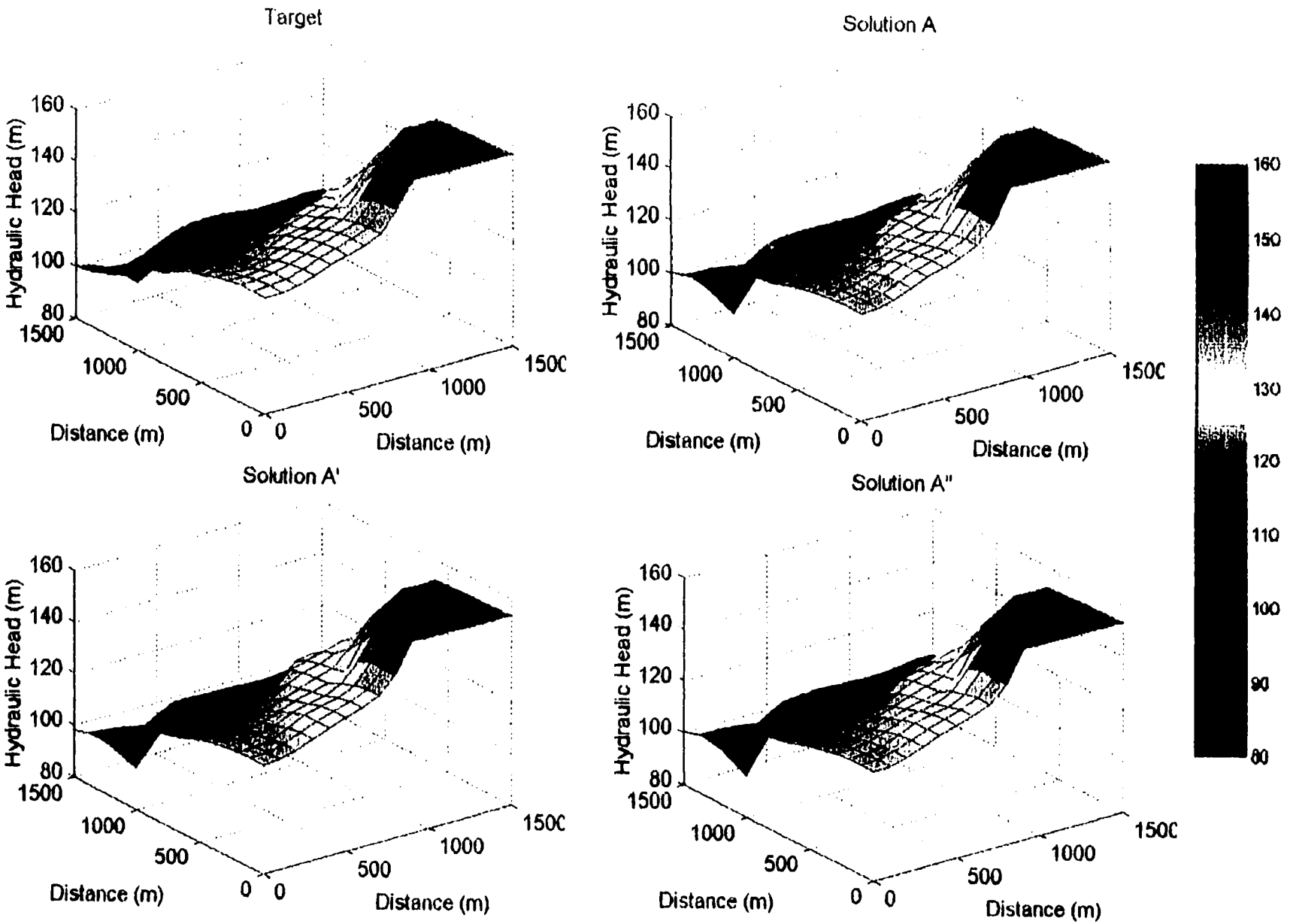


Figure 5.6 Hydraulic head maps three days after the commencement of the stress scenario

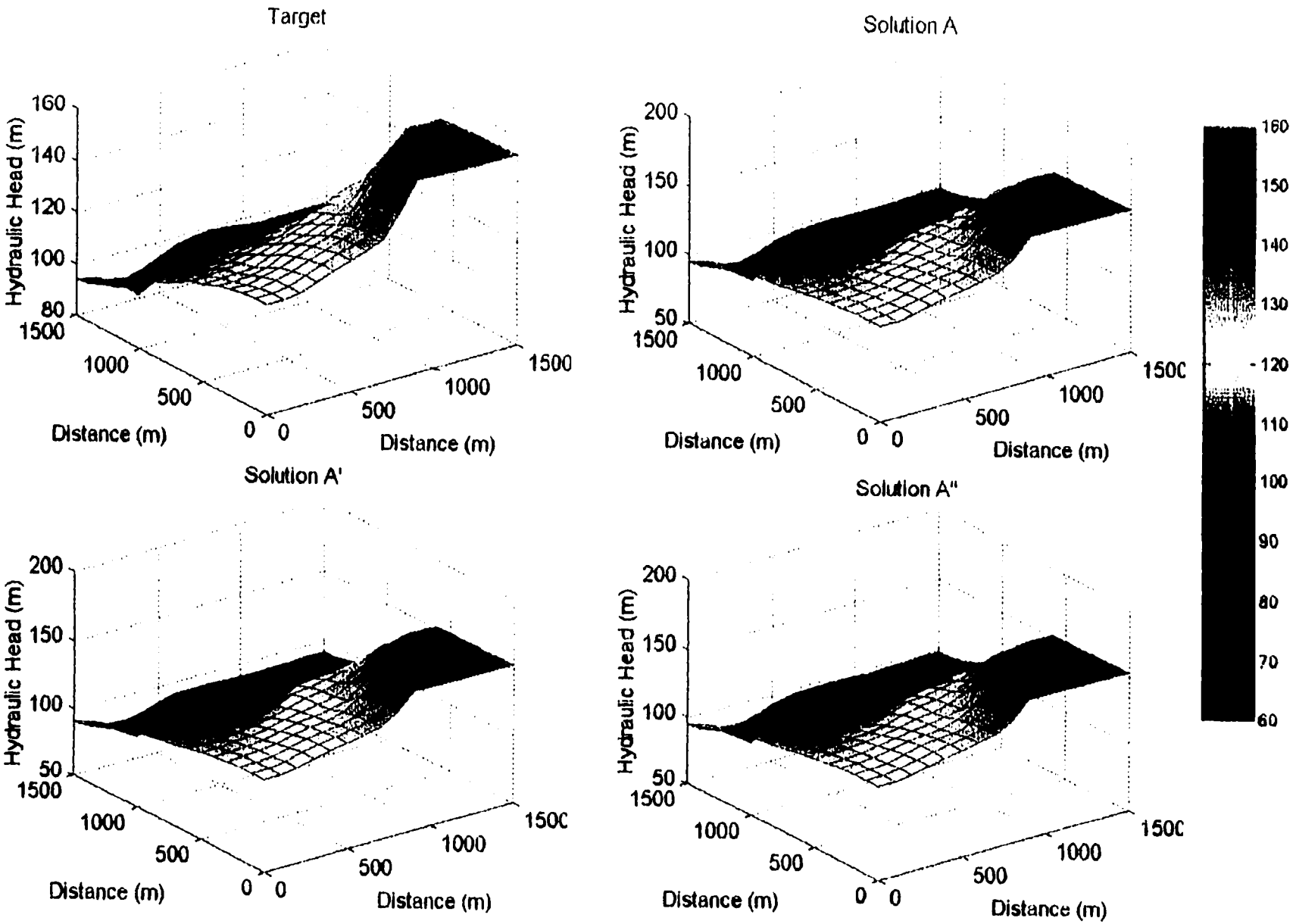


Figure 5.7 Hydraulic head maps seven days after the commencement of the stress scenario

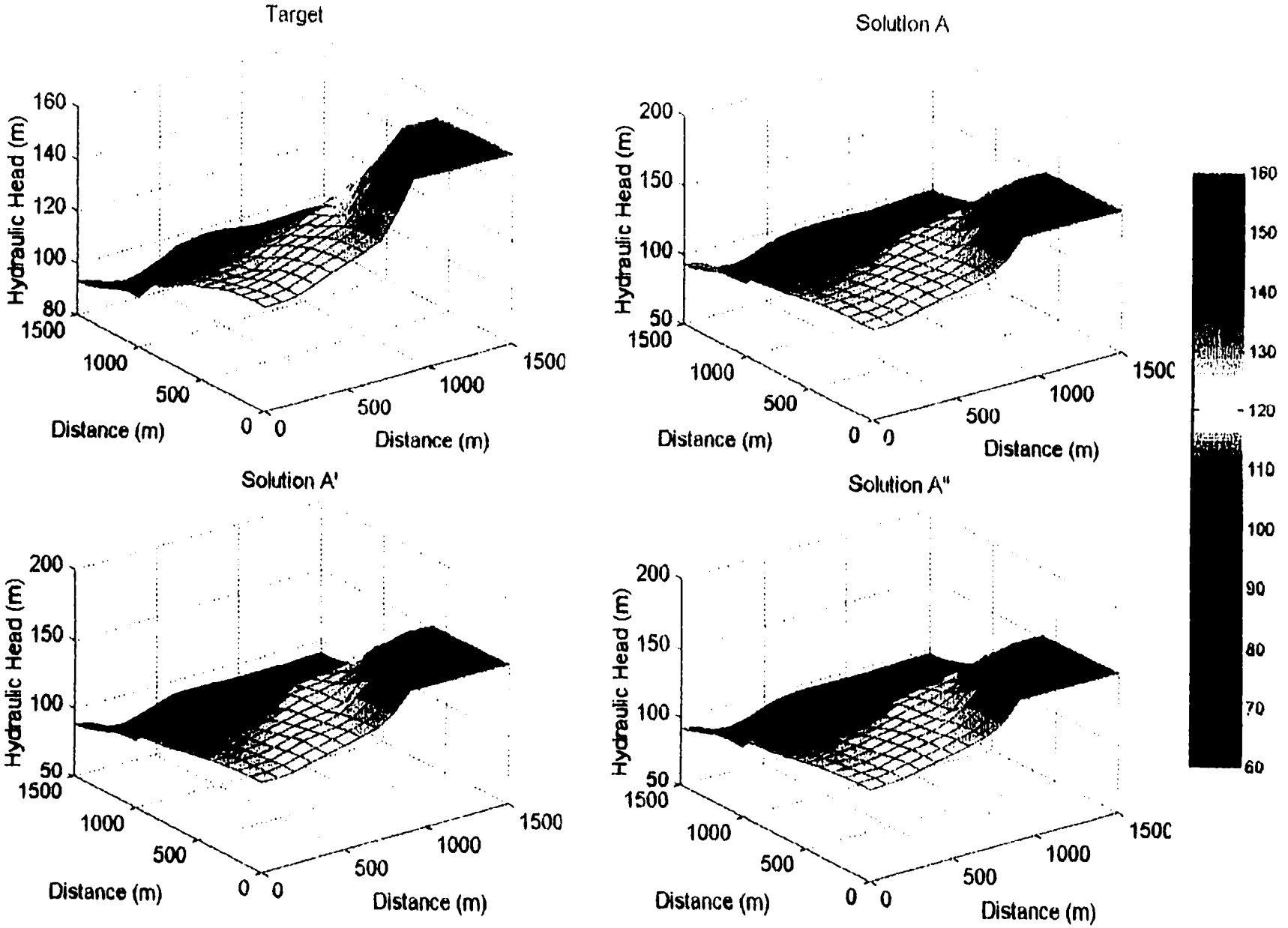


Figure 5.8 Hydraulic head maps ten days after the commencement of the stress scenario

generated by the three inverse solutions were compared to the ideal solution during the different stages of the pumping period. The comparison was conducted by computing the correlation between the responses of the three inverse solutions (**A**, **A'** and **A''**) and the ideal response during the different stages of the stress period. Samples of these results are shown in Figures (5.9-5.11). A summary of the results is given in Table 5.2.

Table 5.2 Evaluation results of the refinement process

Solution Designation	Statistical Analysis Results before the beginning of the stress scenario		Statistical Analysis Results at the end of the stress scenario	
	Slope	Correlation Coefficient	Slope	Correlation Coefficient
A	1.03	0.99	1.01	0.99
A'	1.03	0.99	1.12	0.99
A''	1.03	0.99	1.01	0.99

The statistical analysis results have demonstrated that solutions **A** and **A''** were comparable in that both were able to predict the response of the aquifer to the introduced stresses. The fact the slope and correlation coefficients values obtained by correlating the resulting hydraulic head values to the true values remained virtually unchanged throughout the stress period, is indicative of the ability of solutions **A** and **A''** to correctly represent the aquifer in question. Solution **A'** on the other hand exhibited different behavior. The value of the slope started increasing with time, and a visual inspection of the results at the end of the simulation (Figure 5.10) shows that the resulting hydraulic head values departed from the ideal (true) hydraulic head values.

Based on the results obtained herein and the analysis which was conducted as described in section 5.2.1, solution **A''** was adopted as the final solution to the inverse

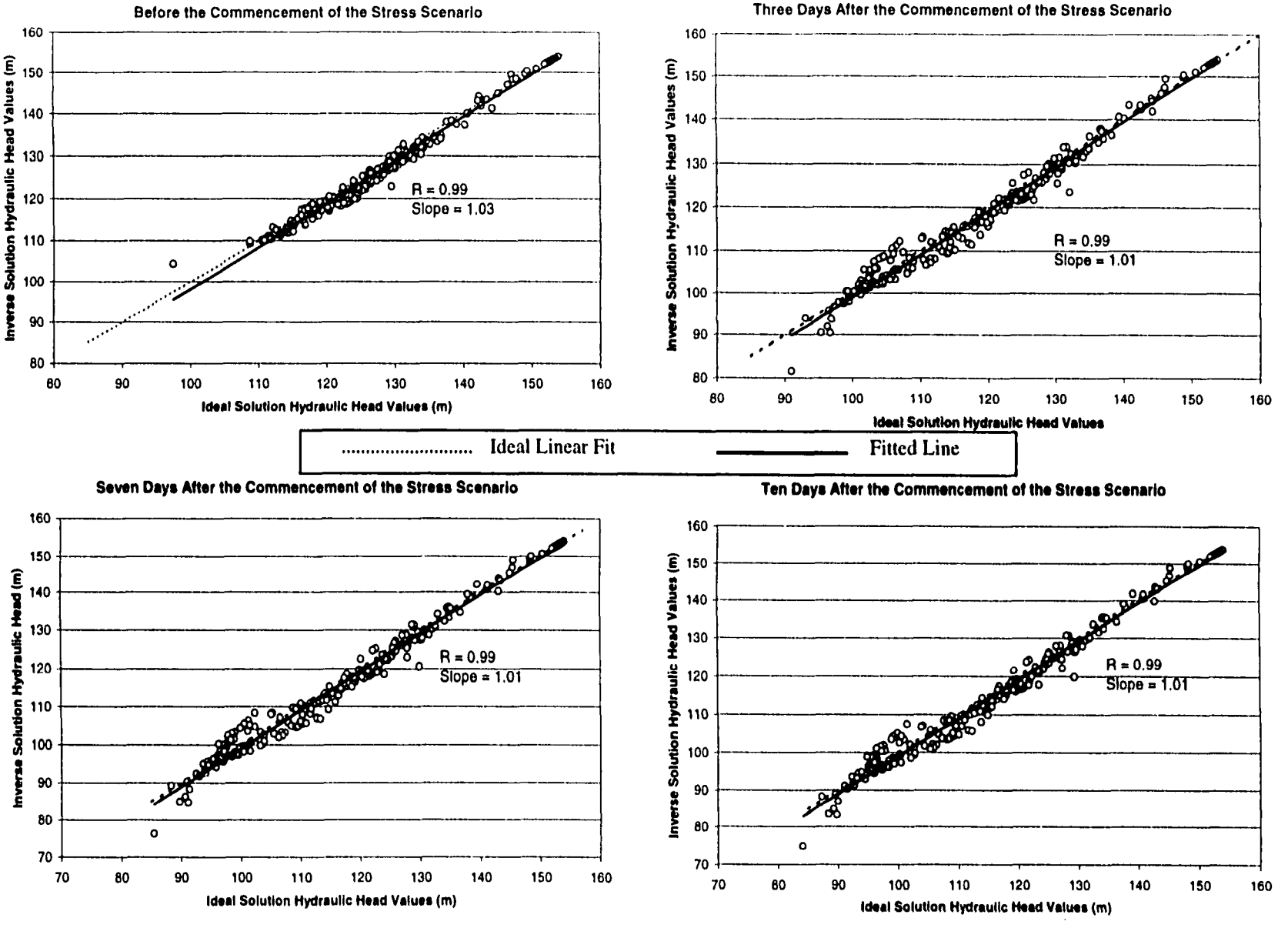


Figure 5.9 Analysis of the performance of solution **A** during the stress scenario

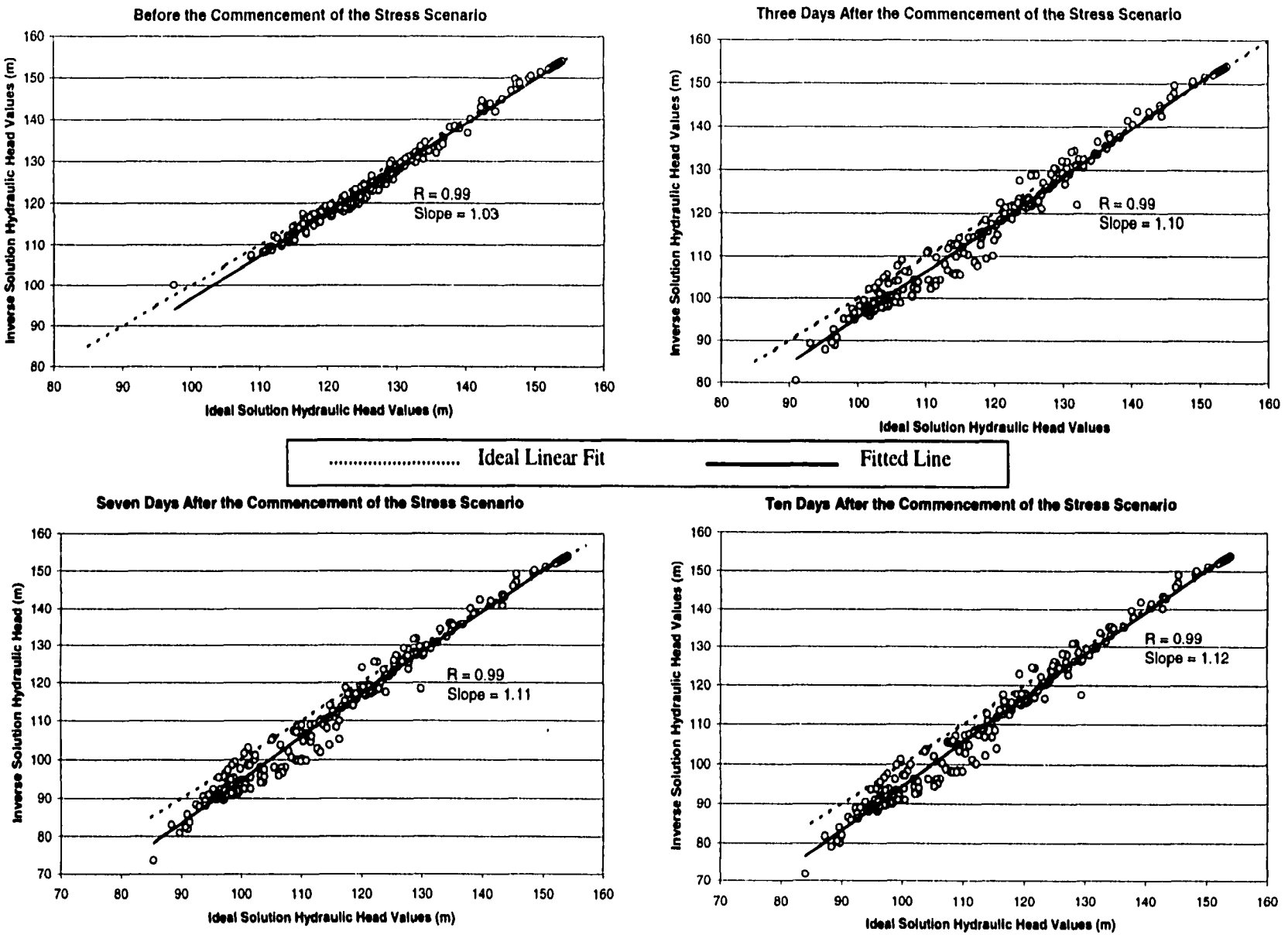


Figure 5.10 Analysis of the performance of solution ***A'*** during the stress scenario

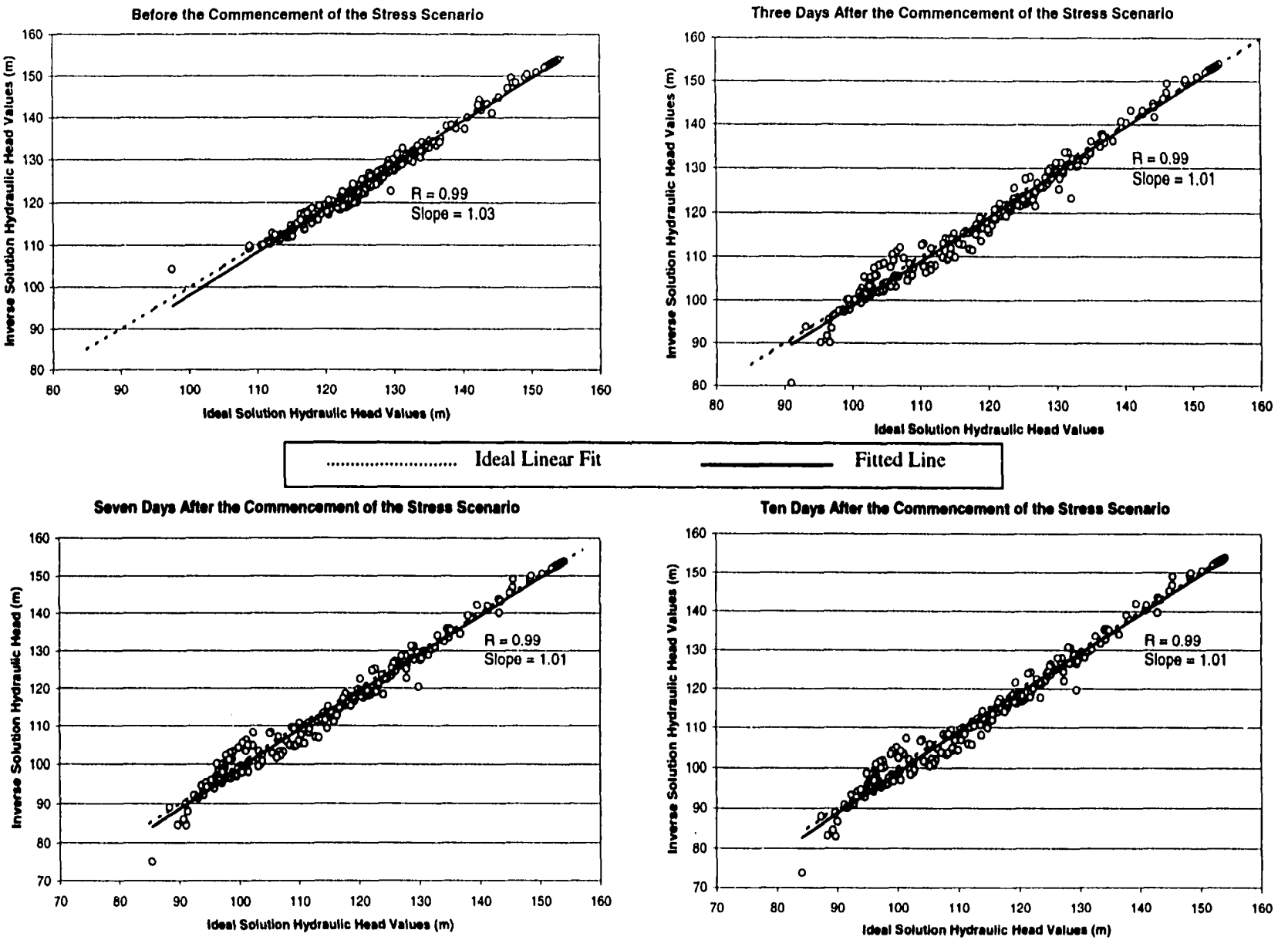


Figure 5.11 Analysis of the performance of solution ***A***'' during the stress scenario

problem. Furthermore, the refinement technique developed and used to produce solution **A''** was adopted as an integral part of the inversion algorithm that is to be used for the remaining part of this research. It is obvious that the solution produced by the refined technique (solution **A''**) did not exhibit a significant improvement over solution **A**. It is the author's opinion however that the refined technique adds to the versatility of the developed inverse technique, and even though its effect was minimal in this particular case, it has the potential to result in significant improvements to the solutions of other inverse problems.

5.4 The Inverse Problem an Ill-Posed Problem

5.4.1 Introduction

Mathematics has always been the vehicle through which engineers have posed and solved the problems they are interested in. No matter how complicated models are, they can be are simply tools that use the language of mathematics to express certain relations satisfied by physical, chemical and/or biological quantities. There is a common held belief among engineers that any physical process can be described and expressed in the form of a mathematical formulation by using a model, and there is a common misconception that all such formulations are well-posed. Ill-posed mathematical problems on the other hand, are often thought of as irrelevant and artificial.

A "well-posed" problem is defined as one that has all of the following properties:

1. There is at least on solution to the problem in question (i.e. a solution exists).
2. There is at most one solution (i.e. the solution is unique).
3. The solution is stable (i.e. slight modification of the input data do not lead to extreme changes in the solution).

A problem is labeled as ill-posed if it lacks at least on of the above mentioned properties. It turns out that the inverse problem presents itself as an ill-posed problem, and depending on the way the problem is posed and the solution technique used, the inverse solution may lack one or more of the features that are associated with the solutions of well-posed problems (i.e. existence, uniqueness, and stability).

It has been shown thus far that using the inverse technique developed in this research an inverse solution can be identified, thus resolving the issue of solution existence. The questions that remain to be answered are whether the methodology being used will produce a unique solution to this problem, and if this solution will be stable or not.

5.4.2 Uniqueness

Considering the methodology developed in this research, the question of solution uniqueness should be posed to address two possibilities. The first possibility is of the neural network's convergence to a different solution each time its is solved using the same initial guess. The second possibility is of the neural network's convergence to different solutions when different initial guesses are used. The effect of the initial transmissivity guess was singled for investigation because it is the only subjective variable in the solution parameters used. It is prudent to assume that if different people are solving the same problem, each one of them will use a different initial guess. Other parameters such as the hydraulic head field data would remain the same.

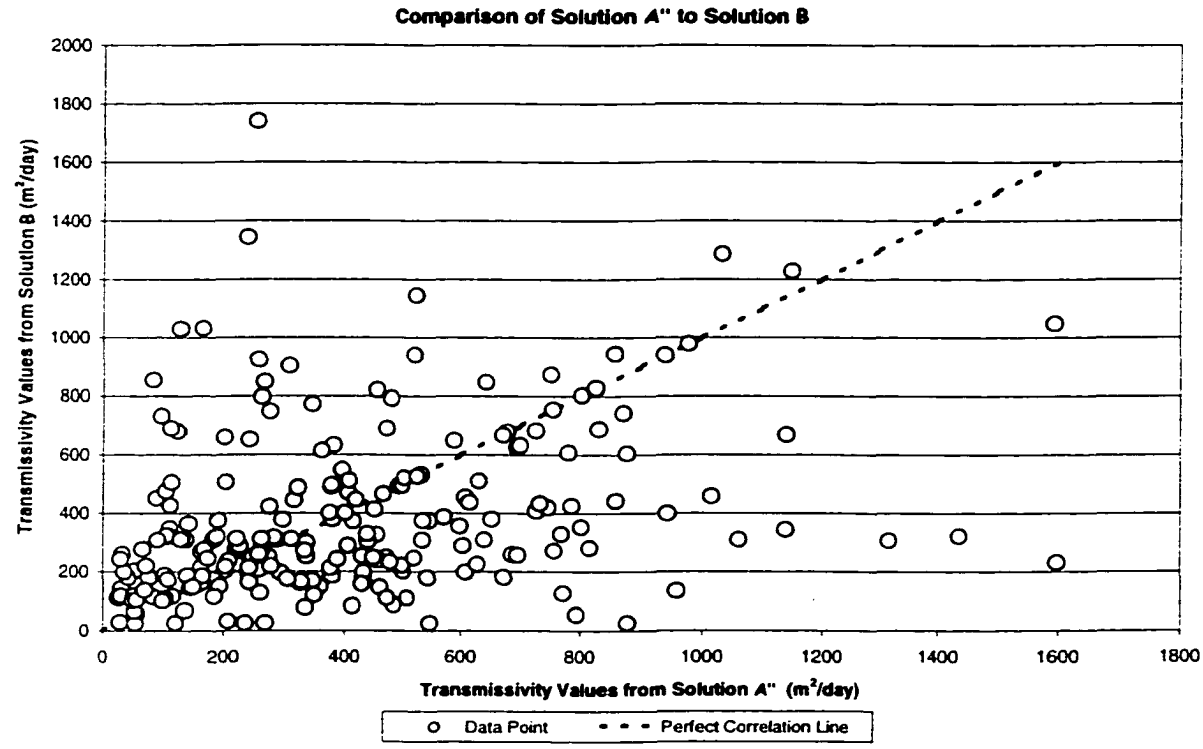
The problem was solved numerous times using the same initial guess that was used to obtain solution ***A***". This investigation has shown that short of prematurely stopping the network inversion algorithm, the inverse algorithm consistently arrived to

solution ***A''*** following exactly the same convergence curve each time. When different initial guesses were used however, the inversion algorithm always converged to a different transmissivity solution. This result gave rise to the question of the ability of these distinct solutions to produce the true hydraulic head map when used in MODFLOW, and whether they will be able to simulate the correct aquifer response to the induced stress scenario. The answers to these questions can be deduced from the sample results shown in Figures (5.12-5.17).

The sample results shown in these figures (Figures 5.12-5.17) were obtained by solving the inverse problem two more times using a different initial transmissivity guess each time. The generated solutions will be referred to hereafter as solutions **B** and ***C***. The transmissivity fields that were produced by these two solutions were used in MODFLOW to produce the hydraulic head map under steady state conditions. Each of the transmissivity fields and their corresponding hydraulic head fields were then used to compute the response they will project to the stress scenario.

Figure 5.12 shows the resulting transmissivity fields from solutions ***A''***, **B**, and ***C*** together with the true transmissivity field. The plot shows that the four solutions exactly honor the field transmissivity data, apart from that they appear to be distinct and uncorrelated. This observation is further corroborated by plotting the transmissivity field from solution ***A''*** versus those obtained from solutions **B** and ***C*** (Figure 5.13).

Figure 5.14 shows the hydraulic head maps that were generated by solutions ***A''***, **B**, ***C*** and for comparison purposes the true (target) hydraulic head map. The plots indicate that inspite of the fact that four distinct transmissivity fields were used to obtain these



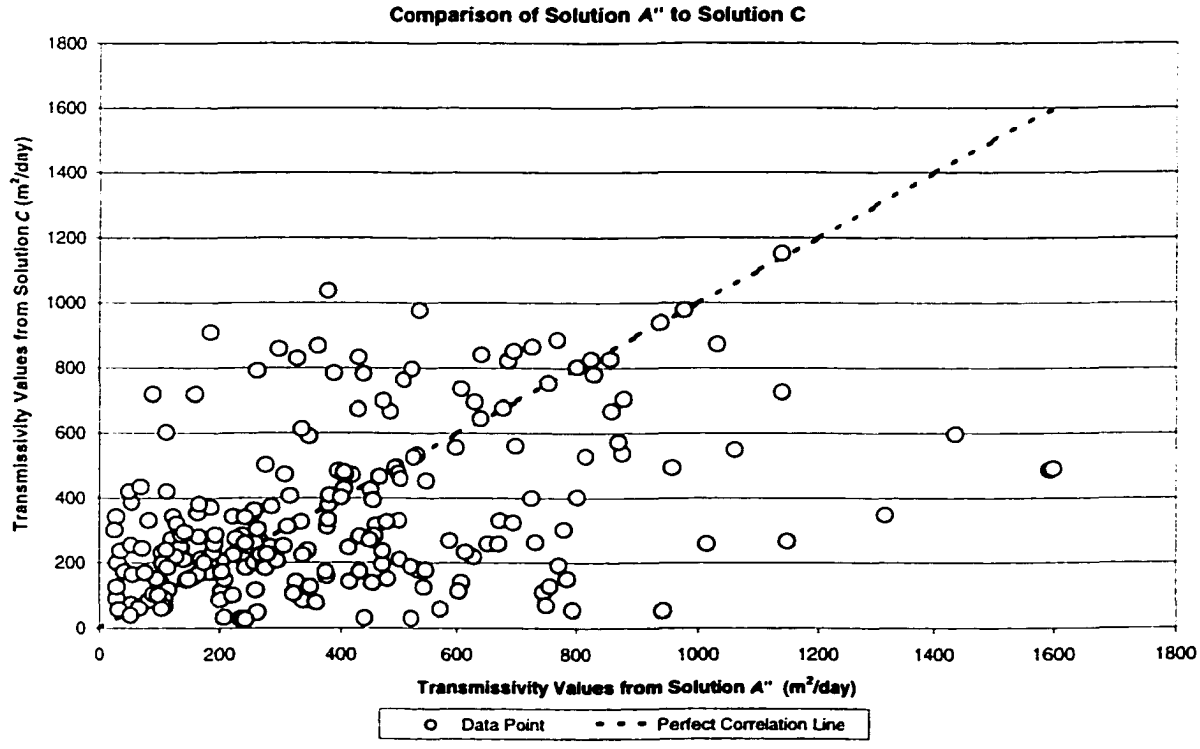
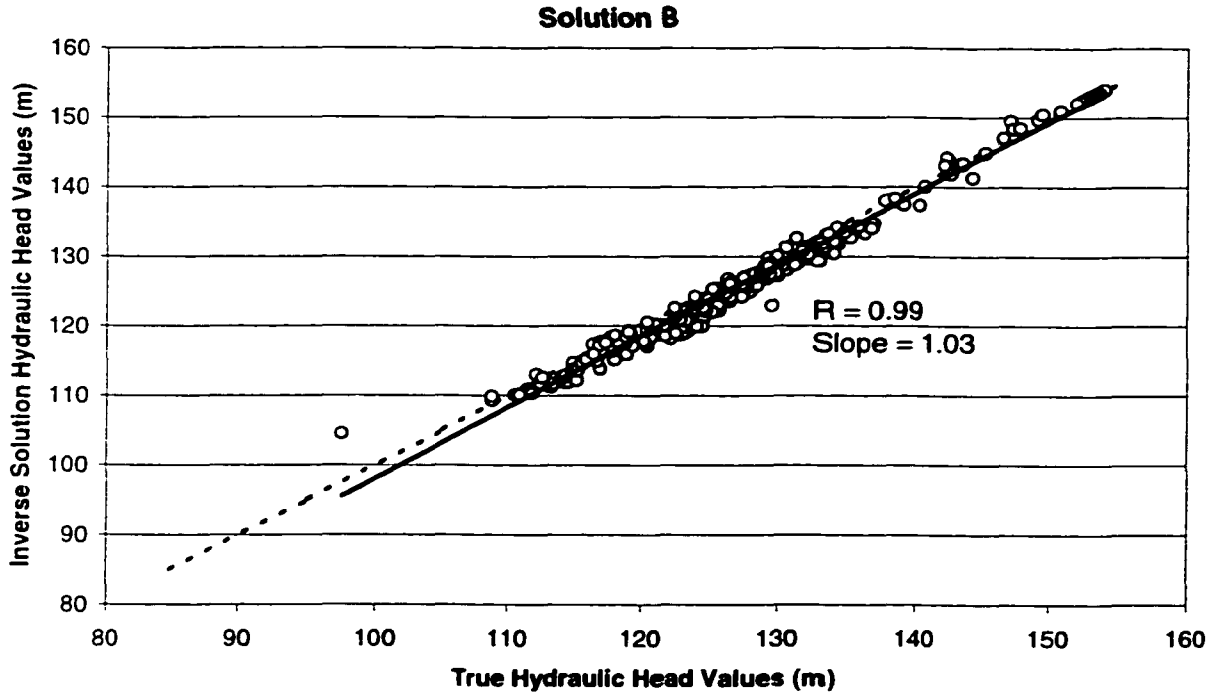


Figure 5.13 Comparison of solutions **B** and ***C*** to ***A***"



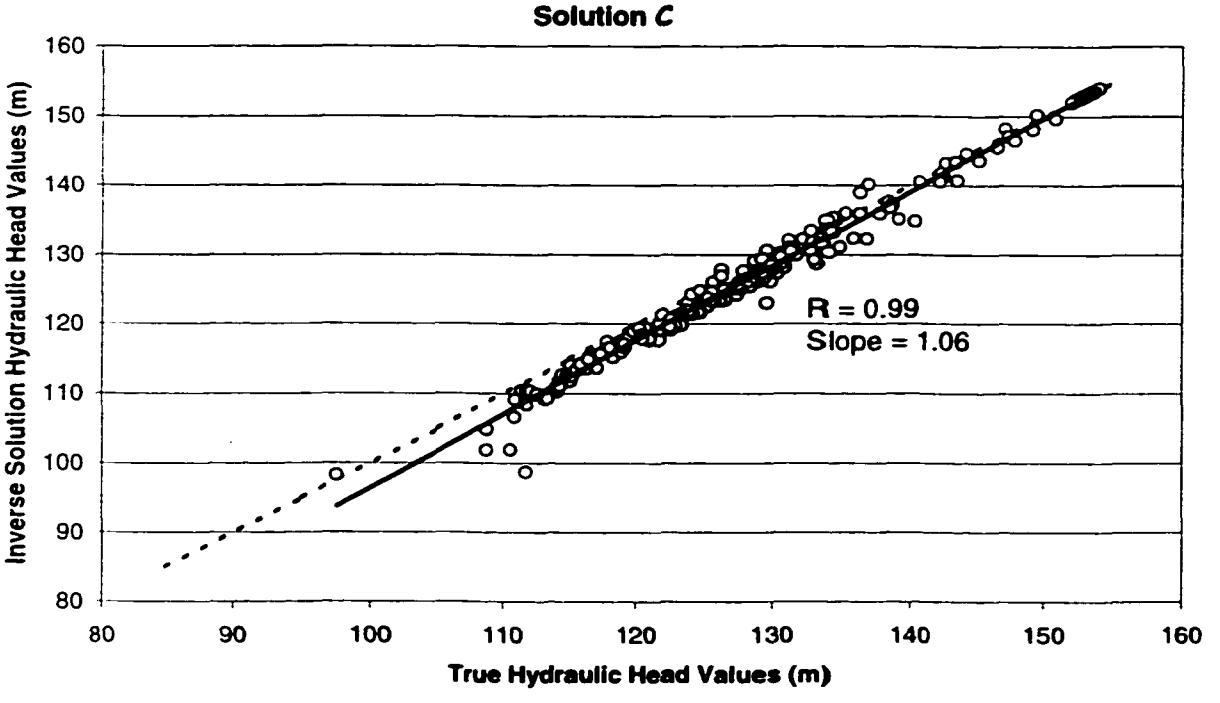


Figure 5.15 Comparison of the hydraulic head maps generated by solutions **B** and ***C*** to the true solution

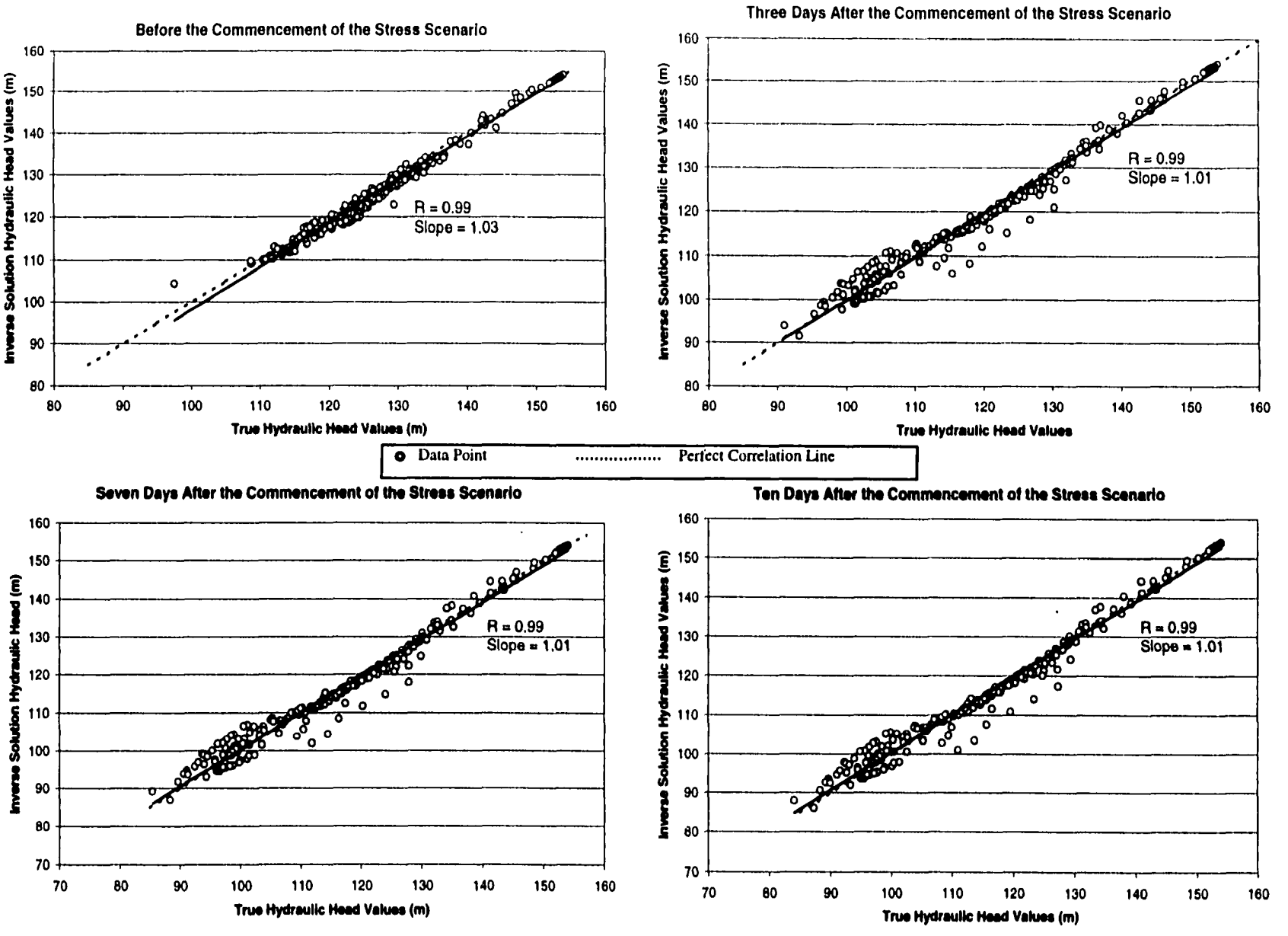


Figure 5.16 Analysis of the performance of solution **B** during the stress scenario

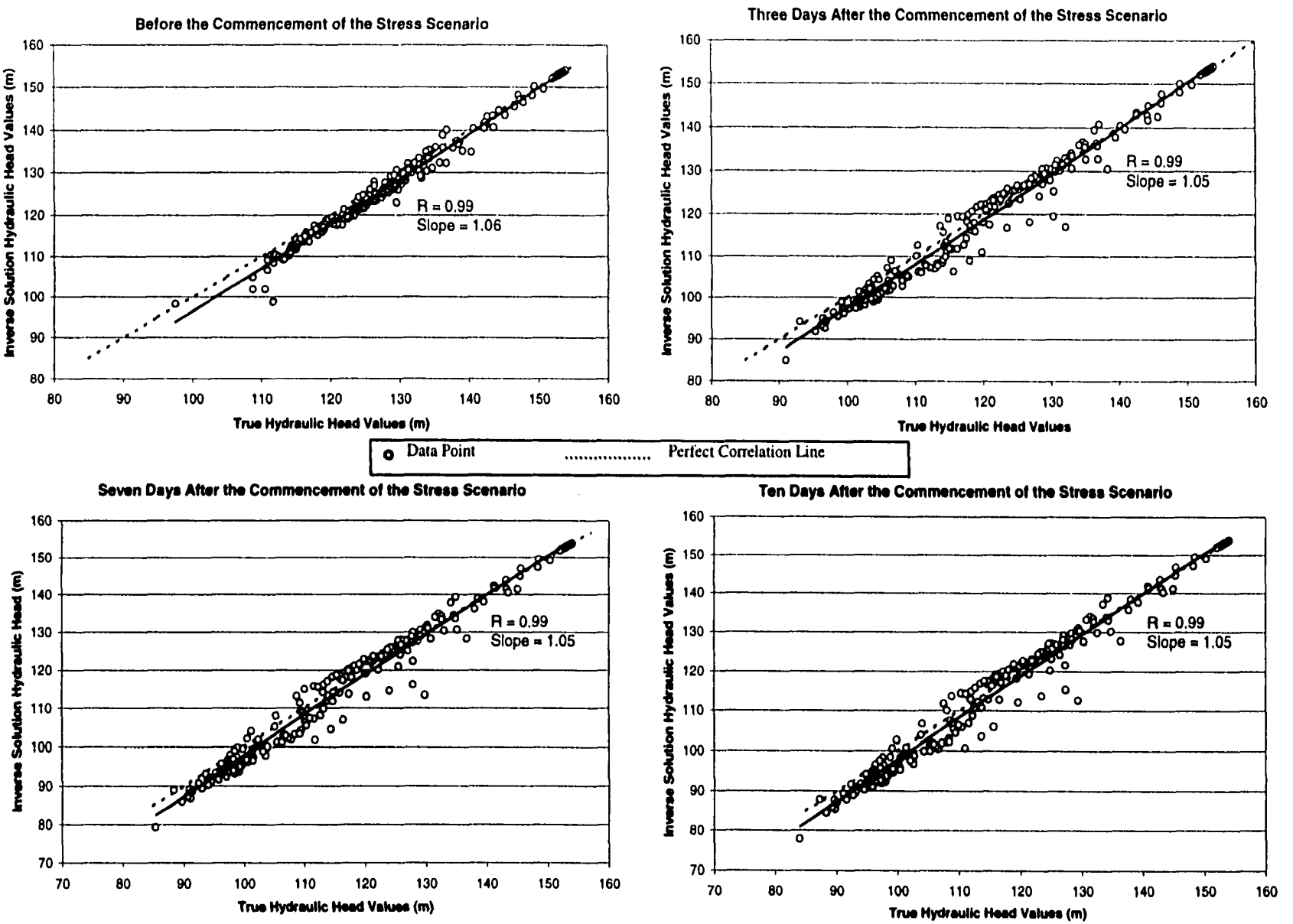


Figure 5. 17 Analysis of the performance of solution ***C*** during the stress scenario

hydraulic head maps, the four maps seems to be identical. This observation is further corroborated by plotting and correlating the true hydraulic values versus those obtained from solutions **B** and ***C***. The results (Figure 5.15) show that just like solution ***A"***, solutions **B** and ***C*** were capable of reproducing the true hydraulic head map with great accuracy.

Thus far it had been demonstrated that there is no unique solution to the posed problem, and that all of the obtained inverse solutions were capable of reproducing the true hydraulic head map with great accuracy. The question that remains to be answered is whether these distinct inverse solutions will be able to simulate the true aquifer response to high transient stresses. Figures 5.16 and 5.17 show the results of comparing the true aquifer response to the responses obtained from solutions **B** and ***C*** during selected times of the stress scenario. The results show that solutions **B** and ***C*** were able to simulate the true aquifer response throughout the period of the stress scenario with great accuracy. This means that even though there is no unique solution to the posed problem, solving the inverse problem by an iterative inversion of a well trained neural network will produce an inverse solution that is capable of simulating the correct aquifer response.

It is most likely that a distinct underlying structure, which characterizes all the inverse solutions that the network converges to, does exist. In an attempt to capture this underlying structure, a statistical analysis of the hydraulic conductivity fields which were obtained from solutions ***A"***, **B** and ***C*** was conducted. The results of the analysis (Table 5.3) reveal that the natural logarithm of the hydraulic conductivity of the inverse solutions has more or less the same mean and standard deviation values.

Table 5.3 Comparison of the statistics of solutions **A"**, **B** and **C**

Solution/Field Designation	**A"**	**B**	**C**	Hypothetical Case Study	Field Data
Mean of ln(K)	2.986	2.946	2.881	3.005	0.3012
Standard Deviation of ln(K)	0.872	0.804	0.802	0.782	0.6990

The analysis was extended to compute the semi-variograms of the obtained inverse solutions. A plot of the semi-variograms of the natural logarithm of the hydraulic conductivity values of solutions **A"**, **B** and **C** (Figure 5.18) shows that the three solutions have a similar correlation structure. Furthermore the parameters of the models that were fitted to the semi-variograms (Table 5.4) indicate that the three fields could be realizations generated from one semi-variogram.

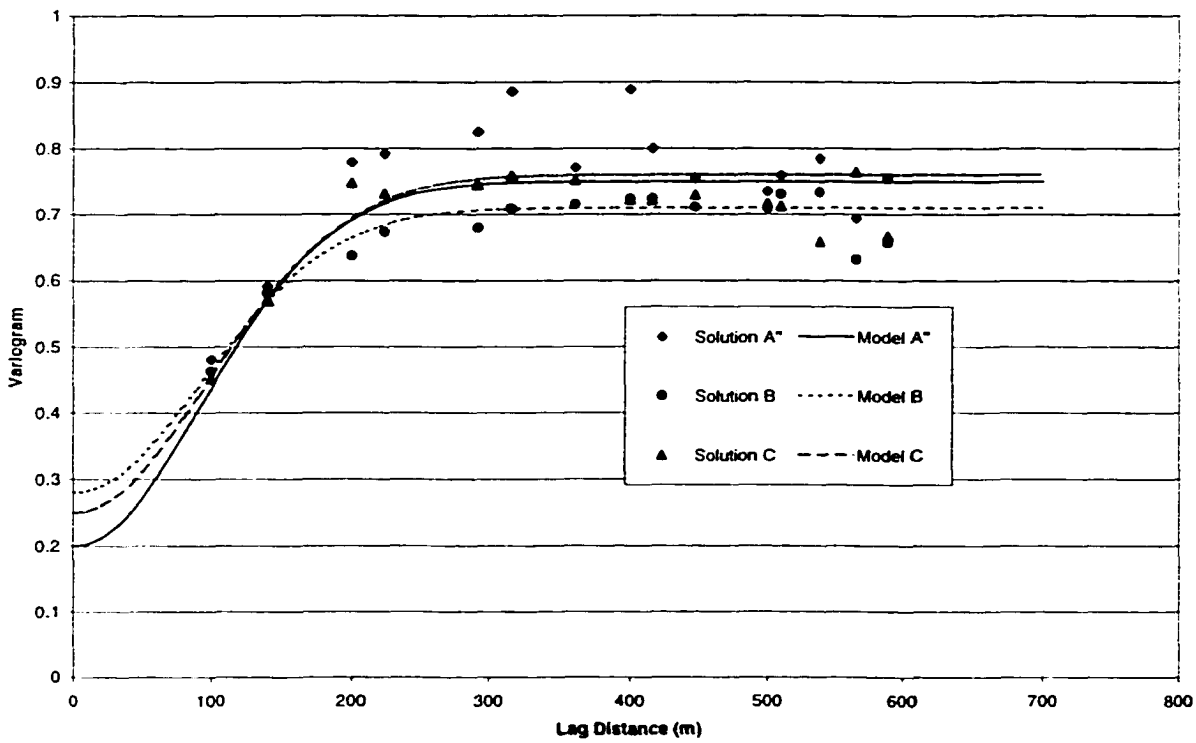


Figure 5.18 Semi-variograms of the inverse solutions

Table 5.4 Comparison of the semi-variograms model parameters

Solution	Model Type	Range (m)	Sill	Nugget
A''	Gaussian	400	0.55	0.20
B	Gaussian	400	0.43	0.28
C	Gaussian	420	0.51	0.25

5.4.3 Stability

The instability of any inverse model will make it very susceptible to small inaccuracies in field hydraulic head measurements. Such inaccuracies are known to occur in practice, and they are usually caused by seasonal fluctuations of the groundwater table. An unstable inverse model is thus rendered as useless and incapable of producing reliable results.

The problem of solution instability plagued the early attempts to solve the inverse problem [Stallman 1956, Nelson 1960, and Kleinecke 1971]. The failure of these early attempts to produce a stable solution has lead to a school of thought, which proclaimed that the inverse problem is hopelessly ill-posed and as such intrinsically unsolvable [Carrera 1986b]. Solution instability can be the result of one of two causes. The first cause is attributed to the way the problem is posed, while the second is attributed to the inherent capability of the inverse technique (model). Modifying the set of a-priori information and incorporating more field observations can easily eliminate instability that is caused by the way the problem is posed [Neuman 1973]. Traditionally the problem arises when the inverse model does not use or incorporate transmissivity field measurements to solve the inverse problem. This particular problem is not of concern

when using the iterative inversion of neural networks because the methodology requires the use of an initial transmissivity guess to start the solution process. The solution technique thus forces the user to pose the problem in a fashion that would eliminate the possibility of instability from the manner in which the problem is posed. The solution instability, which might be caused by the inherent nature of the inversion algorithm, was investigated by adding normal random errors to each of the target hydraulic head values. The problem was then solved by using the same initial guess that was used to arrive at solution **A''**. The investigation was conducted by adding four levels of error values that have a mean of zero and a standard deviation of 0.1, 0.2, 0.5 and 1.0 m (i.e. N(0,0.1), N(0,0.2), N(0,0.5), N(0,1.0)). The goal of the investigation was to compare the four obtained solutions to solution **A''** and to find out if the random errors that were added to the target values would lead to large fluctuations in the solution parameters obtained.

The analysis of the results was conducted by plotting the solution **A''** versus the four solutions that were obtained by adding the random errors (Figure 5.19). The plots show that added errors resulted only in small changes to the inverse solution obtained, which is indicative of the stability of the solution methodology. The four obtained inverse solutions were then used in MODFLOW to compute the hydraulic head field they would generate. By comparing these hydraulic head fields to the true hydraulic head field (Figure 5.20), it was found that adding small random errors to the target hydraulic field data actually resulted in an improvement of the ability of the inverse solution to map the true hydraulic head field. The inverse solutions which were obtained after adding random errors of mean zero and a standard deviation of 0.1 and 0.2 m were better able to map the correct hydraulic head map and aquifer response to hydrologic stresses than the solution

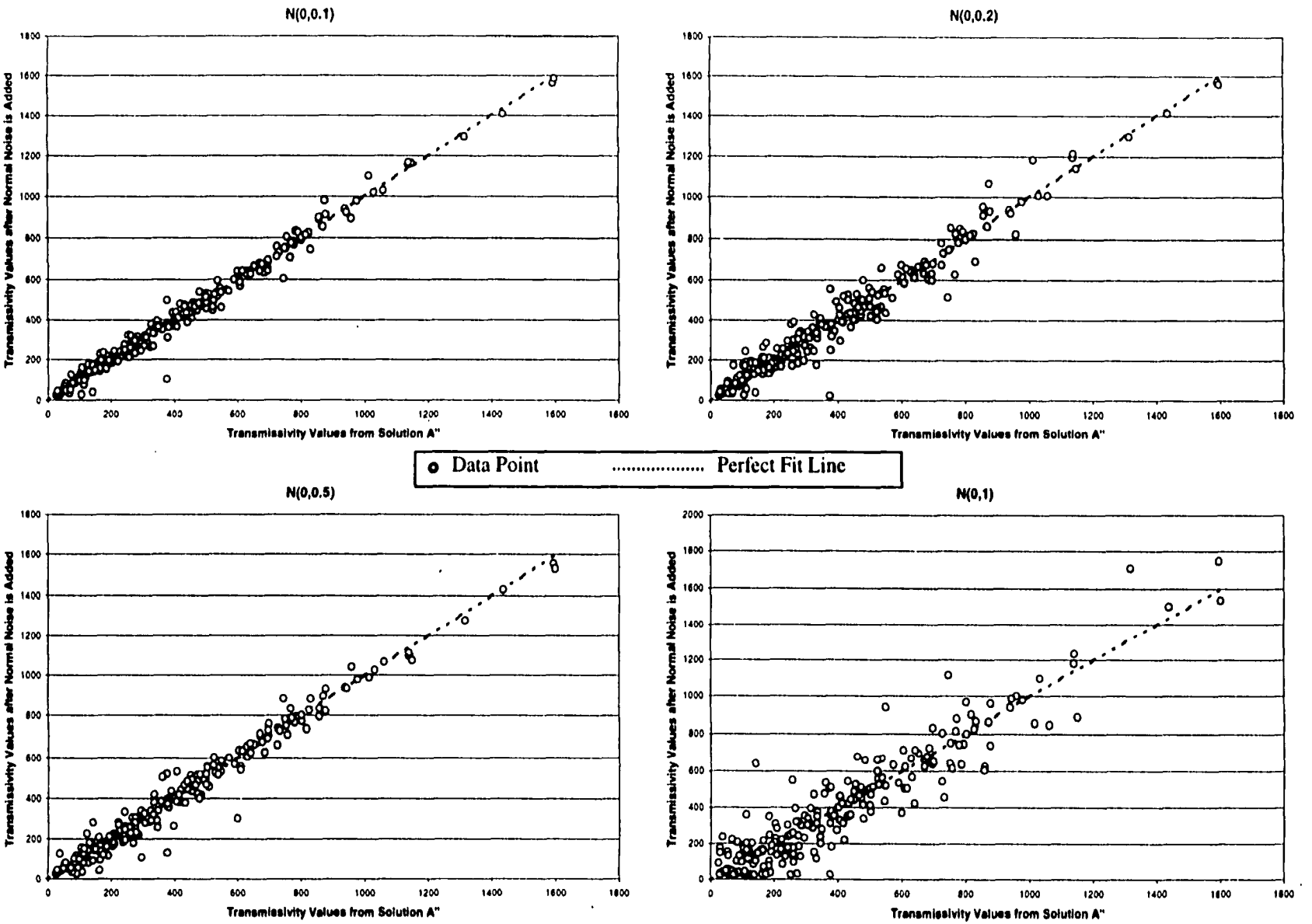


Figure 5.19 Analysis of solution stability to random errors in the field hydraulic head values

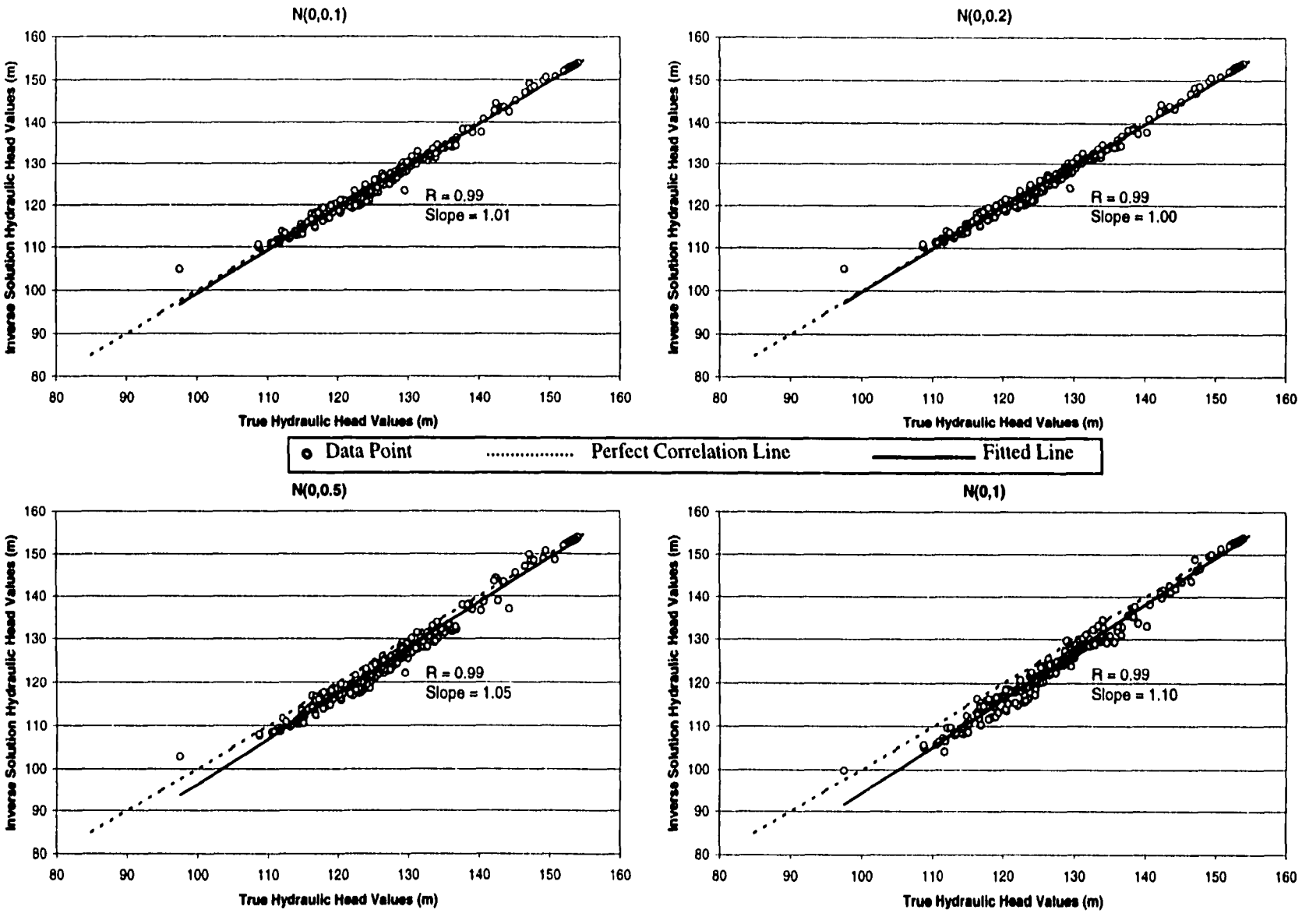


Figure 5.20 Effect of random errors on the hydraulic head map generated by the obtained inverse solution

A". A threshold is apparently reached after which increasing the standard deviation of the random errors results in producing a solution that is incapable of mapping the error free (true) hydraulic head field.

5.5 Effect of Releasing Target Hydraulic Head Constraints

One of the factors that can be of significance when solving an inverse problem by inverting a trained neural network is the number of known or constrained hydraulic head values. The effect of this factor was investigated for solution ***A***" for the problem at hand. The investigation was conducted by solving the inverse problem using the same initial guess that was used to obtain solution ***A***", while releasing some of the constrained (known) hydraulic head values. The investigation focused on the effect of releasing the hydraulic head values on the produced transmissivity field and whether it will lead to a new inverse solution that is distinct from solution ***A***". The subsequent effect on the hydraulic head map that was generated by the obtained solution was also investigated.

Solution ***A***" was reached by using 51 target hydraulic head values distributed randomly in the whole grid. The transmissivity values were presumed known in 29 of these 51 cells, while only the hydraulic head values were presumed known in the remaining 22 cells. It is these 22 cells that were released and assumed unknown for the purpose of the undertaken investigation. The remaining 29 points were used to solve the inverse problem, produce a transmissivity field and a subsequent hydraulic head field. Another solution was obtained by releasing all but one of the 51 target hydraulic head values. The hydraulic head map resulting from this solution was subsequently computed as well. The results of comparing the transmissivity and hydraulic head fields generated

by these two new solutions to those generated by solution **A"** are shown in Figures 5.21 and 5.22.

What is significant is that even with the release of all but one of the 51 target hydraulic head values that were used to obtain solution **A"**, the methodology converged to essentially the same solution (solution **A"**) with little distortion (Figure 5.21). The hydraulic head map that was generated by this solution however, deviated significantly from that which was obtained from solution **A"** (Figure 5.22). The release of 22 target hydraulic head constraints also resulted in a convergence to a transmissivity field that resembles that of solution **A"** (Figure 5.21). This field however, was capable of matching the hydraulic head field generated by solution **A"** with great accuracy (Figure 5.22).

The results clearly indicate that the effect of releasing the hydraulic head constraints is similar to that of adding random noise to the target hydraulic head data. In both instances the methodology converges to a similar solution with a little more distortion every time additional targets are released (or the magnitude of the added error is increased). A point is eventually reached after which releasing more constraints (or increasing the magnitude of the error) would lead to solutions that are incapable of accurately mapping the hydraulic head surface.

In a real life situation, an appropriate approach will be to divide the available hydraulic head data into two sets whereby, one set is used to solve the inverse problem while the second set is used to validate the obtained solution. If the need arises, data can be moved from the validation to the solution set, or more field data can be collected.

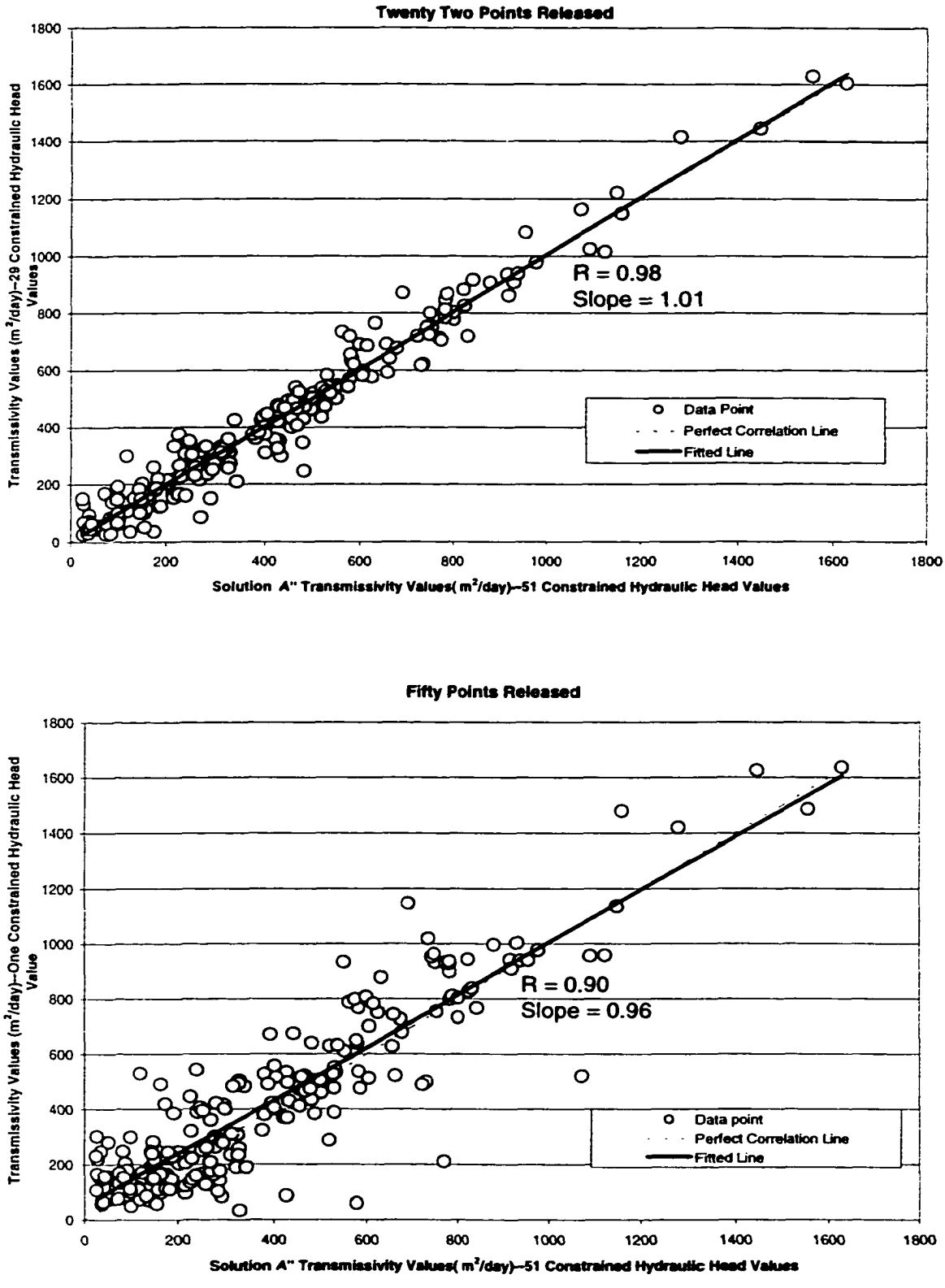
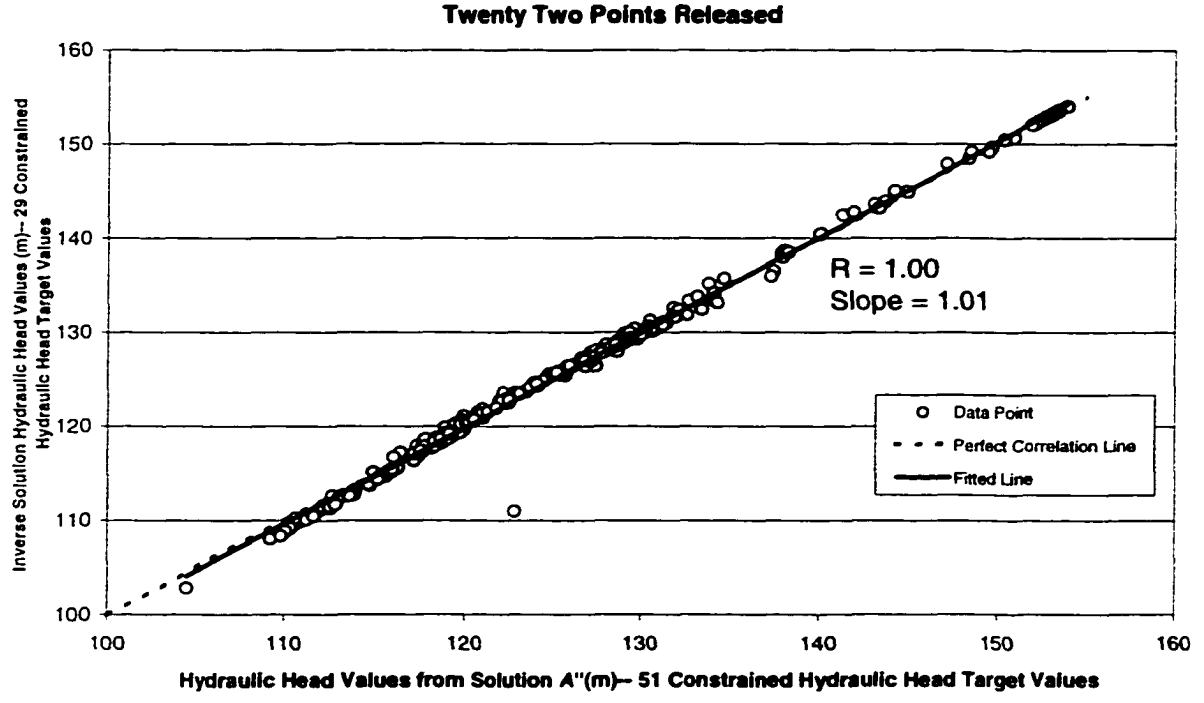


Figure 5.21 Effect of releasing hydraulic head constraints on the transmissivity fields generated by the inverse solutions



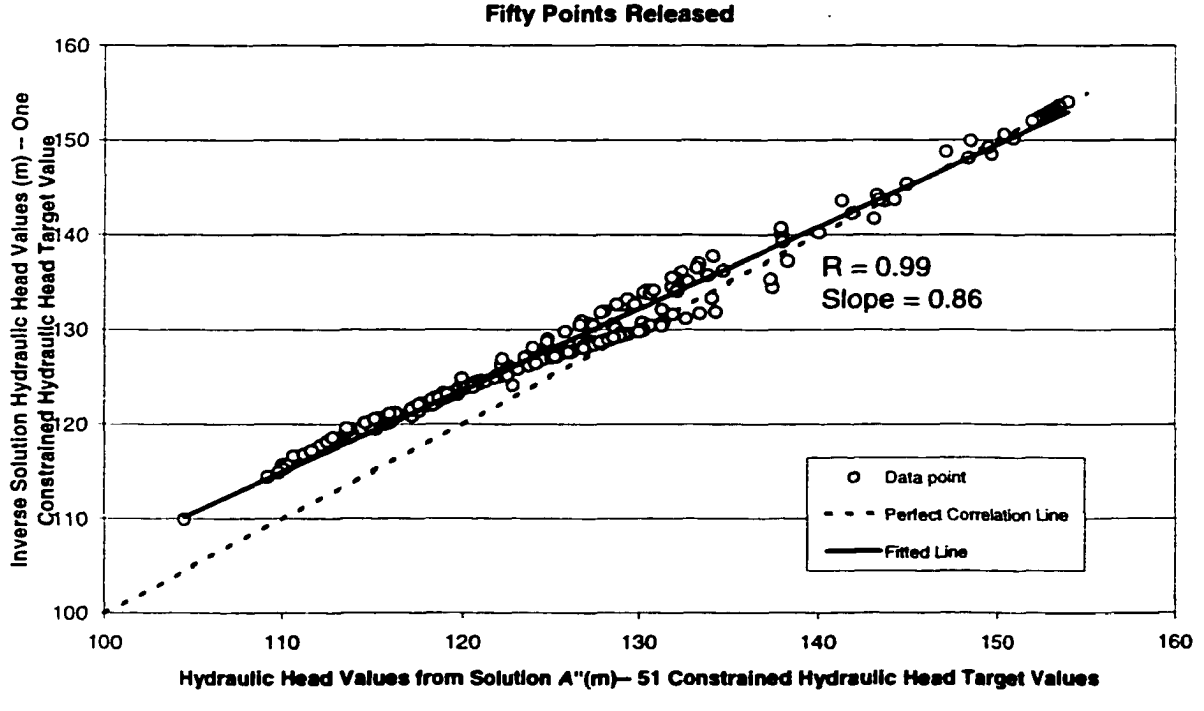


Figure 5.22 Effect of releasing hydraulic head constraints on the hydraulic head map generated by the inverse solution

5.6 Epilogue

The inverse problem in essence is one of estimating aquifer parameters while minimizing the difference (error) between the observed and computed hydraulic head values. This means that the inverse problem can be viewed as a special case of a non-linear optimization problem. An essential requirement of the solution to this optimization problem is the fulfillment of the implicit nonlinear relation which exist between the transmissivity and hydraulic head as described by the groundwater flow equation. This is essential if the computed inverse solution is to yield accurate estimates of the aquifer response to hydrologic stresses. If this element is neglected a solution which reproduces the field observations could be reached, but projections made by such a solution are bound to be erroneous and ultimately meaningless.

The inverse methodology which had been developed and presented in this research, uses a trained neural network to solve the optimization problem and produce an inverse solution. The fact that the neural network can be trained to provide a good approximation of the groundwater flow equation, enables the inverse solution it produces (upon its inversion) to perform well in predicting the aquifer response to various hydrologic stresses. This fact underlines the need to pay special attention to the training process. Using a poorly trained network would lead to inverse solutions that are incapable of correctly simulating the aquifer response. One of the advantages the methodology developed in this research has is the ability to determine before hand the field observations that are poorly mapped by the trained network, and whose use in solving the inverse problem would lead to erroneous results. The methodology also has the ability to

incorporate the local structure of the transmissivity field data, as well as exactly honoring the known transmissivity values.

Adopting the premises that the inverse problem is but a special nonlinear optimization problem leads to the conclusion that this problem would have infinite solutions. This was validated in this study, and it had been demonstrated that by changing the starting point (initial transmissivity guess) a different viable inverse solution can be reached. Each of these solutions is in essence a local minima which satisfies the constraints of the posed optimization problem. The non-uniqueness of the solution might seem very problematic to some people. This however need not be the case, the need is not for a unique solution, but rather for a viable solution that would simulate the correct behavior of the aquifer being modeled. The non-uniqueness of the inverse solution simply means that any prediction of the aquifer responses is best cast in a stochastic framework. A number of inverse solutions could be used to project the aquifer response, and obtain a probability distribution from which the certainty of occurrence of an event can be computed. The fact that different inverse solutions produce approximately the same hydraulic head map means that a relatively limited number of realizations would be needed to reliably cast the solution in a stochastic framework.

One of the questions that might be asked at this stage is of the comparison between the methodology which had been developed in this research and other inverse methodologies. The quest for determining which model is better is one that should be answered with objectivity and without bias. The answer is usually simple if one model gives better results than another, or if it has more desirable features such as stability and the ability to incorporate field measurements. The answer becomes difficult when the two

competing models give comparable results. In this case the model which should be preferred is the one which is simpler, or uses more elegant tools. What is to be noted here is the premise of comparable evaluation. An objective comparison can not be made on the basis of results from non-comparable problems. Unless all the possibilities are checked and a rigorous analysis is conducted, little can be said about the superiority of one inverse methodology to the other.

CHAPTER 6

CONCLUSIONS

The primary objective of this research was to develop a new methodology for solving the inverse problem in groundwater flow. This research developed a methodology which uses an iterative process to invert a neural network that had been trained to approximate the groundwater flow equation so as to produce a solution to the inverse problem. During the course of this research the following conclusions were reached:

1. A two layer neural network was successfully trained to provide a data driven approximation of the relation between transmissivity and hydraulic head as stipulated by the groundwater flow equation.
2. The trained neural network was inverted through an iterative process to produce a viable solution to the inverse problem. The inverse solution produced was not only able to reproduce the true hydraulic head values; it was also capable of simulating the correct aquifer response to a newly introduced stress scenario.
3. The methodology developed is capable of producing an inverse solution that honors the transmissivity field measurements, incorporates information about the expected transmissivity values and also incorporates information about the reliability of different field observations. The methodology is also capable of pointing out the field observations that could lead to erroneous results.

4. The inverse solution produced by the iterative inversion of a neural network has proven to be stable when random errors were added to the hydraulic head observations. The addition of small normal random errors led to an improvement in the ability of the resulting inverse solution to better map the aquifer response.
5. This research has demonstrated that there is no unique solution to the inverse problem. Using the developed methodology, a different solution can be reached for each distinct set of starting points (initial transmissivity guess). Each of these distinct solutions however, is capable of reproducing the true hydraulic head map with good accuracy.
6. Predictions of an aquifer response are best cast in a stochastic framework in which the certainties of the predictions are quantified from realizations that are generated from an array of inverse problems.

CHAPTER 7

RECOMMENDATIONS

In light of the results reached during the course of this research, the following recommendations for future research are made:

1. A study of the effect of ranking and selectively using the training patterns on the number of patterns needed to train the neural network should be conducted. Ranking the training patterns and using some type of sampling could potentially result in a large reduction in the number of patterns needed to train the network.
2. During the course of this research, it was discovered that adding small random normal errors to the hydraulic head field values led to an improvement in the quality of the inverse solution. A full investigation of the causes of this phenomenon and the effect of the error levels on the quality of the inverse solution should be conducted.
3. With the availability of several promising inverse techniques, an objective and comprehensive study to compare these techniques should be conducted. This comparison is best conducted by using different competing inverse techniques to solve several complex hypothetical test cases. Evaluation criteria need to be developed to rank the different methodologies.
4. A study to extend the methodology developed during this research to solve the inverse problem in contaminant transport should be conducted. The iterative inversion of a neural network that approximates the advection dispersion equation

could simultaneously produce the groundwater flow as well as the contaminant transport parameters.

BIBLIOGRAPHY

[1] Alsugair, A. M., and A. Al-Qudrah, Artificial neural network approach for pavement maintenance, *Journal of Computing in Civil Engineering*, Vol. 12, No. 4, pp. 249-255, October 1998.

[2] Bedient, P. B., H. S. Rifai, and C. J., Newell, Ground water contamination, *Prentice Hall*, New Jersey, 541 pp., 1994.

[3] Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions, 1, Maximum likelihood method incorporating prior information, *Water Resources Research*, Vol. 22, No. 2, pp. 199-210, 1986.

[4] Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions, 2, Uniqueness, stability, and solution algorithms, *Water Resources Research*, Vol. 22, No. 2, pp. 211-227, 1986.

[5] Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions, 3, Application to synthetic and field data, *Water Resources Research*, Vol. 22, No. 2, pp. 228-242, 1986.

[6] Chen, X, and J. F. Ayers, Aquifer properties determined from two analytical solutions, *Groundwater*, Vol. 36, No. 5, pp. 783-791, 1998.

[7] Coats, K. H., J. R. Dempsey, and J. H. Henderson, A new technique for determining reservoir description from field performance data, *Society of Petroleum Engineers*, pp. 66-74, 1970.

[8] Cooley, R. L., A comparison of several methods of solving non-linear regression groundwater flow problems, *Water Resources Research*, Vol. 21, No. 10, pp. 1525-1538, 1985.

[9] Cooley, R. L., A method of estimating parameters and assessing reliability for models of steady state groundwater flow, 1, Theory and numerical properties, *Water Resources Research*, Vol. 13, No. 2, pp. 318-324, 1977.

[10] Cooley, R. L., A method of estimating parameters and assessing reliability for models of steady state groundwater flow, 2, Application of statistical analysis, *Water Resources Research*, Vol. 15, No. 3, pp. 603-617, 1979.

[11] Cooley, R. L., Incorporation of prior information on parameters into non-linear regression groundwater flow models, 1, Theory, *Water Resources Research*, Vol. 18, No. 4, pp. 965-976, 1982.

[12] Davis, D. T., Z. Chen, L. Tsang, J-N Hwang, and A. T. C. Chang, Retrieval of snow parameters by iterative inversion of a neural network, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 31, No. 4, pp. 842-852, July 1993.

[13] Deutsch, C. V., and A. G. Journel, GSLIB geostatistical software library and user's guide, *Oxford University Press*, New York, 369 pp., 1998.

[14] Deutsch, C. V., DECLUS: a Fortran 77 program for determining optimum spatial declustering weights, *Computers and Geosciences*, Vol. 15, No. 3, pp. 325-332, 1989.

[15] Emsellem, Y., and G. de Marsily, An automatic solution for the inverse problem, *Water Resources Research*, Vol. 7, No. 5, pp. 1264-1283, 1971.

[16] Englund, E., and A. Sparks, Geostatistical environmental assessment software, *Environmental Monitoring Systems, Laboratory, Office of Research and Development, U.S. Environmental Protection Agency*, Las Vegas, 1991.

[17] Ferraresi, M., E. Todini, and R. Vignoli, A solution to the inverse problem in groundwater hydrology based on Kalman filtering, *Journal of Hydrology*, No. 175, pp. 567-581, 1996.

[18] Fetter, C. W., Applied Hydrogeology, *Macmillan College Publishing Company*, New York, 691 pp., 1994.

[19] Flood, I., and N. Kartam, Neural networks in civil Engineering. I: Principles and understanding, *Journal of Computing in Civil Engineering*, Vol. 8, No. 2, pp. 131-148, April 1994.

[20] Flood, I., and N. Kartam, Neural networks in civil Engineering. II: Systems and applications, *Journal of Computing in Civil Engineering*, Vol. 8, No. 2, pp. 149-162, April 1994.

[21] Freeze, R. A., and J. A. Cherry, Groundwater, *Prentice Hall*, New Jersey, 604 pp., 1979.

[22] Freeze, R. A., and J. A. Cherry, Groundwater, *Prentice-Hall*, New Jersey, 604 pp., 1979.

[23] Gavalas, G. R., P. C. Shah, and J. H. Seinfeld, Reservoir history matching by Bayesian estimation, *Society of Petroleum Engineers Journal*, pp. 337-350, December 1976.

[24] Goh, A. T. C., Modeling soil corelations using neural networks, *Journal of Computing in Civil Engineering*, Vol. 9, No. 4, pp. 275-278, October 1995.

[25] Gomez-Hernandez, J., and R. Srivastava, ISIM3D: An Ansi-C three dimensional multiple indicator conditional simulation program, *Computers and Geosciences*, Vol. 16, No. 4, pp. 395-440, 1990.

[26] Gupta, H. V., and S. Sorooshian, Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information, *Water Resources Research*, Vol. 34, No. 4, pp. 751-763, 1998.

[27] Harbaugh, A. W., and M. G. McDonald, User's documentation for MODFLOW-96, an update to the U.S. geological survey modular finite-difference ground-water flow model, *U.S. Geological Survey*, Virginia, 56 pp., 1996.

[28] Hefez, E., U. Shamir, and J. Bear, identifying the parameters of an aquifer cell model, *Water Resources Research*, Vol. 11, No. 6, pp. 993-1004, 1975.

[29] Hill, M. C., R. L. Cooley, and D. W. Pollock, A controlled experiment in ground water flow model calibration, *Groundwater*, Vol. 36, No. 3, pp. 520-535, 1998.

[30] Hoeksema, R. J., and P. K. Kitanidis, An application of the geostatistical approach to the inverse problem in two-dimensional groundwater modeling, *Water Resources Research*, Vol. 20, No. 7, pp. 1003-1020, 1984.

[31] Hoeksema, R. J., and P. K. Kitanidis, Analysis of the spatial structure of properties of selected aquifers, *Water Resources Research*, Vol. 21, No. 4, pp. 563-572, 1985.

[32] Honjo, Y, and N. Kashiwagi, Matching objective and subjective information in groundwater inverse analysis by Akaike's Bayesian Information Criterion, *Water Resources Research*, Vol. 35, No.2, pp. 435-447, 1999.

[33] Hsieh, W. W., and B. Tang, Applying neural network models to prediction and data analysis in meteorology and oceanography, *Bulletin of the American Meteorological Society*, Vol. 79, No. 9, pp.1855-1870, September 1998.

[34] Hsu, K-l, H. V. Gupta, and S. Sorooshian, Artificial neural network modeling of the rainfall-runoff process, Water Resources Research, Vol. 31, No. 10, pp. 2517-2530, October 1995.

[35] Hwang, J-N, and C. H. Chan, Iterative constrained inversion of neural networks and its applications, *Proceedings of the 24th Conference on Information Sciences and Systems*, pp. 754-759, March 1990.

[36] Hyun, Y, and K-K. Lee, Model identification criteria for inverse estimation of hydraulic parameters, *Ground Water*, Vol. 36, No. 2, pp. 230-239, 1998.

[37] Jacquard, P., and C. Jain, Permeability distribution from field pressure data, *Society of Petroleum Engineers*, pp. 281-294, December 1965.

[38] Karunanithi, N., W. J. Grenney, D. Whitley, and K. Bovee, Neural networks for river flow prediction, *Journal of Computing in Civil Engineering*, Vol. 8, No. 2, pp. 201- 219, April 1994.

[39] Keidser, A., and D. Rosbjerg, A comparison of four inverse approaches to groundwater flow and transport parameter identification, *Water Resources Research*, Vol. 27, No. 9, pp. 2219-2232, 1991.

[40] Kitanidis, P. K., and E. G. Vomvoris, A geostatistical approach to the inverse problem in groundwater modeling (steady state) and one-dimensional simulations, *Water Resources Research*, Vol. 19, No. 3, pp. 677-690, 1983.

[41] Kitanidis, P. K., How observations and structure affect the geostatistical solution to steady-state inverse problem, *Groundwater*, Vol. 36, No. 5, pp. 754-763, 1998.

[42] Kleinecke, D. Use of linear programming for estimating geohydrologic parameters of groundwater basins, *Water Resources Research*, Vol. 7, No. 2, pp. 367-374, 1971.

[43] Kuligowski, R. J., and A. P. Barros, Using artificial neural networks to estimate missing rainfall data, *Journal of the American Water Resources Association*, Vol. 34, No. 6, pp. 1437-1447, December 1998.

[44] La Venue, A. M., and J. F. Pickens, Application of a coupled adjoint sensitivity and kriging approach to calibrate a groundwater flow model, *Water Resources Research*, Vol. 28, No. 6, pp. 1543-1569, 1992.

[45] La Venue, A. M., B. S. RamaRao, G. de Marsily, and M. G. Marietta, Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields, 2, Application, *Water Resources Research*, Vol. 31, No. 3, pp. 495-516, 1995.

[46] Lahkim, M. B., A stochastic approach to environmental exposure assessment related to groundwater contamination, *Colorado State University*, Spring 1998.

[47] Lapedes, A., and R. Farber, How neural networks work, *Proceedings of IEEE Conference on Neural Information Processing Systems*, edited by D. Z. Anderson, pp. 442-456, 1988.

[48] Mackay, R., and P. E. O'Connell, Statistical methods of characterizing hydrogeological parameters, in *Applied Groundwater Hydrology*, edited by R. A. Downing and W. B. Wilkinson, pp. 217-242, Oxford University Press, New York, 1991.

[49] Maier, H. R., and G. C. Dandy, The use of artificial neural networks for the prediction of water quality parameters, *Water Resources Research*, Vol. 32, No. 4, pp. 1013-1022, 1996.

[50] Mc Whorter, D. B., and D. K. Sunada, Ground-water hydrology and hydraulics, *Water Resources Publications, Colorado*, 290 pp., 1979.

[51] McLaughlin, D, and L. R. Townley, A reassessment of the groundwater inverse problem, *Water Resources Research*, Vol. 32, No. 5, pp. 1131-1161, 1996.

[52] Medina, A., and J. Carrera, Coupled estimation of flow and solute transport parameters, *Water Resources Research*, Vol. 32, No. 10, pp. 3063-3076, 1996.

[53] Mehrotra, K., C. K. Mohan, and S. Ranka, Elements of artificial neural networks, *The MIT Press*, Massachusetts, 344 pp.1997.

[54] Murty, V. V. N., and V. H. Scott, Determination of transport model parameters in groundwater aquifers, *Water Resources Research*, Vol. 13, No. 6, pp. 941-947, 1977.

[55] Nelson, R. W., In-place measurement of permeability in heterogeneous media, 1, Theory of a proposed method, *Journal of Geophysical Research*, Vol. 65, No. 6, pp. 1753-1758, 1960.

[56] Nelson, R. W., In-place measurement of permeability in heterogeneous media, 2, Experimental and computational considerations, *Journal of Geophysical Research*, Vol. 66, No. 8, pp. 2469-2478, 1961.

[57] Nelson, R. W., Conditions for determining areal permeability distribution by calculation, Society of Petroleum Engineers, pp. 223-224, September 1962.

[58] Nelson, R. W., In-place determination of permeability distribution for heterogeneous porous media through analysis of energy dissipation, *Society of Petroleum Engineers*, pp. 33-42, March 1968.

[59] Neuman, S. P, Calibration of distributed groundwater flow models viewed as a multi-objective decision process under uncertainty, *Water Resources Research*, Vol. 9, No. 4, pp. 1006-1021, 1971.

[60] Neuman, S. P., and S. Yakowitz, A statistical approach to the inverse problem of aquifer hydrology, 1, Theory, *Water Resources Research*, Vol. 15, No. 4, pp. 845-860, 1979.

[61] Neuman, S. P., Calibration of distributed parameter groundwater flow models viewed as a multi-objective decision process under uncertainty, *Water Resources Research*, Vol. 9, No. 4, pp. 1006- 1021, 1973.

[62] Part-Enander, E, The MATLAB handbook, *Addison-Wesley*, England, 423 pp., 1996.

[63] Peck, A, S. Gorelick, G. de Marsily, S. Foster, and V. Kovalevsky, Consequences of spatial variability in aquifer properties and data limitations for groundwater modelling practice, *IAHS Publication No. 175*, 272 pp., 1988.

[64] Piskunov, N., Differential and integral calculus vol. I, Mir Publications, Moscow, 471 pp., 1974.

[65] Raman, H., and V. Chandramouli, Deriving a general operating policy for reservoirs using neural network, *Journal of Water Resources Planning and Management*, Vol. 122, No. 5, pp. 342-347, September/October 1996.

[66] RamaRao, B. S., A. M. La Venue, G. de Marsily, and M. G. Marietta, Pilot point methodology for automated calibration of an ensemble of conditionally simulated transmissivity fields, 1, Theory and computational experiments, *Water Resources Research*, Vol. 31, No. 3, pp. 475-493, 1995.

[67] Scott, D. M., An evaluation of flow net analysis for aquifer identification, *Groundwater*, Vol. 30, No. 5, pp. 755-764, 1992.

[68] Shah, P. C., G. R. Gavalas, and J. H. Seinfeld, Error analysis in history matching: The optimum level of parametrization, *Society of Petroleum Engineers Journal*, pp. 219-228, June 1978.

[69] Sidauruk, P., and A. H-D. Ouazar, Ground water contaminant source and transport parameter identification by correlation coefficient optimization, *Ground Water*, Vol. 36, N0. 2, pp. 208-214, 1998.

[70] Slater, G. E., and E. J. Durrer, Adjustment of reservoir simulation models to match field performance, *Society of Petroleum Engineers*, pp. 295-305, 1971.

[71] Smith, J., and R. N. Eli, Neural-network models of rainfall-runoff process, *Journal of Water Resources Planning and Management*, Vol. 121, No. 6, pp. 499-508, November/December 1995.

[72] Solley, W. B., R. R. Pierce, and H. A. Perlman, Estimated use of water in the United States in 1995, *U.S. Geological Survey Circular 1200*, Denver, CO, 1998.

[73] Stallman, R.W., Numerical analysis of regional water levels to define aquifer hydrology, *Transactions of American Geophysical Union*, Vol. 37, No. 4, pp. 451-460, 1956.

[74] Strecker, E. W., and W-S. Chu, Parameter identification of a ground-water contaminant transport model, *Ground Water*, Vol. 24, No. 1, 1986.

[75] Sun, N-Z, and W. W-G Yeh, A stochastic inverse solution for transient groundwater flow: Parameter identification and reliability analysis, *Water Resources Research*, Vol. 28, No. 12, pp. 3269-3280, 1992.

[76] Sun, N-Z, S. Yang, and W. W-G. Yeh, A proposed stepwise regression method for model structure identification, *Water Resources Research*, Vol. 34, No. 10, pp. 2561-2572, 1998.

[77] Sureerattanan, S., and H. N. Phein, Backpropagation networks for daily streamflow forecasting, *Water Resources Journal*, ST/ESCAP/SER. C/195, pp. 1-7, 1997.

[78] Todd, D. K., Groundwater hydrology, *John Wiley & Sons*, New York, 535 pp., 1980.

[79] Umari, A., R. Willis, and L-F. L. Philip, Identification of aquifer dispersivities in two-dimensional transient groundwater contaminant transport:, An optimization approach, *Water Resources Research,* Vol. 15, No. 4, pp. 815-831, 1979.

[80] Wagner, B. J., and S. M. Gorelick, A statistical methodology for estimating transport parameters: Theory and applications to one-dimensional advective-dispersive systems, *Water Resources Research,* Vol. 22, No. 8, pp. 1303-1315, 1986.

[81] Wagner, B. J., and S. M. Gorelick, Optimal groundwater quality management under parameter uncertainty, *Water Resources Research,* Vol. 23, No. 7, pp. 1162-1174, 1987.

[82] Wagner, B. J., Simultaneous parameter estimation and contamination source characterization for coupled groundwater flow and contaminant transport modelling, *Journal of Hydrology,* No. 135, pp. 275-303, 1992.

[83] Weiss, R., and L. Smith, Efficient and responsible use of prior information in inverse methods, *Ground Water,* Vol. 36, No. 1, pp. 151-163, 1998.

[84] Weiss, R., and L. Smith, Parameter space methods in joint parameter estimation for groundwater flow models, *Water Resources Research,* Vol. 34, No. 4, pp. 647-661, 1998.

[85] Wilson, J, P. Kitanidis, and M. Dettinger, State and parameter estimation in ground water models, *Proceedings of AGU Chapman Conference*, edited by C-L Chiu, pp. 657-666, 1978.

[86] Yeh, W. W-G, and G. W. Tauxe, Quasilinearization and identification of aquifer parameters, *Water Resources Research,* Vol. 7, No. 2, pp. 375-381, 1971.

[87] Yeh, W. W-G, Review of parameter identification procedures in groundwater hydrology, The inverse problem, *Water Resources Research,* Vol. 22, No. 2, pp. 95-108, 1986.

[88] Yeh, W. W-G. , Optimal identification of parameters in an inhomogenous medium with quadratic programming, *Society of Petroleum Engineers*, pp. 371-375, October 1975.

[89] Yoon, Y. S., and W. W-G. Yeh, Parameter identification in an inhomogenous medium with the finite-element method, *Society of Petroleum Engineers*, pp. 217-226, August 1976.

[90] Zimmerman, D. A., G. de Marsily, C. A. Gotway, M. G. Marietta, C. L. Axness, R. L. Beauheim, R. L. Bras, J. Carrera, G. Dagan, P. B. Davies, D. P. Gallegos, A Galli, J. Gomez-Hernandez, P. Grindord, A. L. Gutjahr, P. K. Kitanidis, A. M. Lavenue, D. McLaughlin, S. P. Neuman, B. S. RamaRao, C. Ravenne, and Y. Rubin, A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow, *Water Resources Research*, Vol. 34, No. 6, pp. 1373-1413, 1998.

APPENDIX A

MATLAB SCRIPT "INVERSE"

```
%%This matlab script (INVERSE) was written to invert the trained network.

%*************************************************************************
%This step is to transform the hydraulic head target values to values between 0 & 1
%in a manner similar to that performed on the head training data
% source {target}, result t0_1
%*************************************************************************
t0_1=(target-newHmin)./(HEADdiff).*JRL_head;


%*************************************************************************
% Strike used in the early stopping loop, it is initialized to zero
% if strike = 2 then the process will stop. (due to error increase)
%*************************************************************************
strike=0;


%*************************************************************************
% Choose the initial guessed transmissivity field (from the post-training data set)
%*************************************************************************
guess=i(:,1001);


%*************************************************************************
%Assign flags to the hydraulic head values at the location of the pumping wells for
% partial set. If the hydraulic head value is known flag=1 otherwise the flag=0.
%*************************************************************************
JRL_head_a=JRL_head;
JRL_head_a(123)=0;
JRL_head_a(116)=0;


%*************************************************************************
%Compute the error between the target and simulated hydraulic heads (using the full set)
%*************************************************************************
err=JRL_head.*abs(sim(net,guess)-t0_1)./(t0_1+1-JRL_head);


%*************************************************************************
%Compute the error from head values excluding the two pumping wells
%(i.e. using the partial set)
%*************************************************************************
err_a=err.*JRL_head_a;
```

```
%*************************************************************************
%Initialize a counter
%*************************************************************************
counter=1;


%*************************************************************************
%Input the number of points in the full set in which the hydraulic head value is known
%*************************************************************************
n_data=55;


%*************************************************************************
%% Start the iterative inversion process
%*************************************************************************
while sum(err)*100/n_data>=0.1

X_tran=guess;


%*************************************************************************
% Track the inversion process progress
%*************************************************************************
[counter sum(err)*100/n_data]
plt_err(counter)=sum(err)*100/n_data;
plt_err_a(counter)=sum(err_a)*100/(sum(JRL_head_a));
iteration(counter)=counter;


%*************************************************************************
%Specify second parameter for convergence criterion
%*************************************************************************
stp1=plt_err(counter);
counter=counter+1;


%*************************************************************************
% Compute the input to the first (hidden) layer neurons {net1} [300x1]
%Multiply (.*) weights  by transmissivity guess (X_tran)     [300x225]
%*************************************************************************
for bbb=1:300
 net_int1(bbb)=sum(net.iw{1,1}(bbb,:)'.*X_tran);
end
net1=net_int1'+net.b{1};
```

```
%*************************************************************************
%Compute the output of the first layer (after the activation function) {m} [300x1]
%*************************************************************************
m=logsig(net1);


%*************************************************************************
%Compute the input to each neuron in the second layer {net2} [225x1]
%*************************************************************************
for ccc=1:225
 net_int2(ccc)=sum(net.lw{2,1}(ccc,:)'.*m);
end
net2=net_int2'+net.b{2};


%*************************************************************************
%Compute the output of the second layer (network output) {o}
%*************************************************************************
o=sim(net,X_tran);


%*************************************************************************
%Check for errors to see if we need to iterate
%*************************************************************************
%*************************************************************************
%Compute error in network
%*************************************************************************
E=2*(t0_1-o)/225.*JRL_head;


%*************************************************************************
%Define the derivative of the transfer function {function}  and
%compute the error propagated past the first (hidden) layer (just before the input
weights)
%*************************************************************************
for ddd=1:300
 interm(ddd)= sum(E.*deriv2(net2).*net.lw{2,1}(:,ddd));
end

interm1=interm.'.*deriv1(net1);


%*************************************************************************
%Compute the error value propagated to each input neuron
%*************************************************************************
for fff=1:225
```

```
 delta(fff)=sum(interm1.*net.iw{1,1}(:,fff));
end

%**************************************************************************
%Update the initial guess (transmissivity value)
%**************************************************************************
 guess=X_tran+(delta'.*JRL_tran);

%**************************************************************************
%Confine results to values between 0 & 1
%**************************************************************************
for nnn= 1:225
 if guess(nnn)<0, guess(nnn)=0; end
 if guess(nnn)>1, guess(nnn)=1; end
end

%**************************************************************************
%Compute the error using the full set
%**************************************************************************
err=JRL_head.*abs(sim(net,guess)-t0_1)./(t0_1+1-JRL_head);

%**************************************************************************
%Compute the error from head values excluding the two pumping wells (using the partial
set)
%**************************************************************************
err_a=err.*JRL_head_a;

%**************************************************************************
% Iteration numbers control (set maximum number of iterations)
%**************************************************************************
if counter == 2500, err=0;, end

%**************************************************************************
%Set stopping criterion (convergence criterion)
%**************************************************************************
stp2=sum(err)*100/n_data;
rate=(stp1-stp2)/stp1*10000;
if ((stp1-stp2)/stp1*1000)<=0.3;, err=0;, end
```

```
%*************************************************************************
%Decide if the full or partial set is to be used to complete the inversion process
%*************************************************************************
new_plt_err_a=sum(err_a)*100/(sum(JRL_head_a));
rate_a=(plt_err_a(counter-1)-new_plt_err_a)/plt_err_a(counter-1)*10000;
if strike==0,
if rate_a<rate,
 JRL_head=JRL_head_a;,
 n_data=53;
 stop_at=counter;
 strike=1
 end
end


%*************************************************************************
%Stop the inversion process
%*************************************************************************
end


%*************************************************************************
%%plot progress graph (Iteration number vs the mean percentage error (MPE)
%*************************************************************************
plot(iteration,plt_err)
```

APPENDIX B

HYPOTHETICAL CASE DATA

Hydraulic Conductivity Values of the Hypothetical Case Study

27.22	3.21	60.34	31.06	36.42	19.89	9.99	34.68	10.94	21.35	12.45	34.97	14.74	31.84	20.15
6.67	10.84	42.49	16.66	35.12	11.44	34.67	18.39	36.60	27.15	30.61	26.08	33.42	20.77	26.70
28.41	60.34	43.07	53.29	2.86	43.37	70.49	9.92	30.51	17.95	18.33	30.05	37.22	16.16	6.59
27.03	62.54	5.20	12.23	32.83	15.16	10.82	15.93	15.14	29.97	27.14	25.37	26.40	53.48	26.86
35.81	60.34	11.10	5.83	13.24	10.89	72.38	26.75	36.57	27.68	10.99	19.46	38.87	18.32	18.09
18.19	7.57	15.93	36.44	7.96	7.61	60.34	4.47	31.84	47.32	32.55	18.72	9.63	2.86	2.92
68.09	16.48	10.97	18.18	60.34	18.23	25.26	17.44	10.91	62.91	10.94	7.57	35.56	27.07	60.34
65.14	34.19	17.67	29.18	10.77	17.29	73.90	35.75	5.35	15.83	35.24	44.08	10.95	10.91	13.18
2.88	62.58	13.57	27.13	5.35	44.71	10.16	17.64	23.30	68.52	35.31	19.34	32.74	38.83	45.18
31.55	32.82	34.25	35.85	9.71	16.73	12.65	60.34	55.09	35.54	7.91	19.42	3.30	26.70	26.44
5.29	32.87	62.16	73.60	5.10	26.89	31.34	10.61	62.89	8.66	25.33	32.50	20.35	4.66	32.27
2.91	34.31	27.42	7.70	7.80	27.62	8.80	11.77	16.68	4.42	29.26	2.88	19.67	18.21	10.84
11.10	33.01	11.79	36.79	9.81	22.58	34.40	10.00	4.87	11.11	50.21	26.44	32.14	13.56	26.56
37.37	6.11	18.47	14.74	60.34	10.96	62.37	20.91	7.93	10.91	52.13	40.84	44.13	62.98	50.33
57.64	31.78	10.30	5.01	14.92	37.50	26.80	31.66	7.58	18.24	18.21	7.92	56.36	24.84	13.93

Units; m/day

True Hydraulic Head Values

115.82	116.40	117.31	118.01	119.08	120.44	122.46	124.01	125.33	126.55	127.54	128.30	128.79	129.12	129.25
115.49	115.93	116.83	117.67	119.10	121.00	122.61	124.09	125.58	126.63	127.62	128.42	128.96	129.25	129.37
114.81	115.14	116.18	116.90	119.15	121.70	122.49	124.17	125.81	126.89	127.94	128.78	129.29	129.69	129.89
114.08	114.26	115.04	117.01	118.88	120.98	122.90	124.63	126.19	127.37	128.27	129.21	129.92	130.29	130.56
113.20	113.36	112.56	114.84	118.27	121.20	123.39	124.80	126.57	127.68	128.80	130.01	130.70	130.97	131.31
111.79	110.86	108.74	112.09	116.35	120.78	123.32	125.27	127.39	128.41	129.34	130.86	133.03	134.71	135.77
110.94	108.76	97.49	111.72	116.47	120.43	123.70	125.68	127.71	128.94	129.83	133.16	136.73	139.08	140.30
111.44	111.84	110.55	114.07	117.76	122.57	124.80	126.38	129.03	130.32	129.52	133.91	138.41	142.23	143.58
114.22	114.46	115.09	116.62	120.21	124.14	126.35	129.25	131.73	132.74	133.46	137.70	142.39	146.59	147.84
116.85	116.86	117.95	118.79	121.94	125.77	128.72	131.27	132.89	134.11	138.60	142.56	147.35	Δ 152.00	Δ 152.00
118.33	118.90	119.36	119.79	123.83	127.93	129.57	132.15	134.21	138.29	145.23	147.13	Δ 152.50	152.38	152.26
120.46	120.45	120.73	122.61	126.41	129.09	131.17	133.87	136.21	143.51	149.46	Δ 153.00	152.82	152.78	152.66
122.16	121.75	123.02	125.15	127.85	130.55	132.77	135.15	140.68	149.17	Δ 153.30	153.23	153.08	153.00	152.96
123.00	123.18	124.52	126.50	128.87	131.39	133.80	136.22	142.71	150.81	Δ 153.50	153.32	153.18	153.11	153.06
123.27	123.57	124.60	127.12	129.98	132.22	134.20	136.83	144.31	Δ 154.00	153.64	153.37	153.20	153.14	153.10

Units ; meters

Δ = constant head cell

APPENDIX C

HYDRAULIC CONDUCTIVITY FIELD DATA

Point num							
1	3	250	50	2.33247	10.3	1.0338	-0.77172
2	25	950	150	2.3894	10.91	1.20258	-0.51406
3	35	450	250	2.28338	9.81	0.91296	-1.2182
4	40	950	250	2.40744	11.11	0.5682	-0.20393
5	41	1050	250	3.9162	50.21	0.5682	1.0606
6	44	1350	250	2.60731	13.56	1.09757	-0.13091
7	50	450	350	2.05424	7.8	0.71157	-1.38149
8	54	850	350	2.81434	16.68	0.5682	0.03082
9	60	1450	350	2.38308	10.84	1.30231	-0.64206
10	62	150	450	3.49243	32.87	0.71829	0.55161
11	84	850	550	4.00898	55.09	0.54135	1.36169
12	85	950	550	3.57079	35.54	0.47422	0.83206
13	92	150	650	4.13638	62.58	1.12106	1.57191
14	106	50	750	4.17661	65.14	1.12106	2.06782
15	111	550	750	2.8504	17.29	0.84583	0.09206
16	118	1250	750	2.39303	10.95	0.89618	-0.3463
17	131	1050	850	2.39201	10.94	0.71829	-0.42142
18	161	1050	1050	2.39696	10.99	0.85255	-0.26707
19	196	50	1350	1.89788	6.67	2.06759	-1.80359
20	200	450	1350	3.55881	35.12	1.10093	0.73936
21	209	1350	1350	3.03342	20.77	1.6581	0.2015
22	214	350	1450	3.43606	31.06	1.10093	0.46219
23	217	650	1450	2.30158	9.99	2.175	-0.97486
24	105	1450	650	3.81064	45.18	1.16806	0.93703
25	158	750	1050	3.28647	26.75	0.96667	0.36482
26	71	1050	450	3.23214	25.33	0.50107	0.2978
27	32	150	250	3.49673	33.01	0.85255	0.63252
28	179	1350	1150	3.97933	53.48	1.12106	1.19876
29	5	450	50	2.70259	14.92	1.0338	-0.03843