Theses and Dissertations

Theses, Dissertations, and Senior Projects

1998

Determining capture zones for three-dimensional, heterogeneous, anisotropic aquifers by Monte Carlo analysis

Jon Paul Jones
University of North Dakota

Follow this and additional works at: https://commons.und.edu/theses

Part of the Geology Commons

DETERMINING CAPTURE ZONES FOR THREE-DIMENSIONAL,
HETEROGENEOUS, ANISOTROPIC AQUIFERS BY MONTE CARLO ANALYSIS

by

Jon Paul Jones
Bachelor of Science, University of Minnesota, 1996

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
July
1998

This thesis, submitted by Jon Paul Jones in partial fulfillment of the requirements for the Degree Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

_____
Philip J. Gerla
(Chairperson)

_____
Scott F. Korom

_____
Ahmad Ghassemi

This thesis meets the standards for appearance, conforms to the style and format requirements of the Graduate School of the University of North Dakota, and is hereby approved.

_____
Dean of the Graduate School

_____
Date

ii

PERMISSION

Title   Determining Capture Zones for Three Dimensional,
         Heterogeneous, Anisotropic Aquifers by Stochastic Analysis

Department   Geology and Geological Engineering

Degree   Master of Science

    In presenting this thesis in partial fulfillment of the requirements for a graduate
degree from the University of North Dakota, I agree that the library of this university
shall make it freely available for inspection. I further agree that permission for extensive
copying for scholarly purposes may be granted by the professor who supervised my
thesis work, or in his absence, by the chairperson of the department or the dean of the
Graduate School. It is understood that any copying or publication or other use of this
thesis or part thereof for financial gain shall not be allowed without my written
permission. It is also understood that due recognition shall be given to me and to the
University of North Dakota in any scholarly use which may be made of any material in
my thesis.

             Signature _____

             Date   _____

iii

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

ABSTRACT

Most methods of delineating capture zones for pump-and-treat remedial design and wellhead protection assume a two-dimensional, homogeneous aquifer. Aquifers, however, are three-dimensional and heterogeneous, thereby introducing uncertainty in capture zone analysis. This study used a Monte Carlo analysis of three sets of statistical parameters defining aquifer heterogeneity. Each set had a different variance for the mean natural log hydraulic conductivity, which varied among the sets by a factor of four. The ensemble means of the capture zones for each set were estimated from 10 randomly generated fields with 12 layers each which were superimposed on each other for a total of 120 realizations. Realizations of the hydraulic conductivity fields were generated using the fast Fourier transforms method (Gutjahr et al., 1996) and incorporated into a confined, 128 m long by 64 m wide grid that included sufficient vertical layers to maintain the vertical correlation length. Constant head and no-flow boundaries were established on the short and long edges, respectively. The U.S. Geological Survey MODFLOW code (MacDonald and Harbaugh, 1988) coupled with MODPATH (Pollack, 1989) were used to simulate the capture zones surrounding a pumping well within this simple flow system. Mapping the capture zones for many aquifer realizations with similar stochastic properties provided the data required to construct 1%, 80%, and 99% quantile intervals. These figures begin to show how heterogeneity reduces the size of capture zones estimated for statistically homogeneous aquifers.

ix

To William and Judy Jones

INTRODUCTION

Capture zones are primarily used in wellhead protection and remediation designs. When ground water becomes contaminated due to a chemical spill or a leaking underground storage tank, remediation may be necessary. One common cleanup strategy is pump-and-treat. In a pump-and-treat design one or more pumping wells are placed in the vicinity of the contaminant plume and polluted water is pumped to the surface where it can be treated. Estimates made 10 years ago for the Superfund Program (Department of Energy, 1988) stated that the average cost of assessment, characterization, and remedial action design per site is close to $1.7 million (1988 dollars). The cost of the remedial action, usually pump-and-treat, averages $12.4 million (1988 dollars) (Lee and Kitanidis, 1991).

The most significant factors in designing a pump-and-treat system are the placement of the pumping well and its pumping rate. Locating the well is typically accomplished by determining ground water flow direction and finding a position whereby the well can capture the contaminant plume. The well should be pumped at a rate that allows capture of all the polluted ground water. If the pumping rate is too low, contaminated water flows by the well, and if the rate is too high, too much clean water will be withdrawn. Either situation adds substantially to the cost of treatment. As the pumping rate increases so does the size of the capture zone. Maximum cost effectiveness is achieved when the pumping rate of the well generates the smallest

1

possible capture zone that can successfully remediate the aquifer. To accomplish this, it is necessary to characterize the flow paths of ground water.

Methods for determining the shape of the capture zone generally assume homogeneity of hydraulic conductivity (K) and porosity, although this always simplifies natural conditions. Natural variation in stratigraphy, lithology, and texture means that aquifers are best characterized as heterogeneous. Most methods of determining capture zones also assume two-dimensional and isotropic conditions. This could be reasonable in some instances, but in most cases, especially with strongly heterogeneous aquifers, it is not because K is typically anisotropic and a vertical component of flow is usually present.

The method described in this thesis considers heterogeneity and anisotropy in estimating capture zones by using a three-dimensional random field generator in conjunction with a Monte Carlo analysis. Each hydraulic conductivity field created by the random field generator is used in a numerical simulation to determine the flow paths of the ground water. By statistically analyzing the resulting ground water flow paths of ten statistically equal but unique models, quantile capture zones can be determined. By increasing or decreasing the pumping rate, the size of quantile capture zones can be changed to meet the needs of a particular design.

Three sets of hydraulic fields were randomly generated. In the first set variance of the K field was 0.29. In the second set the variance was doubled, and in the third, quadrupled. The three sets of K fields were placed in models and the resulting capture zones compared to each other. This application was performed to see the effects, if any, increasing heterogeneity had on capture zone size.

OVERVIEW OF PREVIOUS WORK

Deterministic Methods

A deterministic model is one where a partial differential equation describing ground water flow is solved, either numerically or analytically, for a given set of aquifer parameters and boundary conditions. The result, hydraulic head, has a specific value at each location in the aquifer. Therefore, deterministic methods provide a fast and easy solution for finding capture zones. The method assumes the aquifer parameters are known, which may not be true due to the uncertainty associated with measuring some parameters, such as hydraulic conductivity.

Javandel and Tsang (1986) developed equations to determine the capture zones for two-dimensional, confined aquifers that are homogeneous and isotropic. The resulting capture zones could be used to generate type curves for one or more wells. The equation for a one-well case is

$$y = \pm \frac{Q}{2bu} - \left(\frac{Q}{2\pi bu} \right) \tan^{-1} \frac{y}{x}$$

(1)

where b = aquifer thickness [L], Q= well discharge rate [L^3T^{-1}], x and y are the coordinates of the system, and u = uniform Darcy regional flow velocity [LT^{-1}]. Javandel and Tsang (1986) also developed equations for the 2-, 3-, and n-well cases. The resulting capture zones are assumed to be steady state and the x-axis parallel to the direction of flow. Javandel and Tsang's (1986) work was extended to unconfined aquifers and

3

combined confined and unconfined aquifers by Grubb (1993), who generalized the equations by using discharge potential instead of hydraulic head and specific discharge in the development of the equations. Bakker and Strack (1996) also developed a four step approach for determining two-dimensional capture zones, but the method was not directly applicable to three-dimensional flow fields and the capture zones must be delineated one well at a time.

Faybishenko et al. (1995) created a semianalytical method that described the capture zone in a homogeneous and confined aquifer with a partially penetrating well. The method was also applicable to isotropic, unconfined aquifers but was only of limited value in aquifers that exhibited significant anisotropy.

Bair and Lahm (1996) and Schafer (1996) extended the work of Faybishenko et al. (1995) by developing partially penetrating well capture zone analysis methods that could account for anisotropy. Bair and Lahm (1996) asserted anisotropic aquifers with partially penetrating wells cause capture zones to be shallower and wider than in isotropic conditions. This assertion was similar to the conclusions of Zlotnik (1997) who applied dimensional analysis to find capture zone geometry in aquifers with partially penetrating wells.

Many modern studies (e.g., Hoeksema and Kitanidis, 1985) have shown that there is no possibility of finding an absolutely homogeneous aquifer in nature (Domenico and Schwartz, 1998). Many deterministic methods of capture zone analysis treat the K field as homogeneous, which can lead to questionable results. Deterministic methods that do incorporate heterogeneity involve measuring hydraulic conductivity at a few locations that are used as the basis for estimating values at unmeasured locations (Varljen and

Shafer, 1991). This procedure again produces a solution with significant uncertainty. One way to reduce the uncertainty is to use a stochastic method.

Stochastic Methods

A stochastic approach recognizes the uncertainties involved in parameter estimation. The approach provides a range of possible outcomes rather than a specific value at each point in the aquifer. For example, K can vary by many orders of magnitude in an aquifer over a short distance. Because of this variability, hydraulic conductivity can strongly influence ground water flow paths.

To quantify the uncertainty introduced into a model by spatially varying K, many stochastic approaches use a Monte Carlo simulation. In a Monte Carlo simulation, a deterministic problem is solved using different but statistically identical sets of randomly generated parameter values (i.e., K fields). Results of the simulations are tabulated and used to make statistical interpretations concerning capture zone characteristics (Bair et al., 1991).

The literature has numerous examples of capture zone studies based on Monte Carlo analysis (e.g. Bair et al., 1991; Franzetti and Guadagnini, 1996; Cole and Silliman, 1997). Although each of the studies took a slightly different approach in conducting a Monte Carlo analysis, none of the studies presented a method for producing capture zones in a three-dimensional, anisotropic, heterogeneous aquifer.

Stochastic Representation of Heterogeneous Hydraulic Conductivity Fields

If the vertical and horizontal correlation lengths and variance of the K field in a given aquifer have been estimated, they can be used to generate a random field. The correlation lengths can be thought of as the length of the heterogeneity in the vertical or

horizontal direction, respectively. The K values are usually assumed to be log normally distributed. By using the geostatistical parameters of the aquifer in the random field generator, a K field is produced which, although different, will be statistically identical to the actual K field of the aquifer. Table 1 (modified from Gelhar, 1993 and Anderson, 1997) presents data on aquifers whose three-dimensional geostatistical parameters have been estimated.

The random field generator used in this study produces hypothetical K fields with specified geostatistical values. An actual K field with the same geostatistical values can be thought of as one possible realization from the random field generator. The set of all possible K fields containing identical geostatistical parameters is called the ensemble field. This study assumes that ten 12-layer random fields (in effect, 120 realizations) can adequately estimate the ensemble field's distribution of hydraulic conductivity values. However, the geostatistical parameters originally used to generate the ensemble field were determined from only one realization (the actual K field). This is possible if ergodicity is assumed.

Ergodicity states that averaging over the ensemble is equivalent to averaging over a realization (Bras and Rodriguez-Iturbe, 1985). However, the geostatistical parameters in question must assume stationarity. A stationary ensemble field has a covariance structure that does not vary spatially. The ensemble field is considered homogeneously heterogeneous. Assuming ergodicity facilitates the use of a random field generator. Realizations from a random field generator can be incorporated into a Monte Carlo analysis. Information about the random field generator used in this study is in Appendix A.

Table 1. Variance and correlation scale (natural logarithm) of three-dimensional hydraulic conductivity for several well characterized aquifers (modified from Gelhar, 1993 and Anderson, 1997.)

Source	Medium	Variance	Correlation Scale (m) Horizontal	Vertical
Byers and Stephens (1983)	fluvial sand	0.81	>3	0.1
Goggins et al. (1988)	eolian sandstone outcrop	0.16	0.4	0.8
[1]Hess (1989)	glacial outwash sand	0.25	0.5	5
Hufschmied (1986)	sand and gravel aquifer	3.61	20	0.5
[2]Rehfeldt et al. (1989)	fluvial sand and gravel aquifer	4.41	13	1.5
Smith (1978); Smith (1981)	glacial outwash sand and gravel outcrop	0.64	5	0.4
[3]Sudicky (1986)	glacial lacustrian sand aquifer	0.25	2.8	0.12

[1]Cape Cod site
[2]Columbus site
[3]Borden site

METHODS

MODFLOW (McDonald and Harbaugh, 1988), a finite difference code, was used to determine hydraulic head fields for three sets of 10 randomly generated K fields, each with 12 individual layers. Before the 16-layer K fields were imported into MODFLOW, the bottom 4 layers were truncated. The resulting head fields were imported into MODPATH (Pollock, 1989) models to generate the capture zone at each layer. The resulting capture zones for each set of realizations were superimposed onto a plane. For each data set, this entailed superimposing 120 individual layers (ten 12-layer realizations). These superimposed data represented an estimate of the ensemble capture zone distribution. The distribution was statistically analyzed to determine the y quantiles of each x coordinate position.

Models and Model Parameters

Conceptually, the models simulated a simplified confined aquifer under steady state conditions and uniform flow. All model dimensions were 128 columns by 64 rows (Figure 1) by 12 layers for a total of 98,304 cells. Each cell was 1 m by 1 m wide and 0.1 m thick. The upstream and downstream sides of the model, perpendicular to the direction of flow, had constant head boundaries of 10 m and the 11 m, respectively. The sides parallel to the direction of flow were no-flow boundaries (Figure 1). These parameters gave the models an overall horizontal hydraulic gradient of 0.0078 m/m. The porosity was 0.35 for all cells and all units were in m/day.

8
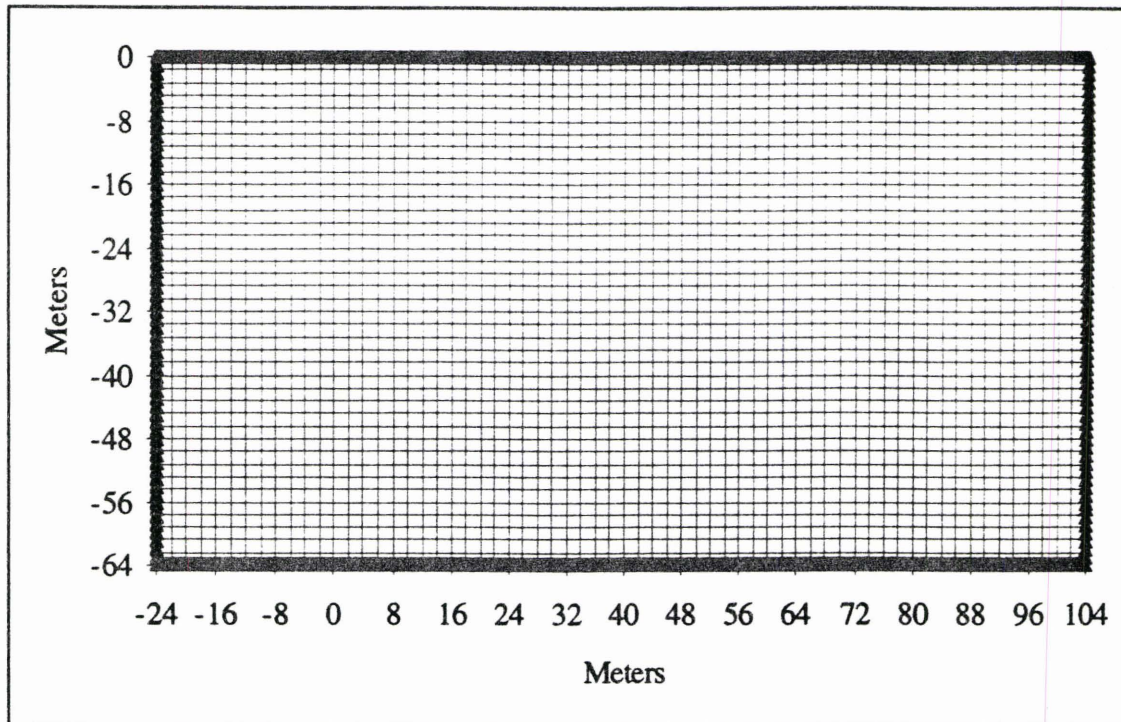
Figure 1. The mesh discretization and boundary conditions used in this study. The top and bottom edges were no-flow boundaries and the left and right sides had a constant hydraulic head of 10 m and 11 m, respectively.

The model coordinates were –24 m to 104 m along the x-axis and 0 m to –64 m along the y-axis (Figure 1). A pumping well was placed at coordinates (0,0) in each model. The well was fully penetrating and it was necessary to define the well at each layer as a boundary condition. The well discharge was 2 m^3/day and was distributed throughout the borehole proportional to the hydraulic conductivity at each layer. The position of the well allowed only half of the actual capture zone to be determined. The full capture zone was resolved by assuming it was symmetrical about the x-axis.

Assuming symmetry about the x-axis in this study was also reasonable if one considers the stochastic analysis performed in this study was an estimation of the ensemble K field. The mean K value at each point in the ensemble field is equivalent to

the K value of the homogeneous field. This being the case, an estimated ensemble K field will produce capture zones symmetrical about the x-axis.

Numerical Model Descriptions

The capture zones in this study are from simulations run using MODFLOW and MODPATH. MODFLOW can be used to determine the steady state head distribution for a given K field. MODPATH uses the head distribution to track the paths of particles at each layer as they are introduced into the system. By releasing 64 evenly spaced particles along the upstream end of the model perpendicular to the direction of flow, MODPATH was used to determine the particle paths. The outermost particle tracks drawn into the pumping well delineated the margin of the capture zone for that layer. This procedure was repeated for each layer in all the models.

The preconditioned conjugate gradient (PCG2) method was used (Hill, 1990) for the matrix solution in MODFLOW simulations. The PCG2 solver was chosen due to potential problems in using either the strongly implicit procedure (SIP) or the slice successive overrelaxation (SSOR) methods. The SIP and SSOR matrix solution methods can affect MODFLOW simulation results under some combinations of matrix solution parameters (Osiensky and Williams, 1997). Trial simulations using PCG2 in conjunction with a head change convergence criterion of 1×10^{-6} consistently produced a volumetric water budget discrepancy of less than 1 percent.

Model Verification

Generation of meaningful capture zones requires reasonable parameters and an independent verification of the model. With all of the basic parameters entered into the model, it was necessary to verify that the numerical model closely approximated an exact
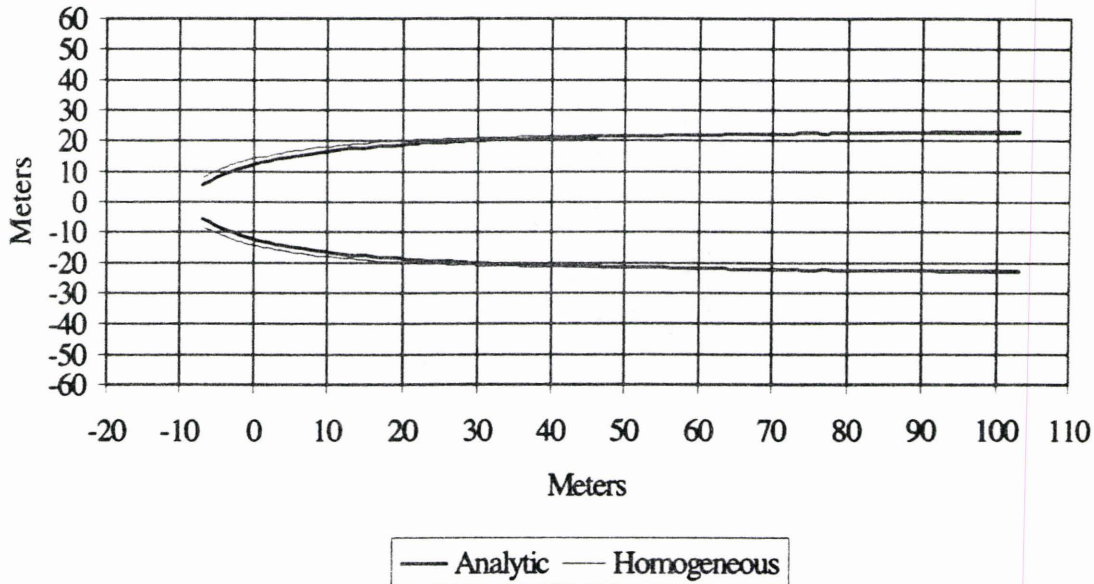
Figure 2. Analytical versus homogeneous solutions for the model discretization.

Testing the Hydraulic Conductivity Field Importation Process

The preprocessor program (GW-VISTAS, Environmental Simulations, Inc., 1996) used to generate the input files for the MODFLOW models has a maximum of 9,999 hydraulic conductivity values, making it necessary to create discrete hydraulic conductivity zones. The horizontal hydraulic conductivity zones ranged from 0.1 m/day to 999.9 m/day, increasing in 0.1 m/day increments. The vertical hydraulic conductivity zones ranged from 0.01 m/day to 99.9 m/day in 0.01 m/day increments. This procedure introduced an anisotropy of 10:1 horizontal to vertical hydraulic conductivity. After truncation, the generated K fields are imported into the model and placed in the closest matching zone.

Tests of the K fields were run to ensure that the process did not significantly alter the statistical properties of the field. Descriptive statistics were calculated before and after the values were imported into a one-layer model. The mesh was discretized using

the parameters described above, with the exception of the layer thickness. The layer thickness was changed from 0.1 m to 1.0 m. In MODFLOW all hydraulic conductivity values are multiplied by the thickness of the aquifer because MODFLOW makes calculations using transmissivity and not hydraulic conductivity. With a layer thickness of 1.0 m the values would not be altered before zonation. A table containing the summary of the statistics is presented in Table 2.

The only significant statistical change that took place after importing the K fields into the models was the mode (Table 2). In statistics, the mode is defined as the value occurring most frequently in a given distribution. It was reasonable to expect zonation of the values (and truncating the values to one decimal place due to the zonation setup) would affect the mode. Because of the statistical similarities of the K field before and after the zonation process, it was concluded GW-VISTAS (GW-VISTAS, Environmental Simulations, Inc., 1996) functioned satisfactorily for this research.

Table 2. Statistics of hydraulic conductivities (m/day) before and after being imported into MODFLOW.

Statistic		
	GRF*	MODFLOW
Mean	10.320	10.320
Standard Error	0.064	0.064
Median	9.040	9.000
Mode	8.565	5.600
Standard Deviation	5.776	5.776
Sample Variance	33.364	33.361
Minimum	1.545	1.500
Maximum	69.120	69.100

* denotes generated random field

Random Field Generator Input Parameters

The random field generator requires the user to enter the covariance type, correlation lengths, spatial step size in all three dimensions, and a mesh discretization. The random field generator code used in this study was incorporated into a program called CRUNCHER, which generated input files for the random field generator and converted the generated fields into a format usable in MODFLOW. The source code for CRUNCHER is in Appendix B.

The vertical correlation length (λ_z) in all three realization sets was 0.4 m. Appendix C contains a sensitivity analysis examining the role vertical correlation length has with respect to grid spacing. A horizontal correlation length of 2.8 m was used for the all three sets in both the x and y directions. An exponential covariance type was chosen for the data sets because the geostatistical parameters of the K fields used in this study are similar to those of the Borden aquifer whose K field exhibited an exponential covariance function (Sudicky, 1986; Woodbury and Sudicky, 1991). Appendix D gives an evaluation of how well the random field generator can produce a field that exhibits exponential covariance.

A variance of 0.25 was used in the first set of generated fields. However, the second and third sets of random fields involve doubling the variance and then doubling the variance again (designated 1σ, 2σ, and 4σ, respectively). The changes in variance in the 2σ and 4σ random fields were incorporated to see how sensitive the resulting capture zones were to changes in variance in K, or in effect, the heterogeneity of the K field.

In addition to the parameters described above, a seed number was also needed to generate each random field. The purpose of a seed number is to give the random field generator a place to begin. A unique seed number yields a unique realization (Dodak, 1996). The seed numbers used in this study were created by using a random function in a spreadsheet.

The random field generator produced natural log K fields whose mean was approximately zero and approximately the desired variance. The mean was then shifted to –9.2 (K in m/day), which was the natural log of the mean hydraulic conductivity value used in this study. The desired variances were 0.29, 0.58, and 1.16 for the 1σ, 2σ, and 4σ random field sets, respectively. The tabulated results of the 1σ (Table 3), 2σ (Table 4), and 4σ (Table 5) random field sets show the random field generator produced an average variance of 85.4, 84.0, and 84.8 percent of the desired variance, respectively. It is not known why the variances produced were consistently below the desired values.

Table 3. Seed numbers, mean, and variance for the 1σ set of realizations.

Realization	Seed	Mean	Variance
1	547976	2.82 E-02	2.47 E-01
2	821572	9.88 E-03	2.46 E-01
3	870878	4.12 E-02	2.54 E-01
4	106356	3.99 E-03	2.49 E-01
5	731047	-4.46 E-02	2.46 E-01
6	821281	-1.52 E-02	2.42 E-01
7	15376	2.82 E-02	2.43 E-01
8	216423	-1.44 E-02	2.53 E-01
9	699034	-8.07 E-03	2.48 E-01
10	427805	6.97 E-03	2.50 E-01
Average	na*	0.003617	0.247651

* denotes not applicable

Table 4. Seed numbers, mean, and variance for the 2σ set of realizations.

Realization	Seed	Mean	Variance
1	326813	-3.64E-02	4.88E-01
2	271290	6.37E-05	5.11E-01
3	586611	-8.94E-02	5.05E-01
4	906076	-3.13E-02	4.73E-01
5	76795	2.64E-02	4.69E-01
6	530390	4.21E-02	4.93E-01
7	925073	-1.99E-02	5.00E-01
8	790725	3.55E-02	5.01E-01
9	956516	-1.22E-02	4.71E-01
10	941775	9.62E-03	4.62E-01
Average	na*	-0.007547	0.487421

* denotes not applicable

Table 5. Seed numbers, mean, and variance for the 4σ set of realizations.

Realization	Seed	Mean	Variance
1	50007	3.64E-03	9.71E-01
2	554768	1.77E-02	9.30E-01
3	320914	-3.11E-02	9.26E-01
4	781679	-4.92E-02	1.01E+00
5	413432	-5.81E-02	1.01E+00
6	167775	-2.95E-02	1.01E+00
7	352773	1.57E-02	9.93E-01
8	921774	-1.96E-02	1.00E+00
9	150466	1.46E-02	1.02E+00
10	380757	7.79E-02	9.67E-01
Average	na*	-0.005796	0.983708

* denotes not applicable

Determining the Appropriate Number of Realizations

A Monte Carlo analysis usually entails using multiple realizations to obtain a

representative distribution of the parameter under consideration. In this study, the

parameter under consideration was the ensemble K field, which was used to produce the

capture zone distribution. The number of realizations used in a Monte Carlo analysis is important; too many realizations will add time and effort to a project while producing little new information, while too few realizations could yield misleading results. The capture zones in this study consisted of the connected given quantile values calculated at each transect.

A sensitivity analysis was performed to determine how the stochastic mean along a transect varied as a function of the number of realizations used in its calculation. The sensitivity analysis used the results of 50 MODFLOW simulations that had been discarded from the study because of a parameter change. Each of the 50 MODFLOW simulations had 12 layers (600 realizations). Three transects were examined where x equaled 4, 26, and 76. The stochastic mean value was calculated at each transect for 10, 20, 30, 40, and 50 random fields; therefore, for 12 layers the stochastic means have at least 120, 240, 360, 480, and 600 realizations included in their calculations, respectively. The resulting means were plotted against the number of realizations included in the calculation of the stochastic mean for the 1σ (Figure 3), 2σ (Figure 4), and 4σ (Figure 5) data sets.

The resulting plots revealed that there was no significant change ($< 0.1\%$) in the stochastic mean at a transect when it is calculated with 120-600 realizations. It is possible that even fewer realizations could produce a similar stochastic mean. However, 10 random fields (120 realizations) were used in each of the data sets to ensure a representative distribution.
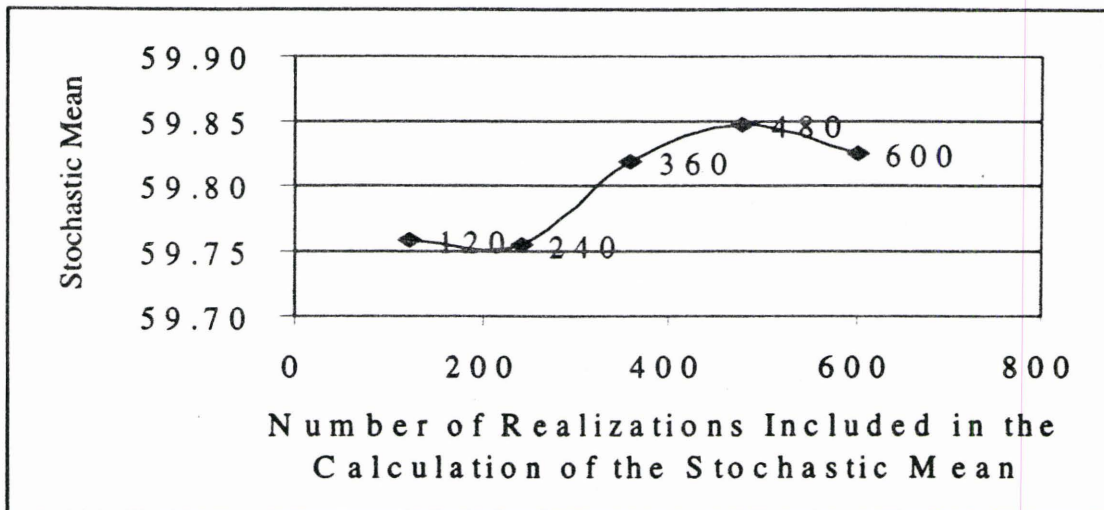
Figure 3. Results of a sensitivity analysis to determine how the stochastic mean changes as a function of the number of realizations used in its calculation at x = 4.
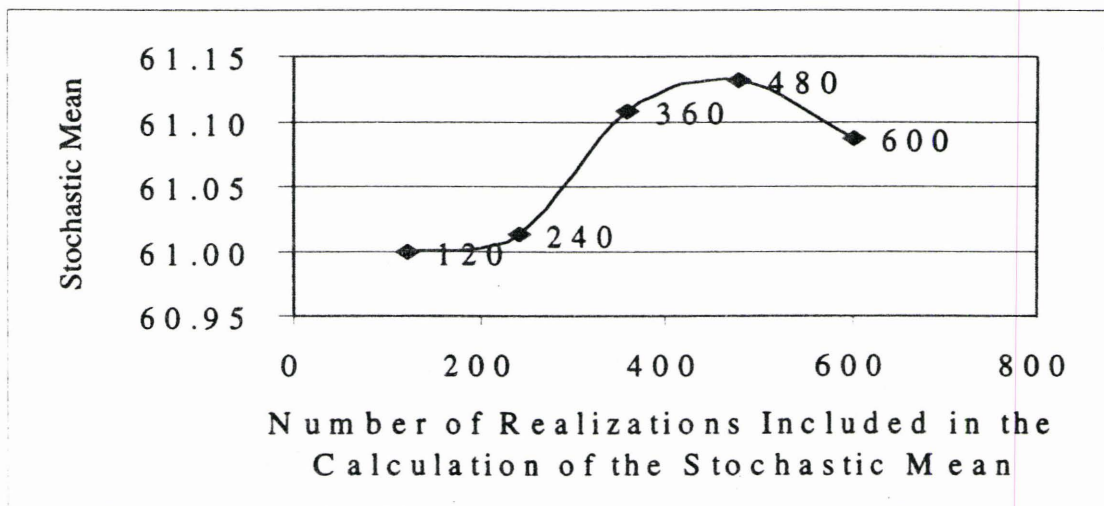
Figure 4. Results of a sensitivity analysis to determine how the stochastic mean changes as a function of the number of realizations used in its calculation at x = 26.
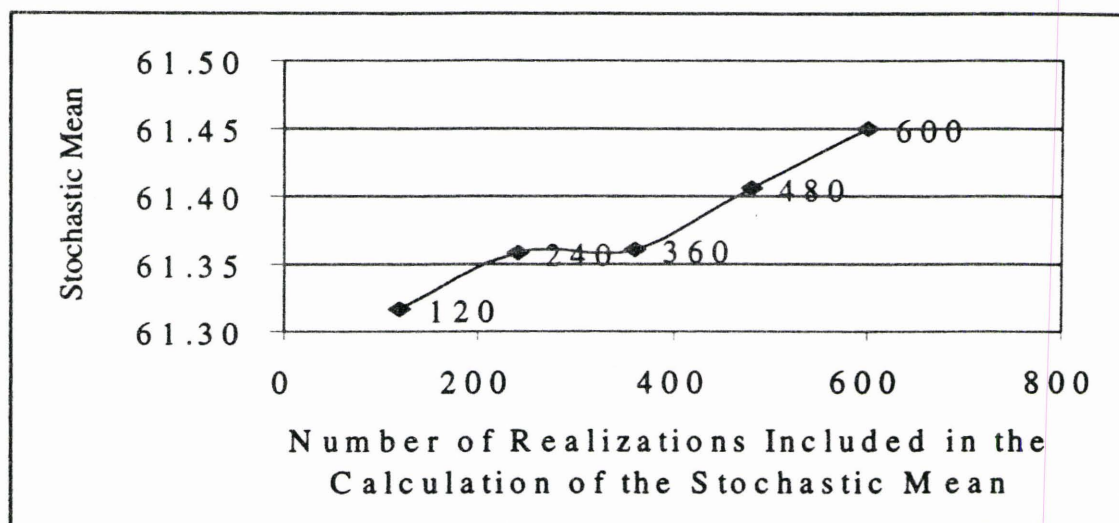
Figure 5. Results of a sensitivity analysis to determine how the stochastic mean changes as a function of the number of realizations used in its calculation at $x = 76$.

An alternative method to determining the number of realizations needed was presented by Chan (1993). In Chan's (1993) study the reliability level of a Monte Carlo simulation was calculated by comparing it to a linear program that determined the optimal pumping strategy. This is similar to this study's goal of developing optimal capture zones. Chan (1993) stated that in previous work using multiple realizations (Gorelick, 1987; Wagner and Gorelick, 1989) the number of stacked realizations was chosen arbitrarily. The reliability level of the realizations had to be tested a posteriori using a Monte Carlo simulation but no prespecification of reliability could be made. Chan's (1993) proposed method used Bayesian analysis and order statistics to develop two predicted reliability estimators: $(N+1)/(N+2)$ and $N/(N+1)$, where N represented the number of realizations in both estimator formulas. Chan (1993) used conditional simulation in the development of the estimators and so they may not be applicable to this work. However, in future site specific applications of the method presented in this study, Chan's (1993) estimators may be an excellent way to determine the number of

realizations required. Other applications that use Monte Carlo simulations could also benefit from Chan's (1993) estimators.

<div align="center">Calculating the Quantiles</div>

After the capture zone boundaries for each realization were extracted from MODPATH and superimposed onto a plane, the quantiles for the capture zones were calculated. The method used to calculate the quantiles depended on the statistical distribution of the data. The data were originally assumed normally distributed, but a Kolmogorov-Smirnov (K-S) test was performed at three transects on the plane to confirm this assumption. The transects tested were located where x equaled 4, 26, and 76. At all three points, the K-S test failed to verify the data were normally distributed. It was decided to consider the capture zone data distribution as unknown and to use a non-parametric procedure in calculating the quantile capture zones (Gilbert, 1987). The procedure involved ranking the data from smallest to largest before beginning calculations. The quantiles at each transect were determined with the formula

$$P (n + 1) = \text{ranked position of desired quantile}$$

<div align="right">(2)</div>

where P equals desired percentile, and n equals the total number of ranked data. This procedure was repeated at each integer x-coordinate along the capture zone distribution.

The particles defining the capture zone in this study moved through the models because of an imposed head gradient. The particles flowed past the well until the influence of the well drew them into the borehole and out of the system. Different particles moved back toward the borehole at different positions (due to the different K

RESULTS AND DISCUSSION

The stochastic mean and the 1%, 80%, and 99% quantiles for each data set were calculated for the flow field using the superimposed capture zones. The aggregate of each calculated quantile was assumed to approximate the corresponding confidence limit of the ensemble capture zone. For example, the aggregate 99% quantile of the ensemble capture zone distribution delineated a capture zone that is smaller than 99% of all possible capture zones. In other words, if the capture zones for 100 realizations were determined, then on the average only one of them will be smaller than the aggregate 99% quantile capture zone.

Plots were made for the stochastic mean, and the 1%, 80%, and 99% quantiles for the 1σ (Figure 6), 2σ (Figure 7), and 4σ (Figure 8) data sets. The plots show that as the quantile percentages get larger, the capture zones they represent become smaller. For example, the 99% quantiles are consistently smaller than the 80% quantiles in the plots.

The application portion of this study examined how heterogeneity influences the size of capture zones. Plots were generated that compared the stochastic means (Figure 9), 80% (Figure 10), and 99% (Figure 11) quantiles of the 1σ, 2σ, and 4σ data sets. The plots show the various quantiles of the 1σ and 2σ data sets track each closely while the 4σ quantiles are consistently smaller. The behavior of the 1σ and 2σ data sets seem to contradict the study's underlying hypothesis of increasing heterogeneity decreasing the size of the capture zone. A possible explanation for this behavior could be that the

22

ensemble fields of the 1σ and 2σ data sets are not different enough to affect the size of

the quantile capture zones. However, the quantile capture zones of the 4σ data set, the

most heterogeneous data set in the study, are consistently smaller than the other two data

sets. This implies that heterogeneity does play a role in capture zone size.

Further evidence that heterogeneity plays a role in capture zone size can be seen

in a comparison of 99% quantiles for the three data sets (Figure 11) and the

corresponding homogeneous and analytical solutions (Figure 2). In all three cases, the

99% quantiles are smaller than the analytical and homogeneous solutions.
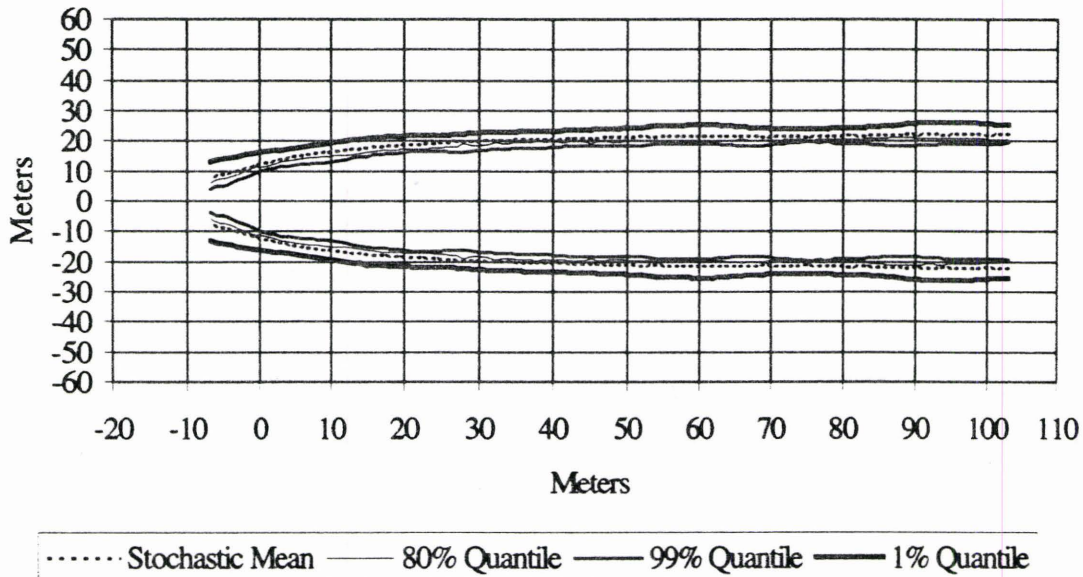
Figure 6. The stochastic means and quantiles for the 1σ data set.
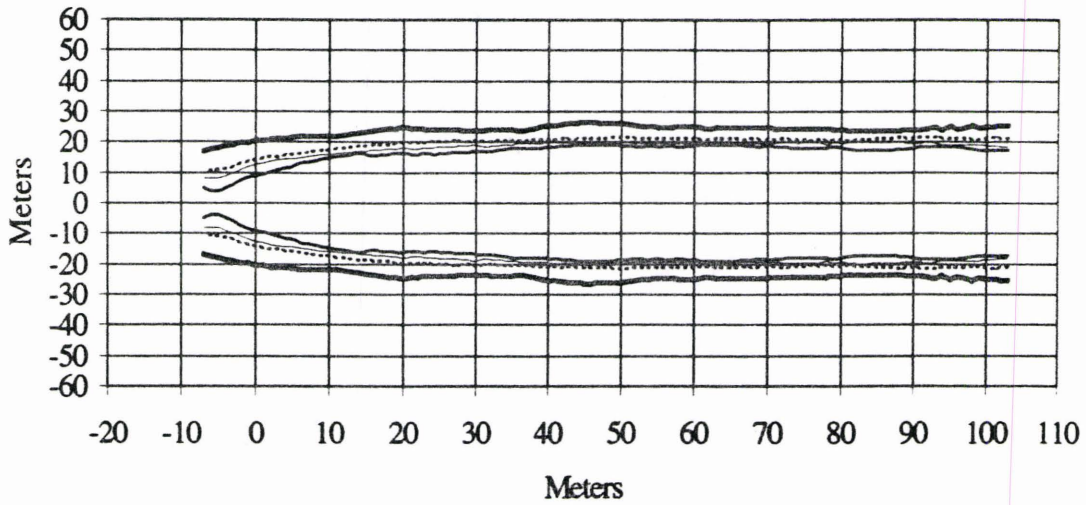
Figure 7. The stochastic means and quantiles for the 2σ data set.
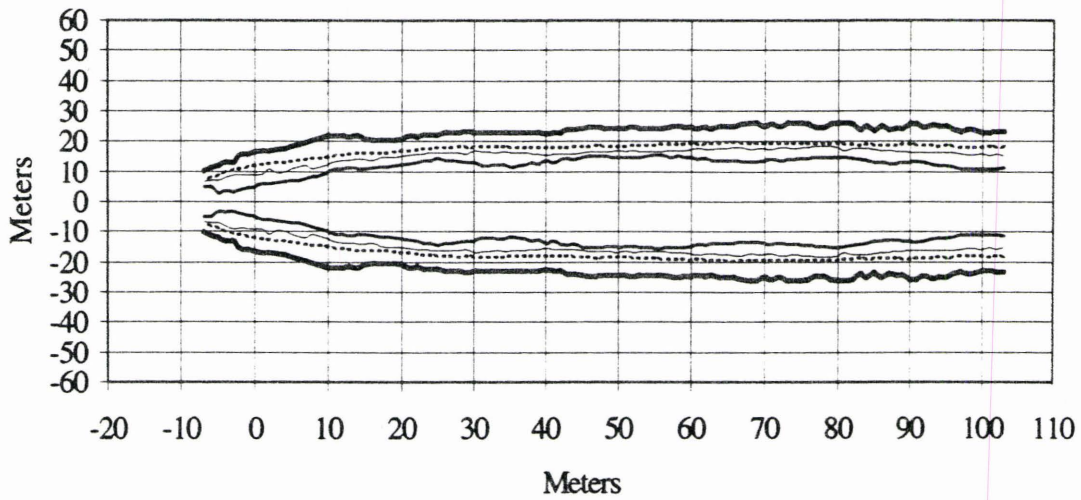
Figure 8. The stochastic means and quantiles for the 4σ data sets.
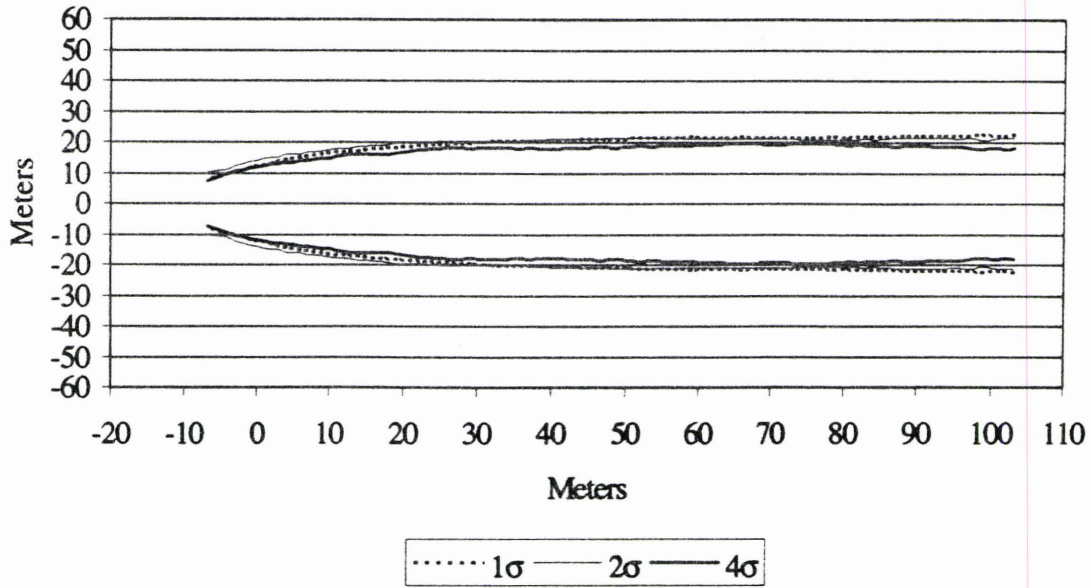
Figure 9. The stochastic means for the 1σ, 2σ, and 4σ sets of realizations.
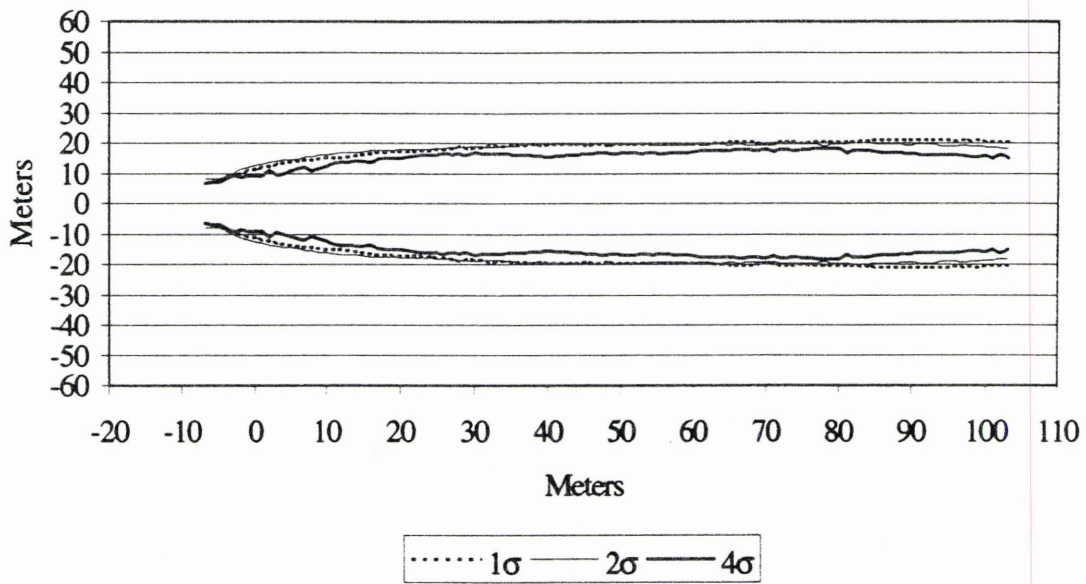
Figure 10. The 80% quantiles for the 1σ, 2σ, and 4σ sets of realizations.
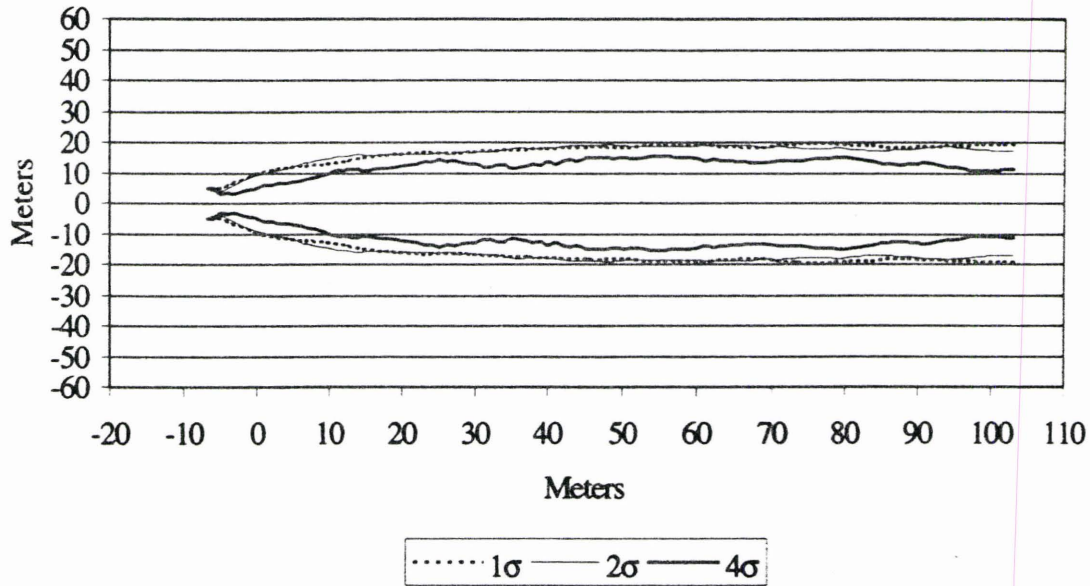
Figure 11. The 99% quantiles for the 1σ, 2σ, and 4σ sets of realizations.

The method presented in this study generates capture zones based on
heterogeneous, anisotropic, and three-dimensional K fields. The heterogeneous K fields
used in this study were produced with a random field generator. Application of the
random field generator assumed the geologic media was spatially periodic and the K field
log normally distributed. However, this assumption has been questioned, and researchers
have proposed alternative models for describing heterogeneity (Anderson, 1997). These
alternative conceptual models include fractal models (e.g. Neuman, 1990, 1994, 1995;
Neuman et al., 1990; Desbarats and Bachu, 1994), geologic facies models (e.g. Miall,
1985; Young et al., 1990), and indicator statistics (e.g. Davis, et al., 1993; Johnson,
1995). More information on alternative conceptual models is available in a review by
Koltermann and Gorelick (1996). Regardless of the conceptual model used to produce
the K fields, the method in this study is still viable.

Random field generators can produce K fields that are geologically unreasonable at a given site. A Monte Carlo analysis employing conditional simulation removes these realizations from the data set. Conditional simulations are constrained to retain field-measured values of parameters at appropriate positions in the generated field (Anderson, 1997). A conditional simulation introduces a statistical bias into the distribution of the K fields and, consequently, can not produce a true ensemble field. Conditional simulation was not used in this study. Instead, the generated K fields are hypothetical, although similar to the parameters of the Borden aquifer. Consequently, there are no field-measured parameters to use in a conditional simulation. However, site specific applications of the method described here, when combined with field-measured parameters, will produce results that are more meaningful if conditional simulation is used.

The results of this study disagree with some of the results of a study performed by Chan (1993). Part of the Chan (1993) study compared how changing the variance from 0.4 to 0.01 would influence the reliability level of the Monte Carlo simulation. Chan (1993) concluded that the reliability level of the Monte Carlo simulation was relatively insensitive to this change in variance.

One possible reason for the differences could be that Chan's (1993) study used conditional simulation while this study did not. However, Chan (1993) also used two-dimensional, isotropic K fields with no vertical component of flow. This study uses a three-dimensional, anisotropic K field with vertical flow. The difference in how the K fields were represented could also explain why the results are not consistent. A more

meaningful comparison could be made if the method presented here also used a

conditioned realization. This is an area for possible future research.

SUMMARY AND CONCLUSIONS

The determination of the capture zone is a critical part of a pump-and-treat design. Overestimation of the capture zone size will result in extracting too much uncontaminated water while underestimation will allow part of the contaminant plume to flow past the influence of the well.

The method presented in this study was used to identify differences between capture zones of a heterogeneous, anisotropic, and three-dimensional K field and a homogeneous, isotropic, and two-dimensional K field with similar parameters. The primary hypothesis of the study was heterogeneity affects the size of a capture zone; as heterogeneity increases, the size of the capture zone decreases for a given Q. In an application of the method, the variance in three data sets was changed to determine if heterogeneity affected the capture zone size.

Three data sets with different variances (1σ, 2σ, and 4σ) were analyzed to test the hypothesis. The 1σ and 2σ data sets produced similarly sized capture zones while the 4σ capture zones were smaller. A possible explanation for this behavior is that the difference in the variance between the 1σ and 2σ was not great enough to generate any significant change in the size of the capture zone. However, the 4σ data set results do suggest that increasing heterogeneity decreased the size of the capture zones for a given Q.

29

APPENDICES

APPENDIX A

THE RANDOM FIELD GENERATOR

Several methods have been developed to generate random fields, including turning bands (Journel and Huijbregts, 1978; Mantoglou and Wilson, 1982), matrix decomposition (Davis, 1987), nearest neighbor (Smith and Freeze, 1979a,b), and spectral (Borgman et al., 1984; Gutjahr, 1989; Gutjahr et al., 1992; Gutjahr et al., 1993; Gutjahr et al., 1994; Weber et al., 1991; Pardo-Iguzquiza and Chica-Olmo, 1993; and Robins et al., 1993). The turning bands method can produce erroneous streaks in both the fields and the covariance (McKay et al., 1988; Thompson et al., 1989). The matrix decomposition method creates immense matrices that are possibly sensitive to numerical error. The nearest neighbor method is quite fast, but can only handle a limited class of covariances (Gutjahr et al., 1996). A comparison of different types random field generators, as well as a discussion of their individual benefits and drawbacks, is presented by Zimmerman et al. (1998).

The random field generator chosen for this work uses a spectral method based on the spectral representation theorem. The method differs from other spectral methods because it incorporates a folded fast Fourier transform algorithm. The algorithm is quick and adaptable for generating large fields in both two and three dimensions for any spectral or cross spectral density (Gutjahr, 1996).

The random field generator creates fields with the approximate geostatistical characteristics provided by the user and a mean of about zero. Because the random field generator uses a fast Fourier transform algorithm, the field dimensions of the mesh are constrained to powers of two. However, if it becomes desirable to strip off unnecessary layers from the mesh (as was the case in this project), the remaining layers retain the geostatistical properties of the entire field (Gutjahr, personal communication, 1998).

The code for the random field generator was incorporated into a FORTRAN program called CRUNCHER, which generated input files for the random field generator and converted the generated fields into a format usable in MODFLOW. CRUNCHER also shifted the ln K means (the K values were assumed to be natural log normally distributed) of the fields from zero to -9.2 m/s. CRUNCHER then switched the hydraulic conductivity values from ln K back to K, and changed the units from m/s to m/day. A typical realization for the 1σ and 4σ data sets produced by CRUNCHER demonstrated how changes in variance affected the resulting K fields (Figure 12).
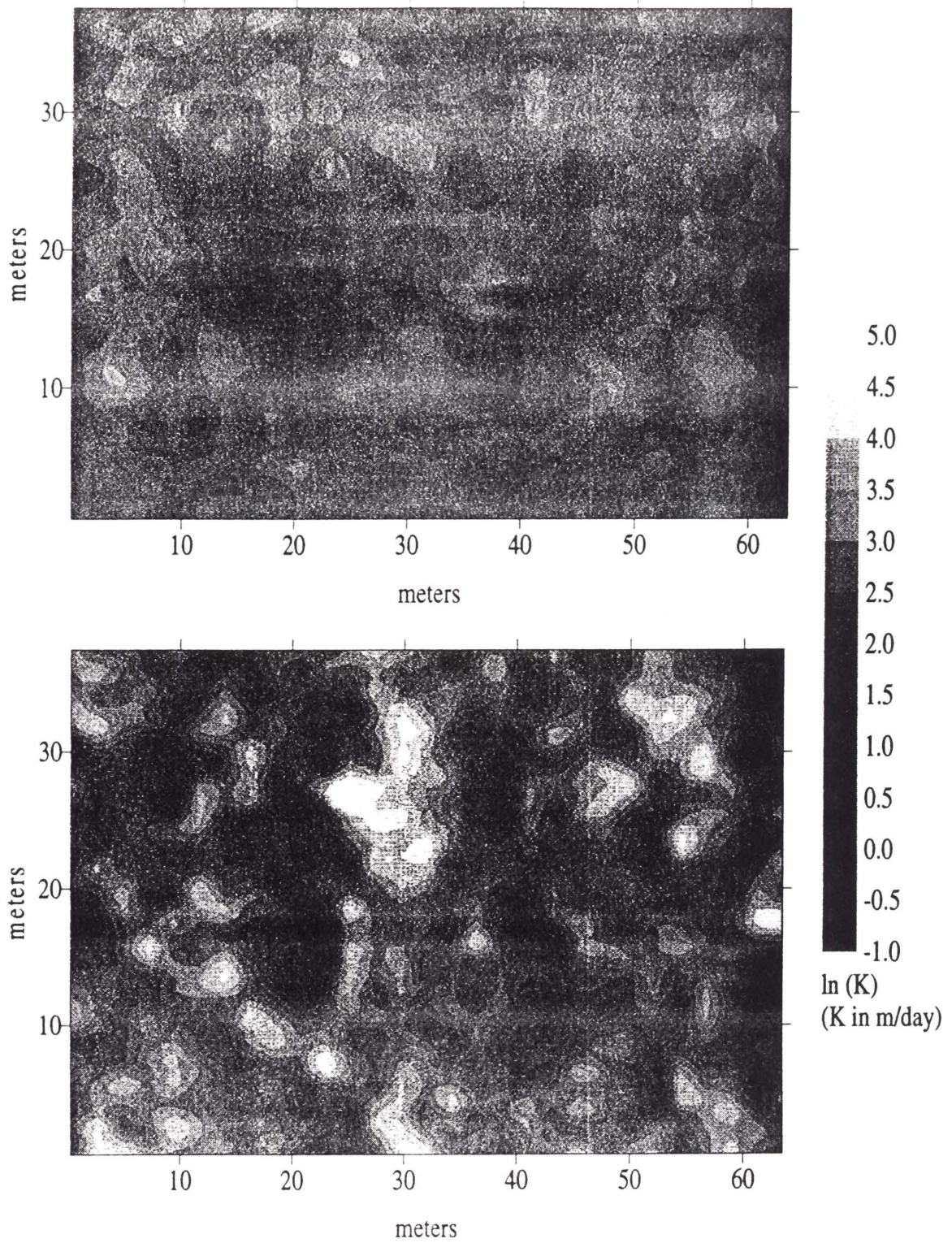
Figure 12. Example of original variance (top) and quadrupled variance (bottom) realizations.

APPENDIX B

THE SOURCE CODE FOR THE PROGRAM CRUNCHER

Program Cruncher

```
*
***********************************************************************
*    This program creates the input file sf3d.in for
*    Dr. Gutjahrs 3-D random field generator (RFG), runs the
*    generator (subroutine sf3d), and converts the output into a
*    form that can be inputted into GW-VISTAS as hydraulic
*    conductivity fields for MODFLOW.
***********************************************************************
*
***********************************************************************
*    Note: To increase the maximum allowed grid size in the R.F.G.
*    you must increase the value of maxn throughout this entire
*    program. Maxn can only be a power of 2 (2, 16, 512, etc.).
***********************************************************************
*
     PARAMETER(maxn=256,ndvice=22)
     REAL f(maxn,maxn,maxn)
     CHARACTER*30,name1
     COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
     COMMON/seed/nseed0,nseed
     COMMON/cons/pi,pi2,itype
     COMMON/cl/clx,cly,clz,sigsq
     COMMON/simu/f,fmean,fvar
     COMMON/name/name1
***********************************************************************
*    ***Define constants.
     pi=3.141592654
     pi2=2*pi
*
*    *****Create random field input file (sf3d.in)
     CALL create
*
*    *****Generate random field
     CALL sf3d
*
*    *****Convert output from a matrix into individual data layers
     CALL convert
*
     STOP
*
     END
*
***********************************************************************
```

```fortran
      SUBROUTINE create
************************************************************************
*     This subroutine queries the user for the data needed to create
*     the file sf3d.in which is needed to run Gutjahr's random field
*     generator.
************************************************************************
*
      OPEN (UNIT=25, FILE='sf3d.in', STATUS='UNKNOWN')
*
************************************************************************
*     Keyboard input for srfg.in
************************************************************************
      WRITE(*,*)'Model type (1, 2 or 3)?'
      READ (*,*)itype
10    WRITE(*,*)'Input the X,Y & Z (integer) dimensions of the model.'
      WRITE(*,*)'All three must be powers of 2 (ex:32,16,4).'
      READ (*,*)nx,ny,nz
************************************************************************
*     Assuring a power of 2 value for nx,ny, and nz
************************************************************************
         xnx=nx
         yny=ny
         znz=nz
      xvar=LOG(xnx)
      yvar=LOG(yny)
      zvar=LOG(znz)
      two=LOG(2.0)
         If((AMOD(xvar,two)).eq.0.0) Then
            goto 20
              else
                 goto 10
         End If
20    If((AMOD(yvar,two)).eq.0.0) Then
            goto 30
              else
                 goto 10
         End If
30    If((AMOD(zvar,two)).eq.0.0) Then
            goto 40
              else
                 goto 10
         End If
************************************************************************
*
40    WRITE(*,*)'Input the (real) X,Y & Z spatial step size'
```

```
      WRITE(*,*)'(ex:10.0,10.0,1.0).'
      READ(*,*)dx,dy,dz
      WRITE(*,*)'Input the desired (real) variance (ex:0.28).'
      READ(*,*) sigsq
      WRITE(*,*)'Input the (real) scale lengths (ex:5.1,4.6,0.28).'
      READ(*,*)clx,cly,clz
      WRITE(*,*)'Input a (integer) seed number (ex:123456).'
      READ(*,*)nseed
***********************************************************************
*     The input file srfg.in has seven lines and contains the
*     information needed to produce a three dimensional grid
*     with user specified geostatistical properties. The input
*     fields are explained below:
***********************************************************************
*     itype             spectral model
*     nx,ny,nz          x,y,z field size
*     dx,dy,dz          x,y,and z spatial discretization
*     sigsq             desired variance for the field
*     clx,cly,clz       x,y, and z scale length
*     nseed             random number generator seed
*     FIELD.TXT            output file name for the field
*                       created by Gutjahr's R.F.G.
***********************************************************************
      WRITE (25,200)itype
200   FORMAT (I1)
*
      WRITE(25,210)nx,ny,nz
210    FORMAT (I3,',',I3,',',I3)
*
      WRITE (25,220)dx,dy,dz
220   FORMAT (F5.2,',',F5.2,',',F5.2)
*
      WRITE(25,230)sigsq
230   FORMAT (F5.2)
*
      WRITE (25,220)clx,cly,clz
*
      WRITE (25,240)nseed
240   FORMAT (I6)
*
      WRITE (25,250)
250   FORMAT ("FIELD.TXT")
*
      CLOSE (UNIT=25)
*
```

```
      RETURN
*
      END
*
**************************************************************************
*     THE BEGINNING OF THE R.F.G. SUBROUTINES
**************************************************************************
**************************************************************************
*     sf3d is Dr. Gutjahr's 3 dimensional random field generator
*     It produces a matrix of values with user supplied geostatistical
*     properties and matrix dimensions. The mean of the field is
*      approximately zero and the variance is input by the user
**************************************************************************
      SUBROUTINE sf3d
*
*     To compile: f77 sf3d.f -o sf3d
*
*     This program generates single 3D random fields.
*     New folded version using FFT, January 1994
*
      PARAMETER(maxn=256,ndvice=22)
      REAL f(maxn,maxn,maxn)
      CHARACTER*30,name1
      COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
      COMMON/seed/nseed0,nseed
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
      COMMON/simu/f,fmean,fvar
      COMMON/name/name1
*
*     *****Define constants.
      pi=3.141592654
      pi2=2*pi
*
*     *****Input data from sf3d.in.
      CALL input

*     *****Call the random generating subroutine.
      CALL fieldgen
*
*     *****Calculate the mean and variance of the field.
      CALL meanvar

*     *****Output the results.
      CALL output
```

```
*
      END
*

**************************************************************
**************************************************************
*

      SUBROUTINE input
*

      PARAMETER(maxn=256,ndvice=22)
      REAL f(maxn,maxn,maxn)
      CHARACTER*30,name1
      COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
      COMMON/seed/nseed0,nseed
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
      COMMON/simu/f,fmean,fvar
      COMMON/name/name1
*
*     *****Set up input data file sf3d.in.
      OPEN(UNIT=25,FILE='sf3d.in')
*
*     *****Types of spectral covariance pairs.
*     *****  For Bell-shaped, input itype=1.
*     *****  For Exponential, input itype=2.
*     *****  For Spherical, input itype=3.
      READ(25,*)itype
*
*     *****Field size 'nx,ny,nz' are  power of 2 <=64.
      READ(25,*) nx,ny,nz
*
*     *****Enter the x, y, z spatial step size (dx,dy,dz).
      READ(25,*) dx,dy,dz
*
*     *****Enter the desired variance of the random field.
      READ(25,*) sigsq
*
*     *****Enter the x-scale, y-scale, and z-scale.
      READ(25,*) clx,cly,clz
*
*     *****Enter the random number generator seed.
      READ(25,*) nseed
      nseed0=nseed
      IF (nseed.ge.0) nseed=-nseed
*
*     *****Enter the name of the field.
```

```
      READ(25,*) name1
*

      RETURN
      END
*
*****************************************************************
*****************************************************************
*

      SUBROUTINE fieldgen
*
*     Subroutine 'fieldgen' will generate the random field
*     f(i,j) with SRT and FFT methods. In this subroutine
*     we generate f(i,j)with mean 'zero' and veriance 'sigsq'.
*

      PARAMETER(maxn=256,ndvice=22)
      REAL xx(2*maxn*maxn*maxn)
      INTEGER nn(3)
      COMPLEX im,zx,zz1,yf(maxn,maxn,maxn)
      REAL f(maxn,maxn,maxn)
      CHARACTER*30,name1
      COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
      COMMON/seed/nseed0,nseed
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
      COMMON/simu/f,fmean,fvar
      COMMON/name/name1
*

      im=(0.0,1.0)
      sq5=sqrt(0.5)
*
*     *****Determine x,y,z steps in the frequency domain.
      mx=nx/2
      my=ny/2
      mz=nz/2
      dfxu=1./(nx*dx)
      dfyu=1./(ny*dy)
      dfzu=1./(nz*dz)
      duxyz=dfxu*dfyu*dfzu
*
*     *****Load the complex array for transformation.
      DO ka=1,mx
        ux=(ka-0.5)*dfxu
*
        DO kb=1,my
          uy=(kb-0.5)*dfyu
```

```
      DO kc=1,mz
        uz=(kc-0.5)*dfzu
        sqtdz=spectral(ux,uy,uz)
        sqtdz=sqrt(sqtdz*duxyz)
        CALL norm01(sq5,rtemp,nseed)
        yreal=rtemp*sqtdz
        CALL norm01(sq5,rtemp,nseed)
        yimage=rtemp*sqtdz
        yf(ka,kb,kc)=cmplx(yreal,yimage)
      END DO
      DO kc=1,mz
        uz=-(kc-0.5)*dfzu
        sqtdz=spectral(ux,uy,uz)
        sqtdz=sqrt(sqtdz*duxyz)
        CALL norm01(sq5,rtemp,nseed)
        yreal=rtemp*sqtdz
        CALL norm01(sq5,rtemp,nseed)
        yimage=rtemp*sqtdz
        yf(ka,kb,nz+1-kc)=cmplx(yreal,yimage)
      END DO
    END DO
  END DO
*
  DO kb=1,my
    uy=-(kb-0.5)*dfyu
    DO kc=1,mz
      uz=(kc-0.5)*dfzu
      sqtdz=spectral(ux,uy,uz)
      sqtdz=sqrt(sqtdz*duxyz)
      CALL norm01(sq5,rtemp,nseed)
      yreal=rtemp*sqtdz
      CALL norm01(sq5,rtemp,nseed)
      yimage=rtemp*sqtdz
      yf(ka,ny+1-kb,kc)=cmplx(yreal,yimage)
    END DO
    DO kc=1,mz
      uz=-(kc-0.5)*dfzu
      sqtdz=spectral(ux,uy,uz)
      sqtdz=sqrt(sqtdz*duxyz)
      CALL norm01(sq5,rtemp,nseed)
      yreal=rtemp*sqtdz
      CALL norm01(sq5,rtemp,nseed)
      yimage=rtemp*sqtdz
      yf(ka,ny+1-kb,nz+1-kc)=cmplx(yreal,yimage)
    END DO
  END DO
```

```
      END DO
*
*     *****Pack the array with indices bigger than mx with 0.
      DO la=1,mx
        DO lb=1,my
          DO lc=1,mz
            yf(mx+la,lb,lc)=(0.0,0.0)
            yf(mx+la,my+lb,lc)=(0.0,0.0)
            yf(mx+la,lb,mz+lc)=(0.0,0.0)
            yf(mx+la,my+lb,mz+lc)=(0.0,0.0)
          END DO
        END DO
      END DO
*
*     *****Convert 3D array to 1D for FFT transform.
*     ***** Get values of nn(3) array for FFT transform.
      nn(1)=nx
      nn(2)=ny
      nn(3)=nz
*
      num=1
      DO k=1,nz
        DO j=1,ny
          DO i=1,nx
            xx(num)=REAL(yf(i,j,k))
            xx(num+1)=AIMAG(yf(i,j,k))
            num=num+2
          END DO
        END DO
      END DO
*
*     *****Perform the transformation.
      CALL fourn(xx,nn,3,1)
*
*     *****Convert 1D array back to 3D.
      num=1
      DO k=1,nz
        DO j=1,ny
          DO i=1,nx
            yf(i,j,k)=cmplx(xx(num),xx(num+1))
            num=num+2
          END DO
        END DO
      END DO
*
```

```
*       *****Extract the f field from the complex array.
*       *****  Multiply Y by W to get the final results.
    DO ja=1,mx
      j1=ja-1
      j11=j1+mx+1
      DO jb=1,my
        j2=jb-1
        j22=j2+my+1
        DO jc=1,mz
          j3=jc-1
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
        DO jc=mz+1,nz
          j3=jc-1-nz
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
      END DO
*
      DO jb=my+1,ny
        j2=jb-1-ny
        j22=j2+my+1
        DO jc=1,mz
          j3=jc-1
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
        DO jc=mz+1,nz
          j3=jc-1-nz
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
      END DO
    END DO
*
    DO ja=mx+1,nx
```

```
      j1=ja-1-nx
      j11=j1+mx+1
      DO jb=1,my
        j2=jb-1
        j22=j2+my+1
        DO jc=1,mz
          j3=jc-1
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
        DO jc=mz+1,nz
          j3=jc-1-nz
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
      END DO
*
      DO jb=my+1,ny
        j2=jb-1-ny
        j22=j2+my+1
        DO jc=1,mz
          j3=jc-1
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
        DO jc=mz+1,nz
          j3=jc-1-nz
          j33=j3+mz+1
          zx=im*pi*(j1/float(nx)+j2/float(ny)+j3/float(nz))
          zz1=2.*cexp(zx)*yf(ja,jb,jc)
          f(j11,j22,j33)=real(zz1)
        END DO
      END DO
      END DO
*
      RETURN
      END
*
********************************************************************
```

```
***************************************************************
*
      SUBROUTINE meanvar
*
*     *****Subroutine meanvar will caculate the mean and variance
*     *****  of the simulated random field
*
      PARAMETER(maxn=256,ndvice=22)
      REAL f(maxn,maxn,maxn)
      CHARACTER*30,name1
      COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
      COMMON/seed/nseed0,nseed
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
      COMMON/simu/f,fmean,fvar
      COMMON/name/name1
*
      sum=0.0
      sum1=0.0
*
      DO i=1,nx
        DO j=1,ny
          DO k=1,nz
            sum=sum+f(i,j,k)
          END DO
        END DO
      END DO
*
      nxyz=nx*ny*nz
      fmean=sum/nxyz
      DO i=1,nx
        DO j=1,ny
          DO k=1,nz
            sum1=sum1+(f(i,j,k)-fmean)**2
          END DO
        END DO
      END DO
*
      fvar=sum1/(nxyz-1)
*
      RETURN
      END
*
***************************************************************
***************************************************************
```

```fortran
*
      SUBROUTINE output
*
      PARAMETER(maxn=256,ndvice=22)
      REAL f(maxn,maxn,maxn)
      CHARACTER*30,name1
      COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
      COMMON/seed/nseed0,nseed
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
      COMMON/simu/f,fmean,fvar
      COMMON/name/name1
*
      PRINT *,fmean,fvar
*
*     *****Output parameters to field.dat.
      OPEN(UNIT=ndvice,FILE='field.dat')
      WRITE(ndvice,*)'Size of the Random Field:',nx,' x',ny,' x',nz
      WRITE(ndvice,*)'Seed for the R.N.G. Used: ',nseed0
      WRITE(ndvice,*)'Covariance-Spectral Type: ',itype
      WRITE(ndvice,*)'Scale of Random Field: x-scale=',clx
      WRITE(ndvice,*)'                       y-scale=',cly
      WRITE(ndvice,*)'                       z-scale=',clz
      WRITE(ndvice,*)'Spacing of the random field: ',dx,' x',dy,' x'
     b    ,dz
      WRITE(ndvice,*)'Input Variance of the Field: ',sigsq
      WRITE(ndvice,*)'Mean of f field = ',fmean
      WRITE(ndvice,*)'Variance of f field = ',fvar
      CLOSE(ndvice)
*
      OPEN(UNIT=ndvice,FILE=name1)
      DO i=1,nx
        DO j=1,ny
          WRITE(ndvice,11)(f(i,j,k),k=1,nz)
 11       FORMAT(2X,64(F8.4,2X))
        END DO
      END DO
      CLOSE(ndvice)
*
      RETURN
      END
*
************************************************************************
************************************************************************
*
```

```
      FUNCTION spectral(ux,uy,uz)
*
*     This function will generate the spectral function values for
*       the specified spectral-covariance pairs.
*
      COMMON/cons/pi,pi2,itype
      COMMON/cl/clx,cly,clz,sigsq
*
      picube=pi2**3
      x=ux*pi2
      y=uy*pi2
      z=uz*pi2
      xyz=(clx*x)**2+(cly*y)**2+(clz*z)**2
      c=clx*cly*clz
*
*     *****Gaussian(Bell-SHaped)
      IF(itype.EQ.1)g=SQRT(pi**3)*c*EXP(-xyz/4.)
*
*     *****Exponential
      IF(itype.EQ.2)g=8.*pi*c/((1.+xyz)**2)
*
*     *****Spherical
      IF(itype.EQ.3) THEN
        xyz1=SQRT(2*pi*xyz)
        c1=c*xyz1
        xk1=(3*xyz1*xyz1*c*c+12)/(c**3*xyz1**5)
        xk2=12*c1*SIN(c1)+(12-3*c1*c1)*COS(c1)
        g=picube*(xk1-xk2/(c**3*xyz1**5))*2./xyz1
      ENDIF
*
      spectral=g*sigsq
*
      RETURN
      END
*
******************************************************************
******************************************************************
*
      SUBROUTINE fourn(data,nn,ndim,isign)
*
*     *****Numerical Recipes in Fortran, 2nd ed., p.518.
*
      INTEGER isign,ndim,nn(ndim)
      REAL data(*)
      INTEGER i1,i2,i2rev,i3,i3rev,ibit,idim,ifp1,ifp2,ip1,ip2,
```

```fortran
     b    ip3,k1,k2,n,nprev,nrem,ntot
      REAL tempi,tempr
      DOUBLE PRECISION theta,wi,wpi,wpr,wr,wtemp
*
      ntot=1
      DO idim=1,ndim
        ntot=ntot*nn(idim)
      END DO
*
      nprev=1
*
      DO idim=1,ndim
        n=nn(idim)
        nrem=ntot/(n*nprev)
        ip1=2*nprev
        ip2=ip1*n
        ip3=ip2*nrem
        i2rev=1
        DO i2=1,ip2,ip1
          IF(i2.LT.i2rev)THEN
            DO i1=i2,i2+ip1-2,2
              DO i3=i1,ip3,ip2
                i3rev=i2rev+i3-i2
                tempr=data(i3)
                tempi=data(i3+1)
                data(i3)=data(i3rev)
                data(i3+1)=data(i3rev+1)
                data(i3rev)=tempr
                data(i3rev+1)=tempi
              END DO
            END DO
          END IF
          ibit=ip2/2
135       IF((ibit.GE.ip1).AND.(i2rev.GT.ibit))THEN
            i2rev=i2rev-ibit
            ibit=ibit/2
            GO TO 135
          END IF
          i2rev=i2rev+ibit
        END DO
        ifp1=ip1
145     IF(ifp1.LT.ip2)THEN
          ifp2=2*ifp1
          theta=isign*6.28318530717959d0/(ifp2/ip1)
          wpr=-2.0D0*DSIN(0.5D0*theta)**2
```

```
      wpi=DSIN(theta)
      wr=1.0D0
      wi=0.0D0
      DO i3=1,ifp1,ip1
        DO i1=i3,i3+ip1-2,2
          DO i2=i1,ip3,ifp2
            k1=i2
            k2=k1+ifp1
            tempr=sngl(wr)*data(k2)-sngl(wi)*data(k2+1)
            tempi=sngl(wr)*data(k2+1)+sngl(wi)*data(k2)
            data(k2)=data(k1)-tempr
            data(k2+1)=data(k1+1)-tempi
            data(k1)=data(k1)+tempr
            data(k1+1)=data(k1+1)+tempi
          END DO
        END DO
        wtemp=wr
        wr=wr*wpr-wi*wpi+wr
        wi=wi*wpr+wtemp*wpi+wi
      END DO
      ifp1=ifp2
      GO TO 145
    END IF
    nprev=n*nprev
  END DO
*
  RETURN
  END
*********************************************************************
*********************************************************************
*
  FUNCTION rnun01(idum)
*
*   Numerical Recipes in Fortran, 2nd ed., p.272.
*   Generate uniform random number between 0 and 1.
*
  PARAMETER (im1=2147483563, im2=2147483399,am=1./im1, imm1=im1-1,
 b   ia1=40014,ia2=40692,iq1=53668,iq2=52774,ir1=12211,
 c   ir2=3791, ntab=32,ndiv=1+imm1/ntab,eps=1.2e-7,rnmx=1.-eps)
  INTEGER iv(ntab)
  REAL rnun01
*
  SAVE iv,iy,idum2
  DATA idum2/123456789/, iv/ntab*0/, iy/0/
*
```

```
      IF(idum.LE.0.)THEN
        idum=max(-idum,1)
        idum2=idum
        DO j=ntab+8,1,-1
          k=idum/iq1
          idum=ia1*(idum-k*iq1)-k*ir1
          IF(idum.LT.0)idum=idum+im1
          IF(j.LE.ntab)iv(j)=idum
        END DO
        iy=iv(1)
      END IF
      k=idum/iq1
      idum=ia1*(idum-k*iq1)-k*ir1
      IF(idum.LT.0.)idum=idum+im1
      k=idum2/iq2
      idum2=ia2*(idum2-k*iq2)-k*ir2
      IF(idum2.LT.0)idum2=idum2+im2
      j=1+iy/ndiv
      iy=iv(j)-idum2
      iv(j)=idum
      IF(iy.LT.1)iy=iy+imm1
      rnun01=MIN(am*iy,rnmx)
*

      RETURN
      END
*
******************************************************************
******************************************************************
*

      SUBROUTINE norm01(sigma,rtemp,nseed)
*
*    Subroutine for generating normal distributed randome numbers
*      with mean zero and variance 0.5 using the central limit
*      theorem.
*
      rtemp=0.0
      DO j=1,12
        rtemp=rtemp+rnun01(nseed)
      END DO
      rtemp=rtemp-6.0
      rtemp=sigma*rtemp
*
      RETURN
      END
*
```

```
**********************************************************************
*    END OF THE R.F.G. SUBROUTINES
**********************************************************************
*
     SUBROUTINE convert
*
**********************************************************************
*    Convert the R.F.G. output into a form usable in GW-VISTAS by
*    decomposing the matrix into individual layers with X-Y
*    coordinates.
**********************************************************************
*
**********************************************************************
*    OUT is an array used to hold the values from the R.F.G.
*    XIJ and YIJ are arravs used to determine model coordinates.
*    SURF is an array that reads the data from OUT and converts it
*    into the form needed in GW-VISTAS for the MODFLOW models
*    This subroutine is designed to handle grids up to 64 layers
*    thick.
**********************************************************************
*
     DIMENSION OUT(70000,64),XIJ(70000),YIJ(70000),GWVISTA(70000)
     CHARACTER*10 mflow(64)
*
**********************************************************************
     PARAMETER(maxn=256,ndvice=22)
     REAL f(maxn,maxn,maxn)
     CHARACTER*30,name1
     COMMON/xyzpar/nx,ny,nz,mx,my,mz,dx,dy,dz
     COMMON/seed/nseed0,nseed
     COMMON/cons/pi,pi2,itype
     COMMON/cl/clx,cly,clz,sigsq
     COMMON/simu/f,fmean,fvar
     COMMON/name/name1
*
**********************************************************************
*    Field.txt is the output from the R.F.G. The *.dat files
*    are the repositories of the converted data contained in
*    the OUT array. Each *.dat file will have information
*    pertaining to only that layer.
**********************************************************************
*
     OPEN (UNIT=8,FILE='FIELD.TXT',STATUS='UNKNOWN')
*
**********************************************************************
```

```
*      Initialize and fill the character array 'mflow'. This is an
*      ugly way to do it but I know of no way of using a DO loop
*      that can sequence character-integer file names.
***********************************************************************
*

       mflow(1)='mflow1.dat'
       mflow(2)='mflow2.dat'
       mflow(3)='mflow3.dat'
       mflow(4)='mflow4.dat'
       mflow(5)='mflow5.dat'
       mflow(6)='mflow6.dat'
       mflow(7)='mflow7.dat'
       mflow(8)='mflow8.dat'
       mflow(9)='mflow9.dat'
       mflow(10)='mflw10.dat'
       mflow(11)='mflw11.dat'
       mflow(12)='mflw12.dat'
       mflow(13)='mflw13.dat'
       mflow(14)='mflw14.dat'
       mflow(15)='mflw15.dat'
       mflow(16)='mflw16.dat'
       mflow(17)='mflw17.dat'
       mflow(18)='mflw18.dat'
       mflow(19)='mflw19.dat'
       mflow(20)='mflw20.dat'
       mflow(21)='mflw21.dat'
       mflow(22)='mflw22.dat'
       mflow(23)='mflw23.dat'
       mflow(24)='mflw24.dat'
       mflow(25)='mflw25.dat'
       mflow(26)='mflw26.dat'
       mflow(27)='mflw27.dat'
       mflow(28)='mflw28.dat'
       mflow(29)='mflw29.dat'
       mflow(30)='mflw30.dat'
       mflow(31)='mflw31.dat'
       mflow(32)='mflw32.dat'
       mflow(33)='mflw33.dat'
       mflow(34)='mflw34.dat'
       mflow(35)='mflw35.dat'
       mflow(36)='mflw36.dat'
       mflow(37)='mflw37.dat'
       mflow(38)='mflw38.dat'
       mflow(39)='mflw39.dat'
       mflow(40)='mflw40.dat'
```

```
      mflow(41)='mflw41.dat'
      mflow(42)='mflw42.dat'
      mflow(43)='mflw43.dat'
      mflow(44)='mflw44.dat'
      mflow(45)='mflw45.dat'
      mflow(46)='mflw46.dat'
      mflow(47)='mflw47.dat'
      mflow(48)='mflw48.dat'
      mflow(49)='mflw49.dat'
      mflow(50)='mflw50.dat'
      mflow(51)='mflw51.dat'
      mflow(52)='mflw52.dat'
      mflow(53)='mflw53.dat'
      mflow(54)='mflw54.dat'
      mflow(55)='mflw55.dat'
      mflow(56)='mflw56.dat'
      mflow(57)='mflw57.dat'
      mflow(58)='mflw58.dat'
      mflow(59)='mflw59.dat'
      mflow(60)='mflw60.dat'
      mflow(61)='mflw61.dat'
      mflow(62)='mflw62.dat'
      mflow(63)='mflw63.dat'
      mflow(64)='mflw64.dat'
************************************************************************
*
************************************************************************
*     Opens the *.dat files for the individual R.F.G. layers
************************************************************************
*
************************************************************************
*     Read the data from the field.txt file into the OUT array.
************************************************************************
      DO 300 K=1,nx*ny,1
      READ (8,290) (OUT(K,L),L=1,nz)
290   FORMAT (2X,128(F8.4,2X))
300   CONTINUE
*
************************************************************************
*     Calculate the X and Y coordinates of each K value in the mesh,
*     and convert the field.txt data matrix into a form that can be
*     used in SURFER and GW-VISTAS.
************************************************************************
*
      DO 320 L=1,nz,1
```

```
      L2=L+29
      OPEN (UNIT=(L2),FILE=MFLOW(L),STATUS='UNKNOWN')
      DO 330 I=1,nx,1
      DO 340 J=1,ny,1
*
************************************************************************
*     Surf(k) is calculated in such a matter that the 0.0 mean
*     produced in the R.F.G. is shifted down by 9.20. The value
*     9.20 was chosen because the mean ln K is .0000975 m/s for
*     the Borden aquifer and .0001 m/s for the Columbus aquifer.
*     By shifting the R.F.G. values down and then taking the e^
*     values, a field containing representative hydraulic
*     conductivities is produced. All of the values are then
*     multiplied by 86,400 to convert the conductivities from
*     m/s to m/day for use in the GW-Vistas input file.
************************************************************************
*
      GWVISTA(K)=(EXP((OUT(((I-1)*ny+J),L)-9.2))*86400)
      XIJ(K)=(I-0.5)
      YIJ(K)=(J-0.5)
      M=L+29
      WRITE (M,350)XIJ(K),YIJ(K),GWVISTA(K)
350   FORMAT (F16.8,F16.8,F16.8)
340   CONTINUE
330   CONTINUE
      CLOSE (L2)
320   CONTINUE
*
      CLOSE (8)
*
      RETURN
*
      END
*
************************************************************************
************************************************************************
*
```

APPENDIX C

A SENSITIVITY ANALYSIS OF HOW VERTICAL DISCRETIZATION INFLUENCES VERTICAL CORRELATION LENGTH

58

A vertical mesh thickness set equal to λ_z cannot replicate the spatial variance of hydraulic conductivity with respect to the vertical direction. A model that divides each λ_z into layers incorporates more information about the vertical conductivity field into the model. However, as the number of layers increases so does the amount of computational time needed to run the model. A sensitivity analysis was performed to determine how the modeled variance varied as a function of the number of correlation lengths in each layer. The analysis also determined if increasing the number of layers in the generated field affected the resulting variance. The goal of the test was to find an optimal configuration that generated a desirable variance while minimizing computational effort.

Four different sets of random fields were generated with each set containing five realizations. Each realization had eight layers with either 1, 0.5, 0.25, or 0.125 λ_z present per layer. The average variance was calculated for each set and plotted against the number of λ_z per layer for an 8-layer model (Figure 13). The procedure was repeated for both 16 (Figure 14) and 32 (Figure 15) layer realizations. The random fields were generated using the parameters of an aquifer whose desired variance was 4.5. The plots indicated that obtaining a variance close to the desired variance was best achieved by adding more layers to the randomly generated field. It was also apparent from the plots there was a relationship between the number of correlation lengths per layer and resulting variance. Based on this sensitivity analysis, a 16-layer random field that had 0.25 λ_z per layer was used.
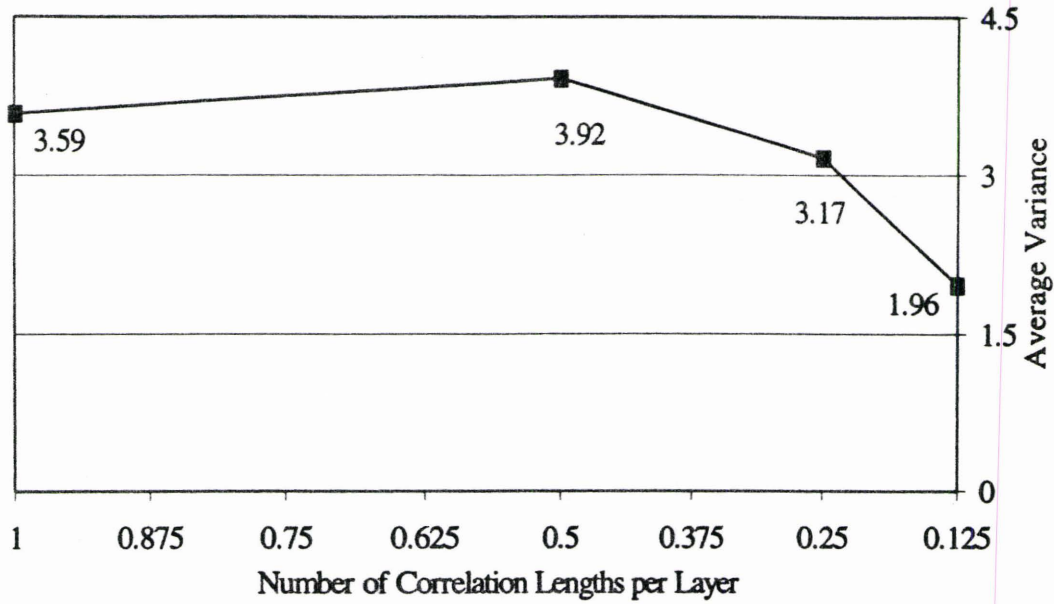
Figure 13. Plot of an eight-layer model of average variance over five runs versus number of correlation lengths per layer.
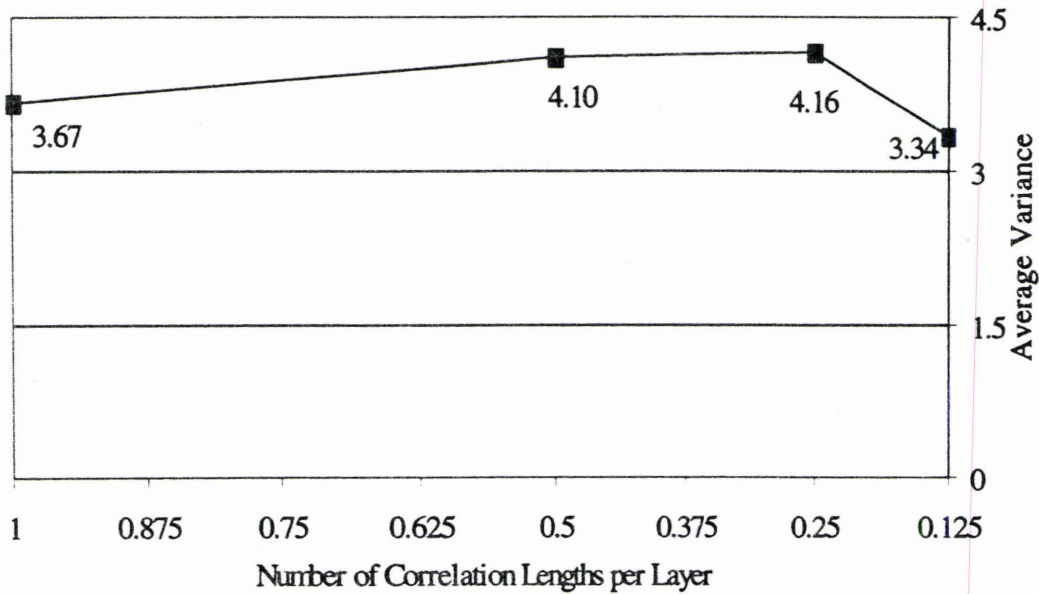
Figure 14. Plot of a 16-layer model of average variance over five runs versus number of correlation lengths per layer.
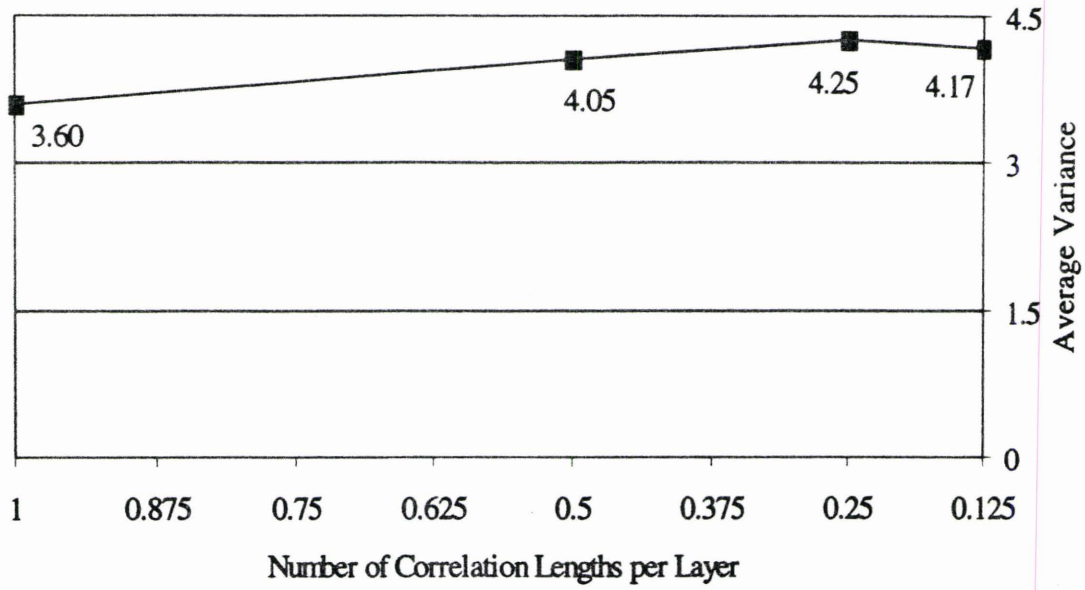
Figure 15. Plot of a 32-layer model of average variance over five runs versus number of correlation lengths per layer.

APPENDIX D

LAG VERSUS GAMMA PLOTS FOR THE RANDOM FIELD GENERATOR

A test was performed to determine if the exponential covariance model option in the random field generator was producing exponentially shaped variograms. A variogram behaving exponentially will trend exponentially upward towards an asymptotic limit. A variogram behaving spherically will trend upward to the asymptote and then flatten out along the asymptote. The source code of the program VARIO (LeFever, 1997) was modified to handle the 8192 cells represented in each layer of the mesh. A 16-layer random field was generated and each layer was analyzed by VARIO. The input parameters for VARIO are given in Table 6.

The resulting gamma versus lag spacing plots (Figures 16-31) exhibited reasonably exponential behavior where gamma represents the variogram function and the lag spacing is the distance between variogram function calculations. Figures 21, 25, 26, 29, and 31 appear to be exhibiting spherical behavior. Some of this spherical behaviour could be accounted by the random variation that would be inherent in using a random field generator.

Table 6. Input parameters for the program VARIO.

Parameter	Value
Number of directions to calculate variograms	1
Length of basic lag	1
Maximum number of lags	30
Minimum value for variable (y or n)	n
Width of distance class	0.5
Width of angle class in degrees	180
Direction to use in degrees counterclockwise	0
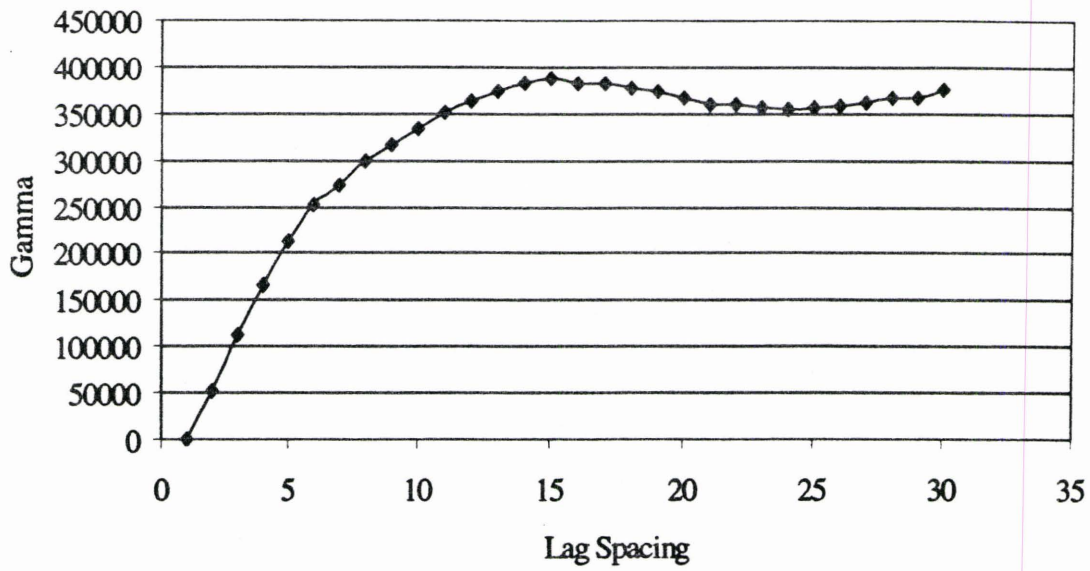
Figure 16. Gamma versus lag spacing for layer 1.
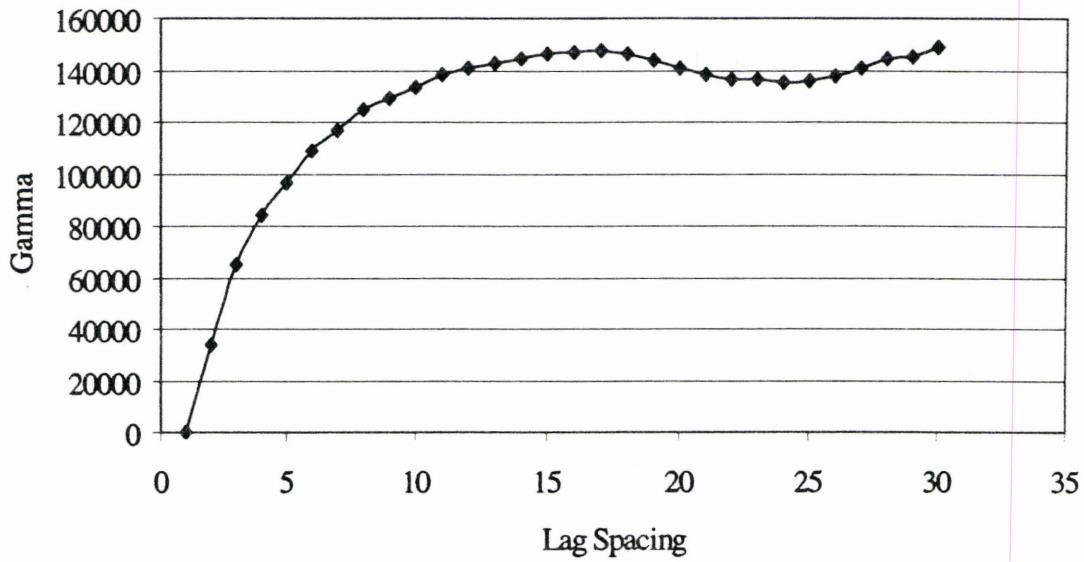
Figure 17. Gamma versus lag spacing for layer 2.
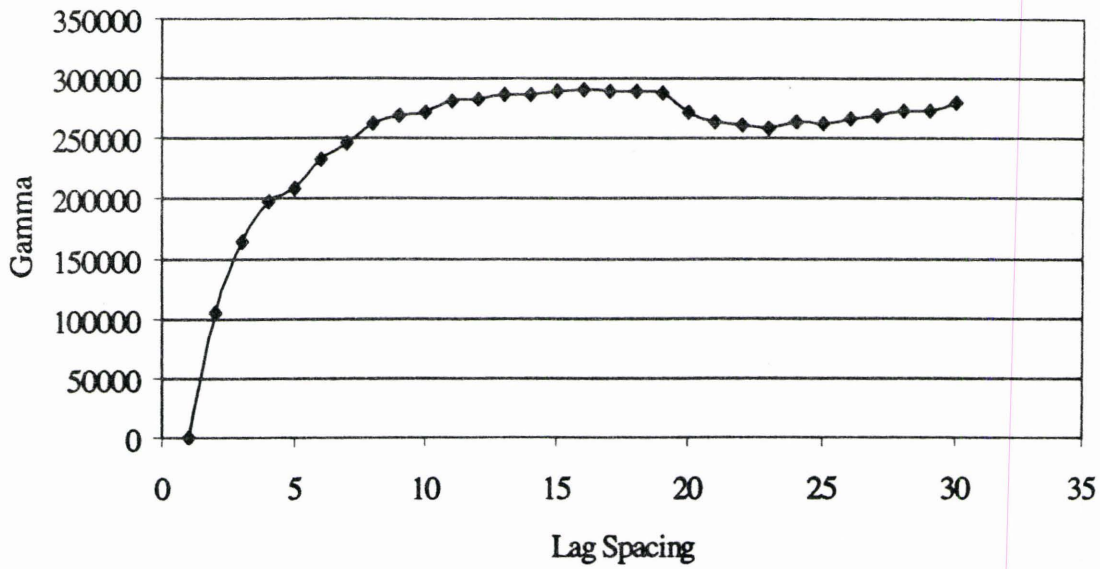
Figure 18. Gamma versus lag spacing for layer 3.
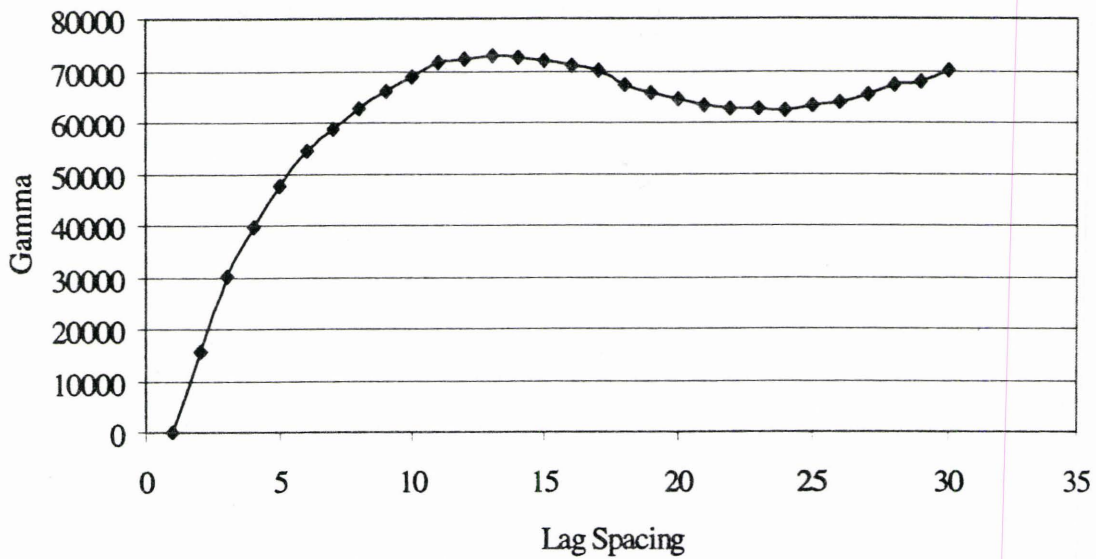
Figure 19. Gamma versus lag spacing for layer 4.
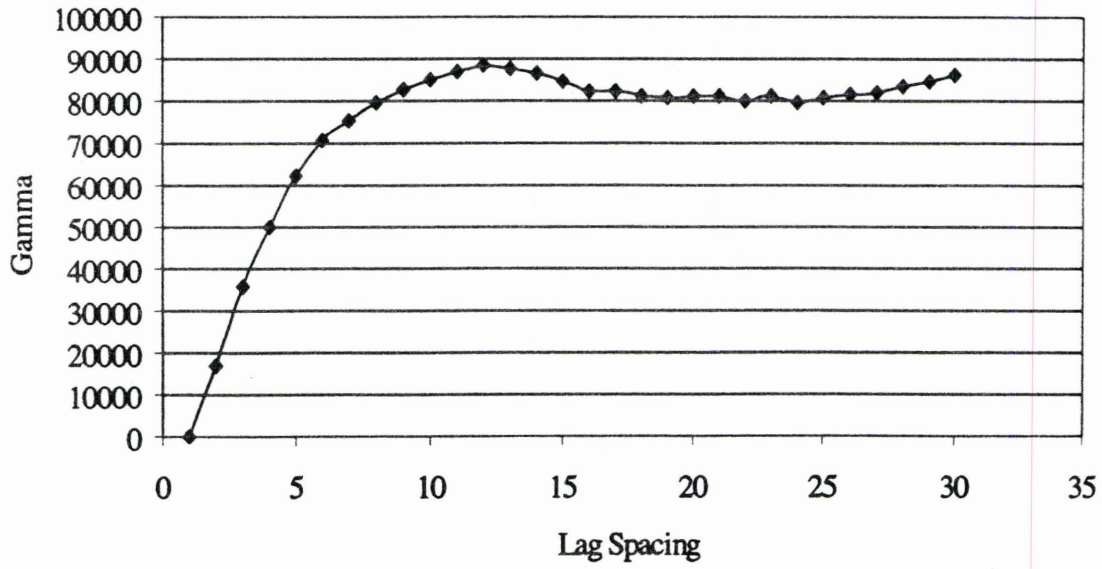
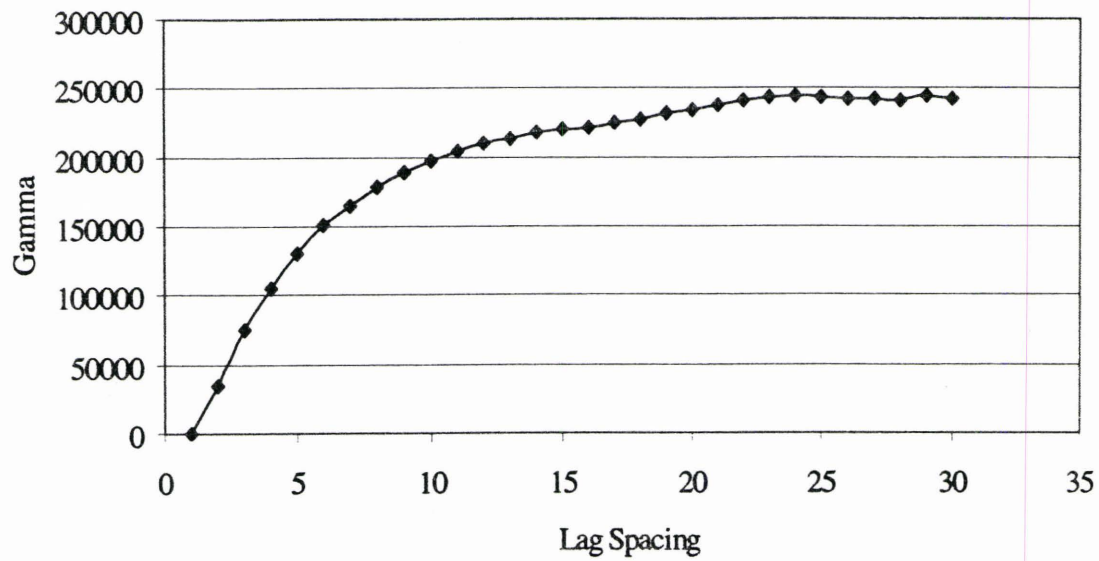Figure 20. Gamma versus lag spacing for layer 5.
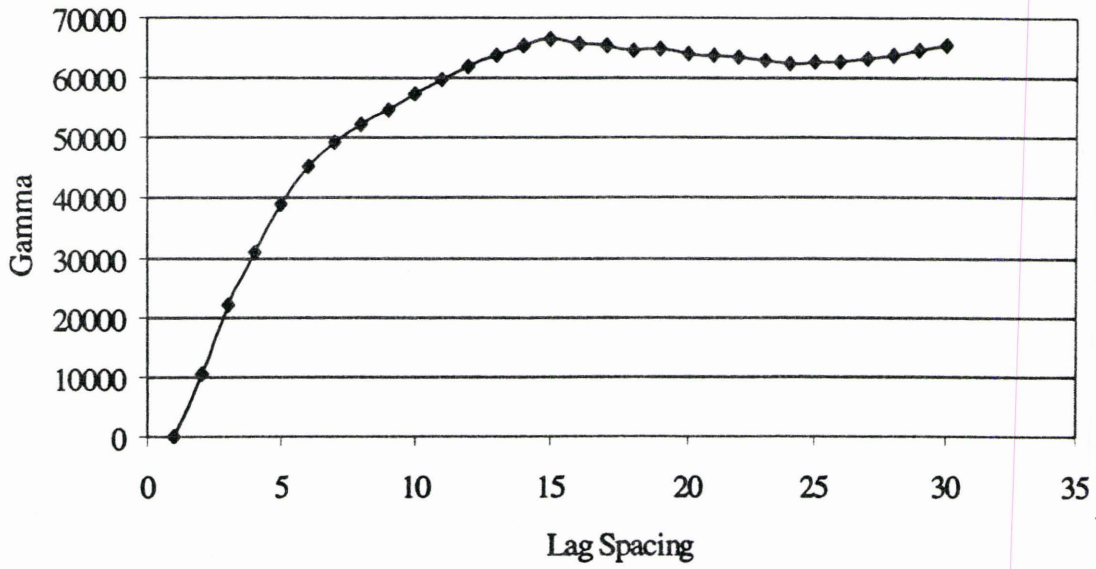

Figure 21. Gamma versus lag spacing for layer 6.
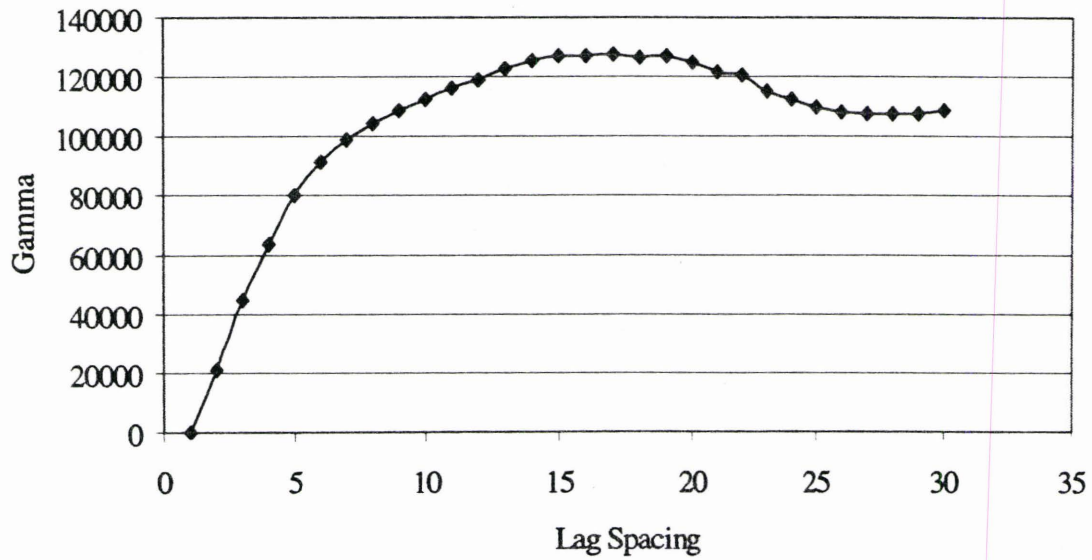
Figure 22. Gamma versus lag spacing for layer 7.
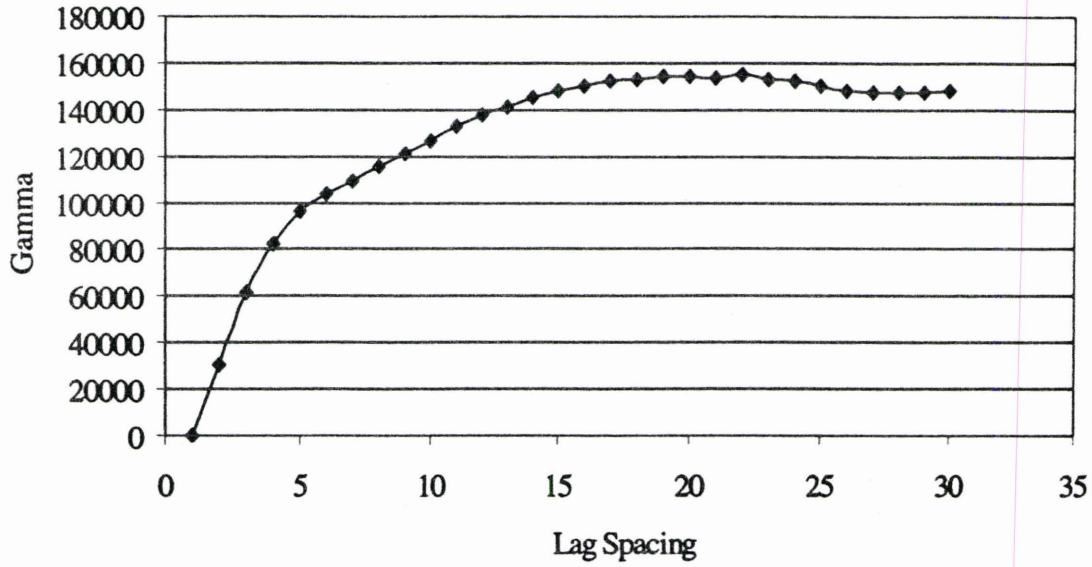

Figure 23. Gamma versus lag spacing for layer 8.
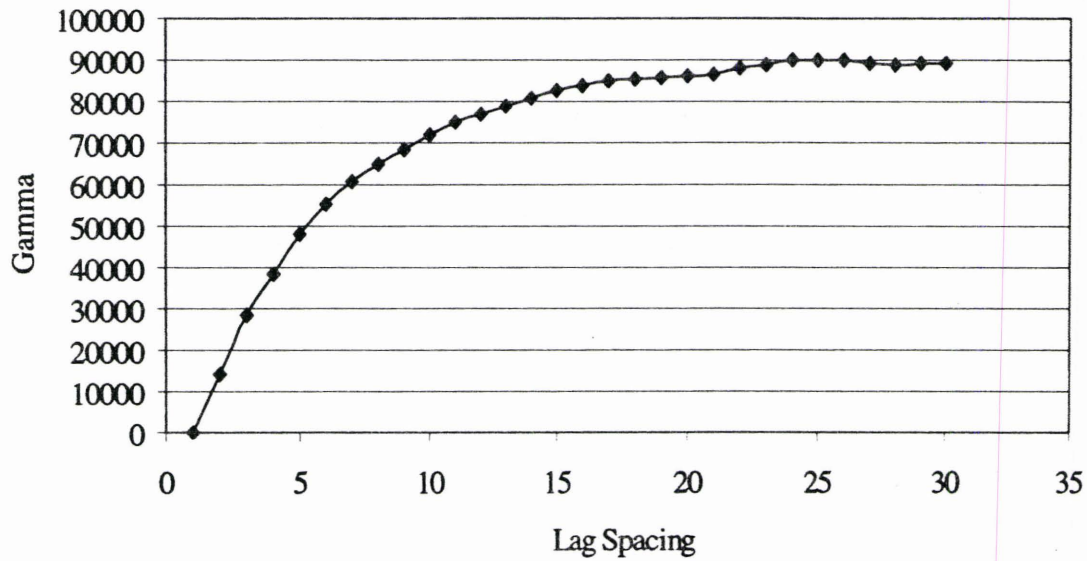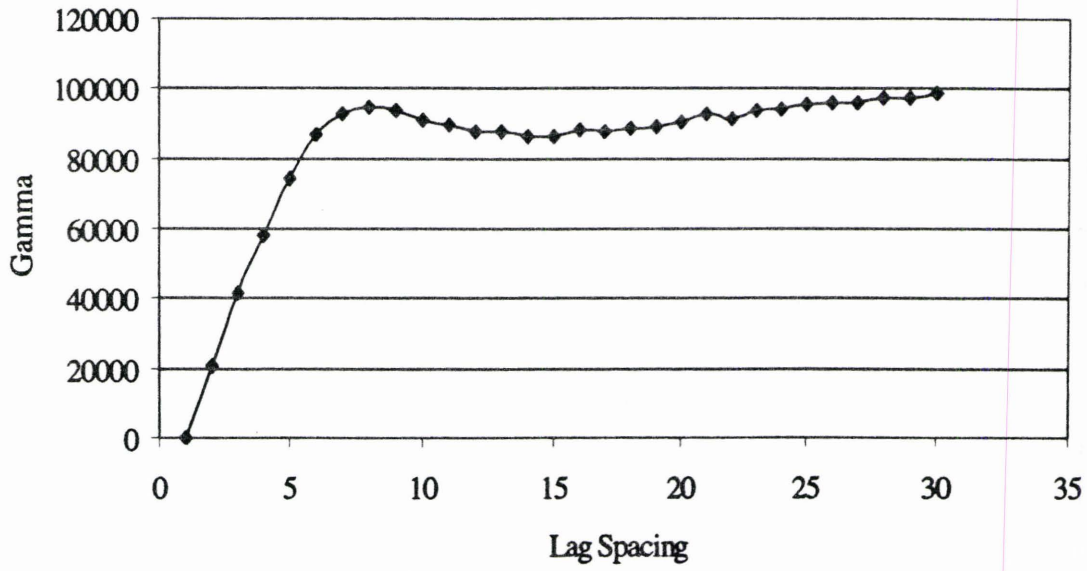
Figure 24. Gamma versus lag spacing for layer 9.

Figure 25. Gamma versus lag spacing for layer 10.
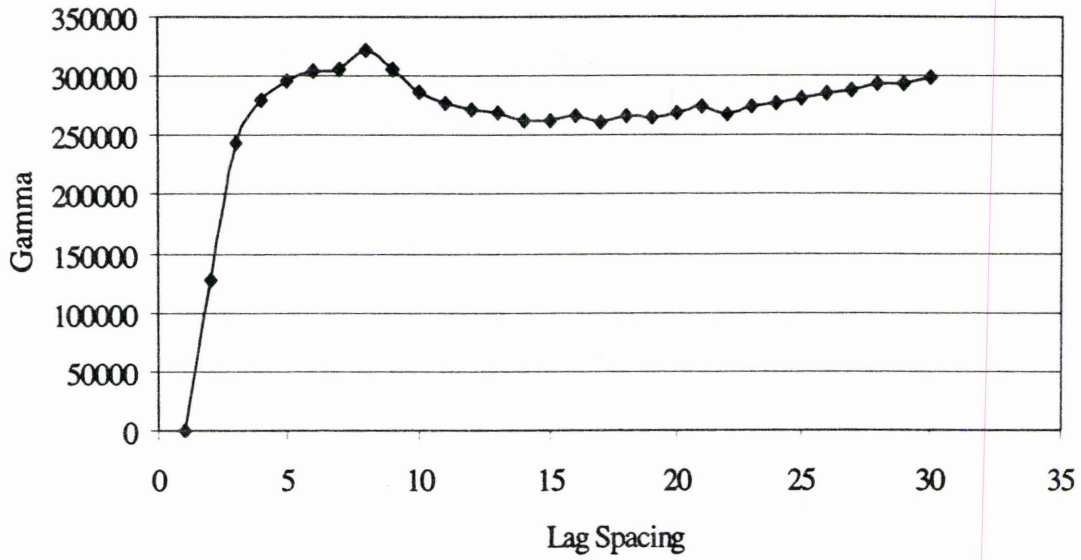
Figure 26. Gamma versus lag spacing for layer 11.

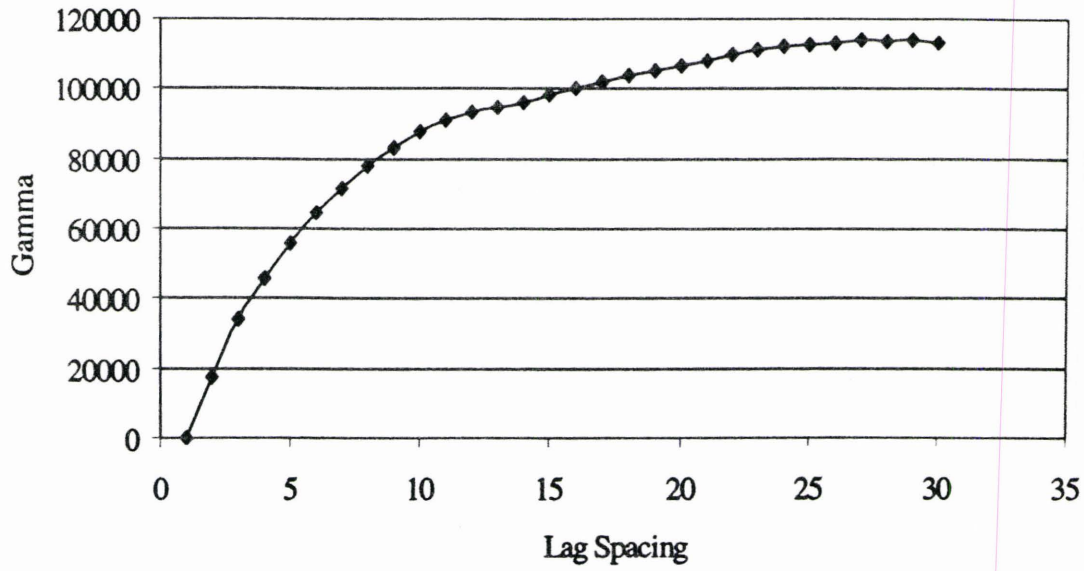Figure 27. Gamma versus lag spacing for layer 12.

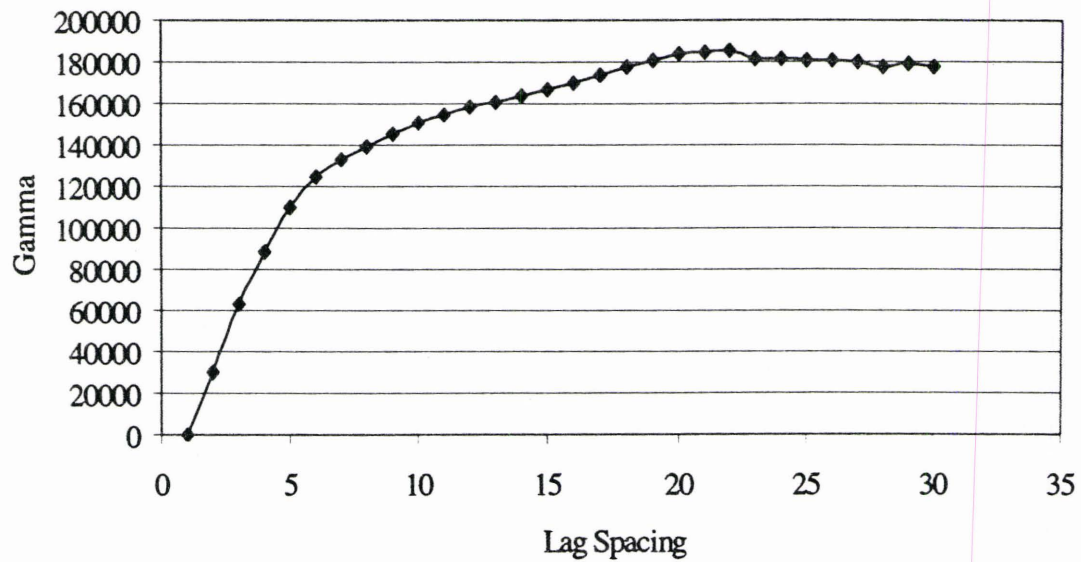Figure 28. Gamma versus lag spacing for layer 13.

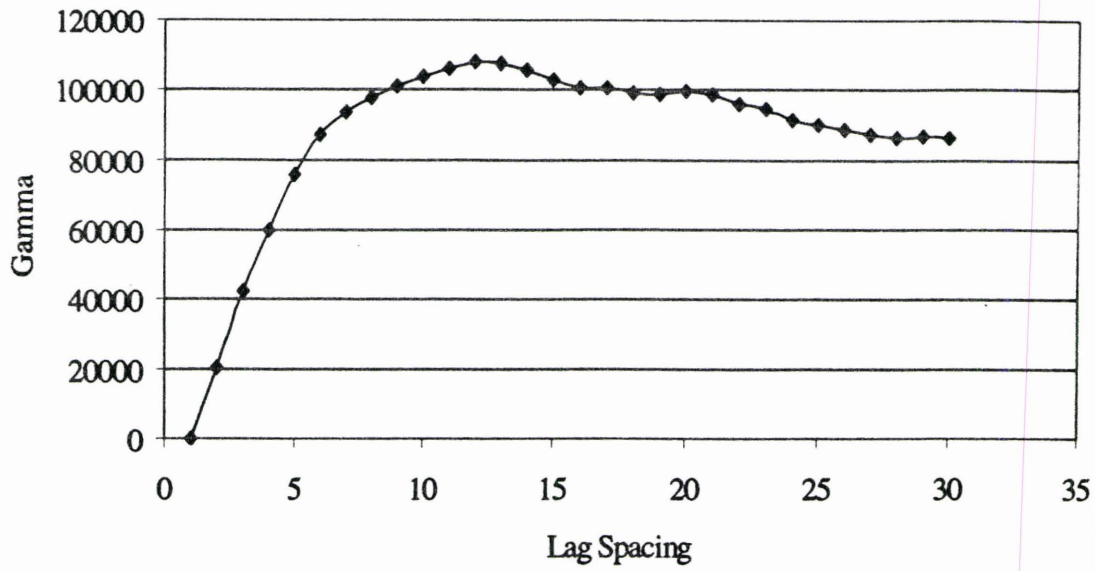Figure 29. Gamma versus lag spacing for layer 14.

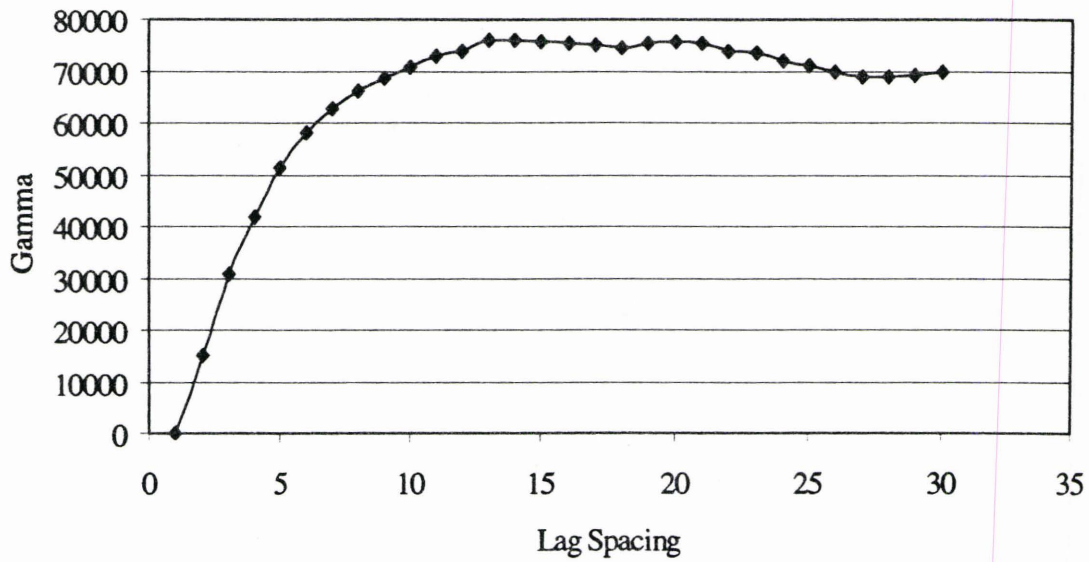Figure 30. Gamma versus lag spacing for layer 15.

Figure 31. Gamma versus lag spacing for layer 16.

REFERENCES

Anderson, M. P., Characterization of geological heterogeneity, In Subsurface Flow and Transport: A Stochastic Approach, eds. G. Dagan, S. P. Neuman, Cambridge University Press, Cambridge, UK, p. 23-43, 1997.

Bair, E. S., Lahm, T. D., Variations in capture-zone geometry of a partially penetrating pumping well in an unconfined aquifer, Ground Water, 34(5), 842-852, 1996.

Bair, E. S., Safreed, C. M., Stasny, E. A., A Monte Carlo-based approach for determining traveltime-related capture zones of wells using convex hulls as confidence regions, Ground Water, 29(6), 849-855, 1991.

Bakker, M., Strack, O. D. L., Capture zone delineation in two-dimensional groundwater flow models, Water Resources Res., 32(5), 1309-1315, 1996.

Borgman, L., Taheri, M., Hagan, R., In Geostatistics for Natural Resources Characterization, part 1, eds. G. Verly et al., Reidel Publ. Co., Dordrecht, The Netherlands, 1984.

Bras, R. L., and Rodriguez-Irurbe, I., Random Functions and Hydrology, Addison-Wesley Publishing Company, Redding, MA, 559 p., 1985.

Byers, E., Stephens, D. B., Statistical and stochastic analyses of hydraulic conductivity and particle size in a fluvial sand, Soil Science Society American Journal, 47, 1072-1081, 1983.

Chan, N., Robustness of the multiple realization method for stochastic hydraulic aquifer management, Water Resources Res., 29(9), 3159-3167, 1993.

Cole, B. E., Silliman, S. E., Capture Zones for passive wells in heterogeneous unconfined aquifers, Ground Water, 35(1), 92-98, 1997.

Davis, J. M., Lohmann, R. C., Phillips, F. M., Love, D. W., Architecture of the Sierra Ladrones Formation, central New Mexico: depositional controls on the permeability correlation structure, Geological Society of America Bulletin, 105, 998-1007, 1993.

Davis, M., Production of conditional simulations via the LU triangular decomposition of the covariance matrix, Math. Geol., 19(2), 91-98, 1987.

Department of Energy, Site-directed subsurface environmental initiative: five-year summary and plan for fundamental research in subsoils and groundwater, DOE/ER-03344/1, Washington, D. C., April 1988.

Desbarats, A. J., Bachu, S., Geostatistical analysis of aquifer heterogeneity from the core scale to the basin scale: a case study, Water Resources Res., 30(3), 673-684, 1994.

Dodak, E. J., The effects of non-linear bromide adsorption on apparent macrodispersivity, Thesis at The University of North Dakota, 1996.

Domenico, P. A., Schwartz, F. W., Physical and Chemical Hydrogeology, 2nd ed., John Wiley & Sons, Inc., New York, NY, 506 p., 1998.

Faybishenko, B. A., Javandel, I., Witherspoon, P. A., Hydrodynamics of the capture zone of a partially penetrating well in a confined aquifer, Water Resources Res., 31(4), 859-866, 1995.

Fetter, C. W., Contaminant Hydrogeology, Prentice-Hall, Inc., Upper Saddle River, NJ, 458 p., 1993.

Franzetti, S., Guadagnini, A., Probablistic estimation of well catchments in heterogeneous aquifers, J. Hydrol., 174(4-6), 149-171, 1996.

Gelhar, L. W., Stochastic Subsurface Hydrology, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.

Gilbert, R. O., Statistical Methods for Environmental Pollution Monitoring, Van Nostrand Reinhold, New York, NY, 1987.

Goggins, D. J., Chandler, M. A., Kacurek, G., Lake, L. W., Patterns of permeability in eolian deposits: Page sandstone (Jurassic), Northeastern Arizona, SPE Formation Evaluation, June, p. 297-306, 1988.

Gorelick, S. M., Sensitivity analysis of optimal ground water contaminant capture curves: spatial variability and robust solutions, in Proceedings, NWWA Conference: Solving Ground Water Problems with Models, National Water Well Association, Dublin, Ohio, p. 133-146, 1987.

Grubb, S., Analytical model for estimation of steady-state capture zones of pumping wells in confined and unconfined aquifers, Ground Water, 31(1), 27-32, 1993.

Gutjahr, A. L., Fast Fourier transforms for random field generation, New Mexico Tech. Project Report, 1989.

Gutjahr, A. L., Bai, Q., Hatch, S., Conditional simulation applied to contaminant flow modeling, Technical Completion Report, Project Number WERC 91-38, 1992.

Gutjahr, A. L., Bullard, B., Hatch, S., Joint conditional simulations and flow modeling, Geostatistics for the Next Century, ed. R. Dimitrikopoulous, Kluwer Academic Publishers, 185-196, 1993.

Gutjahr, A. L., Bullard, B., Hatch, S., Hughson, L., Joint conditional simulations and the spectral approach for flow modeling, Stochastic Hydrology and Hydraulics, March, 79-108, 1994.

Gutjahr, A. L., Hatch, S., Bullard, B., Random field generation using the Fast Fourier transform, (unpublished), 1996.

Hess, K. M., Use of a borehole flow meter to determine spatial heterogeneity of hydraulic conductivity and macrodispersion in a sand and gravel aquifer, Cape Cod, Massachusetts, In Proceedings of the Conference on New Field Techniques for Quantifying the Physical and Chemical Properties of Heterogeneous Aquifers, eds. F. J. Molz et al., National Water Well Association, Dublin, Ohio, p. 497-508, 1989.

Hill, M. C., Preconditioned Conjugate-Gradient 2 (PCG2), a computer program for solving ground-water flow equations, U.S. Geological Survey Water-Resources Investigations Report 90-4048, 1990.

Hoeksema, R. J., Kitanidis, P. K., Analysis of spatial structure of properties of selected aquifers, Water Resources Res., 21(4), 563-572, 1985.

Hufschmied, P., Estimation of three-dimensional, statistically anisotropic hydraulic conductivity field by means of single well pumping tests combined with flow meter measurements, Hydrogelogie, 2, 163-174, 1986.

Javandel, I. C., Tsang, C. F., Capture-zone type curves: a tool for aquifer cleanup, Ground Water, 24(5), 616-625, 1986.

Johnson, N. M., Characterization of alluvial hydrostratigraphy with indicator semivariograms, Water Resources Res., 31(12), 3217-3227, 1995.

Journel, A. G., Huijbregts, C. J., Mining Geostatistics, Academic Press, Inc., New York, NY, 594 p., 1978.

Koltermannn, C. E., Gorelick, S. M., Heterogeneity in sedimentary deposits: a review of structure-imitating, process-imitating and descriptive approaches, Water Resources Res., 32(9), 2617-2658, 1996.

Lee, S., Kitanidis, P. K., Optimal estimation and scheduling in aquifer remediation with incomplete information, Water Resources Res., 27(9), 2203-2217, 1991.

LeFever, R., VARIO: a program for calculating variograms, University of North Dakota, 1997.

Mantoglou, A., Wilson, J. L., The turning bands method for simulation of random fields using line generation by a spectral method, Water Resources Res., 18(5), 1379-1394, 1982.

McDonald, M. G., Harbaugh, A. W., A modular three-dimensional ground-water flow model, U.S. Geological Survey Techniques of Water Resources Investigations, Book 6, chapter A1, 1988.

McKay, D., Gutjahr, A. L., Wilson, J. L., Fast Fourier transform algorithm for random field generation, Poster presentation, AGU Meeting, San Francisco, Ca. Fall, 1988.

Miall, A., Architectural-element analysis: a new method of facies analysis applied to fluvial deposits, Earth-Science Reviews, 22, 261-308, 1985.

Neuman, S. P., Universal scaling of hydraulic conductivities and dispersivities in geologic media, Water Resources Res., 26(8), 1749-1758, 1990.

Neuman, S. P., Generalized scaling permeabilities: validation and effect of support scale, Geophysical Research Letters, 21(5), 349-352, 1994.

Neuman, S. P., On advective transport in fractal permeability and velocity fields, Water Resources Res., 31(6), 1455-1460, 1995.

Neuman, S. P., Zhang, Y-K., Levin, O., Quasilinear analysis, universal scaling, and Lagrangian simulation of dispersion in complex geological media. In Dynamics of Fluids in Hierarchical Porous Media, ed. J.H. Cushman, Academic Press, San Diego, CA, p. 349-391, 1990.

Osiensky, J. L., Williams, R. E., Potential inaccuracies in MODFLOW simulations involving the SIP and SSOR methods for matrix solution, Ground Water, 35(2), 229-232, 1997.

Pardo-Iguzquiza, E., Chica-Olmo, M., The Fourier integral method: an efficient spectral method for simulation of random fields, Math. Geol., 25(2), 177-217, 1993.

Pollack, D. W., Documentation of computer programs to compute and display pathlines using results from the U.S. Geological Survey modular three-dimensional finite-difference ground-water flow model, U.S. Geological Survey Open-file Report 89-381, 118 p., 1989.

Rehfeldt, K. R., Gelhar, L. W., Southard, J. B., Dasinger, A. M., Estimates of macrodispersivity based on analyses of hydraulic conductivity and variability at the MADE site, EPRI EN-6405, Project 2485-5, Palo Alto, CA: Electric Power Research Institute Report, 1989.

Robins, M. J. L., Gutjahr, A. L., Sudicky, E. A., Wilson, J. L., Cross-correlated random field generation with the direct Fourier transform method, Water Resources Res., 29(7), 2385-2397, 1993.

Schafer, D. C., Determining 3D capture zones in homogeneous, anisotropic aquifers, Ground Water, 34(4), 628-639, 1996.

Smith, L., A stochastic analysis of steady state groundwater flow in a bounded domain, Ph.D. thesis, University of British Columbia, 1978.

Smith, L., Freeze, R. A., Stochastic analysis of steady state groundwater flow in bounded domain, 1: one dimensional simulations, Water Resources Res., 15(6), 1521-1528, 1979a.

Smith, L., Freeze, R. A., Stochastic analysis of steady state groundwater flow in bounded domain, 2: two dimensional simulations, Water Resources Res., 15(6), 1543-1559, 1979b.

Smith, L., Spatial variability of flow parameters in a stratified sand, Mathematical Geology, 13(1), 1-21, 1981.

Sudicky, E. A., A natural gradient experiment on solute transport in a sand aquifer: spatial variability of hydraulic conductivity and its role in the dispersion process, Water Resources Res., 22(13), 2069-2082, 1986.

Thompson, A. F. B., Ababou, R., Gelhar, L. W., Implementation of the three-dimensional turning bands random field generator, Water Resources Res., 25(10), 2227-2243, 1989.

Varljen, M. D., Shafer, J. M., Assessment of uncertainty in time-related capture zones using conditional simulation of hydraulic conductivity, Ground Water, 29(5), 737-748, 1991.

Wagner, B. J., Gorelick, S. M., Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: from data to design, Water Resources Res., 25(10), 2211-2225, 1989.

Weber, D., Probability of plume interception using conditional simulation of hydraulic head and inverse modeling, Math. Geol., 23, 291-340, 1991.

Woodbury, A. D., Sudicky, E. A., The geostatistical characteristics of the Borden aquifer, Water Resources Res., 27(4), 533-546, 1991.

Young, S. C., Herweijer, J., Benton, D. J., Geostatistical evaluation of a three-dimensional hydraulic conductivity field in an alluvial terrace aquifer. In Fifth Canadian/American Conference on Hydrogeology Proceedings, National Water Well Association, Dublin, Ohio, p. 116-137, 1990.

Zimmerman, D. A., de Marsily, G., Gotway, C. A., Marietta, M.G., Axness, C. L., Beauheim, R. L., Bras, R. L., Carrera, J., Dagan, G., Davies, P. B., Gallegos, D. P., Galli, A., Gomez-Hernandez, J., Grindrod, P., Gutjahr, A. L., Kitanidis, P. K., Lavenue, A. M., McLaughlin, D., Neuman, S. P., RamaRao, B. S., Ravenne, C., Rubin, Y., A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow, Water Resources Res., 34(6), 1373-1414, 1998.

Zlotnik, V. A., Effects of anisotropy on the capture zone of a partially penetrating well, Ground Water, 35(5), 842-847, 1997.