STOCHASTIC MODELING OF WATER FLOW THROUGH A VARIABLY-SATURATED,

HETEROGENEOUS FIELD AT IDAHO NATIONAL LABORATORY:

UNCERTAINTY ANALYSIS

By

LIMIN YANG

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING

WASHINGTON STATE UNIVERSITY
College of Engineering and Architecture

August 2005

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of LIMIN YANG find it satisfactory and recommend that it be accepted.

_____

Chair

_____

_____

_____

ACKNOWLEDGMENTS

STOCHASTIC MODELING OF WATER FLOW THROUGH A VARIABLY-SATURATED,

HETEROGENEOUS FIELD AT IDAHO NATIONAL LABORATORY:

UNCERTAINTY ANALYSIS

Abstract

by Limin Yang, M.S.
Washington State University
August 2005

Chair: Joan Q. Wu

Water flow through variably-saturated, heterogeneous field at the Idaho Nuclear

Technology and Engineering Center (INTEC), Idaho National Laboratory (INL), which has been

used to receive, store, and process spent nuclear fuel, is of great concern. Adequate flow models

that can be used for reliable predictions are needed. In general, the performance and accuracy of a

flow model depend largely on the accuracy of input data, which is affected by the spatial

heterogeneity of geological materials (soil, rock) and the density of field samples.

The goal of this study was to examine the uncertainty associated with water flow through

the vadose zone at the INTEC. Specific objectives were (i) to model the spatial distribution of

basalt and the interbedding sediments using indicator kriging, (ii) to analyze the uncertainty in

water flow as a function of the spatial distribution of sediment interbedding by applying

sensitivity analysis on several key physical parameters (geological media classification; relative

position of water sources; geostatistical model parameterization), and (iii) to assess uncertainty in

water flow as affected by local random variability of the geological material using conditional

simulation.

Indicator kriging was use to characterize the subsurface geological media using specified probability cutoff. A two-component exponential empirical variogram model was determined based on the borehole data from the study site. TOUGH2, a comprehensive software package for three-dimensional numerical simulation of transport processes in porous and fractured media was applied to simulate water flow through variably-saturated, heterogeneous fields at study site based on the kriged stratigraphy. Multiple prediction runs were carried out to examine the uncertainty in subsurface water flow regime due to the uncertainty of (i) geological media (basalt and sediments) categorization, (ii) the relative location of water source to the impeding sediment interbeds, and (iii) geostatistical model parameters, specifically, the ranges of variogram model. Additionally, a total of 30 conditional simulations were made to evaluate the effect of random local variability on the water flow.

Results from the flow modeling indicate that all tested parameters affect the variably-saturated water flow to varying degrees. The uncertainties due to different geological media classifications were most sensitive. Similarly, the uncertainty due to uncertainty of relative locations of water source to interbeds was also significant. When the adjustment to the range values of the variogram model was minor, the resultant impact on water flow was not substantial. However, if the adjustment is significant, the impact may be considerable. Local random variability of the geological media may have great influence on certain characteristics of the water flow regime, in particular, the positions of flow peaks across flow bottom boundary as well as the peak values.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

Water flow and contaminant transport through the vadose zone are of great concern in nuclear and hazardous waste disposal (Evans et al., 2001). Computer modeling of vadose-zone transport processes has been recognized as an efficient means for obtaining a better understanding of the hydrologic systems in question, evaluating various remedial alternatives, and providing a rationale for developing monitoring plans (Fogg et al., 1995; Kramer and Cullen, 1995).

By and large, hydrologic model performance depends heavily on the accuracy of input data, which are used to define the hydrologic and chemical transport parameters. Uncertainty in the definition of these parameters in turn causes uncertainty in model predictions (Gureghian and Sagar, 1993; Bagtzoglou et al., 1994). Spatial heterogeneity of geological materials (soil, rock) and their hydraulic and transport properties are natural phenomena (Russo et al., 1981; Vieira et al., 1981; Rautman et al., 1993). The ability to properly characterize the spatial variability is crucial to parameterizing hydrologic models accurately. However, this ability is often hampered by the limited amount of sample data (soil cores or borehole logs) available.

Uncertainty and sensitivity analysis of subsurface water flow and solute transport has been attempted in a number of studies. Nichols et al. (1993) applied Monte Carlo techniques to Richards' equation to analyze the uncertainty in water travel time through the unsaturated, non-welded zeolitic layers at Yucca Mountain, Nevada, as a function of the uncertainty in key input parameters. They found that the most important parameters affecting the water travel time were the recharge rate and matrix porosity. Samuel and Cawlfield (1999) performed an uncertainty analysis of two-dimensional unsaturated flow and solute transport by applying the first-order

1

reliability method to Richards' equation. Their results indicated that saturated water content was the most crucial variable. Gureghian and Sagar (1993) presented a two-dimensional sensitivity and uncertainty analysis of leakage through a hypothetical nuclear waste repository in a layered and fractured, unconfined aquifer. They used the first-order second-moment approach in combination with Lagrangian method to estimate the uncertainty and sensitivity of the steady-state solutions of the ground-water flow. Both hydraulic conductivity and porosity were critical factors governing the water flow.

At the Idaho National Laboratory (INL), a US Department of Energy (DOE) complex located in Idaho Falls, ID, cleanup of the legacy nuclear wastes generated from weapons production and energy research are currently underway. These nuclear wastes have been buried near surface, released in spills, and leaks to the subsurface that is composed of several hundred meters of sediments and basalts. Migration of the nuclear wastes through the vadose zone over time to cause ground-water contamination is the concern.

The INL is situated in south-central Idaho on the Eastern Snake River Plain (ESRP), which is composed of multiple-layered, plains-style, late-Cenozoic basalt flows over a rhyolitic basement (Greeley, 1982; Malde, 1991). The volcanism in the ESRP has been episodic, distinguished by lava flows emplaced over relatively short periods with long periods of volcanic quiescence. Several volcanic events have left multiple volcanic rift zones as characterized by aligned vents, dikes, and fissures in the vicinity of the INL (Knutson et al., 1990). These rift zones are likely sources of basalt flows at the INL (Knutson et al., 1990). During the volcanic quiescence, loess, clay, silt, sand and gravel were deposited on the top of the lava to form interbeds (INEEL, 2003). The surface of the plain near the INL is primarily composed of alluvial, lacustrine, and eolian deposits (Kuntz et al., 1990).

The Idaho Nuclear Technology and Engineering Center (INTEC), covering roughly 0.85-km^2, is where fuel reprocessing took place (Fig. 1). The INTEC was primarily used to receive, store, and process spent nuclear fuel from prototype, commercial, and naval reactors. High-level radioactive liquid wastes generated by processing were solidified by calcination and are stored in solid-form in stainless steel bins. Low-level radioactive liquid wastes were discharged through an injection well to the underlying Snake River Plain Aquifer from 1953 to 1984. From 1984 to 1993, these low-level liquid wastes were discharged to an infiltration pond. Since 1993, radioactive liquid wastes have been concentrated by evaporation and stored for disposal and only industrial waste water has been discharged to the pond. During reprocessing of the spent fuel, a number of leaks and spills occurred resulting in small volumes of highly contaminated sediments within the facility. Radionuclides of concern for ground-water contamination at the facility are strontium-90 and technetium-99 (Schafer et al., 1997).

During the past few decades, more than 140 wells have been drilled in and around the INTEC to obtain information of the spatial distribution of basalt flows and sedimentary interbeds (INEEL, 2003). The subsurface geology at the INTEC is highly complex, which is likely the result of the occurrence of hundreds of basalt flows combined with inferred vent corridors associated with the rift zones (Anderson et al., 1999; S.R. Anderson, USGS, Pocatello, ID, personal communication, 2003). Consequently, the spatial variation of the hydraulic properties at INTEC is extremely large, with the permeability values differing by up to six orders of magnitude between basalt and sediments (Schafer et al., 1997; Anderson, 1999).
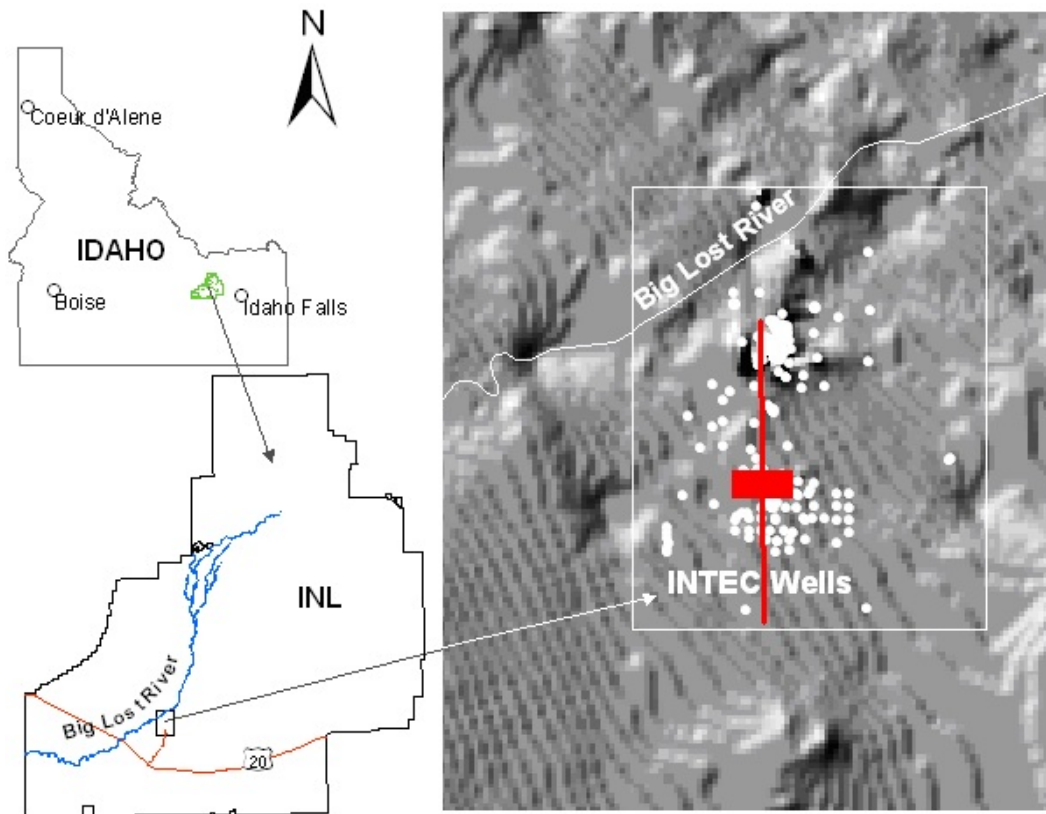
Figure 1. Site map of INTEC, INL.

Several subsurface modeling studies have been conducted at the INTEC. Schafer et al. (1997) developed a three-dimensional, variably-saturated model to evaluate vadose zone water flow and solute transport in support of the assessment of remediation alternatives. In their model, a layered structure of geological materials was assumed and 13 sediment interbeds were combined to form five effective sediment layers. The thicknesses of these layers were estimated from kriged thicknesses of the geological units determined from borehole logs. The flow through the basalt matrix was regarded as insignificant in comparison to flow through the fractures, and only fractured basalt was considered as an anisotropic, single-porosity medium with a horizontal hydraulic conductivity of 9×10^{-6} m s^{-1} (9,000 millidarcy of permeability) and a vertical hydraulic conductivity of 3×10^{-7} m s^{-1} (300 millidarcy of permeability). Recent field data collected at the INTEC have further confirmed the significant spatial variability of the hydraulic properties of the geological media, manifested by the highly irregular presence of saturated areas or the perched aquifers in the vadose zone (INEEL, 2003). Stratigraphic studies also show that, in general, there exist more, thinner, and less continuous sediment interbeds at the INTEC than at other part of the INL (Schafer et al., 1997; S.R. Anderson, USGS, Pocatello, ID, personal communication, 2003).

Contaminant transport through the vadose zone at the INTEC is a serious problem as the radionuclides (e.g., strontium-90, tritium, technetium-99) that are present in the perched waters may migrate along preferential paths and eventually reach the ground-water body. The major sources of these contaminants are the spills, leaks and infiltration ponds, which have led to the formation of the upper and lower perched waters over time. Understanding how water flow and contaminant movement are affected by the spatial heterogeneity of the geological materials and their hydraulic and transport properties at the INTEC is critical to establishing future monitoring and remediation strategies.

The main purpose of this study was to examine the uncertainty associated with water flow through the variably-saturated, heterogeneous field at the INTEC. Specific objectives were (i) to model the spatial distribution of basalt and the interbedding sediments using indicator kriging, (ii) to analyze the uncertainty in water flow as a function of the spatial distribution of sediment interbedding by applying sensitivity analysis on several key physical parameters (geological media classification; relative position of water sources; geostatistical model parameterization), and (iii) to assess uncertainty in water flow as affected by local random variability of the geological material using conditional simulation.

The deterministic code chosen for this study is the US Department of Energy's TOUGH2-EOS9 code. TOUGH2 is a comprehensive software package for three-dimensional numerical simulation of transport processes in porous and fractured media (Pruess et al., 1999). The EOS9 module is an integral finite-difference program for solving Richards' flow equation as well as the solute and heat transport equations. The code has been used in numerous simulation applications that involve multi-phase, transient and variably-saturated flow problems (Eaton et al., 1996; Rutqvist et al., 2002).

CHAPTER TWO

METHODOLOGY

2.1. Geostatistical characterization

Based on the analysis of the borehole data and the general geological characteristics of the INTEC site from the literature, vertical discretization of the log data was made at a 0.3-m interval, which represents the smallest thickness of the interbeds that can be resolved. This vertical discretization yielded a total of 33,420 samples for a domain of roughly 2,420 m by 1,710 m encompassing the entire INTEC and extending to the depth of 229 m. The characteristics of the borehole logs as well as the discretized samples are summarized in Table 1. Basalt, in combination with the interbedding sediment and alluvium that comprises the surface layer at the INTEC, accounted for more than 92% of the total samples while rubble and mixed samples, together with missing data, accounted for less than 8%. Consequently, as a first step, all the samples were generalized as basalt (including basalt, rubble and mixed) and sediment (including both the interbedding sediment and the surficial alluvium).

Among the variogram model we selected the exponential model. The binary system (basalt vs. sediment) excludes both the liner and Gaussian type variogram models. A nested empirical variogram model was developed by using an optimization utility code adapted from GSLIB (Deutsch et al., 1997) to represent both short- and long-range structures reflected in the empirical variograms,

$$\gamma(h) = Nugget + C_1 \mathrm{Exp}(-\frac{h}{a_1}) + C_2 \mathrm{Exp}(-\frac{h}{a_2}) \tag{1}$$

where $\gamma(h)$ represents the degree of spatial continuity of data points separated by vector h, and a_1

Table 1. Characteristics of the borehole log and discretized samples for geostatistical analysis[a].

	Number of Samples	Percentage of Total Samples
Well depth < 50 m	16399 (161)[b]	49.1
Well depth 50–100 m	7007 (8)	21.0
Well depth 100–150 m	5690 (15)	17.0
Well depth 150–229 m	4324 (26)	12.9
Total	33420 (210)	100
Basalt	21836	65.3
Interbedding sediment	6510	19.5
Alluvium	2520	7.5
Rubble	172	0.5
Mixed	1278	3.8
Missing data[c]	1105	3.3
Total	33420 (210)	100

[a] All borehole log was discretized at a 0.3-m interval.
[b] Shown in parentheses are the numbers of wells with corresponding depths.
[c] Missing data were for those intervals within a well, for which there were no characterization.

and a_2 are the effective ranges for the two exponential models. For this study, a convention was set such that a low (close to zero) probability value would indicate the likely occurrence of sediment while a high probability value (approaching 1) would represent a high likelihood of the occurrence of basalt.

Table 2 lists the detailed parameters of this empirical variogram model. The two exponential components of the model have different effective ranges reflecting the anisotropic structure of the geological media.

Under the assumption of second-order stationarity, ordinary kriging (Deutsch et al., 1997) was performed on the indicator variable to obtain estimates of the probability of the occurrence of geological media (basalt vs. sediment) for individual grid points. The threshold value of 0.5 was taken as the cutoff, and a basalt-sediment binary geological stratigraphy was then generated from the kriged occurrence probability map. For the kriging interpolation, a 20 m by 20 m by 0.3 m grid was chosen, which was appropriate for the distribution of boreholes within the study domain of INTEC and compatible with the grid configuration used in the subsequent water flow simulation. Cross-validation and jackknife validation were performed and the results were regarded as satisfactory (Table 3). Subsequently, the alluvium that is present primarily within the 19 m below the ground surface was separated from the sediment interbeds. Within the alluvium, however, there exists a wedge of basalt between 6 m and 12 m depths.

2.2. Development of the numerical model

2.2.1. Hydrogeologic setting

A north-south vertical transect crossing over a percolation pond (roughly 150 m by 150 m), which, together with another pond of the same size, formed the major water source at the INTEC from 1984 to 1993, was chosen for this study (Fig. 2). The cross-section is 2,000 m long

Table 2. Parameters of the empirical variogram model.

Nugget	C_1	C_2	Structure	Rotation[a], deg			Range[b], m		
				x	y	z	x	y	z
0.074	0.041	0.539	1	9.22	−3.13	18.63	472.53	297.76	17.04
			2	0.03	0.11	3.51	2415.08	1709.65	12.41

[a] Rotations follow the GSLIB definitions.
[b] The ranges follow the GSLIB definition; i.e., they are the effective ranges.

Table 3. Cross validation and jackknife validation.

	Mean Error, %	Variance
Cross Validation[a]	1.3	0.034
Jackknife Validation[b]	7.8	0.162

[a] In cross validation actual data are dropped one at a time and re-estimated from some of the remaining neighboring data. (Deutsch et al., 1997)

[b] Jackknife validation was made with 20 percent of data taken out.
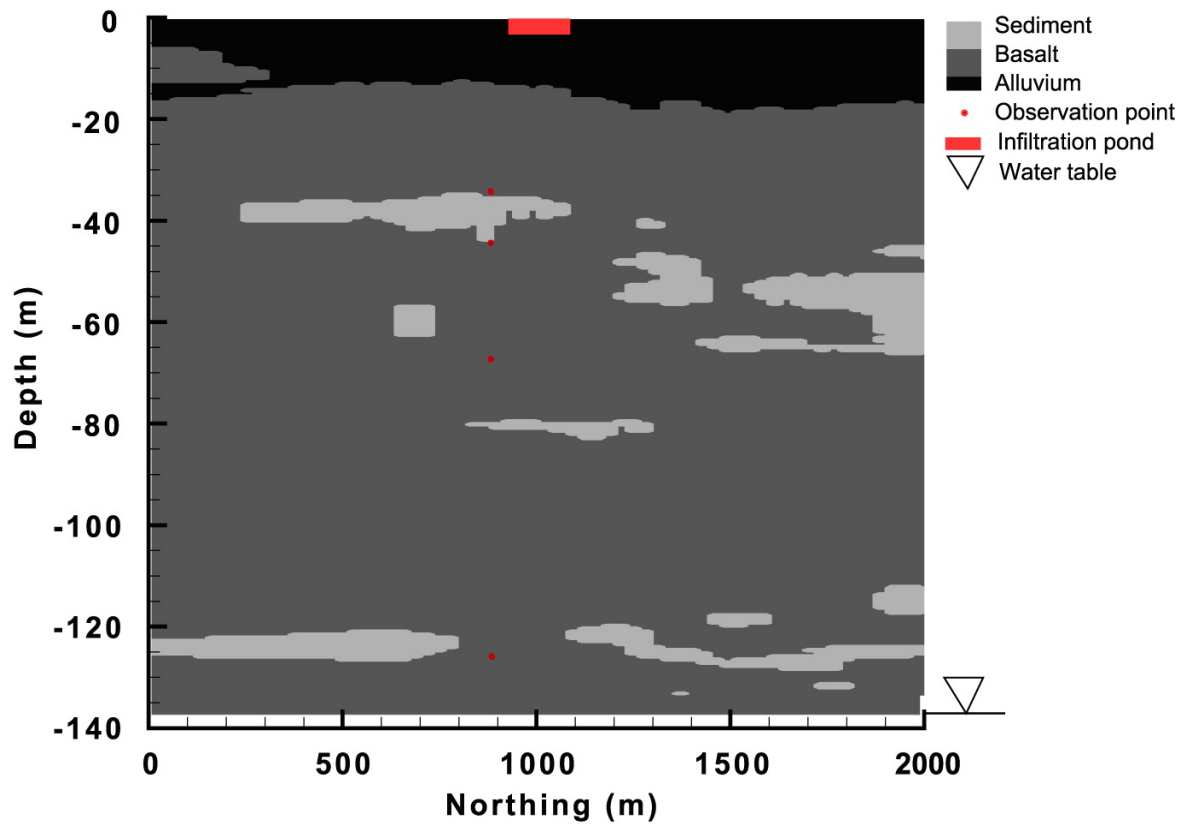
Figure 2. Kriged cross-section at INTEC.

by 137 m deep extending to the ground-water table. The long-tern average annual precipitation at

the INTEC is 0.22 m, and the estimated effective evapotranspiration is roughly half of the annual

precipitation, or about 0.11 m (INEEL, 2003). The other major water source is the Big Lost River

to the northwest of the INTEC, which had no strong hydraulic connections with the model

domain from the results of preliminary model runs.

2.2.2. Modeling grid for base run

TOUGH2, which is a 3-dimensional code, was used in a 2-dimensional form. The third

dimension, i.e., the thickness of the cross-section, was set to 1 m. An adaptive-size mesh was

used in the TOUGH2 code to better characterize the sediment-basalt interfaces as well as the

model boundaries. A number of utility codes were developed for automatically defining the

integral finite-deference meshes under various model runs. A major criterion in determining the

finite-difference mesh was to retain an adequate ratio (e.g., 2) of the sizes of the adjacent grid

cells following Courant (1928). For the purpose of comparison, we set up a baseline simulation,

in which all model inputs represent the "best available" estimates, the percolation pond was

located at its original location, and the probability cutoff value of 0.5 was used to classify the

subsurface media. For the base run, 41,076 grid cells were used with the smallest grid size being

0.3 m by 0.3 m and the largest 15.1 m by 3.0 m. All interfaces between basalt and sediment as

defined by kriging were retained, and for grids smaller than the kriging grid size of 20 m, the

geological media were assigned by simple discretization.

2.2.3. Boundary and initial conditions

A Neumann (second-type) boundary with time-constant flux was used for the top

boundary of the model domain. For those grids across the percolation pond, the influx was

1.129×10^{-3} kg m^{-2} s^{-1} or 2200 t d^{-1}, representing the average flux rate (per unit area) of the water

infiltrating below the two ponds (INEEL, 2003). The flux rate for a 1-m width strip of the

percolation pond, instead of the total flux from the two ponds, was used as the input flux

considering the relative small lateral dimension of 1 m compared to the total width of 300 m

spanned by the two ponds. For the remaining top grids, the influx was 3.171×10^{-7} kg m^{-2} s^{-1}

corresponding to a net recharge from the long-term average annual precipitation of 0.11 m.

The bottom boundary was a constant-head Dirichlet (first-type) boundary representing the

ground-water table, of which the seasonal fluctuation is insignificant (INEEL, 2003). Preliminary

model runs and field observations (INEEL, 2003) showed that the lateral flux across a 2000-m

modeling domain as affected by the influx from the percolation ponds was negligible. Therefore,

no-flux boundaries were used for both sides of the model domain.

The initial conditions were obtained by running the model until the gravity-capillary

equilibrium was reached in order to mimic the steady state after the water content redistribution

in the subsurface system was complete.

2.2.4. Model parameter adjustment

The key model parameters were first defined following relevant literature (Schafer et al.,

1997; INEEL, 2003). These parameters were further adjusted. Specifically, the predicted water

contents at the locations where observation wells exist were compared with the monitored water

contents. Horizontal and vertical hydraulic conductivities for all three major media, namely,

alluvium, basalt and sediment, were then adjusted. For example, horizontal hydraulic

conductivity was increased relative to the vertical hydraulic conductivity if the simulated

saturation degree at the observed points were lower than the observations. Additionally, from the

chemistry analysis of water samples taken from the perched aquifer, the likely lateral extent of

the migration of the percolated water could be characterized, which further helped to restrain the

values of the horizontal hydraulic conductivities. The adjusted hydraulic conductivities as well as other parameters in the van Genuchten equations for water release characteristics (van Genuchten, 1980; Pruess, 1999), are listed in Table 4. Note that the hydraulic conductivities of the alluvium are the recently reported values (INEEL, 2003), while other properties of this material are from previous literature (Schafer et al., 1997) for which no new data are available.

2.3. Model uncertainty assessment

A major focus of this study was to elucidate the uncertainty in water flow simulations as a consequence of uncertainty in subsurface characterization. Four groups of simulations were preformed using the TOUGH2 code, to evaluate the prediction uncertainty as a result of the uncertainties in (i) categorization of the geological media, (ii) location of the interbeds, (iii) geostatistical model parameters, and (iv) conditional simulations.

2.3.1. Uncertainty in basalt and sediment categorization

There was inconsistency in classifying the individual geological materials during the borehole logging. For instance, the specification of sediment and cinder was not always consistent as both materials are relatively loose and can be difficult to distinguish. As a result, certain cinder samples were categorized as basalt and others as sediment. The uncertainties in media categorization were further increased in this study by generalizing the five different categories of materials into three classes of alluvium, interbedding sediment and basalt. In order to evaluate the impact of these uncertainties, we varied the cutoff values for the kriged occurrence probability map and examined the generated binary subsurface as affected by the cutoff value. The base line cutoff value was 0.5, meaning that a probability value less than 0.5 would indicate the occurrence of sediment whereas others represent the occurrence of basalt. Other tested cutoff values were 0.1, 0.3, 0.7, and 0.9. For each cutoff, a finite-difference grid

Table 4. Model hydraulic input parameters after adjustment.

Media	θ^a (m^3 m^{-3})	θ_r (m^3 m^{-3})	λ	α (m^{-1})	ρ (kg m^{-3})	K_{sh} (m s^{-1})	K_{sv} (m s^{-1})
Basalt	0.05	0.001	0.45	0.3	2518	7.70E−05	2.60E−06
Interbed	0.277	0.0175	0.297	0.258	1560	6.70E−07	6.70E−07
Alluvium	0.277	0.0175	0.297	0.258	1560	4.10E−04	4.10E−04

[a] The van Genuchten equation parameters include the curve fitting parameters λ and α, saturated soil moisture θ, and the residual soil moisture θ_r. ρ is dry bulk density, and K_{sh} and K_{sv} are horizontal and vertical saturated hydraulic conductivities, respectively.

with the distribution of alluvium, basalt and sediment was generated and the TOUGH2 code was

run using the same refined hydraulic parameters as in the base run. Further, to understand the

prediction uncertainty and sensitivity to the uncertainty of the media hydraulic parameters, an

additional simulation using a set of different hydraulic parameters for the sediment interbeds was

performed. Although the focus of this study was not on a systematic analysis of the influence of

hydraulic parameters on flow and solute transport, this additional simulation may provide insight

on the extent to which the hydraulic properties of basalt and sediments affect the flow conditions

and thus the direction of future research efforts. For all the TOUGH2 runs, specific outputs

evaluated were (i) the characteristics (peak and spread) of the flux across the bottom boundary;

(ii) travel time of water from the center of the major water source, i.e., the percolation pond, to

the ground water; (iii) the time for the system to reach equilibrium; and (iv) the cumulative mass

of water leaving the system at certain times (0.2, 0.5 years) and upon reaching equilibrium.

2.3.2. Uncertainty of the location of interbeds

Subsurface geometric uncertainty may also result from the difficulty in adequately

determining the precise location of the sediment interbeds due to limited borehole data. Yet the

uncertainty in geometric location of these interbeds may be critical to flow and transport

modeling. The uncertainty in the location of interbeds may be transformed to the uncertainty in

source location. Therefore, we evaluated this uncertainty by shifting the major water source, i.e.,

the percolation ponds, 150 m left and right, respectively, to the original position. The specific

outputs were as previously described in section 2.3.1.

2.3.3. Uncertainty of the geostatistical model parameters

In general, the use of estimated or optimized mathematical parameters, such as the range

value (a surrogate for correlation length) in the empirical semi-variogram model, can also

contribute to predictive uncertainty. In principle, the range values reflect the spatial extent or continuity of geological media. In this study, we evaluated the effect of the uncertainty in the estimated horizontal and vertical ranges on water flow simulation. The sensitivity of the model results to these parameters was analyzed based on the following formula

$$S = \frac{\Delta y}{\Delta x} \qquad (2)$$

where S is the sensitivity with respect to the parameter x, Δy is the change in a specific output of interest, Δx is the change in the parameter, i.e., the estimated horizontal or vertical range.

2.3.4. Conditional simulation

Sequential simulation is a commonly used conditional simulation technique. It is especially preferred for describing the effect of local variability in studies including flow simulation (Deutsch et al., 1997; Gomez-Hernandez et al., 1997). In essence, sequential indicator simulation applies a Monte Carlo procedure to spatially correlated data. It differs distinctly from estimation or indicator kriging even though it utilizes kriging, but for different purposes. Indicator kriging provides a single probability estimate for unsampled locations that are weighted averages of neighboring (0 or 1 coded) samples. If performed well, indicator kriging results in a single map that is correct on average, meaning the sample data mean is equal to the mean of all the kriged estimates. However, kriged maps do not honor the sample variograms due to kriging's smoothing effect. Indicator simulation, in contrast, not only honors the sample data and mean as does kriging, but it also honors the sample variograms. Indicator simulation generates multiple, equally plausible probability maps of the phenomenon of interest. At each unsampled location, these multiple simulated probabilities comprise a conditional probability distribution that is a quantification of the uncertainty what the samples and variograms could generate.

Specifically, indicator simulation begins by following a random path through the unsampled nodes. At the first simulation node, the neighboring samples are used and simple kriging is performed. This probability estimate defines the probability that the location belongs to the class coded 1. By default, it also defines the probability that the location belongs to the class coded 0 since the probabilities must sum to 1. A uniform random number is then generated. The class (0 or 1) corresponding to the generated random number is determined based on a specified cut off, and the location is assigned that code value until all remaining locations are simulated. For the second simulation location the first simulated value is treated as an actual, hard sampled value. For all subsequent simulations along the random path, all actual samples as well as any previously simulated values are used to condition the remaining simulation values. This process is repeated to create multiple simulated maps.

In our study, sequential indicator simulations (Deutsch et al., 1997) were performed and 30 realizations were generated. For ease of comparison and mathematical aggregation of the results from these realizations, the size of the grids in the horizontal direction was fixed and the total number of grids in this direction was 163 (as compared to 164 in the base run). The 30 realizations were then used in subsequent uncertainty analysis of flow simulation.

CHAPTER THREE

RESULTS AND DISCUSSION

3.1. Model uncertainty due to the uncertainty in basalt and sediment categorization

Using different probability cutoff values for differentiating sediment and basalt yielded different cross-sections (Fig. 3). Compared to the cross-section obtained using the cutoff value of 0.5, the cross-section with a cutoff of 0.1 contained much greater amount of basalt while the cross-section with a cutoff of 0.9 contained substantially more sediment. The categorization of the geological material, in turn, had a remarkable influence on the flow regime. The resultant spatial distribution of water saturation in the subsurface was different in each case (Fig. 4), as readily demonstrated by the different sizes of the areas and the different locations of those areas where water content is near or at saturation (perched water). The characteristics of the flux across the bottom boundary also exhibited differences in terms of both the shape of the flux curves and the time for the water flow to start to break through and to reach equilibrium (Fig. 5). More irregularities in the flux curves would be developed corresponding to more complex geometric configuration of the geological materials. The earliest time for noticeable flux to occur at the bottom boundary generally increases in the order of increasing cutoff value, or increasing amount of sediment interbeds. Similarly, the time to equilibrium increases with the cutoff value and ranges 308–407 years (Table 5). Quantitative analysis showed that the output sensitivity S to cutoff values is greater than 150% for some of the indices of Table 5. The results are regarded as reasonable because the increased presence of sediment interbeds that are of lower permeability certainly will cause more resistence to water flow and create more complex flow conditions, increasing the time for the system to reach equilibrium.
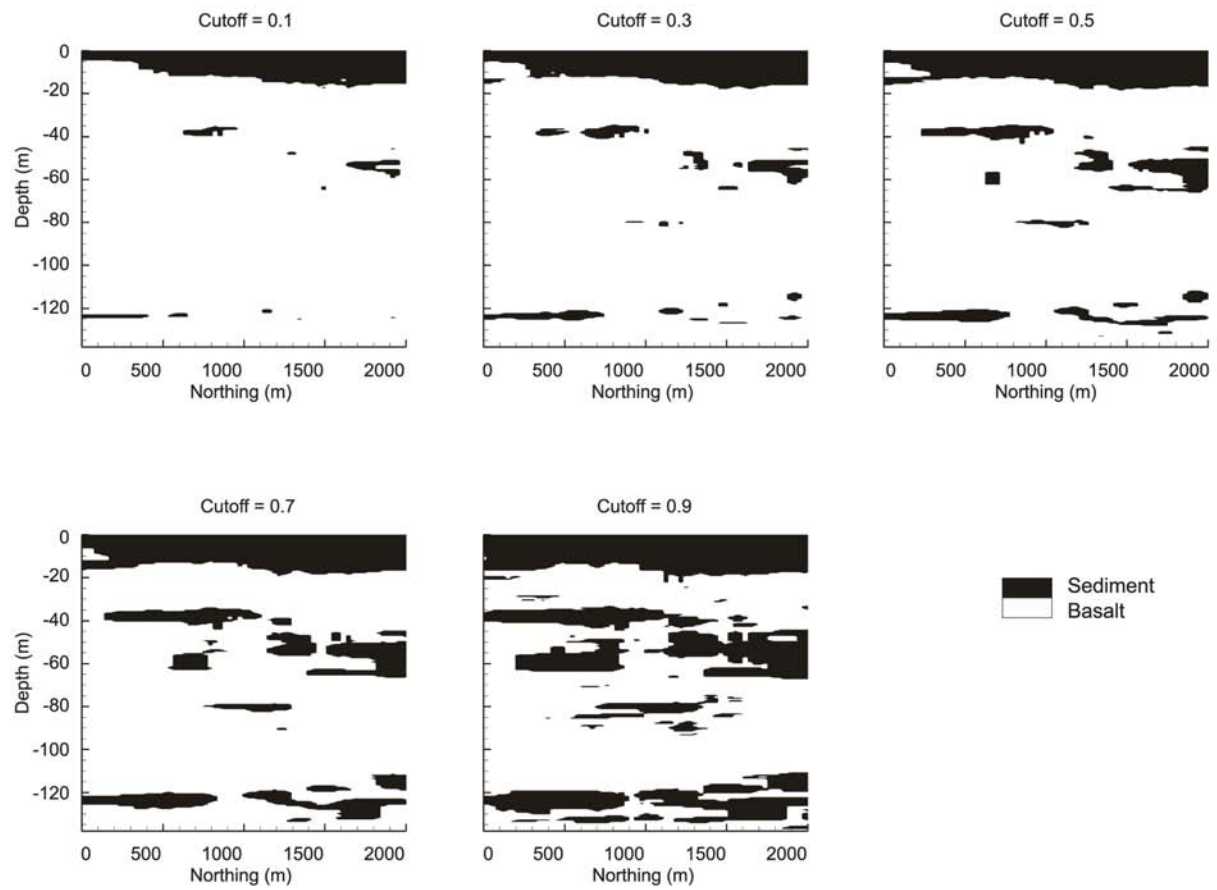
Figure 3. Cross-section maps for different cutoff values.
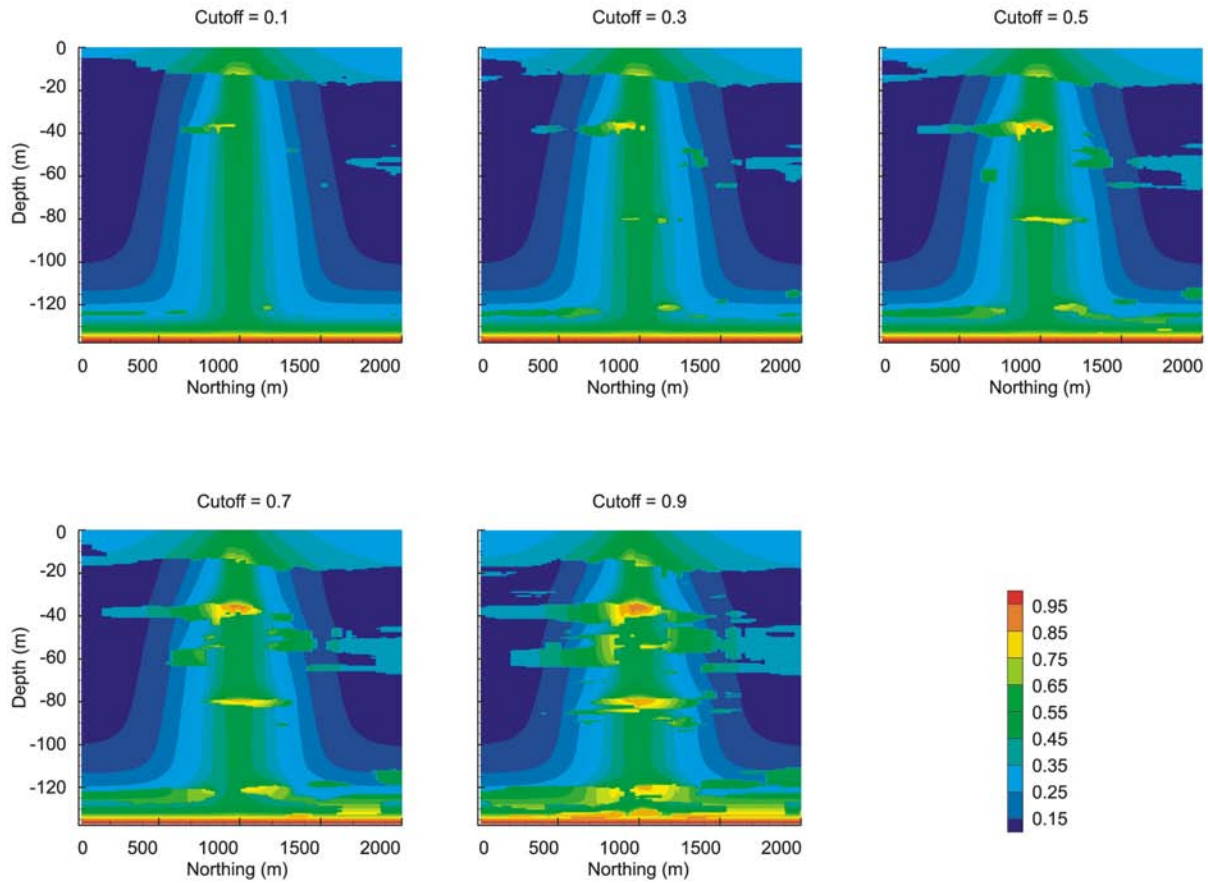
Figure 4. Subsurface moisture contour plot for different cutoffs.
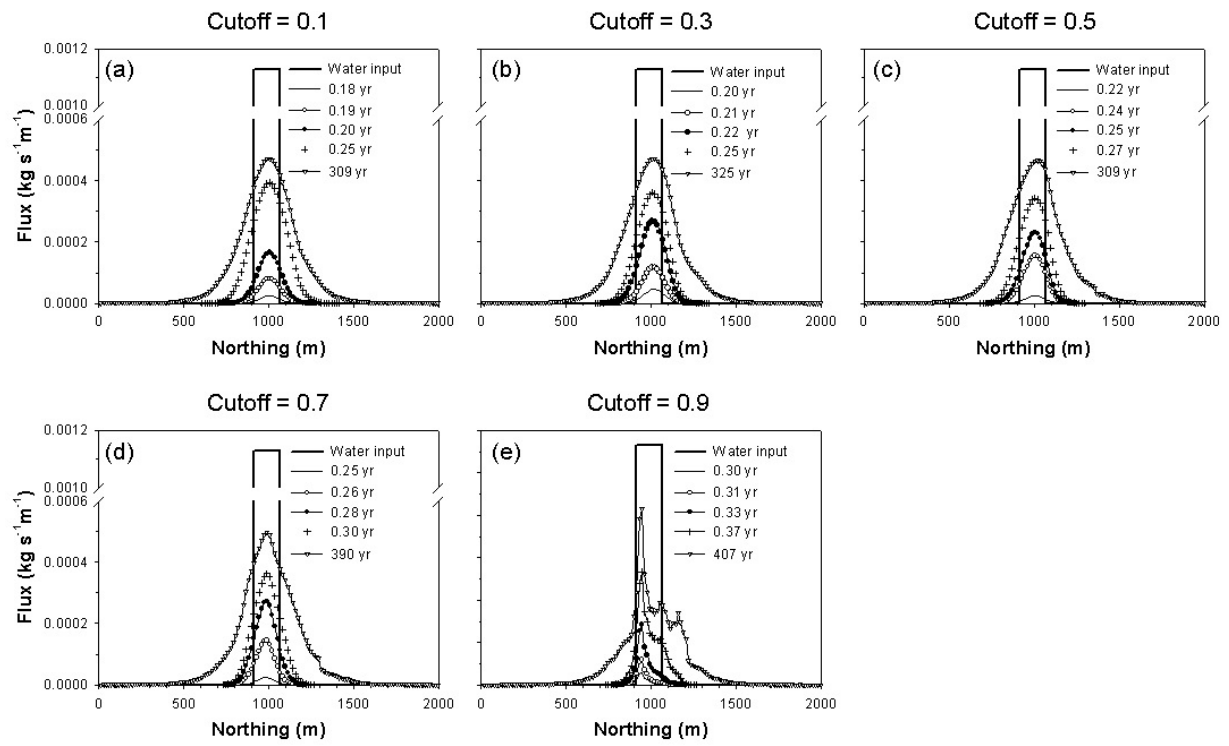
Figure 5. Simulated flux across the bottom boundary for different cutoffs.

Table 5. Simulation results for different cutoff values.

Cutoff	Time to (yr)		Total flow across the bottom boundary (kg)	
	equilibrium	2×10^6 kg flow	at 0.2 yr	at 0.5 yr
0.1	309.2 (109)[a]	0.621	12805 (0.166)[b]	1.3×10^6
0.3	325.5 (112)	0.640	2407 (0.174)	1.3×10^6
0.5	308.6 (129)	0.676	0 (0.198)	1.0×10^6
0.7	390.0 (139)	0.724	0 (0.230)	8.6×10^5
0.9	406.6 (156)	0.842	0 (0.277)	5.0×10^5

[a] Included in parentheses are time steps needed before the system reaches equilibrium.
[b] Included in parentheses are the breakthrough times for water flow.

Both the geometric configuration of the sediment interbeds and their hydraulic properties play crucial roles in water movement and their interactive impact could be highly complex. The effect of the hydraulic properties of the geological materials, though not a focus of this study, could be readily seen from the results of changing hydraulic parameters. In this simulation, the vertical hydraulic conductivity of the basalt was reduced by 10 times to 2.6×10^{-7} m s^{-1}, and the hydraulic conductivity of sediment was decreased 100 times to 6.7×10^{-9} m s^{-1}, while all other parameters were kept the same as in the base run. The results appeared significantly different from those of the base run (Fig. 6a,b). The resultant degree of saturation is in general higher, especially near the interbeds with the difference in water saturation reaching 0.55. The pattern of the flux across the bottom is different from that of the base run, with a broader spread of the flux and more local peaks. The breakthrough time is about 0.44 year, longer than the 0.2 year from the base run.

3.2. Model uncertainty due to the uncertainty in the location of interbeds

For the three tested scenarios, the spatial distribution of sediment and basalt (stratigraphy) was the same. The flow patterns within the model domain were affected only by the relative position of the water source to the sediment interbeds (Fig. 7). Major flow paths and perched aquifer developed primarily beneath the water source in each case, yet the relative location of the interbeds played evidently an important role in the water movement. The characteristics of flux across the bottom boundary were different in both shape and the maximum (Fig. 8). The peak flux in the base run was 0.00046 kg m^{-1} s^{-1} and was 0.00050 kg m^{-1} s^{-1} and 0.00045 kg m^{-1} s^{-1} when the pond was shifted to the left and right, respectively. Discontinuous interbeds, together with other factors, including the flux intensity and the media hydraulic properties, interactively affected the flow pattern in a complex manner.
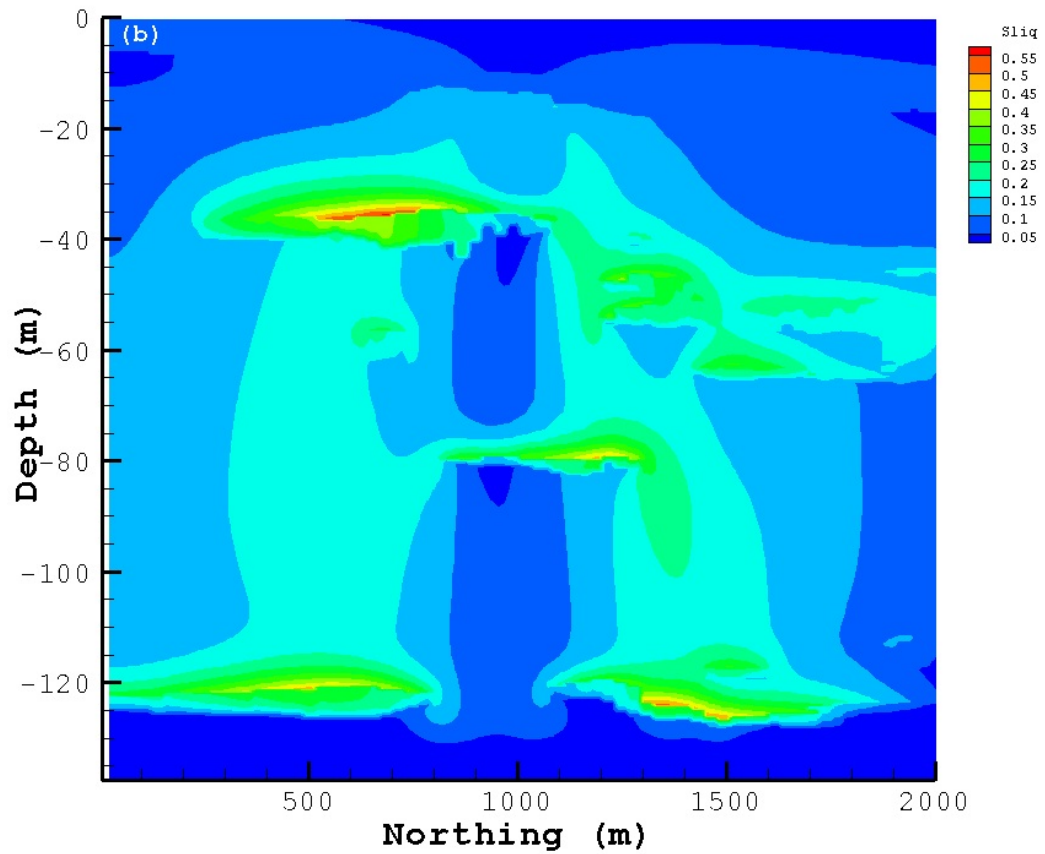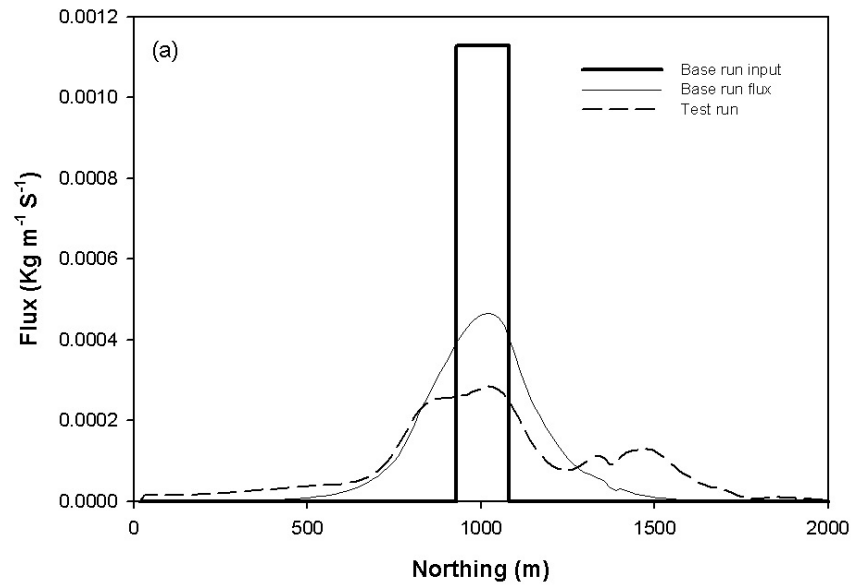
Figure 6. (a) Flux across the bottom boundary comparison between the test run and the base run.

(b) The water saturation degree difference of the test run and the base run.
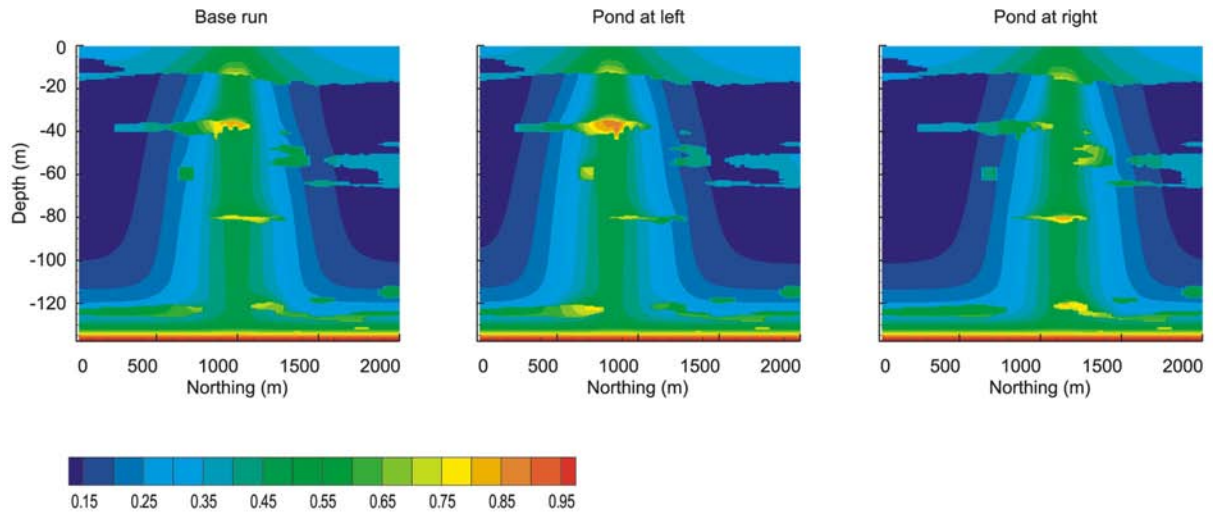
Figure 7. Subsurface moisture contour plot for different pond locations.
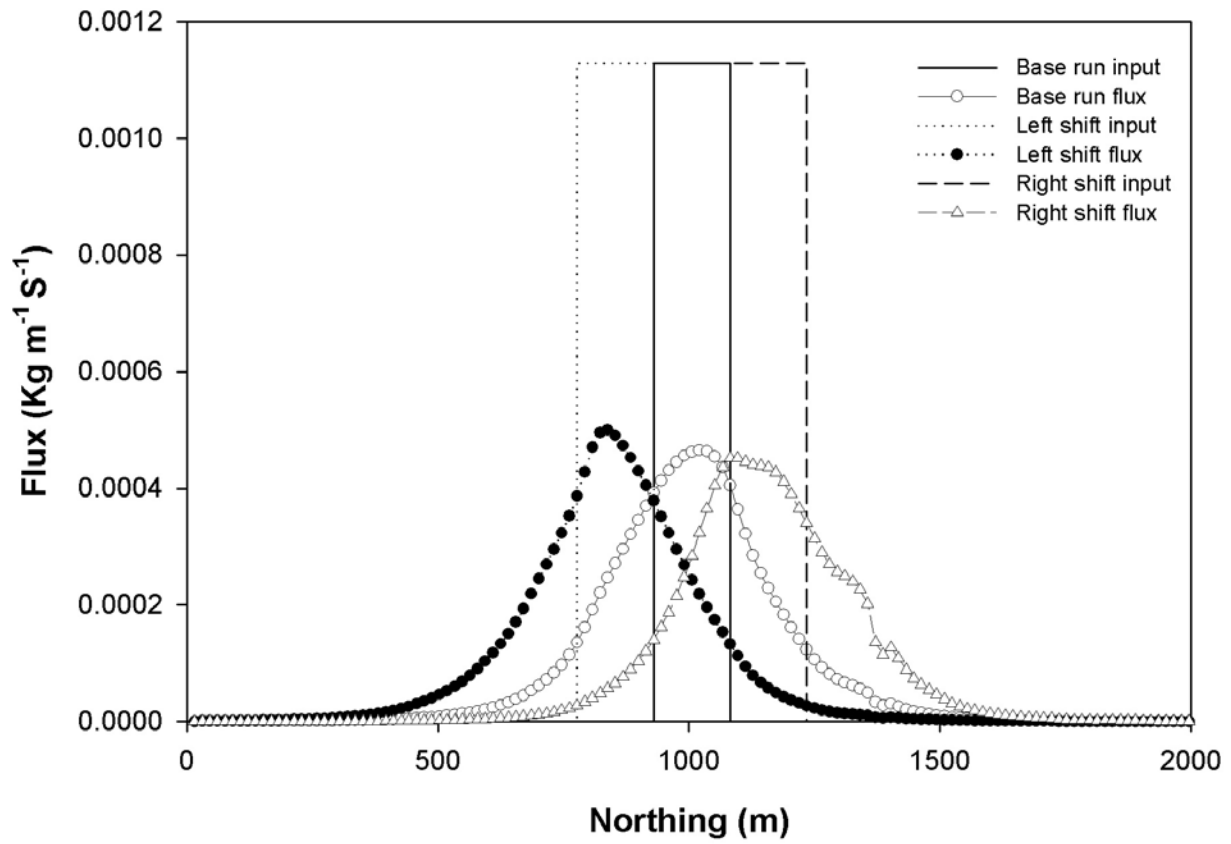
Figure 8. Bottom flux for different pond locations.

Using Eq. 2 to assess the change in the location of the flux peak (Δy) in response to the change in the relative location of water source to the interbeds (Δx), we found a linear relationship with S close to 1. Thus, uncertainty of the water source relative to the interbeds is essentially reflected in the change of the location of the flux peak with the shift approximating the shift of the water source. However, the change in the magnitude of the flux was not sensitive to the shift of the water source in the cases simulated in this study.

3.3. Model uncertainty due to the uncertainty of the geostatistical model parameters

We evaluated eight scenarios, including increasing or decreasing the horizontal and vertical ranges of the base-run variogram model by 20% and 40%, respectively (Table 6). Simulated water flow characteristics were not particularly sensitive to these parameters (Table 7, Fig. 9). The relatively sensitive result is time to equilibrium, with about 17% of change corresponding to a change of 40% in horizontal range (Fig. 9) or a S value of 42.5%. The pattern of the flux across the bottom boundary in all scenarios was virtually the same. The lack of sensitivity may be attributed to multiple reasons: the rather small changes in the tested geostatistical model parameters; the smoothing effect of kriging; and the quantity of the available data.

3.4. Conditional simulation results

Statistical analysis of the results of 30 sequential indicator simulations showed that the maximum flux across the bottom boundary varies from 5.18×10^{-4} to 8.18×10^{-4} kg m^{-1} s^{-1} with mean of 6.08×10^{-4} kg m^{-1} s^{-1} and standard deviation of 5.95×10^{-5} kg m^{-1} s^{-1} (Fig. 10). The average flux across the bottom boundary was obtained by averaging individual results from the 30 simulations. The mean and standard deviation for the averaged flux across the bottom (with a sample size of 163) are identical to those of the base run (sample size 164), but the maximum is

Table 6. Eight scenarios with horizontal and vertical effective ranges of the base-run variogram model decreasing or increasing by 20% and 40%.

| | | Horizontal ranges (m) | | | | Vertical range (m) | |
| | | Structure 1 | | Structure 2 | | | |
Scenario	Change (%)	X	Y	X	Y	Structure 1	Structure 2
Base run	0	472.53	2415.08	297.76	1709.65	17.04	12.41
1	−40	--[a]	--	--	--	10.22	7.45
2	−20	--	--	--	--	13.63	9.93
3	20	--	--	--	--	20.44	14.89
4	40	--	--	--	--	23.85	17.37
5	−40	283.52	1449.05	178.65	1025.79	--	--
6	−20	378.03	1932.06	238.20	1367.72	--	--
7	20	567.04	2898.09	357.31	2051.58	--	--
8	40	661.55	3381.11	416.86	2393.52	--	--

[a] -- denotes the value same as that of base run.

Table 7. Simulation results from changing horizontal and vertical ranges.

Scenario number		Time to equilibrium (yr)	Break Thru. (yr)	Total flow across the bottom (kg) at 0.5yr	Time to 2×10^6 kg water flow (yr)
Horizontal	1	258.8 (124)[a]	0.198	1.02E+06	0.679
	2	299.0 (130)	0.206	1.03E+06	0.707
	3	283.1 (131)	0.198	1.03E+06	0.677
	4	309.2 (133)	0.198	1.02E+06	0.677
	AVG	287.5 (130)	0.200	1.03E+06	0.685
Vertical	1	277.1 (130)	0.206	1.02E+06	0.677
	2	270.5 (131)	0.198	1.03E+06	0.676
	3	282.6 (130)	0.198	1.03E+06	0.675
	4	288.0 (129)	0.198	1.04E+06	0.674
	AVG	279.6 (130)	0.200	1.03E+06	0.676
Base run		308.6 (129)	0.198	1.03E+06	0.677

[a] Included in parentheses are time steps needed before the system reaches equilibrium.
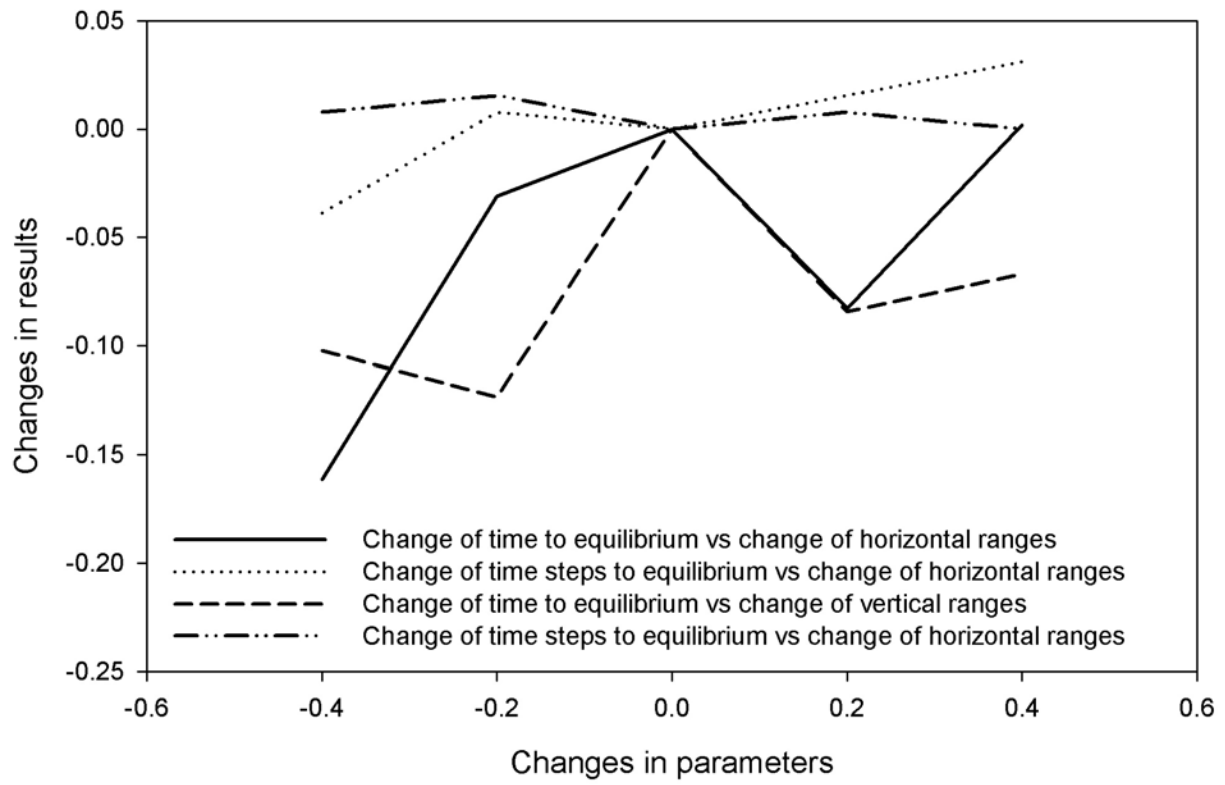
Figure 9. Sensitivity of flow due to changes in geostatistical variogram ranges.
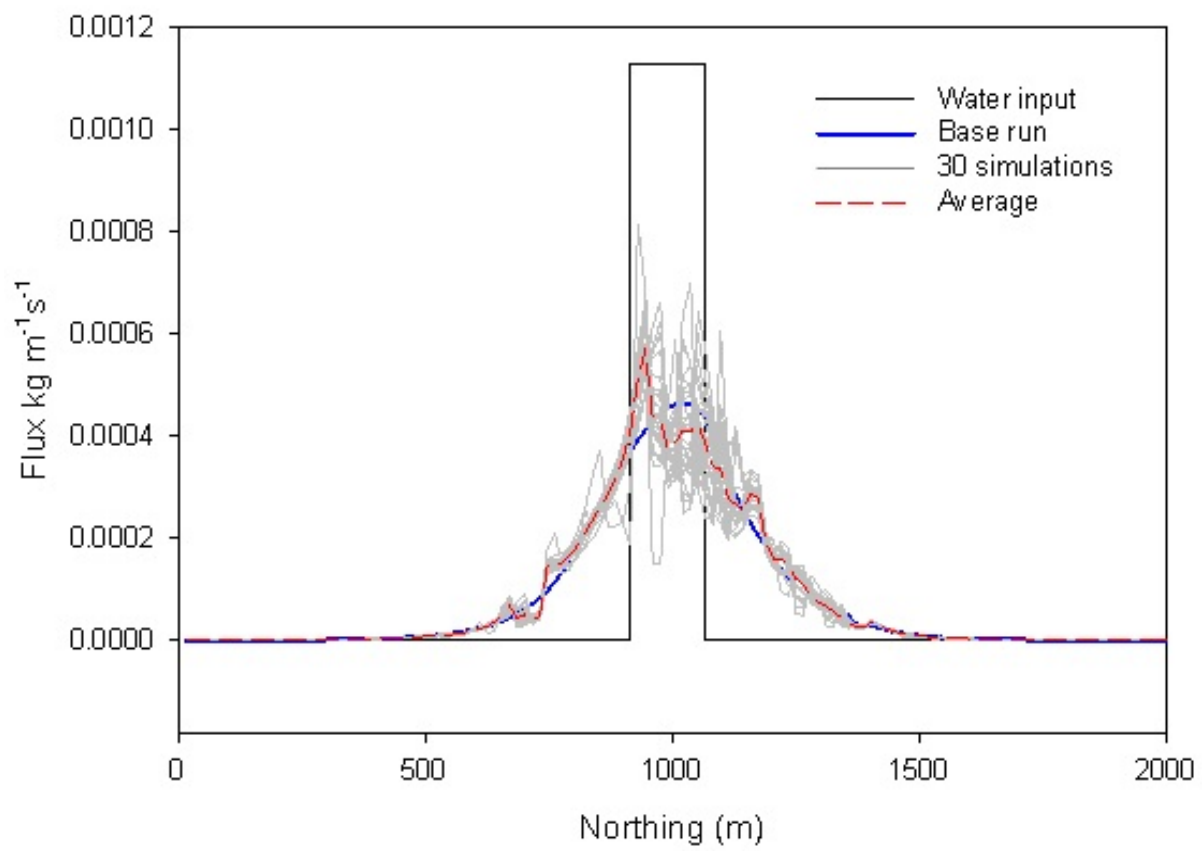
Figure 10. Flux across the bottom boundary comparison for conditional simulation.

5.71×10^{-4} kg m^{-1} s^{-1} compared to 4.65×10^{-4} kg m^{-1} s^{-1} or 22% larger.

All the conditional simulations generated higher maximum flux at the bottom boundary than the base run. Conditional simulations tended to reveal local heterogeneity smoothed by kriging. Therefore, the flux across the bottom boundary was much less smooth compared to that of the base run. Additionally, the position of the averaged peak flux shifted about 76 m to the left compared to that of the base run. The peak flux increased by 75.9% in one extreme case, and less than 30% in most cases as compared to that of the base run. The mean travel time upon break through is 0.22 year (compared to 0.20 year from the base run) with the standard deviation of 0.0075 year (Fig. 11a). The mean time to equilibrium is 292.2 years (258.8 years in the base run) and the standard deviation is 34.6 years (Fig.11b). The generally longer advective travel time and the time to equilibrium resulting from the conditional simulations are likely a consequence of the more heterogeneous cross-sections represented in these simulations.

The assumptions made on the boundary conditions may influence or even reverse the results of uncertainty analysis. No-flow lateral boundaries, though set far away from the water source, may still cause higher water contents in the study domain. Simplified constant-flux top boundary and fixed water table both will smooth out any actual fluctuations.
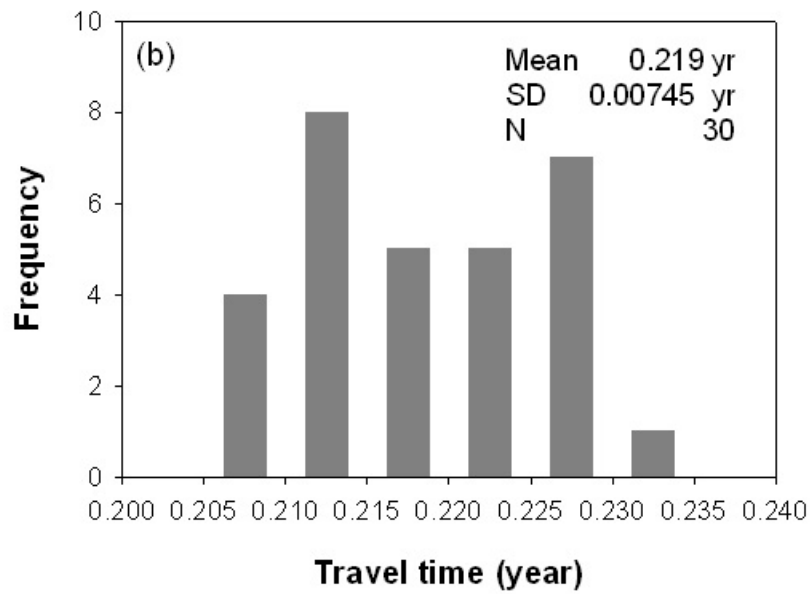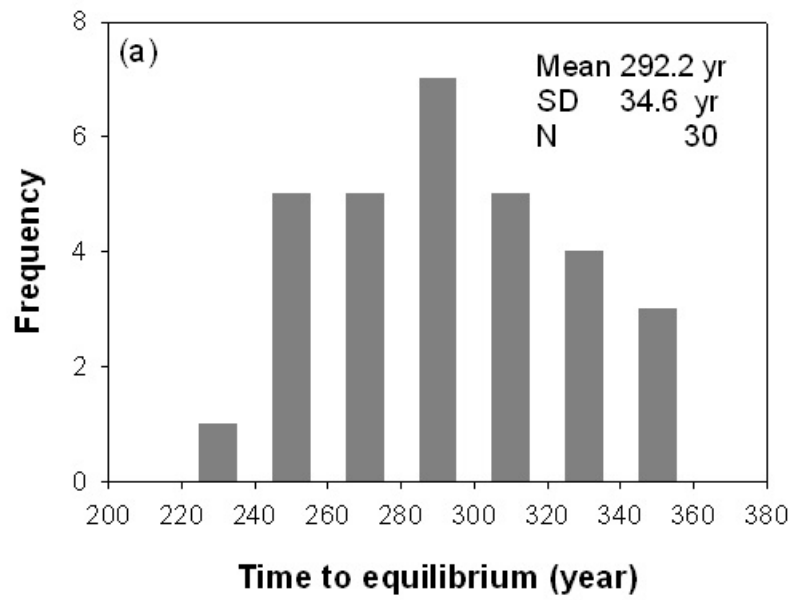
Figure 11. Ground water travel time and time to equilibrium

comparison for 30 conditional simulations.

CHAPTER FOUR

SUMMARY AND CONCLUSIONS

In this study, we developed a two-dimensional model for water flow through variably-saturated, heterogeneous fields at INTEC, INL. Indicator kriging was use to characterize the subsurface geological media using specified probability cutoff. A two-component exponential empirical variogram model was obtained based on borehole data from the study site. TOUGH2, a comprehensive software package for three-dimensional numerical simulation of transport processes in porous and fractured media was applied to simulate water flow through variably-saturated, heterogeneous fields at this study site based on the kriged stratigraphy. Multiple prediction runs were carried out to examine the uncertainty in subsurface water flow regime due to the uncertainty of (i) geological media (basalt and sediments) categorization, (ii) the relative location of water source to the impeding sediment interbeds, and (iii) geostatistical model parameters, specifically, the ranges of variogram model. Additionally, conditional simulations were made to evaluate the effect of random local variability on the water flow.

The uncertainty in geological media categorization was represented by using different cutoffs in assigning basalt or sediment based on the probability value from the indicator kriging. Due to the limitation on available borehole data, it is often difficult to identify the exact locations of the interbeds that have relatively low hydraulic conductivities compared to the basalt and tend to impede downward flow movement. This uncertainty of the relative location of the interbeds to input water source was evaluated by changing the location of the water source (infiltration pond) itself. Under current setting in this study, any change in the relative location of water source would directly lead to change in the location of peak flux across the bottom boundary of the

model domain. However, the magnitude of the flux remains rather stable.

Parameterization of a kriging model to a large extent depends on the quantity and quality of the available data as well as the user expertise. In this study, a two-structure exponential variogram model was obtained using GSLIB (Deutsch et al., 1997) and an optimization program. Both horizontal and vertical ranges of the variogram model were tested for their sensitivity.

The uncertainty caused by random local variability of the geological media was evaluated through conditional simulations. In general, kriging tends to generate smooth interpolation of geological properties. In contrast, conditional simulation helps to reveal local heterogeneity. A total of 30 generated realizations of the model domain from conditional simulation using GSLIB (Deutsch et al., 1997).

Results from the flow modeling indicate that all tested parameters affect the variably-saturated water flow to varying degrees. The uncertainties due to different geological media classifications were most sensitive. Similarly, the uncertainty due to uncertainty of relative locations of water source to interbeds was also significant. When the adjustment was not substantial (with an increase or decrease of 20% or 40%) the change in flow regime was insignificant. However, if the adjustment is significant, the impact may be considerable. Local random variability of the geological media had great influence on the water flow regime. All generated realizations of the model domain exhibited more complex distribution of the subsurface media. As a consequence, the flow regimes were more complicated as reflected in generally longer advective travel time and time to equilibrium. Additionally, the positions of flow peaks across the bottom boundary as well as the peak values varied from case to case.

This study investigated the uncertainty of subsurface water flow due primarily to the characterization of geological media. Yet, the physical and chemical properties of the porous

media play an equally, if not more, important role in flow and contaminant transport. Quantitative analysis of these uncertainties is important and imperative before a vadose zone hydrological model can be used for reliable prediction. Future efforts may be devoted to (i) examining the impact of hydraulic properties of geological material (as random processes) on subsurface flow at the INTEC, INL by applying Monte Carlo techniques to flow models; (ii) investigating the interactive effect of subsurface characterization as well as porous media properties on both flow and contaminant transport; (iii) expanding current two-dimensional simulation to three-dimensional modeling; and (iv) incorporating findings and information from model uncertainty analysis into designing and establishing field monitoring plans.

REFERENCES

Anderson, S.R., M.A. Kuntz, and L.C. Davis. 1999. Geologic controls of hydraulic conductivity in the Snake River Plain Aquifer at and near the Idaho National Engineering and Environmental Laboratory, Water-resources Investigations Report 99-4033 (DOE/ID-22155), U.S. Geological Survey.

Bagtzoglou, A.C., and R.G. Baca. 1994. Probabilistic calculations of groundwater travel time in heterogeneous three-dimensional porous media. In *Scientific basis for nuclear waste management XVII*, ed. A. Barkatt, and R.A. Van Konynenburg, 849–854. Pittsburgh, PA: Materials Research Society.

Courant, R., K. Friedrichs, and H. Lewy. 1928. On the Partial Difference Equations of Mathematical Physics. Mathematische Annalen 100, 32-74.

Deutsch, C., A. Journel. 1997. GSLIB: Geostatistical Software Library and User's Guide. Oxford University Press.

Eaton, R.R., C.K. Ho, R.J. Glass, M.J. Nicholl, and B.W. Arnold. 1996. Modeling of flow through fractured tuff at Fran, High Level Radioactive Waste Management - Proceedings of the Annual International Conference, 76–78.

Evans, D.D., T.C. Rasmussen, and T.J. Nicholson. 2001. Flow and transport through unsaturated fractured rock: An overview. In *Flow and Transport Through Unsaturated Fractured Rock*, ed. D.D. Evans, T.J. Nicholson, and T.C. Rasmussen, 1–18. Washington, DC: American Geophysical Union.

Fogg, G.E., D.R. Nielsen, and D. Shibberu. 1995. Modeling contaminant transport in the vadose zone: Perspective on state of the art. In *Handbook of Vadose Zone Characterization & Monitor*, ed. L.G.Everett and S.J. Cullen. Lewis Publishers, Boca Raton.

Goovaerts, P.. 1997. Geostatistics for Natural Resources Evaluation. Oxford University Press. 483p.

Gomez-Hernandez, J.J., A. Sahuquillo, and J.E. Capilla. 1997. Stochastic simulation of transmissivity fields conditional to both transmissivity and piezometric data-I. Theory. Journal of Hydrology 203, 162–174.

Greeley, R. 1982. The style of basaltic volcanism in the Eastern Snake River Plain, Idaho. In *Cenozoic Geology of Idaho*, ed. B. Bonnichsen, and R.M. Breckenridge. Idaho Bureau of Mines and Geology Bulletin 26, 407–421.

Gureghian, A.B., and B. Sagar.1993. Uncertainty and sensitivity analyses of groundwater travel time in a two-dimensional variably-saturated fractured geologic medium. In *Scientific Basis for Nuclear Waste Management XVI*, ed. C.G. Interrante, and R.T. Pabalan, 921–928. Pittsburgh, PA: Materials Research Society.

Idaho National Engineering and Environmental Laboratory (INEEL), 2003. Phase I monitoring well and tracer study report for operable unit 3–13, group 4, perched water, revision 1. DOE/ID-10967, U.S. DOE Operations Office.

Kramer, J.H., and S.J. Cullen. 1995. Review of vadose zone flow and transport models. In *Handbook of Vadose Zone Characterization & Monitor*, ed. L.G.Everett and S.J. Cullen. Lewis Publishers, Boca Raton.

Knutson, C.F., K.A. McCormick, R.P. Smith, W.R. Hackett, J.P. O'Brien, and J.C. Crocker. 1990. FY-89 RWMC vadose zone basalt characterization, Report EGG-WM-8949, Idaho National Engineering and Environmental Laboratory, EG&G Idaho, Inc.

Kuntz, M.A., Skipp, B., Lanphere, M.A., Scott, W.E., Pierce, K.L., Dalrymple, G.B., Morgan, L.A., Champion, D.E., Embree, G.F., Smith, R.P., Hackett, W.R. Rodgers, D.W., and

Page, W.R.. 1990. Revised geologic map of the Idaho National Engineering Laboratory and adjoining areas, Eastern Idaho. U.S. Geological Survey Open File Report 90-333, 1-35.

Malde, H.E. 1991. Quaternary geology and structural history of the Snake River Plain, Idaho and Oregon. In *Quaternary Nonglacial Geology: Conterminous US*, ed. R.B. Morrison. Boulder, Colorado, Geological Society of America, The Geology of North America, k-2.

Nichols, W.E., and M.D. Freshley. 1993. Uncertainty analyses of unsaturated zone travel time at Yucca mountain. Ground Water 31, no. 2: 293–301.

Pruess, K., C. Oldenburg, and G. Moridis. 1999. TOUGH2 User's Guide, version 2.0. Lawrence Berkeley National Laboratory, Report LBNL-43134, Berkeley, CA.

Rautman, C.A., A.L. Flint, J.D. Istok, L.E. Flint, and M.P. Chornack. 1993. Influence of deterministic geologic trends on spatial variability of hydrologic properties in volcanic tuff. In *Proceedings of the Fourth Annul International Conference on High Level Radioactive Waste Management*, 921–929. Las Vegas, NV: American Society of Civil Engineers.

Russo, D., and E. Bresler. 1981. Soil hydraulic properties as stochastic processes, I, An analysis of field spatial variability. Soil Science Society of America Journal 45, 682–687.

Rutqvist, J., Y.-S. Wu, C.-F. Tsang, G. Bodvarsson. 2002. A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock, International Journal of Rock Mechanics and Mining Sciences, 39, n 4, 429–442.

Samuel, B., and J.D. Cawlfield. 1999. Two-dimensional sensitivity analysis of contaminant

transport in the unsaturated zone. Ground Water 37, no. 2: 185–193.

Schafer, A.L., P. Martian, J.M. McCarthy, and T. Honeycutt. 1997. Appendix F: WAG-3 vadose

zone and aquifer RI/FS contaminant source identification and fate and transport modeling

results.

van Genuchten, M.Th. 1980. A closed-form equation for predicting the hydraulic conductivity of

unsaturated soils. Soil Sci. Soc. Am. J. 44:892–898.

Vieira, S.R., D.R. Nielsen, and J.W. Biggar. 1981. Spatial variability of field measured

infiltration rate. Soil Science Society of America Journal 45, 1040–1048.

APPENDIX

A. INL BOREHOLE DATA

(included in the attached CD):

1. Original data from INL

=us-asciiQintec-lithology.xls=.xls

2. Corrected data set that was used in this study

us-asciiQintec-lithologynew.xls

B. VARIOGRAM

All rotations were made according to GSLIB (Deutsch et al., 1997) conventions. First, rotate around the Z axis according to the left hand rule. Second, rotate around the rotated X axis according to the right hand rule. Third, rotate around the rotated Y axis according to the right hand rule.

The right and left hand rules are easy to follow. For example, to rotate around the Z axis according to the left rule, imagine grabbing the Z axis with your left hand so that your thumb is pointing in the direction of increasing Z values. Then the fingers will be pointing in the direction of a positive rotation. A negative rotation angle indicates a rotation in exactly the opposite direction. Also the exponential ranges are the GSLIB convention, i.e., one third of the practical range.

The GSLIB modeling criteria include:

Minimum number of pairs ==>1000

Max allowable drift on head and tail means ==> 1.0000

Sample variogram points weighted by number of pairs.

1. Vertical variogram
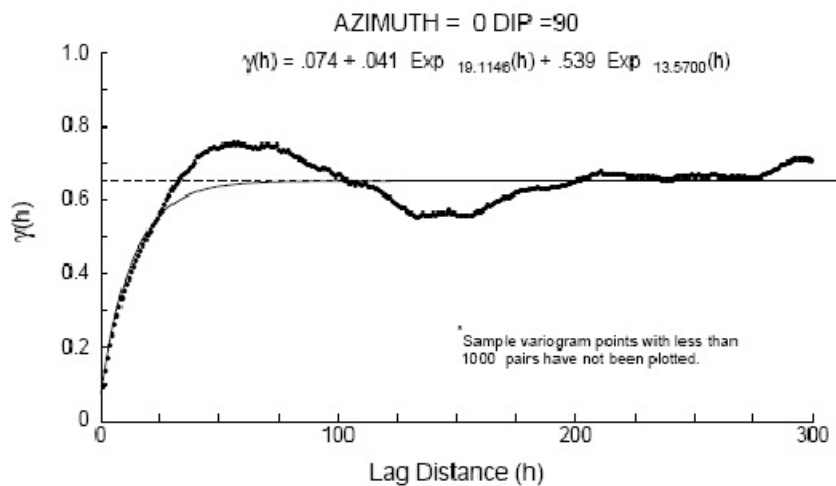
Figure B1-1. Sample vertical variogram for azimuth = 0

45

2. Horizontal variogram
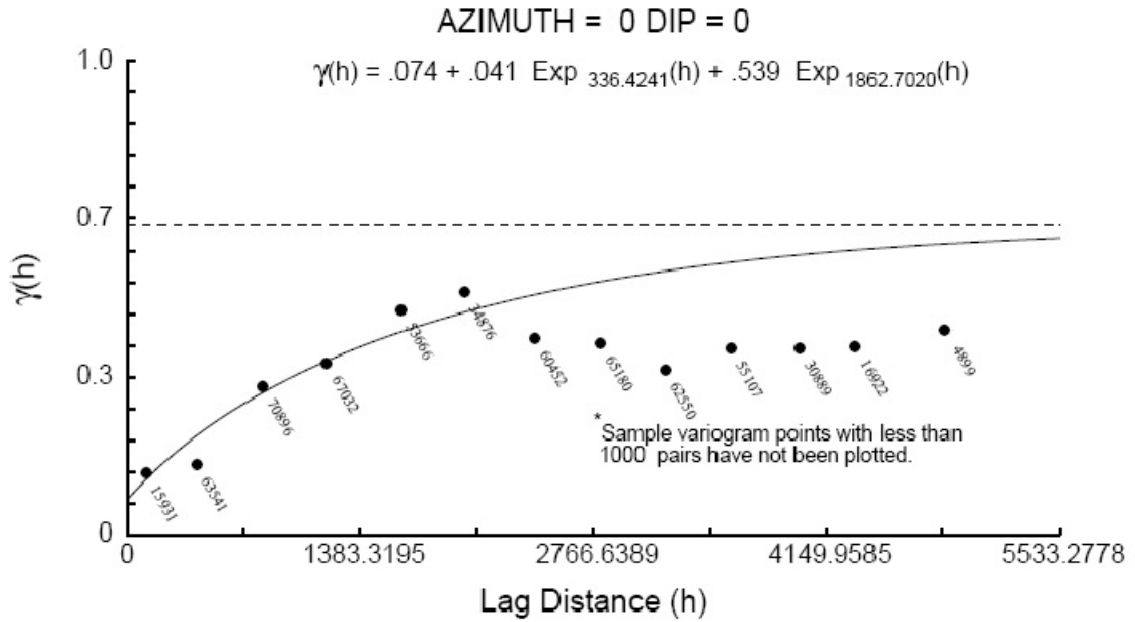
Figure B2_1. Sample horizontal variogram for azimuth = 0 and dip = 0.
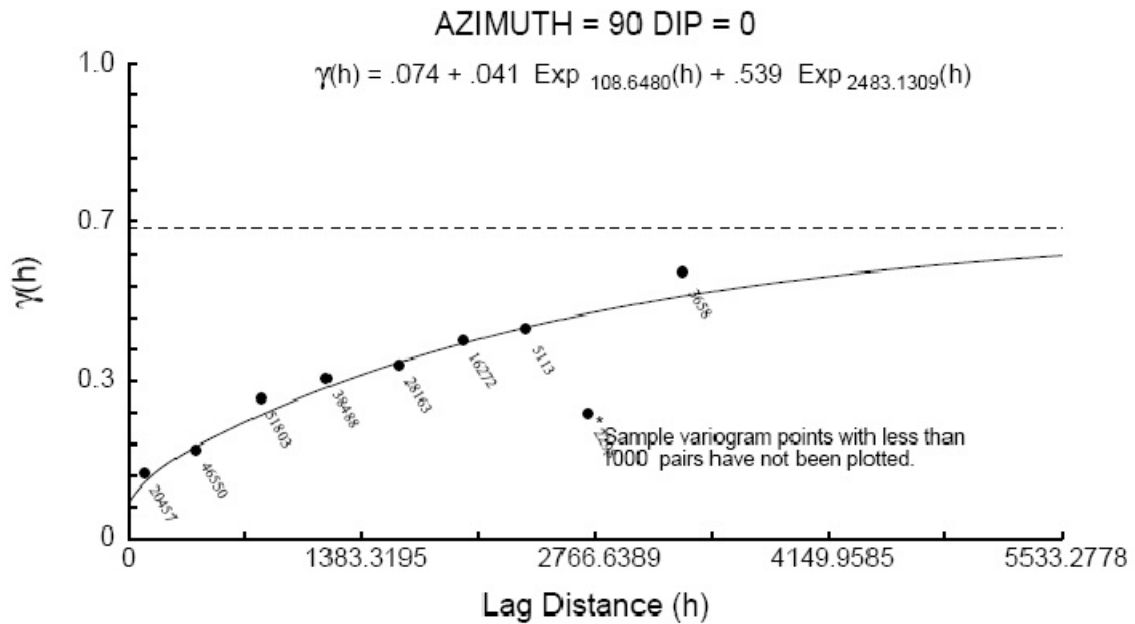
Figure B2-2. Sample horizontal variogram for azimuth = 90 and dip = 0.

C. SECTIONAL VIEWS THOUGH THE ELLIPSOIDS
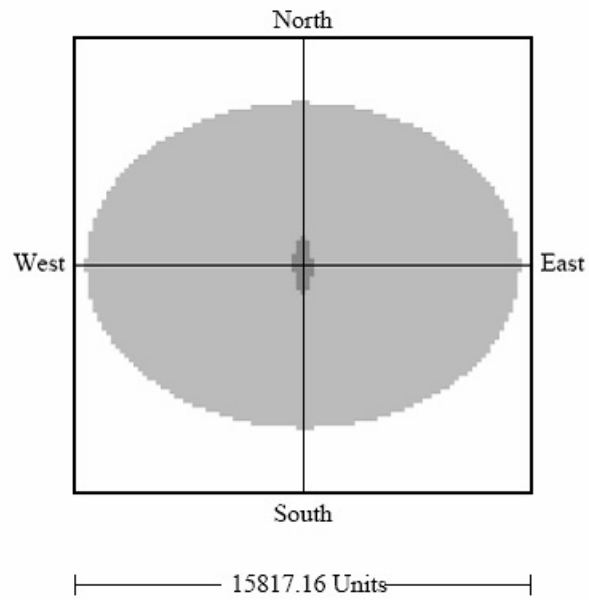
Figure C-1. X-Y plane through origin looking down.
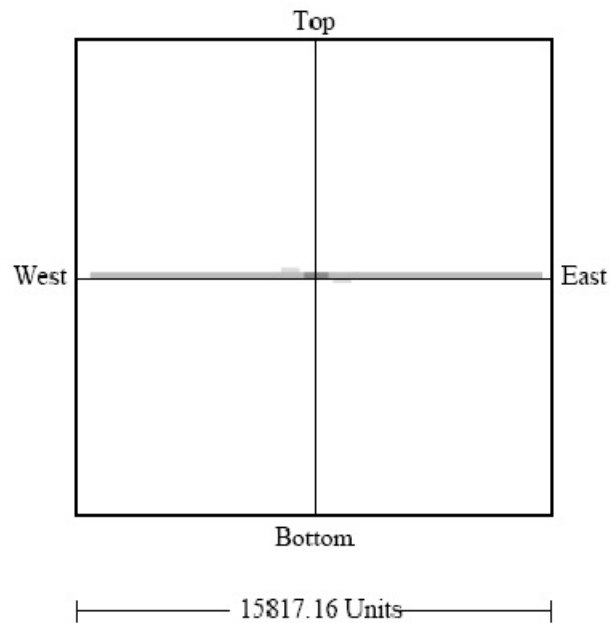
Figure C-2. X-Y plane through origin looking north.
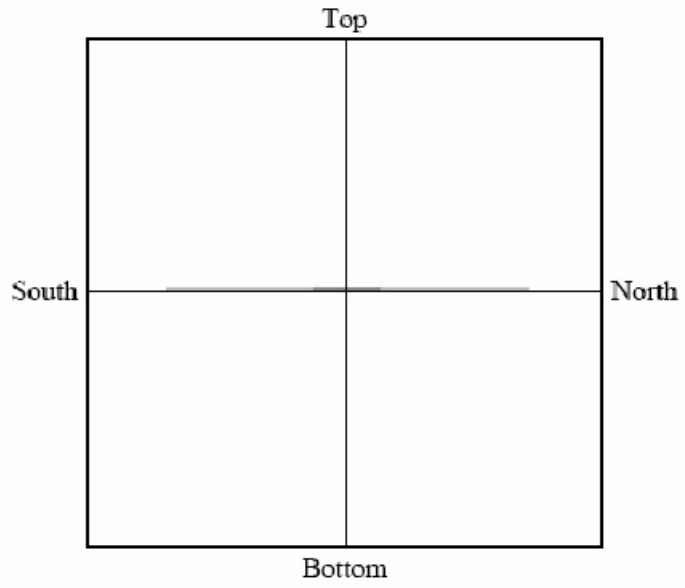
Figure C-3. X-Y plane through origin looking west.

D. ROSE DIAGRAM OF RANGES

333.3333 Units
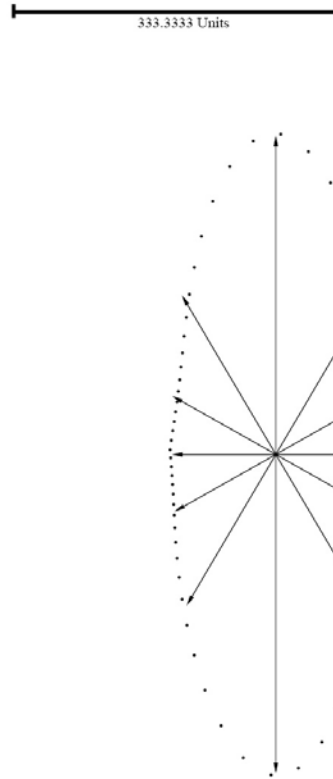
Figure D-1. Rose diagram of ranges of Structure-1 dipping 0 degrees

Scale:

333.3333 Units
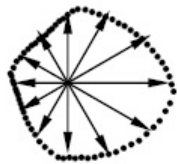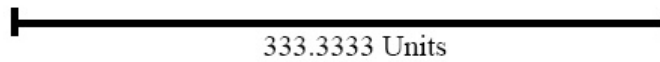
Dipping 30 degrees   Dipping 60 degrees

Figure D-2. Rose diagram of ranges of Structure 1 dipping 30 and 60 degrees

2333.3333 Units

Dipping 30 degrees

Dipping 60 degrees
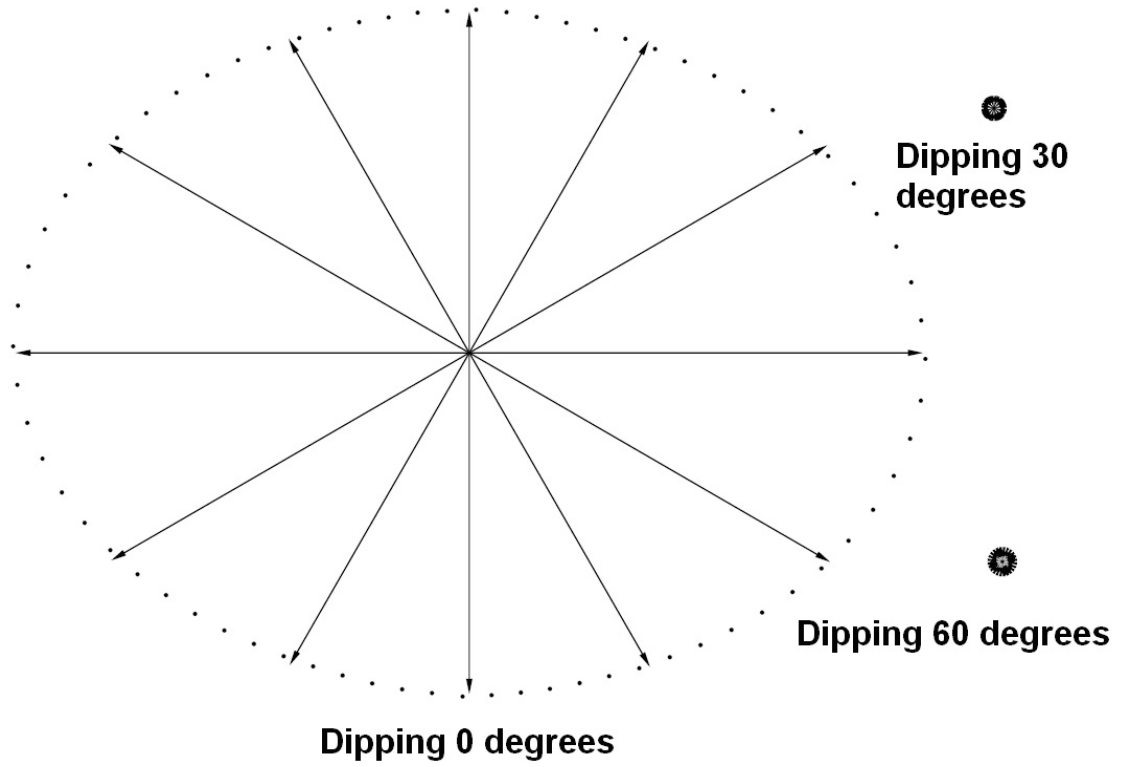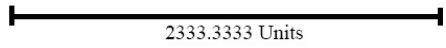
Dipping 0 degrees

Figure D-3. Rose diagram of ranges of Structure 2 dipping 0, 30 and 60 degrees

E. TOUGH2 CODE MODIFICATION

TOUGH2 codes are primarily written for the IBM UNIX system, though they were

claimed to be independent of any hardware or operational systems. Modifications were need for a

successful compiling of TOUGH2 on a PC platform. The limitations of the codes also include

the use of limited digits for the grid dimensions. The modifications to TOUGH2 are shown

below. Note: all lines with line numbers were changed by Limin Yang. Changed files (EOS9.f,

T2CG2.f, T2F.f, and MASHM.f) are included in the attached CD.

1. EOS9.f

```
906              SUBROUTINE MULT9
                 SUBROUTINE MULTI
907      c       *het/mul9.f*, from MULT9 of dvlp/eos9.f
         c       *het/mul9.f*, from MULTI of dvlp/eos9.f
1040     c 899 FORMAT(/6X,'MULT9    1.0 S    15 January   1993',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    15 January   1993',6X,
1041     c 899 FORMAT(/6X,'MULT9    1.0 S    20 April     1994',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    20 April     1994',6X,
1042     c 899 FORMAT(/6X,'MULT9    1.0 S     3 May       1994',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S     3 May       1994',6X,
1043     c 899 FORMAT(/6X,'MULT9    1.0 S    17 May       1994',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    17 May       1994',6X,
1044     c 899 FORMAT(/6X,'MULT9    1.0 S    15 December  1994',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    15 December  1994',6X,
1045     c 899 FORMAT(/6X,'MULT9    1.0 S     2 November  1995',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S     2 November  1995',6X,
1046     c 899 FORMAT(/6X,'MULT9    1.0 S    28 June      1996',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    28 June      1996',6X,
1047     c 899 FORMAT(/6X,'MULT9    1.0 S    11 October   1996',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    11 October   1996',6X,
1048     c 899 FORMAT(/6X,'MULT9    1.0 S    12 February  1997',6X,
         c 899 FORMAT(/6X,'MULTI    1.0 S    12 February  1997',6X,
1049       899 FORMAT(/6X,'MULT9    1.0 PM    4 March     1997',6X,
           899 FORMAT(/6X,'MULTI    1.0 PM    4 March     1997',6X,
1067       201 FORMAT(/39H $$$$$$$$$$ SUBROUTINE MULT9 $$$$$$$$$$,24H   ---
           201 FORMAT(/39H $$$$$$$$$$ SUBROUTINE MULTI $$$$$$$$$$,24H   ---
```
2. T2CG2.f

```
866      c    timing function for DVF
         c    timing function for IBM RS/6000
867        real*4 s, sold
868        save sold
873        899 FORMAT(6X,'SECOND   1.0       7 February  2001',6X,
           899 FORMAT(6X,'SECOND   1.0       6 September 1994',6X,
874        &          'CPU timing function for DVF')
           x'CPU timing function for IBM RS/6000')
875         call cpu_time(s)
           time=float(mclock())/100.
876         if (icall.eq.1) sold=s
877         time=s-sold
```

3. T2F.f

```
80        common/fgt1/ioft,iofu,igoft,igofu,noft(10000),ngoft(10000)
          common/fgt1/ioft,iofu,igoft,igofu,noft(100),ngoft(100)
81        common/fgt2/eoft(10000),egoft(10000),ecoft(10000)
          common/fgt2/eoft(100),egoft(100),ecoft(100)
82        common/fgt3/icoft,icofu,ncoft(10000)
          common/fgt3/icoft,icofu,ncoft(100)
312    10 FORMAT(A3,I2,2I5,A3,A2,2E10.4,E10.4,3F10.4
       10 FORMAT(A3,I2,2I5,A3,A2,2E10.4,4E10.4)
313    IF(EL.EQ.'   '.AND.NE.EQ.0) GOTO40  !ylm 9/28/04
       IF(EL.EQ.'   '.AND.NE.EQ.0) GOTO40
336    14 FORMAT(A3,I2,10X,I5,2E10.4,E10.4,3F10.4
       14 FORMAT(A3,I2,10X,I5,2E10.4,4E10.4)
345    114 FORMAT(A3,I2,13X,A2,2E10.4,E10.4,3F10.4)
       114 FORMAT(A3,I2,13X,A2,2E10.4,4E10.4)
361    20 FORMAT(A3,I2,A3,I2,4I5,2F10.4,3E10.4) !5E10.4 to 2F10.4,3E10.4
       20 FORMAT(A3,I2,A3,I2,4I5,5E10.4)
372    24 FORMAT(A3,I2,A3,I2,15X,I5,2F10.4,3E10.4) !5E10.4 to 2F10.4,3E10.4
       24 FORMAT(A3,I2,A3,I2,15X,I5,5E10.4)
595        do81 i=1,10000
           do81 i=1,100
610        do181 i=1,10000
           do181 i=1,100
629        do281 i=1,10000
           do281 i=1,100
798        common/fgt1/ioft,iofu,igoft,igofu,noft(10000),ngoft(10000)
           common/fgt1/ioft,iofu,igoft,igofu,noft(100),ngoft(100)
799        common/fgt2/eoft(10000),egoft(10000),ecoft(10000)
           common/fgt2/eoft(100),egoft(100),ecoft(100)
800        common/fgt3/icoft,icofu,ncoft(10000)
           common/fgt3/icoft,icofu,ncoft(100)
1952       common/fgt1/ioft,iofu,igoft,igofu,noft(10000),ngoft(10000)
           common/fgt1/ioft,iofu,igoft,igofu,noft(100),ngoft(100)
1953       common/fgt3/icoft,icofu,ncoft(10000) !CHANGED FROM 100 TO 10000.YLM
           common/fgt3/icoft,icofu,ncoft(100)
2169       common/fgt1/ioft,iofu,igoft,igofu,noft(10000),ngoft(10000)
           common/fgt1/ioft,iofu,igoft,igofu,noft(100),ngoft(100)
2170       common/fgt3/icoft,icofu,ncoft(10000) !CHANGED FROM 100 TO 10000.YLM
           common/fgt3/icoft,icofu,ncoft(100)
2171       dimension xd(510000),ind(10000)
           dimension xd(500),ind(100)
2212   81 format(I5,' , ',E12.6,10000(' , ',I5,4(' , ',E10.4))) !100 to 10000.
       81 format(I5,' , ',E12.6,100(' , ',I5,4(' , ',E10.4)))
2238   281 format(I5,' , ',E12.6,10000(' , ',I5,3(' , ',E10.4))) !100 to 10000.
       281 format(I5,' , ',E12.6,100(' , ',I5,3(' , ',E10.4)))
2307   86 format(I5,' , ',E12.6,3000(' , ',I5,5(' , ',E10.4)))!100 to 3000.ylm
       86 format(I5,' , ',E12.6,100(' , ',I5,5(' , ',E10.4)))
2311       SUBROUTINE MULTIxx        !changed by ylm from MULTI to MULTIxx
           SUBROUTINE MULTI
2312       !to avoid conflict to the same module in EOS9.f
```

4. MESHM.f

```
318         5 FORMAT(3A1,I2,10X,'    1',2E10.4,10X,E10.4,10X,F10.4) !E10.4 TO
F10.4
            5 FORMAT(3A1,I2,10X,'    1',2E10.4,10X,E10.4,10X,E10.4)
373         9 FORMAT(3A1,I2,3A1,I2,19X,1H1,E10.4, 2F10.4)
            9 FORMAT(3A1,I2,3A1,I2,19X,1H1,3E10.4)
395        41 FORMAT(3A1,I2,3A1,I2,19X,1H3,E10.4,2F10.4,2H1.)
           41 FORMAT(3A1,I2,3A1,I2,19X,1H3,3E10.4,2H1.)
511       101 FORMAT(8F10.4) !FROM E10.4 TO F10.4. YLM 9/27/04
```

52

```
        101 FORMAT(8E10.4)
623      10 FORMAT(3A1,I2,10X,A5,2E10.4,10X,3F10.4)!FROM E10.4 TO F10.4
         10 FORMAT(3A1,I2,10X,A5,2E10.4,10X,3E10.4)
639      11 FORMAT(2(3A1,I2),19X,A1,2F10.4, 2E10.4)!FROM E10.4 TO F10.4
         11 FORMAT(2(3A1,I2),19X,A1,4E10.4)
1337     10 FORMAT(A1,A4,10X,I5,E10.4,E10.4,10X,3F10.4)!FROM E10.4 TO F10.4
         10 FORMAT(A1,A4,10X,I5,E10.4,E10.4,10X,3E10.4)
1349      7 FORMAT(A5,10X,I5,2E10.4,10X,3F10.4)!FROM E10.4 TO F10.4
          7 FORMAT(A5,10X,I5,2E10.4,10X,3E10.4)
1404     20 FORMAT(2(A1,A4),15X,I5,2F10.4,2E10.4)
         20 FORMAT(2(A1,A4),15X,I5,4E10.4)
1437    105 FORMAT(2A5,15X,I5,2F10.4,2E10.4) !FROM 4E10.4 TO 2F10.4, 2E10.4
        105 FORMAT(2A5,15X,I5,4E10.4)
1488    102 FORMAT(2(A1,A4),19X,'1',2F10.4,E10.4)
        102 FORMAT(2(A1,A4),19X,'1',3E10.4)
1489        EL10=NA(M)                    !BY YLM ON 9/27/04.
            EL10=NA(M)
```

F. UTILITY CODES

This section contains utility codes for pre-processing the TOUGH2 input files and post-

processing the output files.

F.1. FORTRAN 90 codes for pre-processing (see beginning of the program for further

description)

```fortran
Program T2GridxzCoft
!     This program is to
!     1. read kriged grid file,
!     2. regenerate grid in TOUGH2 (lbl program) format,
!     3. and gradually make adaptive grids at each interface,
!     4. write to ELEM and CONNE files in required formats.
!     by Limin Yang 07/16/2004.
!     Last modified time: 9/18/2004. 9/19/2004.
!     Change Mesh, Conne part x,y,z from E10.4 to F10.4. ylm 9/28/04.
!     Change to X-Z.  ylm 10/04/04
!     Fix interface search problem by 1; LocaY(i)+1 method
!     ylm 10/05/04
!     Fix big Gmin resorting negative grid size in Z direction problem.
!     01/05/05, ylm.
implicit none

Character*60 DataFile
Character Title1        !read title from input file.
Character*2 WORD        !denotes 'NX', 'NY', and 'NZ'
Integer i, j, k, nX, nZ, nGx, nGz, nGxx, nGzz, N, N2
Integer Xk, Zk, nTotal, Max, flag, flag1
Real Gx, Gz, Gmin, GminX,Xmax, Zmax, a, b, cc, Sn     !Gx = grid size in X,
Gz=grid size in Z.
REAL, ALLOCATABLE :: DataSet(:,:)
Integer, ALLOCATABLE ::LocaX(:), LocaZ(:),LocaX1(:), LocaZ1(:),Intf(:)
!LocaX() & LocaZ() store grids' sizes in X & Z directions
!for MESHMAKER to generate varied grids. Intf() stores temp interfaces.
Real, ALLOCATABLE :: nG1(:,:), nG2(:,:), nG3(:,:), nGtemp(:,:)

DataFile =  'c:\krigged2.txt'
N = 1         ! repeat number
N2= 2         !Logarithm base
Gmin = 0.7   !meter
GminX = 20.0       !meter
Max    = 300000
Xmax = 60.0                                !meter
Zmax = 3.0                                 !meter
! Read krigged grid file (currently 2D) into matrix DataSet(i,j)  [j=1,2,3]
! and get nX (total grid in X direction), nZ, Gx (grid size), Gz.
OPEN (1, FILE = DataFile)
ALLOCATE (DataSet(MAX/3,3))

DO WHILE (.NOT. EOF(1))
     i=i+1
     READ( 1, *) (DataSet(i,j),j=1,3)
END DO
CLOSE (1)

DEALLOCATE (DataSet)
PRINT*,i,j
```

```
ALLOCATE (DataSet(i,3))
OPEN (1, FILE = DataFile)
i=0
DO WHILE (.NOT. EOF(1))
       i=i+1
       READ( 1, *) (DataSet(i,j),j=1,3)
END DO
CLOSE (1)
nTotal = i
!------------------------end of reading data----------------
!change x, z from ft to meter, ylm on Sept. 06, 2004
do i =1, nTotal
       DataSet(i,1)=DataSet(i,1)*0.3048
       DataSet(i,2)=DataSet(i,2)*0.3048
End do
!-----end of transformation --------


a = DataSet(1,1)
Gx = DataSet(2,1) - a
Do k = 2, nTotal
       if (DataSet(k,1) == a) then
       goto 2000
       end if
end Do
2000 nX = k - 1
!////////// following can be used in 3D case /////////
!a = DataSet(k-1,2)
!Do while (DataSet(k,1) == a)
!      k= k +nX
!end Do
!nZ = k/nX - 1
!nX = nTotal/nX/nZ
!///////////////////////////////////////////////////
nZ = nTotal/nX
Gz = DataSet(nX+1,2)-DataSet(1,2)
!--------------end of finding nX and nZ--------------------
allocate (LocaX((nZ-1)*(nX-1)))
allocate (LocaZ((nZ-1)*(nX-1)))         !Therotical maximum
!--Search for interfaces in X----------------------------------
k= 1
Xk = 0
do i=1, nX-1
       do j = 1, nZ
             if (DataSet((j-1)*nX+i,3) .NE. DataSet((j-1)*nX+i+1,3)) then
             ! compare to left element
                   Xk = Xk+1
                   LocaX(Xk) = i
                   Xk = Xk+1
                   LocaX(Xk) = i+1
             end if
       end do
end do

If (Xk == 0) then
       Print *, "The media is homogeneous, end of calculation."
       Goto 10000       !goto end
End if
!--End of searching for interfaces------------------------------------

!--Search for interfaces in Z------------------------------------
k= 1
Zk = 0
do j=1, nZ-1
```

```
      do i = 1, nX
            if (DataSet((j-1)*nX+i,3) .NE. DataSet(j*nX+i,3)) then
            ! compare to below element
                  Zk = Zk+1
                  LocaZ(Zk) = j
                  Zk = Zk+1
                  LocaZ(Zk) = j+1
            end if
      end do
end do
!--End of searching for interfaces-------------------------------------

!----------delete invalid data (zeros) in LocaX() and LocaZ()
Xk=0
Do i = 1, (nZ-1)*(nX-1)
      if (LocaX(i)> 0) then
            Xk=Xk+1
      end if
end do
Zk=0
Do i = 1, (nZ-1)*(nX-1)
      if (LocaZ(i)> 0) then
            Zk=Zk+1
      end if
end do

allocate (LocaX1(Xk))
allocate (LocaZ1(Zk))

Do i = 1, Xk
      LocaX1(i)=LocaX(i)
end do

Do i = 1, Zk
      LocaZ1(i)=LocaZ(i)
end do
deallocate(LocaX)
deallocate(LocaZ)
!-----------------delete repeat interfaces------------
Do i = 1, Xk-2,2
      LocaX1(i+1)=LocaX1(i)
end do

Do i = 1, Zk-2,2
      LocaZ1(i+1)=LocaZ1(i)
end do

allocate (LocaX(Xk))
allocate (LocaZ(Zk))
LocaX(1)=LocaX1(1)
LocaZ(1)=LocaZ1(1)
k=1
Do i = 2, Xk
      LocaX(i)=0
      LocaZ(i)=0
end do
Do i = 1, Xk-1
      if (LocaX1(i+1)>LocaX1(i)) then
            k = k+1
            LocaX(k)=LocaX1(i+1)
      end if
end do
Xk = k
k=1
```

```
Do i = 1, Zk-1
     if (LocaZ1(i+1)>LocaZ1(i)) then
          k = k+1
          LocaZ(k)=LocaZ1(i+1)
     end if
end do
Zk =k
deallocate(LocaX1)

allocate (LocaX1(Xk+1))
LocaX1(1)= 1                    !add left bdry
Do i = 2, Xk
     LocaX1(i)=LocaX(i-1)+1
end do
LocaX1(Xk+1)= nX+1 !right bdry

!--------end of adding bdry in X--------
deallocate(LocaZ1)

allocate (LocaZ1(Zk+1))
LocaZ1(1)= 1                              !add top bdry
Do i = 2, Zk
     LocaZ1(i)=LocaZ(i-1)+1
end do
LocaZ1(Zk+1)= nZ+1           !bottom bdry
!--------end of adding bdry in Z--------

Xk = size(LocaX1)
Zk = size(LocaZ1)

deallocate(LocaX)
deallocate(LocaZ)
allocate (LocaX(Xk))
allocate (LocaZ(Zk))

Do i = 1, Xk
     LocaX(i)=LocaX1(i)
end do
Do i = 1, Zk
     LocaZ(i)=LocaZ1(i)
end do
deallocate(LocaX1)
deallocate(LocaZ1)
!-----------end of deleting repeat interfaces------------


! assign adaptive grids in X direction XXXXXXX Grid-center Method XXXXXXXXXXXX
allocate (nG1(50*Xk,2))
nGxx = 0
cc= N*N2*Xmax/(N2-1)    !cc is the possible length of all grids not bigger
than Xmax
Do i = 1, Xk-1
     a = (LocaX(i+1) - LocaX(i))*Gx      !length between columns

     If (a <= 2*cc) then
     !Xmax = 60 m. N is repeat #.
          b = a/2/N/GminX
          nGx = Int(LOG10(b)/LOG10(real(N2)))+1     !half space will be
calculated
          nG1(nGxx+1,1) = N                         !nG1(1,1) is # of grid in the
length
          nG1(nGxx+1,2) = GminX                !of nG1(1,2)=GminX.
          Do j = 2, nGx     !Serial start as 1*N*GminX, N2**1*N*GminX,
N2**2*N*GminX,...
```

```
                nG1(nGxx+j,1) = N                        !nG1(i,1) is # of grid
in the length
                nG1(nGxx+j,2) = N2**(j-2)*GminX      !of nG1(i,2). N2, base
of Logarithm
        End do
        Sn = N/(N2-1)*GminX*N2**Int(LOG10(b)/LOG10(real(N2)))
        if (nG1(nGxx+nGx,2)<a/2-Sn) then    !if there is room for bigger
grids
            IF (Int((a/2-Sn)/N2**(nGx-1)/GminX)>0) then
                nG1(nGxx+nGx+1,1) = Int((a/2-Sn)/N2**(nGx-1)/GminX)

                nG1(nGxx+nGx+1,2) = N2**(nGx-1)*GminX
                nG1(nGxx+nGx+2,1) = 1
                nG1(nGxx+nGx+2,2) = a/2-Sn-
nG1(nGxx+nGx+1,1)*nG1(nGxx+nGx+1,2)
                nGx = nGx +2
            ELSE
                nG1(nGxx+nGx+1,1) = Int((a/2-Sn)/nG1(nGxx+nGx,2))

                nG1(nGxx+nGx+1,2) = nG1(nGxx+nGx,2)
                nG1(nGxx+nGx+2,1) = 1
                nG1(nGxx+nGx+2,2) = a/2-Sn-
nG1(nGxx+nGx+1,1)*nG1(nGxx+nGx+1,2)
                nGx = nGx +2
            END IF
        else            !if left space <= biggest grid size
nG1(nGxx+nGx,2)
                nG1(nGxx+nGx+1,1) = 1
                nG1(nGxx+nGx+1,2) = a/2-Sn
                nGx = nGx +1
        end if

        Allocate(nG2(nGx,2))
        Do k =nGxx+1, nGxx+ nGx      !copy this piece of nG1 to nG2
                nG2(k-nGxx,1)= nG1(k,1)
                nG2(k-nGxx,2)= nG1(k,2)
        End Do

        CALL Quick_Sort( nG2, 1, nGx)      ! quick sort now
        Allocate(nGtemp(2*(nGx),2))                !prepare to mirror nG2
        Do k = 1, nGx
                nGtemp(k,1) = nG2(k,1)
                nGtemp(k,2) = nG2(k,2)
        End Do
!----------mirror and copy mirror to nGtemp()
        Do j=1, nGx
                nGtemp(nGx+j,1)=nGtemp(nGx-j+1,1)
                nGtemp(nGx+j,2)=nGtemp(nGx-j+1,2)
        end Do
!----------End of mirrorring ------------------
        Do k =nGxx+1, nGxx+2*nGx      !Copy nGtemp back to nG1
                nG1(k,1)= nGtemp(k-nGxx,1)
                nG1(k,2)= nGtemp(k-nGxx,2)
        End Do

        nGxx = nGxx + 2*nGx                 !commulative index for next
loop
        Deallocate (nGtemp)
        Deallocate (nG2)
!----------------------------------------
    else  !distance is great between two interfaces (a>2cc)
                !half space will be calculated
        nGx = Int(LOG10(Xmax/GminX)/LOG10(real(N2))) +2
        nG1(nGxx+1,1) = N
```

```fortran
            nG1(nGxx+1,2) = GminX
            Do j = 2, nGx
                  nG1(nGxx+j,1) = N
                  nG1(nGxx+j,2) = N2**(j-2)*GminX
            End do
            Sn = N/(N2-1)*GminX*N2**Int(LOG10(Xmax/GminX)/LOG10(real(N2)))+1
            If (a/2-Sn>=Xmax) then
                  nG1(nGxx+nGx+1,1) = Int((a/2-Sn)/Xmax)
                  nG1(nGxx+nGx+1,2) = Xmax
                  IF  ((a/2-Sn)/Xmax - Int((a/2-Sn)/Xmax) .NE. 0) then
                        nG1(nGxx+nGx+2,1) = 1
                        nG1(nGxx+nGx+2,2) = (a/2-Sn) - Xmax*Int((a/2-Sn)/Xmax)
                        nGx = nGx +2
                  ELSE
                        nGx = nGx +1
                  END IF
            else  !impossible case, since a>2*cc
                  If (Int((a/2-Sn)/Xmax) .NE. 0) then
                        nG1(nGxx+nGx+1,1) = 1
                        nG1(nGxx+nGx+1,2) = (a/2-Sn) - Xmax*Int((a/2-Sn)/Xmax)
                        nGx = nGx +1
                  End If
            End If

            Allocate(nG2(nGx,2))
            Do k =nGxx+1, nGxx+ nGx         !copy this piece of nG1 to nG2
                  nG2(k-nGxx,1)= nG1(k,1)
                  nG2(k-nGxx,2)= nG1(k,2)
            End Do

            CALL Quick_Sort( nG2, 1, nGx)      ! quick sort now
            Allocate(nGtemp(2*(nGx),2))                    !prepare to mirror nG2
            Do k = 1, nGx
                  nGtemp(k,1) = nG2(k,1)
                  nGtemp(k,2) = nG2(k,2)
            End Do
!----------mirror and copy mirror to nGtemp()
            Do j=1, nGx
                  nGtemp(nGx+j,1)=nGtemp(nGx-j+1,1)
                  nGtemp(nGx+j,2)=nGtemp(nGx-j+1,2)
            end do
!----------End of mirrorring ------------------
            Do k =nGxx+1, nGxx+2*nGx       !Copy nGtemp back to nG1
                  nG1(k,1)= nGtemp(k-nGxx,1)
                  nG1(k,2)= nGtemp(k-nGxx,2)
            End Do

            nGxx = nGxx + 2*nGx                        !commulative index for next
loop
            Deallocate (nGtemp)
            Deallocate (nG2)
      End if
End do
nGx = nGxx
! End of assigning adaptive grids in X direction
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!                                                                !
!                                                                !
! assign adaptive grids in Z direction  !ZZZ Grid-center Method
ZZZZZZZZZZZZZZZZZZZ
allocate (nG3(100*Zk,2))
nGxx = 0
cc= N*N2*Zmax/(N2-1)
Do i = 1, Zk-1
```

```fortran
        a = (LocaZ(i+1) - LocaZ(i))*Gz        !depth between layers

        If (a <= 2*cc) then
        !Zmax = 1 m. N is repeat #.
        if (a/2<=Gmin) then
                nG3(nGxx+1,1) = N                        !nG3(1,1) is # of grid
in the length
                nG3(nGxx+1,2) = a/2
                Sn = a/2
                nGz=1
        else
                b = a/2/N/Gmin
                nGz = Int(LOG10(b)/LOG10(real(N2)))+1      !half space will
be calculated
                nG3(nGxx+1,1) = N                        !nG3(1,1) is # of grid
in the length
                nG3(nGxx+1,2) = Gmin                     !of nG3(1,2)=Gmin.
                Do j = 2, nGz       !Serial start as 1*N*Gmin, N2**1*N*Gmin,
N2**2*N*Gmin,...
                        nG3(nGxx+j,1) = N
                        nG3(nGxx+j,2) = N2**(j-2)*Gmin
                End do
                Sn = N/(N2-1)*Gmin*N2**Int(LOG10(b)/LOG10(real(N2)))
        end if

        if (nG3(nGxx+nGz,2)<a/2-Sn) then
        !if there is room for bigger grids
                IF (Int((a/2-Sn)/N2**(nGz-1)/Gmin)>0) then
                ! if we can use a little bigger grid
                        nG3(nGxx+nGz+1,1) = Int((a/2-Sn)/N2**(nGz-1)/Gmin)
                        nG3(nGxx+nGz+1,2) = N2**(nGz-1)*Gmin
                        nG3(nGxx+nGz+2,1) = 1
                        nG3(nGxx+nGz+2,2) = a/2-Sn-
nG3(nGxx+nGz+1,1)*nG3(nGxx+nGz+1,2)
                        nGz = nGz +2
                ELSE
                        nG3(nGxx+nGz+1,1) = Int((a/2-Sn)/nG3(nGxx+nGz,2))
                        nG3(nGxx+nGz+1,2) = nG3(nGxx+nGz,2)
                        nG3(nGxx+nGz+2,1) = 1
                        nG3(nGxx+nGz+2,2) = a/2-Sn-
nG3(nGxx+nGz+1,1)*nG3(nGxx+nGz+1,2)
                        nGz = nGz +2
                End IF
        else            !if left space <= biggest grid size
nG3(nGxx+nGz,2)
                if (a/2 .NE. Sn) then
                        nG3(nGxx+nGz+1,1) = 1
                        nG3(nGxx+nGz+1,2) = a/2-Sn
                        nGz = nGz +1
                end if
        end if

        Allocate(nG2(nGz,2))
        Do k =nGxx+1, nGxx+ nGz                          !copy this piece of nG3
to nG2
                nG2(k-nGxx,1)= nG3(k,1)
                nG2(k-nGxx,2)= nG3(k,2)
        End Do

        CALL Quick_Sort( nG2, 1, nGz)       ! quick sort now
        Allocate(nGtemp(2*(nGz),2))                      !prepare to mirror nG2
        Do k = 1, nGz
                nGtemp(k,1) = nG2(k,1)
                nGtemp(k,2) = nG2(k,2)
```

```
            End Do
!----------mirror and copy mirror to nGtemp()
            Do j=1, nGz
                  nGtemp(nGz+j,1)=nGtemp(nGz-j+1,1)
                  nGtemp(nGz+j,2)=nGtemp(nGz-j+1,2)
            end Do
!----------End of mirrorring -----------------
            Do k =nGxx+1, nGxx+2*nGz        !Copy nGtemp back to nG3
                  nG3(k,1)= nGtemp(k-nGxx,1)
                  nG3(k,2)= nGtemp(k-nGxx,2)
            End Do

            nGxx = nGxx + 2*nGz                        !commulative index for next
loop
            Deallocate (nGtemp)
            Deallocate (nG2)
!----------------------------------------
      else   !distance is great between two interfaces (a>2cc)
            nGz = Int(LOG10(Zmax/Gmin)/LOG10(real(N2))) +2
            nG3(nGxx+1,1) = N
            nG3(nGxx+1,2) = Gmin
            Do j = 2, nGz
                  nG3(nGxx+j,1) = N
                  nG3(nGxx+j,2) = N2**(j-2)*Gmin
            End do
            Sn = N/(N2-1)*Gmin*N2**(Int(LOG10(Zmax/Gmin)/LOG10(real(N2)))+1)

            If (a/2-Sn>=Zmax) then
                  nG3(nGxx+nGz+1,1) = Int((a/2-Sn)/Zmax)
                  nG3(nGxx+nGz+1,2) = Zmax
                  If ((a/2-Sn)/Zmax - Int((a/2-Sn)/Zmax).NE. 0) then
                        nG3(nGxx+nGz+2,1) = 1
                        nG3(nGxx+nGz+2,2) = (a/2-Sn) - Zmax*Int((a/2-Sn)/Zmax)
                        nGz = nGz +2
                  else
                        nGz = nGz +1
                  END IF
            else   !impossible case, since a >2cc
                  If (Int((a/2-Sn)/Zmax) .NE. 0) then
                        nG3(nGxx+nGz+1,1) = 1
                        nG3(nGxx+nGz+1,2) = (a/2-Sn) - Zmax*Int((a/2-Sn)/Zmax)
                        nGz = nGz +1
                  End If
            End If

            Allocate(nG2(nGz,2))
            Do k =nGxx+1, nGxx+ nGz        !copy this piece of nG3 to nG2
                  nG2(k-nGxx,1)= nG3(k,1)
                  nG2(k-nGxx,2)= nG3(k,2)
            End Do

            CALL Quick_Sort( nG2, 1, nGz)        ! quick sort now
            Allocate(nGtemp(2*(nGz),2))                    !prepare to mirror nG2
            Do k = 1, nGz
                  nGtemp(k,1) = nG2(k,1)
                  nGtemp(k,2) = nG2(k,2)
            End Do
!----------mirror and copy mirror to nGtemp()
            Do j=1, nGz
                  nGtemp(nGz+j,1)=nGtemp(nGz-j+1,1)
                  nGtemp(nGz+j,2)=nGtemp(nGz-j+1,2)
            end Do
!----------End of mirrorring -----------------
            Do k =nGxx+1, nGxx+2*nGz        !Copy nGtemp back to nG3
```

```fortran
                nG3(k,1)= nGtemp(k-nGxx,1)
                nG3(k,2)= nGtemp(k-nGxx,2)
            End Do

            nGxx = nGxx + 2*nGz                    !commulative index for next
loop
            Deallocate (nGtemp)
            Deallocate (nG2)
        End if
End do
nGz = nGxx
! End of assigning adaptive grids in Z direction
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

OPEN (4, FILE = 'c:\meshxyz.txt')
WORD = 'NY'
write (4, 100) WORD, 1, 1.0
100 FORMAT(A2,3X,I5,E10.4)
WORD = 'NX'

Do i =1, nGx
write (4, 100) WORD, INT(nG1(i,1)), nG1(i,2)
End do

WORD = 'NZ'
Do i =1, nGz
write (4, 100) WORD, INT(nG3(i,1)), nG3(i,2)
End do

call GXYZ(nX, Gx, nZ, Gz, DataFile)

CONTAINS
  RECURSIVE SUBROUTINE Quick_Sort( X, L, R )
    REAL, DIMENSION(:,:), INTENT(INOUT) :: X ! 2D matrix, sort by the 2nd
vector
    INTEGER, INTENT(IN)                 :: L, R ! left, right bounds
    INTEGER L1, R1                              ! etc

    IF (L < R)THEN
      L1 = L
      R1 = R

      DO
        DO WHILE (L1 < R .and. X(L1,2) <= X(L,2))  ! shift L1 right
          L1 = L1 + 1
        END DO

        DO WHILE (L < R1 .and. X(R1,2) >= X(L,2))  ! shift R1 left
          R1 = R1 - 1
        END DO

        IF (L1 < R1) then
                CALL Swop( X(L1,2), X(R1,2) )       ! swop
                CALL Swop( X(L1,1), X(R1,1) )
          End if
        IF (L1 >= R1) EXIT                        ! crossover -
                                                  ! partition
      END DO

      CALL Swop( X(L,2), X(R1,2) )      ! partition with X(L) at R1
        CALL Swop( X(L,1), X(R1,1) )      ! partition with X(L) at R1

      CALL Quick_Sort( X, L, R1-1 ) ! now attack left subproblem
      CALL Quick_Sort( X, R1+1, R ) ! don't forget right subproblem
```

```fortran
      END IF
    END SUBROUTINE Quick_Sort

    SUBROUTINE Swop( A, B )
      REAL, INTENT(INOUT) :: A, B
      REAL              :: Temp

      Temp = A
      A = B
      B = Temp

    END SUBROUTINE Swop
!---------------------------eND OF qUICKsORT------------------------

10000 END

SUBROUTINE GXYZ(nna, a, nnb, b, DFile) !changed from GXYZ to GXYZ(D, a, b) by
ylm on 8/10/2004.
!
!-----PROGRAM FOR GENERATING 3-D X-Y-Z MESHES.
!
!     GRID BLOCK DIMENSIONS ARE DX, DY, AND DZ.
!
        INTEGER, INTENT(IN) ::  nna, nnb   ! nna is grid # in X, nnb in Z. By
ylm on 8/10/2004.
!       REAL, DIMENSION(:,:), INTENT(IN) :: D  ! D is DataSet().
        REAL, INTENT(IN) :: a, b                          !a is Gx (Grid size in
X). b is Gz.
        Character*60 DFile
        CHARACTER NA*1,NW*2,NXYZ*2
        CHARACTER*5 DOM,E, MatName(4)          !Material names by ylm on
8/11/2004.!9/18/2004.
        Integer I, J, K, ii, jj, kk       !By ylm on 8/11/2004.
        Integer TempX, TempZ              !denote mother block center in which
given block is.
        Integer NX, NY, NZ, ICALL, NO, N, NX1, NY1, NZ1, MAX
        REAL PI, DEL, RAD, BET, BETA, V, AIJ,DEG
        REAL, ALLOCATABLE:: DataSet(:,:)
        LOGICAL lBotBCFlag, lTopBCFlag    !denote the need of adding DRICHILET
bdries, 8/17/04
        LOGICAL lBotNBCFlag, lTopNBCFlag,lLTNBCFlag, lRTNBCFlag   !denote the
need of adding nEUMANN bdries, 9/13/04
        Real cutoff                 !denote cutoff value for geological media
dividing. ylm on Sept. 6, 2004
        COMMON/SECPAR/E(1)
        COMMON/N/NA(35)
!
!     BEGIN ADDITION FOR T2DM
!
        COMMON/NUMGBXYZ/NGB(3)
!
!     END OF ADDITION FOR T2DM.
!
        REAL DX(5000),DY(5000),DZ(5000)   !changed from 2500 to 5000 by ylm on
8/10/2004
        DIMENSION D1(2),D2(2),D3(2)
        DIMENSION NW(3)
!       Integer, ALLOCATABLE :: IMat(:)         !IMat stores material types
for each grid
                                                                !by ylm on
8/10/2004.
        SAVE ICALL,NW,NX,NY,NZ,PI
```

```
        DATA
NA/'1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M'

        DATA NW/'NX','NY','NZ'/
        DATA NX,NY,NZ/0,0,0/
        DATA PI/3.14159265/
        DATA ICALL/0/
        Max  = 300000
!----    dIRICHLET BDRY Conditions  ----------------------------
        lBotBCFlag = .False.        !No adding Bottom boundary, by ylm on
8/17/04.
        lTopBCFlag = .False.        !No adding top boundary
!----    Neumann No flux BDRY Conditions  ----------------------------
        lBotNBCFlag = .False.       !No adding Bottom boundary, by ylm on
9/13/04.
        lTopNBCFlag = .False.       !No adding top boundary
        lLTNBCFlag = .False.        !No adding Left boundary
        lRTNBCFlag = .False.        !No adding Right boundary
!---- , by ylm on 9/13/04        ----------------------------
        lBotBCFlag = .True.         !Now add bottom CONSTANT HEAD boundary
!        lTopBCFlag = .True.        !Now add top CONSTANT HEAD boundary
!        lTopNBCFlag = .True.       !Now add top No flux Neumann boundary
        lLTNBCFlag = .True.         !adding Left No flux Neumann boundary
        lRTNBCFlag = .True.         !adding Right No flux Neumann boundary, by
ylm on 9/13/04.
!----                            ----------------------------
        cutoff = 0.5+0.00001        !can be changed to anX number between 0~1,
same as used in krigging.
                                              !Here use it to aviod
geological material losses as real to integer transfermation.
        ICALL=ICALL+1
        MatName(4) = 'SALLU'  !Surfacial alluvium (Depth from 0~19m or 0~62ft,
9/18/2004
        MatName(1) = 'SAND1'  !Sediment by ylm on 8/11/2004.

        MatName(2) = 'BASLT'  !Basalt by ylm on 8/11/2004.
        MatName(3) = 'BOUND'  !Boundary material by ylm on 8/17/2004.
          ! Read krigged grid file (currently 2D) into matrix DataSet(i,j)
[j=1,2,3]
          ! and get NY (total grid in X direction), nZ, Gx (grid size), Gz.
          OPEN (1, FILE = DFile)
          ALLOCATE (DataSet(MAX/3,3))

          DO WHILE (.NOT. EOF(1))
               i=i+1
               READ( 1, *) (DataSet(i,j),j=1,3)
          END DO
          CLOSE (1)

          DEALLOCATE (DataSet)
          ALLOCATE (DataSet(i,3))
          OPEN (1, FILE = DFile)
          i=0
          DO WHILE (.NOT. EOF(1))
               i=i+1
               READ( 1, *) (DataSet(i,j),j=1,3)
          END DO
!         nTotal = i
          !-------------------------end of reading data----------------
        IF(ICALL == 1) WRITE(*,899)

  899 FORMAT(/6X,'GXYZ     1.0      16 June      1999',6X,'GENERATE 1, 2, OR
3-D CARTESIAN MESH')
!
```

```
!       DOM='    1'              by ylm on 8/11/2004.
        REWIND 4

!       READ 101,DEG                              !commented by ylm
!
        DEG = 0.0                                 !add by ylm @ 8/5/2004
  103 DO WHILE (.NOT. EOF(4))        !add by ylm @ 8/5/2004
            READ (4, 100) NXYZ,NO,DEL   !delete line # 103 before 'READ'
                                        !Following indented by ylm on
8/11/2004.
            100 FORMAT(A2,3X,I5,E10.4)
                DO 102 N=1,3
                IF(NXYZ.EQ.NW(N)) GOTO(111,112,113),N
            102 CONTINUE
                GOTO 120
          !
            111 NX1=NX+1
                NX=NX+NO
                IF(DEL.NE.0.) GOTO 104
                READ 101,(DX(I),I=NX1,NX)
            101 FORMAT(8E10.4)
                GOTO 103
            104 DO 105 I=NX1,NX
            105 DX(I)=DEL
                GOTO 103
          !
            112 NY1=NY+1
                NY=NY+NO
                IF(DEL.NE.0.) GOTO 106
                READ 101,(DY(I),I=NY1,NY)
                GOTO 103
            106 DO 107 I=NY1,NY
            107 DY(I)=DEL
                GOTO 103
          !
            113 NZ1=NZ+1
                NZ=NZ+NO
                IF(DEL.NE.0.) GOTO 108
                READ 101,(DZ(I),I=NZ1,NZ)
                GOTO 103
            108 DO 109 I=NZ1,NZ
            109 DZ(I)=DEL
                GOTO 103
        END DO                                    !Add by ylm
!
  120 CONTINUE
        RAD=PI*DEG/180.
        BET=SIN(RAD)
        BETA=COS(RAD)
!
!     BEGIN ADDITION FOR T2DM.
!
        NGB(1)=NX
        NGB(2)=NY
        NGB(3)=NZ
!
!     END OF ADDITION FOR T2DM.
!
!
!-----MAKE PRINTOUT OF MESH SPECIFICATIONS.
        NN=NX*NY*NZ     !Note: NY*NZ <= 26*35^2=31850 for 2D.
        !Note: NX*NZ<=35*99*26=90090. NX*NY*NZ<=26*99*35^2=3153150

        PRINT 40,NX,NY,NZ,NN,DEG
```

```
   40 FORMAT(/' GENERATE CARTESIAN MESH WITH NX*NY*NZ = ',I4,' *',I4,' *',I4,'
= ',I5,'  GRID BLOCKS'/1X,131('*')//'   THE X-AXIS IS HORIZONTAL; THE Y-AXIS
IS ROTATED BY ',E10.4,' DEGREES AGAINST THE HORIZONTAL'//'   THE GRID
INCREMENTS ARE AS FOLLOWS')
      PRINT 41,NX,(NA(I),I=1,9),(DX(I),I=1,NX)
   41 FORMAT(//'   DELTA-X  (',I4,'
INCREMENTS)'//13X,9(A1,10X),'10'//(10X,10(1X,E10.4)))

      PRINT 42,NY,(NA(I),I=1,9),(DY(I),I=1,NY)
   42 FORMAT(//'   DELTA-Y  (',I4,'
INCREMENTS)'//13X,9(A1,10X),'10'//(10X,10(1X,E10.4)))

      PRINT 43,NZ,(NA(I),I=1,9),(DZ(I),I=1,NZ)
   43 FORMAT(//'   DELTA-Z  (',I4,'
INCREMENTS)'//13X,9(A1,10X),'10'//(10X,10(1X,E10.4)))

      PRINT 44
   44 FORMAT(/' ',131('*'))
!
      ILOOP=0
   30 CONTINUE
      ILOOP=ILOOP+1
!
      IF(ILOOP.EQ.1) WRITE(4,20), NX, NY, NZ, NN
   20 FORMAT('ELEME---NX=', I4, '   NY=', I4, '   NZ=', I4, '   Total=', I9)
      IF(ILOOP.EQ.2) WRITE(4,21)
   21 FORMAT('CONNE')
!-------------------Following comments were created by ylm on
8/10/2004.-----------
      XI=DX(1)/2.    !XI denotes X-coordinate of the center of the 'block
I'
      DO 1 I=1,NX
      IM=MOD(I-1,99)+1      !IM denotes the left of I mod by 99.! Since
TOUGH uses
      NOVI=(I-1)/99         !NOVI denotes the times of I of 99. ! A__99
format for X.
      IF(I.GT.1) XI=XI+DX(I)/2.+DX(I-1)/2.
     ! A is leading # that
      I1=I+1
               ! can be combined with
      IM1=MOD(I1-1,99)+1
         ! the # from Y &/or Z.
      NOVI1=(I1-1)/99
      D1(1)=DX(I)/2.
      IF(I.LT.NX) D1(2)=DX(I1)/2.
!
      YJ=DY(1)/2.    !YJ denotes Y-coordinate of the center of the 'block
J'
      DO 1 J=1,NY
      JM=MOD(J-1,35)+1      !JM denotes the left of I mod by 35.! Since
TOUGH uses
      NOVJ=(J-1)/35         !NOVJ denotes the times of J of 35. ! A35__
format for Y & Z
      IF(J.GT.1) YJ=YJ+DY(J)/2.+DY(J-1)/2.
      J1=J+1
      JM1=MOD(J1-1,35)+1
      NOVJ1=(J1-1)/35
      D2(1)=DY(J)/2.
      IF(J.LT.NY) D2(2)=DY(J1)/2.
!
      ZK=-DZ(1)/2.
      DO 1 K=1,NZ
      KM=MOD(K-1,35)+1
      NOVK=(K-1)/35
```

66

```
          IF(K.GT.1) ZK=ZK-DZ(K)/2.-DZ(K-1)/2.
          K1=K+1
          KM1=MOD(K1-1,35)+1
          NOVK1=(K1-1)/35
!     SET CUMULATIVE "OVERFLOW" PARAMETERS, TO START AT NA(10) = 'A'
          NOV=NOVK+((NZ-1)/35+1)*NOVJ+((NZ-1)/35+1)*((NY-1)/35+1)*NOVI+10
          NOV1=NOVK+((NZ-1)/35+1)*NOVJ+((NZ-1)/35+1)*((NY-1)/35+1)*NOVI1+10
          NOV2=NOVK+((NZ-1)/35+1)*NOVJ1+((NZ-1)/35+1)*((NY-1)/35+1)*NOVI+10
          NOV3=NOVK1+((NZ-1)/35+1)*NOVJ+((NZ-1)/35+1)*((NY-1)/35+1)*NOVI+10
          D3(1)=DZ(K)/2.
          IF(K.LT.NZ) D3(2)=DZ(K1)/2.
!
          IF(ILOOP.EQ.2) GOTO 31
          V=DX(I)*DY(J)*DZ(K)
!
          AIJ=0.
          IF(K.EQ.1.) AIJ=AIJ+DX(I)*DY(J)
          IF(K.EQ.NZ) AIJ=AIJ+DX(I)*DY(J)
          !-----Find material of this block by ylm on 8/11/2004.---------
          TempX = Int(XI/a) +1   !changed from XI/a +0.5
          TempZ = Int(-ZK/b)     !changed from -ZK/b +0.5
          If ((-ZK < 19.0).and. (Int(DataSet(nna*TempZ + TempX,3))+1)==1) then
              DOM = MatName(4)  !Surfacial sediments, 9/18/2004
          Else
              DOM = MatName(Int(DataSet(nna*TempZ + TempX,3))+1)
          End if
!------    by ylm on 8/17/2004  for ELEME   --------------------
          If (lBotBCFlag == .True.) then
              If (K==NZ) then
                  DOM = MatName(3)  !add bottom boundary to ELEME section
                  V  = .1000E+51    !change volume to a huge number
              End If
          End if
          If (lTopBCFlag == .True.) then
              If (K==1) then
                  DOM = MatName(3)   !add top boundary to ELEME section
                  V  = .1000E+51    !change volume to a huge number
              End If
          End if
!------    end of modification  -------------------------
!     WRITE(4,10) NA(NOV),NA(KM),NA(JM),IM,DOM,V,AIJ,XI,YJ,ZK
      IF (K==1) WRITE(4,1022) NA(NOV),NA(KM),NA(JM),IM,"wat","COM1"
      1022 FORMAT(3A1,I2,A3,27X,A4)
   10 FORMAT(3A1,I2,10X,A5,2E10.4,10X,3F10.4)
          GOTO 1
   31 CONTINUE
!     ----   Writting CONNE section starts now-----------
          AYZ=DY(J)*DZ(K)
          AXZ=DX(I)*DZ(K)
          AXY=DX(I)*DY(J)
!------    by ylm on 8/18/2004 for CONNE  X-Z 2D case
-----------------------
!-------   Dirichlet Conditions   --------------------
          If (lBotBCFlag == .True.) then
              If (K==NZ-1) then            !@Bottom, TOUGH uses above to below
connection
                                           !(eg. NZ-1 grid to NZ grid)
                  D3(2) = 1.000E-10 !change the distance from bottom grid
                                           !to direct above one to very
small.
              End If
          End if
          If (lTopBCFlag == .True.) then
              If (K==1) then         !@Top
```

```
                      D3(1) = 1.000E-10 !change the distance from top grid
                                               !to direct below one to very
small.
                End If
          End if
!------    end of Dirichlet Conditions
-------------------------------------------
!------    end of modification for CONNE  X-Z 2D case
------------------------
!     If (lBotBCFlag .AND. lLTNBCFlag .AND. lRTNBCFlag) then      !by ylm on
08/18/2004, 9/14/04

                                                          !for this very special
case only.
!            If (K < NZ .and. I>1) then           !lateral connection deleted @
bottom bdry, Dirichlet and Neumann
!              IF(I<NX-1) WRITE(4,11)
NA(NOV),NA(KM),NA(JM),IM,NA(NOV1),NA(KM),NA(JM),IM1,NA(1),D1(1),D1(2),AYZ
!              END If
!        END If

!        IF(J.LT.NY) WRITE(4,11)
NA(NOV),NA(KM),NA(JM),IM,NA(NOV2),NA(KM),NA(JM1),IM,NA(2),D2(1),D2(2),AXZ,BET

!        If (lTopNBCFlag) then         ! lTopNBCFlag = .Ture., No flux BC at Top,
delete verticle connections between top and below layers.
!                 IF(K.LT.NZ .and. K>1) WRITE(4,11)
NA(NOV),NA(KM),NA(JM),IM,NA(NOV3),NA(KM1),NA(JM),IM,NA(3),D3(1),D3(2),AXY,BETA
!        Else
!                 IF(K.LT.NZ) WRITE(4,11)
NA(NOV),NA(KM),NA(JM),IM,NA(NOV3),NA(KM1),NA(JM),IM,NA(3),D3(1),D3(2),AXY,BETA
!do not delete connection, since there are source terms.
!        End if
!        !THE FOLLOWING IS FOR WRITTING BOTTOM CONNECTION OF GOFT SECTION
         IF(K.LT.NZ .and. K>NZ-2) WRITE(4,1122)
NA(NOV),NA(KM),NA(JM),IM,NA(NOV3),NA(KM1),NA(JM),IM
      1122 FORMAT(2(3A1,I2))
    11 FORMAT(2(3A1,I2),19X,A1,2F10.4, 2E10.4)    !FROM 4E10.4 TO 2F10.4,
2E10.4, YLM 9/28/04.
!
       1 CONTINUE
         WRITE(4,22)
    22 FORMAT('      ')
         IF(ILOOP.LT.2) GOTO 30

       CLOSE (1)
       RETURN
END SUBROUTINE GXYZ
```

2. MS Excel macros (for post-precessing, descriptions see headlines of each subroutine)

```
Sub coftx()
' This module is to calculate flow and search for characteristic times.
    Dim i As Integer
    Dim k As Integer
    Dim nY As Integer
    Dim nX As Integer                'denotes maximum column number
    Dim nXeff As Integer   'denotes column number at about 2000 m
    Dim BT As Integer              'BT denotes Break Thru time
    Dim EQ As Integer      'EQ denotes time to equilibrium

    Dim Totmass As Double
    Dim Totmasstemp As Double
```

```vba
    Dim Totmassgiven As Double     'denotes any given amount flow value to
find corresponding time
    Dim diffcells As Double        'denotes difference between to up and below
cells' values
    Dim Time As Double             'denotes time of given flow reached
    Totmassgiven = 2000000#        'kg water flow thru bottom bdry
    nX = 161
    nXeff = 132
'finding total time step
    j = 0
    For i = 1 To 3000
        If (Cells(i, 1).Value > 0#) Then
            j = j + 1
        End If
    Next i
    Nmax = j
    i = 0
    j = 0


'finding time step close to 0.5 yr
    k = 0
    For i = 2 To Nmax         '0.5 yr = 15768000 seconds
        If (Cells(i, 1) < 15768000) Then
            k = k + 1
        End If
    Next i
    nY = k + 1
'finding break thru time
    For j = 1 To Nmax
        For i = 2 To nX     'maximum column number
            If (Cells(j, i).Value < 0#) Then
                GoTo 10
            End If
        Next i
    Next j
10  BT = j
    Cells(Nmax + 6, 2).Value = "Maximum time step"
    Cells(Nmax + 6, 3).Value = Nmax
    Cells(Nmax + 7, 3).Value = "Time step"
    Cells(Nmax + 7, 4).Value = "In seconds"
    Cells(Nmax + 7, 5).Value = "In year"
    Cells(Nmax + 8, 2).Value = "Break Through time"
    Cells(Nmax + 8, 3).Value = BT
    Cells(Nmax + 8, 4).Value = Cells(BT, 1).Value     'seconds
    Cells(Nmax + 8, 5).Value = Cells(BT, 1).Value / (15768000 * 2) 'yr

'finding equilibrium time and step
    diffcells = 0#
    For j = Nmax - 1 To BT Step -1
        For i = 2 To nX
            diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1,
i).Value)
        Next i
        If (diffcells <> 0#) Then
            GoTo 5
        End If
        diffcells = 0#
    Next j
5   EQ = j + 1
    i = 0
    j = 0
    Cells(Nmax + 9, 2).Value = "Time to EQ"
    Cells(Nmax + 9, 3).Value = EQ
    Cells(Nmax + 9, 4).Value = Cells(EQ, 1).Value     'seconds
```

```
    Cells(Nmax + 9, 5).Value = Cells(EQ, 1).Value / (15768000 * 2) 'yr

'finding total flow from BT to 0.5 yr
    Totmass = 0#
    For k = BT To nY
        Totmasstemp = 0#
        For i = 2 To nXeff '163    'column number
            Totmasstemp = Totmasstemp + Cells(k, i).Value
        Next i
        Totmass = Totmass + Totmasstemp * (Cells(k + 1, 1).Value - Cells(k,
1).Value)
    Next k
    Totmass = Totmass * (-1)
    Cells(Nmax + 10, 2).Value = "Totmass1"
    Cells(Nmax + 10, 3).Value = Totmass
    Totmasstemp = 0#
    'calculate the flow difference to 0.5 yr.
    For i = 2 To nXeff
        Totmasstemp = Totmasstemp + Cells(nY, i).Value
    Next i
    Totmasstemp = Totmasstemp * (-1) * (15768000 - Cells(nY, 1).Value)
    Cells(Nmax + 11, 2).Value = "Totmass"
    Cells(Nmax + 11, 3).Value = Totmass + Totmasstemp
'finding time and step for given total flow amount across the bottom bdry
    Totmass = 0#
    For k = BT To Nmax - 1
        Totmasstemp = 0#
        For i = 2 To nXeff
            Totmasstemp = Totmasstemp + Cells(k, i).Value
        Next i
        Totmass = Totmass + Totmasstemp * (Cells(k + 1, 1).Value - Cells(k,
1).Value)
        If (Abs(Totmass) >= Totmassgiven) Then
            GoTo 20
        End If
    Next k
20   k = k - 1   'previous time step close to the objective time step
    Totmass = 0#
    For j = BT To k
        Totmasstemp = 0#
        For i = 2 To nXeff '163    'column number
            Totmasstemp = Totmasstemp + Cells(j, i).Value
        Next i
        Totmass = Totmass + Totmasstemp * (Cells(j + 1, 1).Value - Cells(j,
1).Value)
    Next j
    'determining complementary part of time for flow equal exactly
Totmassgiven
    Totmasstemp = 0#
    For i = 2 To nXeff
        Totmasstemp = Totmasstemp + Cells(k, i).Value
    Next i
    Time = (Totmassgiven - Abs(Totmass)) / ((-1) * Totmasstemp)
    Cells(Nmax + 12, 2).Value = "Time of given flow reached"
    Cells(Nmax + 12, 3).Value = (Cells(k, 1).Value + Time) / (15768000 * 2)
End Sub

=====================================================
Sub foft()
' This module is to calculate flow and search for characteristic times for
each observation points.
    Dim i As Integer
    Dim istart As Integer
    Dim j As Integer
```

```
    Dim nX As Integer          'denotes maximum column number
    Dim BT1 As Integer         'BT denotes Break Thru time
    Dim BT2 As Integer         'BT denotes Break Thru time
    Dim BT3 As Integer         'BT denotes Break Thru time
    Dim BT4 As Integer         'BT denotes Break Thru time
    Dim EQ1 As Integer         'EQ denotes time to equilibrium
    Dim EQ2 As Integer         'EQ denotes time to equilibrium
    Dim EQ3 As Integer         'EQ denotes time to equilibrium
    Dim EQ4 As Integer         'EQ denotes time to equilibrium

    Dim diffcells As Double          'denotes difference between to up and below
cells' values
    Dim Time As Double      'denotes time of given flow reached
    nX = 22
'finding total time step
    j = 0
    For i = 1 To 2000
        If (Cells(i, 1).Value > 0#) Then
            j = j + 1
        End If
    Next i
    Nmax = j
    istart = 4    'starting column
    i = istart
    j = 0
    diffcells = 0#
'finding break thru time for each obs. point.
'Since the foft format is fixed, special columns will be picked.
    For j = 1 To Nmax
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 10
        End If
        diffcells = 0#
    Next j
10  BT1 = j + 1
    i = i + 5
    diffcells = 0#
    For j = 1 To Nmax
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 11
        End If
        diffcells = 0#
    Next j
11  BT2 = j + 1
    i = i + 5
    diffcells = 0#
    For j = 1 To Nmax
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 12
        End If
        diffcells = 0#
    Next j
12  BT3 = j + 1
    i = i + 5
    diffcells = 0#
    For j = 1 To Nmax
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 13
        End If
        diffcells = 0#
```

71

```
    Next j
13  BT4 = j + 1
    diffcells = 0#
    Cells(Nmax + 6, 2).Value = "Maximum time step"
    Cells(Nmax + 6, 3).Value = Nmax
    Cells(Nmax + 7, 2).Value = "Break Through time"
    Cells(Nmax + 7, 3).Value = "Time step"
    Cells(Nmax + 7, 4).Value = "In seconds"
    Cells(Nmax + 7, 5).Value = "In year"
    Cells(Nmax + 7, 6).Value = "Time to EQ"
    Cells(Nmax + 9, 2).Value = "Obs#2"
    Cells(Nmax + 9, 3).Value = BT1
    Cells(Nmax + 9, 4).Value = Cells(BT1, 2).Value    'seconds
    Cells(Nmax + 9, 5).Value = Cells(BT1, 2).Value / 31536000        'yr
    Cells(Nmax + 10, 2).Value = "Obs#3"
    Cells(Nmax + 10, 3).Value = BT2
    Cells(Nmax + 10, 4).Value = Cells(BT2, 2).Value                  'seconds
    Cells(Nmax + 10, 5).Value = Cells(BT2, 2).Value / 31536000       'yr
    Cells(Nmax + 11, 2).Value = "Obs#4"
    Cells(Nmax + 11, 3).Value = BT3
    Cells(Nmax + 11, 4).Value = Cells(BT3, 2).Value  'seconds
    Cells(Nmax + 11, 5).Value = Cells(BT3, 2).Value / 31536000       'yr
    Cells(Nmax + 8, 2).Value = "Obs#1"
    Cells(Nmax + 8, 3).Value = BT4
    Cells(Nmax + 8, 4).Value = Cells(BT4, 2).Value  'seconds
    Cells(Nmax + 8, 5).Value = Cells(BT4, 2).Value / 31536000  'yr
'finding equilibrium time and step
    i = istart
    For j = Nmax - 1 To BT1 Step -1
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 5
        End If
        diffcells = 0#
    Next j
5   EQ1 = j + 1
    i = i + 5
    diffcells = 0#
    For j = Nmax - 1 To BT2 Step -1
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 4
        End If
        diffcells = 0#
    Next j
4   EQ2 = j + 1
    i = i + 5
    diffcells = 0#
    For j = Nmax - 1 To BT3 Step -1
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 3
        End If
        diffcells = 0#
    Next j
3   EQ3 = j + 1
    i = i + 5
    diffcells = 0#
    For j = Nmax - 1 To BT4 Step -1
        diffcells = diffcells + Abs(Cells(j, i).Value - Cells(j + 1, i).Value)
        If (diffcells <> 0#) Then
            GoTo 2
        End If
        diffcells = 0#
```

```
    Next j
2    EQ4 = j + 1

    Cells(Nmax + 9, 6).Value = "OB#2"
    Cells(Nmax + 9, 7).Value = EQ1
    Cells(Nmax + 9, 8).Value = Cells(EQ1, 2).Value  'seconds
    Cells(Nmax + 9, 9).Value = Cells(EQ1, 2).Value / 31536000        'yr
    Cells(Nmax + 10, 6).Value = "OB#3"
    Cells(Nmax + 10, 7).Value = EQ2
    Cells(Nmax + 10, 8).Value = Cells(EQ2, 2).Value  'seconds
    Cells(Nmax + 10, 9).Value = Cells(EQ2, 2).Value / 31536000        'yr
    Cells(Nmax + 11, 6).Value = "OB#4"
    Cells(Nmax + 11, 7).Value = EQ3
    Cells(Nmax + 11, 8).Value = Cells(EQ3, 2).Value    'seconds
    Cells(Nmax + 11, 9).Value = Cells(EQ3, 2).Value / 31536000         'yr
    Cells(Nmax + 8, 6).Value = "OB#1"
    Cells(Nmax + 8, 7).Value = EQ4
    Cells(Nmax + 8, 8).Value = Cells(EQ4, 2).Value       'seconds
    Cells(Nmax + 8, 9).Value = Cells(EQ4, 2).Value / 31536000         'yr


End Sub
```
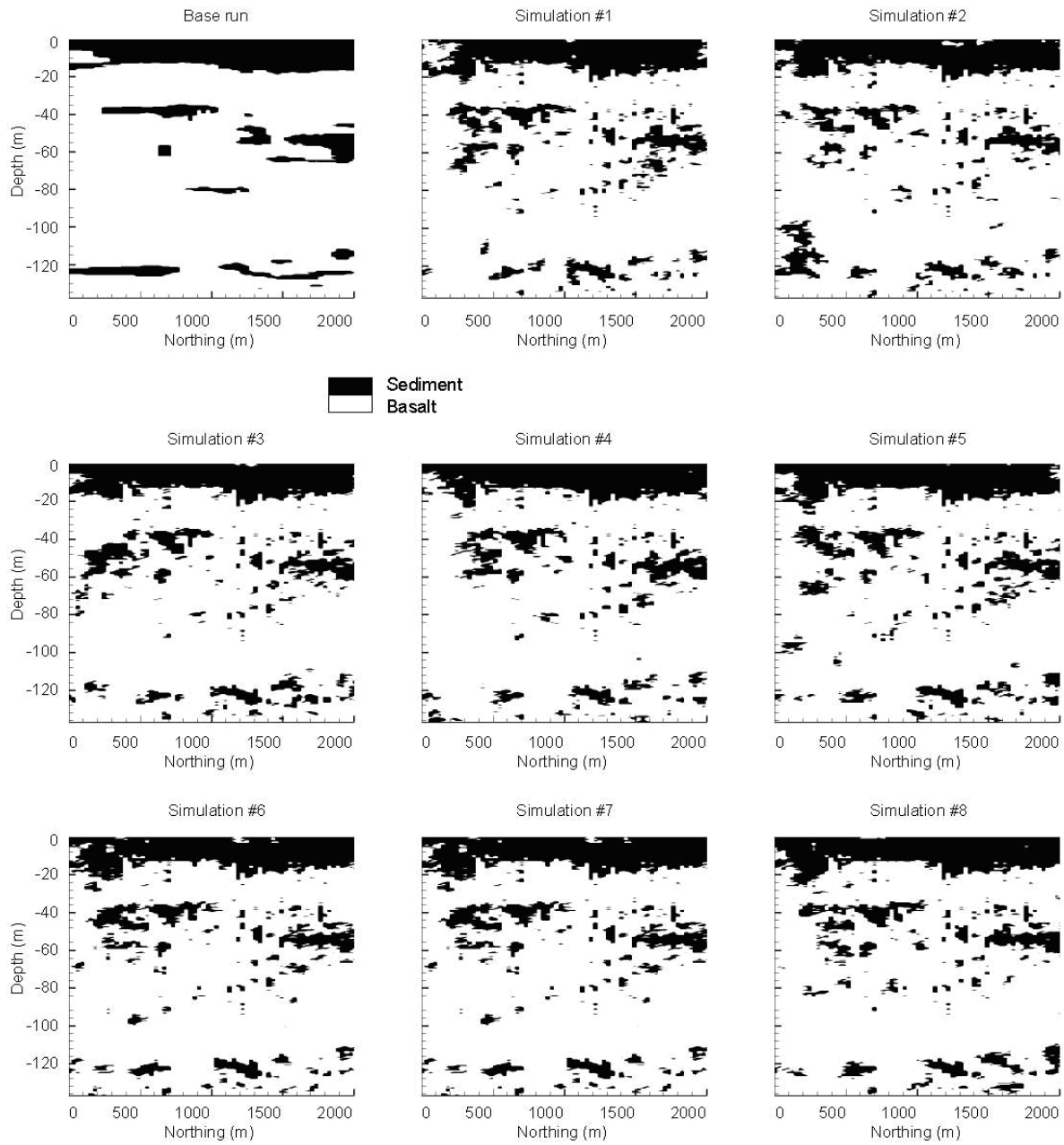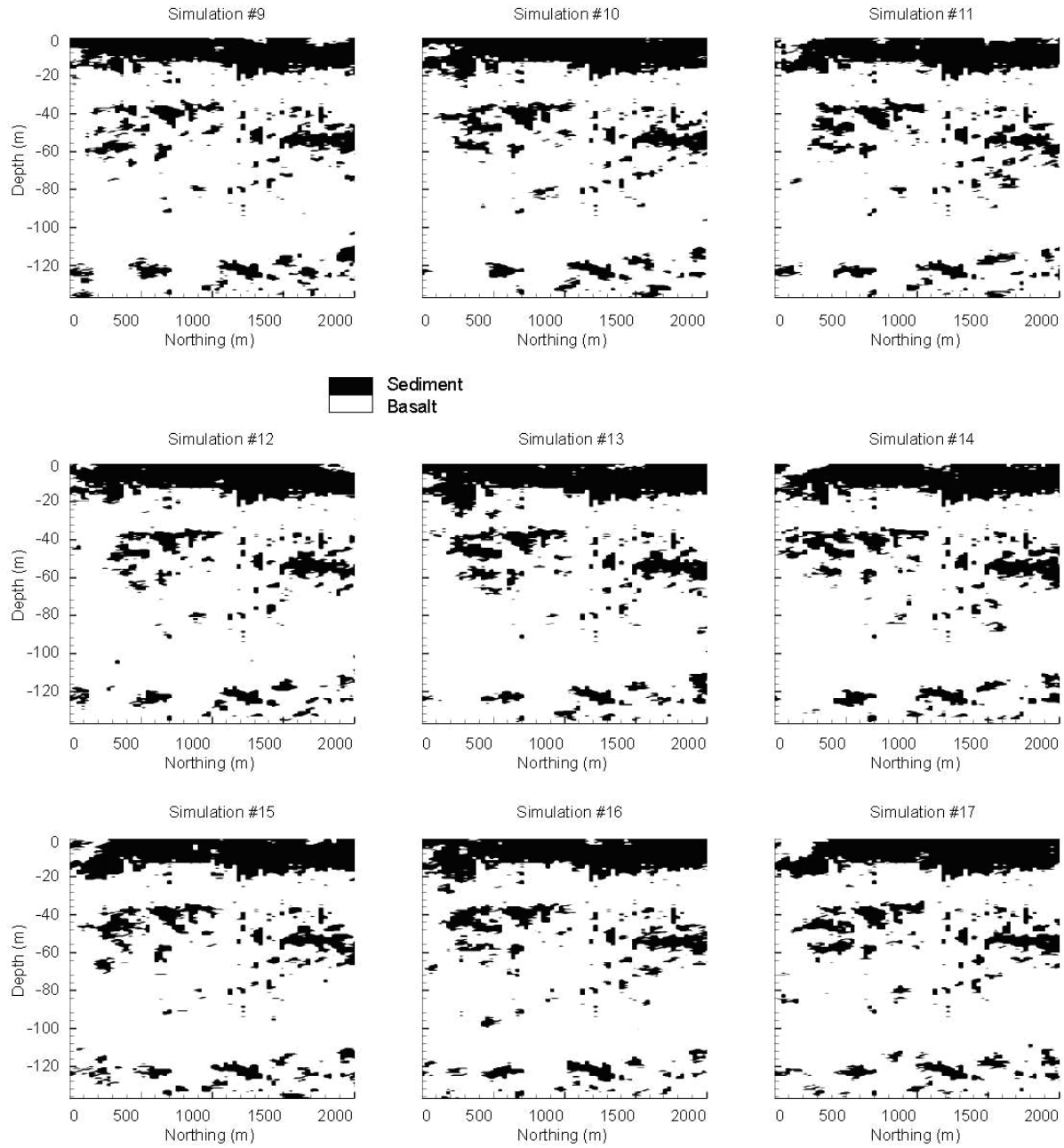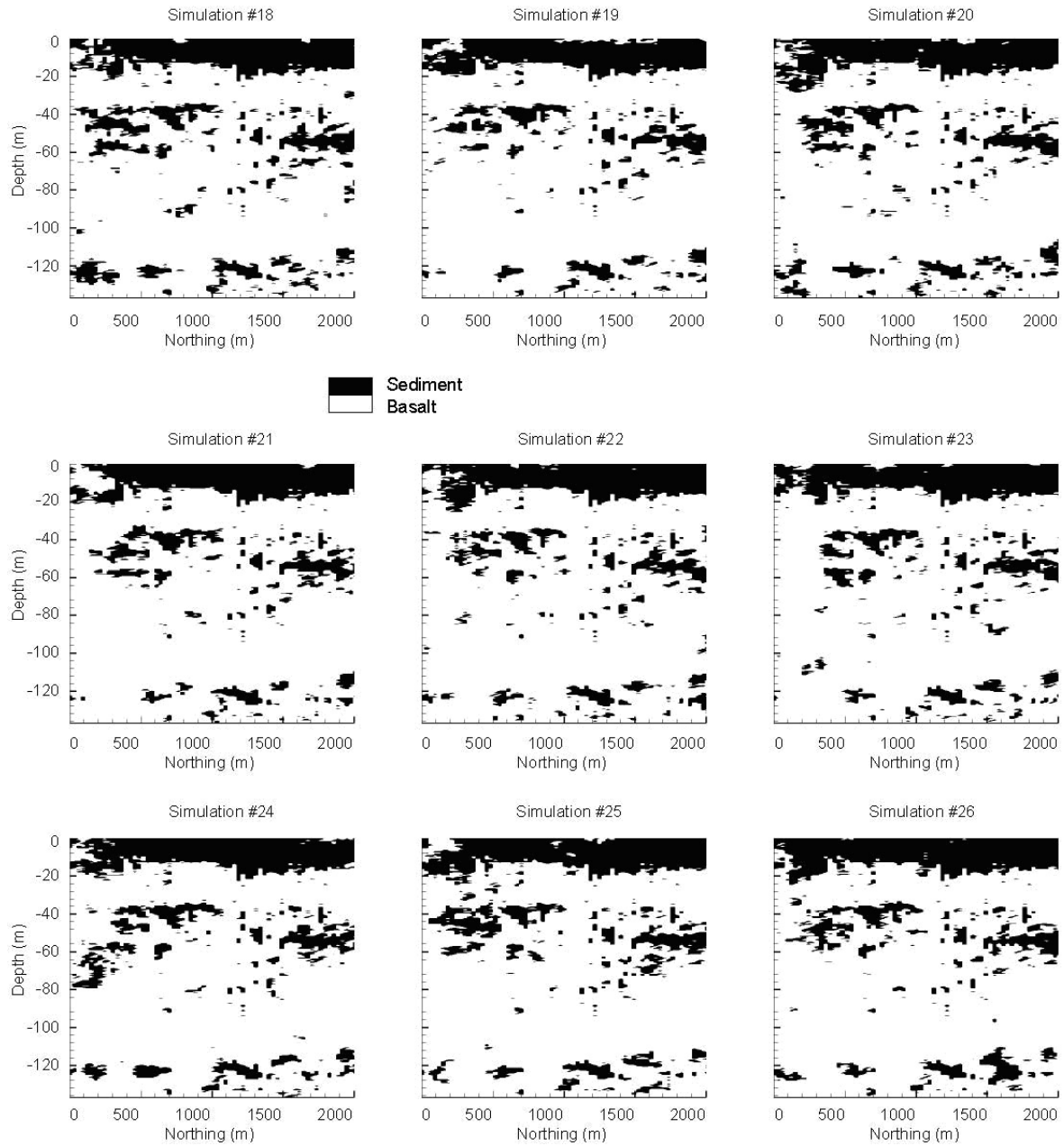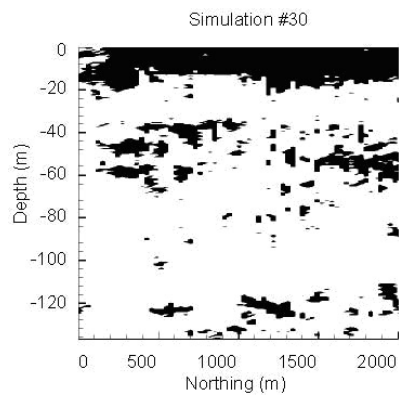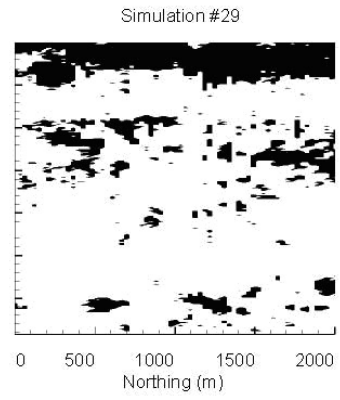
G. CONDITIONAL SIMULATIONS
1. All 30 generated cross-sections' maps

Simulation #9   Simulation #10   Simulation #11

Sediment
Basalt

Simulation #12   Simulation #13   Simulation #14

Simulation #15   Simulation #16   Simulation #17

75

Simulation #27

Simulation #28

Simulation #29
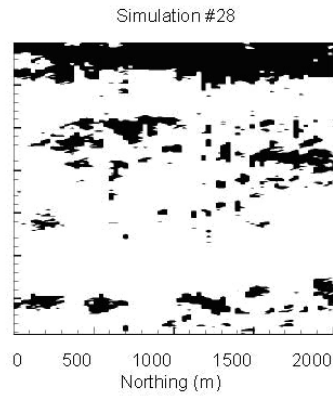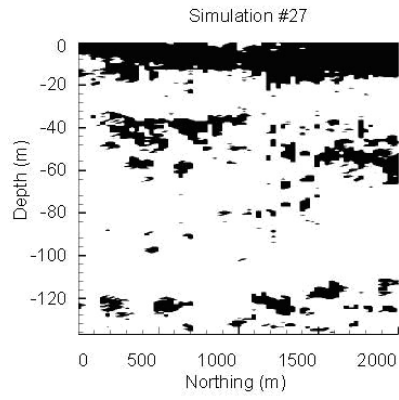
Simulation #30

Sediment
Basalt

2. First four moisture contour maps of 30 simulations