

# Final Report

## Optimal Search Strategy for the Definition of a DNAPL Source

SERDP Project ER-1347

AUGUST 2009

George Pinder  
**University of Vermont**

James Ross  
**University of Vermont**

Zoe Dokou  
**University of Vermont**

Distribution Statement A: Approved for Public Release,  
Distribution is Unlimited



Strategic Environmental Research and  
Development Program

This report was prepared under contract to the Department of Defense Strategic Environmental Research and Development Program (SERDP). The publication of this report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official policy or position of the Department of Defense. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Department of Defense.

## Abstract

DNAPL (Dense Non-Aqueous Phase Liquid) contamination poses a major threat to the groundwater supply; thus, successful remediation of the contaminated sites is of paramount importance. Delineating and removing the DNAPL source is an essential step that renders remediation successful and lowers the estimated remediation time and cost significantly.

This work addresses the issue of identifying and delineating DNAPL at its source. The methodology employed here is based upon the rapidly evolving realization that it is unlikely to identify and adequately define the extent of a DNAPL source location using field techniques and strategies that focus exclusively on directly locating separate phase DNAPL.

The goal of this work is to create an optimal search strategy in order to obtain, at least cost, information regarding a DNAPL source location. The concept is to identify, prior to a detailed site investigation, where to initially sample the subsurface to determine the DNAPL source characteristics and then to update the investigative strategy in the field as the investigation proceeds.

The search strategy includes a stochastic groundwater flow and transport model that is used to calculate the concentration random field and its associated uncertainty. The model assumes a finite number of potential source locations. Each potential source location is associated with a weight that reflects our confidence that it is the true source location. After a water quality sample is selected, an optimization algorithm is employed that finds the optimal set of magnitudes that corresponds to the set of potential source locations.

The simulated concentration field is updated using the real data and the updated plume is compared to the individual plumes (that are calculated using the groundwater flow and transport simulator considering only one source at a time). The comparison provides new weights for each potential source location. These weights define how the concentration realizations calculated by the stochastic groundwater flow and transport model will be combined. The higher the weight for a specific source location, the more concentration realizations generated by this source will be included in the calculation of the mean concentration field. The steps described above are repeated until the weights stabilize and the optimal source location is determined.

The algorithm has been successfully tested using various synthetic example problems of increasing complexity. The effectiveness of the search strategy in identifying a DNAPL source at two field sites is also demonstrated. The sites chosen for the test are the Anniston Army Depot (ANAD) in Alabama and Hunters Point Shipyard in California. The contaminant of interest at both sites is trichloroethene (TCE).

## Table of Contents

1. Objective .....	10
1.1. Overview.....	10
2. Background.....	12
2.1. Source identification background.....	12
2.1.1. Source identification problem types .....	12
2.1.1.1. Reconstruction of source release history .....	12
2.1.1.2. Identification of source location or release time of contaminant.....	13
2.1.1.3. Identification of source location and magnitude.....	14
2.1.1.4. Identification of source location and release time of contaminant .....	14
2.1.1.5. Identification of location, magnitude of source and release time of contaminant.....	15
2.2. Forward vs. backward models.....	15
2.3. Brief introduction and background of tools used in this work .....	16
2.3.1. Random field generation – Latin hypercube sampling.....	16
2.3.2. Kalman filter.....	17
2.3.3. Monotone measures and Choquet Integral .....	17
3. Methods.....	19
3.1. Motivation .....	19
3.2. Assumptions .....	19
3.3. Methodology overview .....	19
3.4. Mathematical toolbox .....	22

3.4.1. Initial weighting of potential source locations - Choquet integral .....	22
3.4.1.1. Application for synthetic examples .....	23
3.4.2. Flow and transport equations.....	27
3.4.3. Random hydraulic conductivity field generation – Latin hypercube sampling .....	28
3.4.3.1. Statistical definitions.....	28
3.4.3.2. Variogram analysis .....	30
3.4.3.3. Latin hypercube sampling.....	31
3.4.4. Concentration plume statistics calculation .....	33
3.4.5. Water quality sampling location selection.....	33
3.4.5.1. Linear Kalman filter.....	34
3.4.6. Optimization problem – solving for the source strength .....	39
3.4.6.1. Optimization problem formulation .....	40
3.4.7. Comparison of composite and individual plumes – $\alpha$ -cut method .....	42
3.4.8. Iteration procedure.....	44
4. Results and Discussion .....	45
4.1 Synthetic example.....	45
4.2. Sensitivity analysis results.....	52
5. Field Applications.....	54
5.1. Anniston Army Depot .....	54
5.1.1. Site description .....	54
5.1.2. Groundwater flow and transport model.....	57
5.1.3. Source search algorithm .....	59
5.1.4. Test results .....	63
5.2. Hunters Point Shipyard.....	70
5.2.1. Site description .....	70
5.2.2. Hydrogeologic characterization.....	72

5.2.3. Groundwater flow and transport model .....	73
5.2.4. Source search algorithm application.....	74
5.2.5. Test results .....	76
6. Conclusions.....	80
6.1. Summary.....	80
6.2. Conclusions .....	80
6.3. Contributions to the field.....	81
6.4. Future work.....	81
References.....	82
Appendix A: List of Publications .....	94
Appendix B: DNAPL Source Finder Code Documentation .....	95

## List of Tables

Table 1. Distances and corresponding membership degrees for all potential source locations and all features.....	25
Table 2. Partial and global scores for each potential source location.....	27
Table 3. Choquet integral results for 15 preliminary potential source locations.....	63
Table 4. Sampling sequence information .....	63
Table 5. Available water quality measurements and their locations in the vicinity of Building 134; greyed out wells provided infeasible solutions and were eliminated from consideration.....	75
Table 6. The order in which water quality data were selected reveals a proclivity of the source finder to select water quality samples nearer to potential sources. ....	78

## List of Figures

Figure 1. Flow chart of the source search algorithm .....	22
Figure 2. Location of manufacturing facility, waste dump and potential source locations (not to scale).....	24
Figure 3. Membership function representing the meaning of ‘near the manufacturing facility or waste dump.....	25
Figure 4. Membership function representing the meaning of ‘near’ water table.....	25
Figure 5. Three important model variogram types: spherical, Gaussian and exponential. ....	30
Figure 6. Intervals used with a Latin hypercube sample in terms of a normal probability density function.....	32
Figure 7. Intervals used with a Latin hypercube sample in terms of a normal cumulative distribution function.....	32
Figure 8. Strategy for the selection of a water quality sampling location. ....	34
Figure 9. Kalman filter as part of the search algorithm. ....	34
Figure 10. Normalized concentration plume presented as a fuzzy set and its 0.5 $\alpha$ -cut.	43
Figure 11. Comparison of $\alpha$ -cuts. The common area of the 0.4 $\alpha$ -cuts is shown in purple. ....	44
Figure 12. a) Synthetic aquifer for example 1, b) Potential water quality sampling locations. ....	45
Figure 13. True plume generated by a single realization of hydraulic conductivity for single source problem. ....	46
Figure 14. Simulated plume obtained using the initial source location weights for single source problem.....	47
Figure 15. Updated plumes and obtained weights after taking 1 concentration sample for single source problem.....	47



Figure 16. Updated plumes and obtained weights after taking 2 concentration samples for single source problem.....	48
Figure 17. Updated plumes and obtained weights after taking 3 concentration samples for single source problem.....	48
Figure 18. Updated plumes and obtained weights after taking 4 concentration samples for single source problem.....	49
Figure 20. Updated plumes and obtained weights after taking 6 concentration samples for single source problem.....	50
Figure 21. Updated plumes and obtained weights after taking 7 concentration samples for single source problem.....	50
Figure 22. Individual plumes of mean concentration for each potential source location for single source problem.....	51
Figure 23. Contaminant concentration uncertainty after taking each sample for single source problem.....	52
Figure 25. SWMU 12 location (black rectangle) and model domain (red boundary) (After SAIC, 2006). .....	56
Figure 26. Potentiometric map (After SAIC, 2006). .....	57
Figure 27. Vertical discretization of the model domain. ....	57
Figure 28. Finite element grid and location of monitoring wells. Green circles represent wells screened in the residuum interval and blue circles wells screened at the weathered bedrock interval. ....	58
Figure 29. Flow field results for stochastic model (colored contours) and potentiometric map created by hydrogeologist using well water level measurements (black contours). .....	59
Figure 30. Preliminary potential source locations. ....	60
Figure 31. Membership function for ‘close’ to the SWMU 12 boundary. ....	61
Figure 32. Membership function for ‘close’ to the high soil concentration locations... ..	61

Figure 33. Membership function for ‘close’ to the average TCE contour greater than 10,000 µg/L.....	61
Figure 34. Locations with high soil concentrations (red blocks).....	62
Figure 36. Search algorithm results for case 2 – real data before taking any samples. .	64
Figure 37. Search algorithm results for case 2 – real data after taking 1 sample. ....	65
Figure 38. Search algorithm results for case 2 – real data after taking 2 samples. ....	65
Figure 39. Search algorithm results for case 2 – real data after taking 3 samples. ....	66
Figure 40. Search algorithm results for case 2 – real data after taking 4 samples. ....	66
Figure 41. Search algorithm results for case 2 – real data after taking 5 samples. ....	67
Figure 42. Search algorithm results for case 2 – real data after taking 6 samples. ....	67
Figure 43. Search algorithm results for case 2 – real data after taking 7 samples. ....	68
Figure 45. Search algorithm results for case 2 – real data after taking 9 samples. ....	69
Figure 46. Pumping well drawdown area (After SAIC, 2006) .....	70
Figure 47. Hunters Point Shipyard is located on San Francisco Bay in southern San Francisco; image courtesy of (SulTech, 2008) .....	71
Figure 48. RU-C5 is the most northwestern remedial unit at Hunters Point Shipyard; Building 134 is located in the center of RU-C5; image courtesy of TetraTech (TetraTech, 2004) .....	72
Figure 49. The flow and transport model of Hunters Point Shipyard was comprised of 6 mathematical layers and 1054 nodes; boundary conditions were specified to be either constant head or no flow. ....	73
Figure 50. A potentiometric map drawn from 2002 measurements reveals unique head contours (blue lines) and suggested groundwater flow directions (blue arrows). ....	74
Figure 51. Calibrated model hydraulic heads correspond to measurement-based head contours very well.....	75

Figure 52. Originally, 13 small areas around the sump and dip tank were considered as possible locations for the true TCE source. ....	76
Figure 53. Search algorithm results – after taking one sample; concentration in $\mu\text{g/L}$ . ....	77
Figure 54. Search algorithm results –after taking two samples (same results after taking 3 through 5 samples) ; concentration in $\mu\text{g/L}$ .....	77
Figure 55. Search algorithm results – after taking six samples (remains unchanged for samples 7 through 10) ; concentration in $\mu\text{g/L}$ .....	78
Figure 56. Measurements of TCE in groundwater are predominantly located below and around the sump and dip tank. ....	79

## 1. Objective

This work addresses the issue of identifying and delineating DNAPL at its source. More specifically the goal of this work is to create an optimal search strategy to obtain, at least cost, information regarding a DNAPL source magnitude and location. The concept is to identify, prior to a detailed site investigation, where to initially sample the subsurface to determine the DNAPL source characteristics and then to update the sampling strategy in the field as the investigation proceeds. The overall technical objective of this project is to develop, test and evaluate a computer assisted analysis algorithm to help groundwater professionals identify, at least cost, the location, magnitude and geometry of a DNAPL source.

The technical approach of this work is based upon the rapidly evolving realization that it is unlikely to identify and adequately define the extent of a DNAPL source location using field techniques and strategies that focus exclusively on directly locating separate phase DNAPL. In essence, the target DNAPL is generally too small and filamentous to be identified efficiently via borings or geophysical methods, even using state of the art techniques. On the other hand, the plume emanating from a DNAPL source is typically quite large and consequently easily discovered, although identification of its extent and its concentration topology may, depending upon the nature of the groundwater flow field, require the collection of considerable field data. Water quality, lithological and permeability information constitute the primary field data used in this work.

### 1.1. Overview

Chapter 2 is comprehensive literature review of research related to source identification problems. A distinction between four different source identification problem types is made and two modeling approaches (forward vs. backward models) are presented and compared. The second part presents a literature review on the various tools used in this work.

Chapter 3 provides a detailed presentation of the methodology employed in this work. An extensive overview of the various tools used in the search algorithm is provided along with a flow diagram of the sequence of steps involved.

Chapter 4 is devoted to the demonstration of the effectiveness of the proposed DNAPL search strategy by the use of various synthetic example problems. These problems include a single source homogeneous aquifer, the addition of a pumping well, multiple true DNAPL sources, larger DNAPL source targets and two dimensional and three dimensional problems. Chapter 4 also includes a sensitivity analysis of various input parameters such as: the initial weights that correspond to each potential source location, the actual true source location chosen for the synthetic examples, the hydraulic conductivity correlation length, the number of Monte Carlo simulations and the weights of importance that correspond to features related to the selection of the optimal water quality sampling location and the number and type of  $\alpha$ -cuts used at the plume comparison step of the algorithm. The above parameters are described in detail in Chapter 3.

Chapter 5 describes the application of the proposed methodology to the field. Two real world problems were used as ‘blind tests’ of the proposed algorithm. The sites chosen for the implementation of the search algorithm are the Anniston Army Depot (ANAD) and Hunters Point Shipyard (HPS), located in northeast Alabama and San Francisco, California, respectively. The results and challenges of the field application are presented and discussed in Chapter 5. Conclusions resulting from the various synthetic and field applications are presented in Chapter 6.

## 2. Background

In this chapter, a comprehensive literature review is provided that is comprised of two parts. The first part offers a review of past and current approaches for groundwater contaminant source identification. The second part provides background knowledge on the various tools that were used in this work.

### 2.1. Source identification background

In recent years, hydrogeologists have focused a lot of attention on the problem of groundwater contaminant source identification. There are three important questions that need to be answered regarding a contaminant source. When was the contaminant released from the source (release history)? Where is the contamination source (source location)? At what concentration was the contaminant released from the source (source magnitude)? Depending on which of these questions one tries to answer, there exist different types of source identification problems.

#### 2.1.1. Source identification problem types

##### *2.1.1.1. Reconstruction of source release history*

One type of problem that has been extensively studied in past years is the reconstruction of contaminant source release history. In this case, the contaminant source location is assumed known and researchers seek to identify the release time of the contaminant as well as the magnitude of the source.

One of the very first attempts to reconstruct the release history of a contaminant source was performed by Skaggs and Kabala (1994). They applied a method called Tikhonov Regularization (TR) to solve a one dimensional, saturated, homogeneous aquifer problem with a complex contaminant release history. In their work they assumed no prior knowledge of the release function. Their method was found to be highly sensitive to errors in the measurement data. Liu and Ball, (1999) tested Skaggs and Kabala's method at a low permeability site at Dover Air Force Base, Delaware. They performed tests for two primary contaminants, PCE and TCE, and found that the results matched the measured data well in most cases. Skaggs and Kabala (1998) used Monte Carlo numerical simulations to determine the ability to recover various 'test functions'. These test functions were designed to provide insight into the effect of transport parameters on the ability to recover the true source release history.

Skaggs and Kabala (1995) applied a different method called Quasi-Reversibility (QR) to the same problem and argued that it is potentially superior to the TR approach because of its improved computational efficiency, its easier implementation and the fact that it allows for space and time dependent transport parameters. However, the results showed that the above advantages of the QR method come at the expense of accuracy.

An inverse problem approach was proposed by Woodbury and Ulrych (1996) that uses a statistical inference method called Minimum Relative Entropy (MRE). The authors applied this method to the same problem as Skaggs and Kabala (1994) and demonstrated that, for noise-free data, the reconstructed plume evolution history matched

the true history very well. For noisy data, their technique was able to recover the salient features of the source history.

Neupauer et al. (2000) evaluated the relative effectiveness of the TR and MRE methods in reconstructing the release history of a conservative contaminant in a one-dimensional domain. They concluded that in the case of error-free concentration data, both techniques perform well in reconstructing a smooth source history function. In the case of error-free data the MRE method is more robust than TR when a non-smooth source history function needs to be reconstructed. On the other hand, the TR method proved to be more efficient in the case of data that contain measurement error.

Snodgrass and Kitanidis (1997) developed a probabilistic method for source release history estimation that combines Bayesian theory with geostatistical techniques. The efficiency of their method was tested for transport in a simple, one-dimensional, homogeneous medium and it produced a best estimate of the release history and a confidence interval. Their method is an improvement on previous solutions to the source identification problem because it is more general, it incorporates uncertainty and it makes no assumptions about the nature and structure of the unknown source function.

Another approach in recovering the source release history was developed by Alapati and Kabala (2000). A non-linear least-squares (NSL) method without regularization was applied to the same problem addressed earlier by Skaggs and Kabala (1994). The performance of the method was affected mostly by the amount of noise in the data and the extent to which the plume is dissipated. In the case of a gradual source release, the NSL method was found to be extremely sensitive to measurement errors; however, it proved effective in resolving the release histories for catastrophic release scenarios, even for data with moderate measurement errors.

#### *2.1.1.2. Identification of source location or release time of contaminant*

Another type of source delineation problem is the identification of the location or release time of the source. Wagner (1992) developed a strategy that performs simultaneous parameter estimation and contaminant source characterization by solving the inverse problem as a non-linear maximum likelihood estimation problem. In the examples presented, the unknown source parameter estimated was the contaminant flux at given locations and over specific times.

Wilson and Liu (1994) used a heuristic approach to solve the stochastic transport differential equations backwards in time. They obtained two types of probabilities: location and travel time probabilities. Liu and Wilson (1995) extended their previous study to a two-dimensional heterogeneous aquifer. Their results were very similar to those obtained by traditional forward-in-time methods. Neupauer and Wilson (1999) proposed the use of the adjoint method as a formal approach for obtaining backward probabilities and verified the results of the study by Wilson and Liu (1994). Neupauer and Wilson (2001) extended their previous work to multidimensional systems and later applied their methodology to a TCE plume at the Massachusetts Military Reservation (Neupauer and Wilson, 2005). Under the assumption that their model is properly calibrated, their results verify the existence of the two suspected contamination sources and suggest that one or more additional sources is likely. Recently, Neupauer and Lin (2006) extended the work by Neupauer and Wilson (1999, 2001, and 2005) by

conditioning the backward probabilities on measured concentrations. The results show that when the measurement error is small and as long as the samples are taken from throughout the plume, the conditioned probability density functions include the true source location or the true release time.

#### *2.1.1.3. Identification of source location and magnitude*

A third type of source identification problem involves the simultaneous identification of the source location and magnitude, which is the type of problem addressed in this work. Among the first to attempt solving this type of source identification problem were Gorelick et al. (1983). Their strategy involves forward-time simulations coupled with a linear programming model or least squares regression. In their work, they assumed no uncertainty in the physical parameters of the aquifer. Their source identification models were tested for two different problems, a steady state and a transient case. The method was found to be successful in solving both problems, in the presence of minimal measurement errors in the first problem, and when there was an abundance of data in the second problem. Datta et al. (1989) employed a statistical pattern recognition technique to solve problems similar to those considered by Gorelick et al. (1983) and found that it required less data than the optimization approach to achieve similar results.

Another study whose goal was to identify the location and magnitude of the contamination source was recently performed by Mahinthakumar and Sayeed (2005). They compared several popular optimization methods and proved that a hybrid genetic algorithm – local search approach was more effective than using individual approaches, identifying the source location and concentration to within 1% of the true values for the hypothetical, single source identification problems they investigated.

One recently proposed approach in identifying the source location and recovering the concentration distribution of contaminant sources is that of Hayden et al. (2007). Their strategy involves the use of an extended Kalman filter in conjunction with the adjoint state method and was successfully applied in both experimental and synthetic problems.

#### *2.1.1.4. Identification of source location and release time of contaminant*

Another type of source characterization problem targets the identification of both the source location and release time of the contaminant of interest. Atmadjia and Bagtzoglou (2001) tackled this problem by using a method called Marching – Jury Backward Beam Equation (MJBBE) to solve the inverse problem. Using examples involving deterministic heterogeneous dispersion coefficients, the authors were able to reconstruct the time history and spatial distribution of a one-dimensional plume. Baun and Bagtzoglou (2004) extended the aforementioned study by coupling the MJBBE method with Discrete Fourier Transform processing techniques to significantly improve the computational efficiency of the method and enhanced it by implementing an optimization algorithm to overcome difficulties associated with the ill-posed nature of the inverse problem. They applied their method to a two-dimensional, advection-dispersion problem with homogeneous and isotropic coefficients. Their results showed that even when only one measurement location is available, as long as it is close to the centroid of the plume, the



algorithm will perform very well. They also noted that the results become less reliable as one goes further into the past.

#### *2.1.1.5. Identification of location, magnitude of source and release time of contaminant*

The final and most challenging category of source characterization problems is the simultaneous identification of all three source characteristics (location, magnitude and release time). Mahar and Datta (1997) formulated a methodology that combines an optimal groundwater quality monitoring network design and an optimal source identification model. Their results show that the addition of an optimally designed monitoring network to the existing network of monitoring wells improves the source identification model results. Mahar and Datta (2000) applied a non-linear optimization model with embedded flow and transport simulation constraints to solve an inverse transient transport problem. They found that the estimated source fluxes differ from the true ones by approximately 10% in the case of no missing data and 30% in the case of missing data. One of their most important observations was the fact that results were best when the observation wells were located downstream in close proximity to the sources.

Aral et al. (2001) used a progressive genetic algorithm (PGA) to solve the optimization problem. Their method proved to be very computationally efficient and it was successfully applied on a single-source identification problem in a heterogeneous aquifer. The authors observed that the measurement errors affected the reconstruction of the source release history more than they affected the source location identification.

The interested reader is referred to Morrison et al. (2000) and Atmadja and Bagtzoglou (2001) for an extensive literature review of methods that focus on groundwater contaminant source identification.

## **2.2. Forward vs. backward models**

Source locations and historical contaminant release histories are assumed in this discussion to be unknown inputs to the groundwater contaminant transport model. Therefore, the source identification problem is a problem whose solution requires the collection of contaminant concentration data from monitoring wells. Groundwater contaminant transport is an irreversible process because of its dispersive nature. This makes modeling contaminant transport backwards in time an ill-posed problem. Ill-posed problems exhibit discontinuous dependence on data and high sensitivity to measurement errors. A problem is considered ill-posed if its solution does not satisfy the following conditions: existence, uniqueness and stability. In the case of a source identification or release history problem, the condition of existence is satisfied since the contamination has to originate from someplace. Thus, researchers have to deal with the issues associated with instability and non-uniqueness.

There are two different approaches to solving the source identification problem. One approach aims to solve the differential equations backwards in time (inverse problem) by using techniques that will overcome the problems of non-uniqueness and instability. These techniques include: the random walk particle method (Bagtzoglou et al., 1991, 1992), the Tikhonov regularization method (Skaggs and Kabala, 1994), the quasi-reversibility technique (Skaggs and Kabala, 1995), the minimum relative entropy method

(Woodbury and Ulrych, 1996), the Bayesian theory and geostatistical techniques (Snodgrass and Kitanidis, 1997), the adjoint method (Neupauer and Wilson, 1999, Hayden et al., in review, Li et al., 2007), the non-linear least-squares method (Alapati and Kabala, 2000), the marching-jury backward beam equation method (Atmadjia and Bagtzoglou, 2001) and the genetic algorithm (Aral et al., 2001; Mahinthakumar and Sayeed, 2005).

A very different approach to solving the source identification problem is a simulation-optimization approach, which couples a forward-time contaminant transport simulation model with an optimization technique. The work presented here employs a simulation-optimization model. Some of the optimization techniques included in this category are: linear programming and least squares regression analysis (Gorelick et al., 1983), non-linear maximum likelihood estimation (Wagner, 1992), and statistical pattern recognition (Datta et al., 1989). This approach avoids the problems of non-uniqueness and stability associated with formally solving the inverse problem but the iterative nature of the simulation model usually requires increased computational effort. Mahar and Datta (1997, 2000) used non-linear programming with an embedding method that eliminates the necessity of external simulation since the governing equations of flow and solute transport are directly incorporated in the optimization model as binding constraints. The use of artificial neural networks (Singh et al., 2004; Li et al., 2006) offers an alternative way of simulating the model results which proves to be very computationally effective. Mirghani et al., (2006) proposed a grid-enabled simulation-optimization approach as a method to solve problems that require a large number of model simulations.

### **2.3. Brief introduction and background of tools used in this work**

A stochastic groundwater flow and transport model lies at the foundation of the methodology employed in this work. The crux of this model is a random hydraulic conductivity field, whose generation requires the availability of field data. Usually the available information on the model parameters is limited, thus the hydrogeologic parameters are associated with considerable uncertainty. The stochastic groundwater flow and transport model, with uncertain hydraulic conductivity, provides the means for generating a random contaminant concentration field. There are many different techniques for achieving this; perturbation methods, stochastic equation methods and Monte Carlo methods are among the most popular ones. Herrera (1998) provides a comprehensive review of these methods. The Monte Carlo approach is the method used in this work. Recently, there was a new method developed by Kunstmann et al. (2002), called first-order second moment (FOSM) that reduces the computational effort required by the Monte Carlo approach, but its application is restricted to a very limited uncertainty space (Wu and Zheng, 2004).

The Monte Carlo simulation method has become increasingly more appealing due to its easy implementation combined with the development of faster computers. One of the most important steps of the Monte Carlo approach is the selection of a random field generation technique.

#### **2.3.1. Random field generation – Latin hypercube sampling**

In past years, various random field generators have been developed, including: 1) the turning bands algorithm (Matheron, 1973; Journel and Huijbregts, 1978); 2) spectral decomposition methods (Mejia and Rodriguez-Iturbe, 1974; Gutjahr, 1989 and Robin et al., 1993); covariance decomposition based methods, such as LU decomposition (Davis, 1987; Alabert, 1987)) and Latin hypercube sampling (McKay et al., 1979; Zhang and Pinder, 2003); 3) kriging and sequential simulation based methods, such as sequential Gaussian simulation; 4) optimization based methods, such as simulated annealing (Goovaerts, 1997).

The Latin hypercube sampling (Lhs) algorithm is the random field generator used in this work. The Latin hypercube sampling technique was first introduced by McKay et al., 1979. Their algorithm was later combined with a distribution free approach to induce a desired rank correlation among the input variables (Iman and Conover, 1982). The Latin hypercube sampling strategy is a stratified sampling technique where the assumed probability density function is divided into a number of non-overlapping, equal-probability intervals. Samples are taken, one from each area, and they are permuted in a way such that the correlation of the field is accurately represented. The effectiveness of Lhs as a hydraulic conductivity random field generator was demonstrated in the work of Zhang (2002) and Zhang and Pinder (2003).

### 2.3.2. Kalman filter

The Kalman filter is an optimal linear estimator whose use in this work is twofold: 1) it provides a means of quantifying the concentration field uncertainty reduction that results from taking a groundwater quality sample and 2) it performs the updating of the mean and covariance matrix of the concentration random field after taking a contaminant concentration sample.

Since Kalman (1960) first described his filtering technique, it has been applied to various fields, especially in control systems engineering. Although its potential application to groundwater modeling has long been recognized (McLaughlin, 1976; Bras, 1978), Kalman filtering was seldom applied to groundwater problems (van Geer, 1987; Graham and McLaughlin, 1989) until the early nineties. Since then, the Kalman filter has been successfully used in groundwater problems to improve prior state estimates of hydraulic head (Zhou et al., 1991; Graham and Tankerskey, 1993; Ross et al, 2006, 2008) and contaminant concentration (Yu et al., 1989; Graham and McLaughlin, 1989; Zou and Parr, 1995). The Kalman filter has also been used as a parameter estimation tool by Ferraresi et al., (1996) and Eppstein and Dougherty, (1996). There have been many applications of the filter in optimal design of long term monitoring networks (Zhou et al., 1991; Andrisevic, 1993, Herrera, 1998; Rizzo et al., 2000; Zhang, 2002).

For an extended discussion of the Kalman filter derivation, use and applications the interested reader is referred to Jazwinski (1970).

### 2.3.3. Monotone measures and Choquet Integral

Since Sugeno (1974) introduced the concept of monotone measures and integrals, they have gone through important development, both from a theoretical and applied point of view. From an applied point of view, monotone measures can be considered in two ways:

1) as a general uncertainty measure; 2) as a tool for representing weights (or importance) of groups of elements (Dubois et al., 1996). In this work, we use the second interpretation of the monotone measures and combine them with a Choquet integral.

A Choquet integral is an innovative aggregation operator that has been successfully applied to multicriteria decision making, pattern recognition, image processing, etc. The main advantage of using a Choquet integral, instead of the traditional weighted arithmetic mean, lies in its ability to represent the interaction between criteria in a flexible way (Marichal, 2000).

The Choquet integral is used here as a tool for assigning initial weights to the potential DNAPL source locations and for selecting new measurement locations. In the former case, these weights represent our confidence that a potential source location is the true one, and in the latter case they assign a degree of importance to the selection of a sample near the source as compared to the reduction in overall concentration distribution uncertainty. A detailed description of the theory of the Choquet integral and its applications can be found in Dubois et al., 1996; Grabisch, 1996, 2000; Klir et al., 1997; Marichal, 2000; Dubois and Prade, 2004.

## 3. Methods

### 3.1. Motivation

The current approach to locating DNAPL sources in contaminated field sites is a heuristic combination of expert opinion, computer simulation of potential sources and institutional knowledge. The source search algorithm presented here combines these elements into an integrated optimal predictor of the DNAPL source location.

The specific goal of this work is to identify the source of DNAPL contamination using a search algorithm that exploits the observation that plumes emanating from a DNAPL source are typically quite large and consequently easily discovered, as opposed to the actual DNAPL source targets. This algorithm seeks to identify the DNAPL source location by using the least amount of water quality data. Such an algorithm can assist groundwater professionals in identifying and dealing with DNAPLs. If the correct DNAPL source location is identified and removed from the site, the remediation and monitoring costs are significantly reduced.

### 3.2. Assumptions

The basic assumptions used in this work are the following:

1. A groundwater plume has been identified and a preliminary field investigation has been conducted.
2. There is reason to believe that the plume is generated by a suspected DNAPL source.
3. Enough hydrological site information on the site exists to construct a groundwater flow and transport model, assuming that the hydraulic conductivity is known with uncertainty.
4. The primary introduction of uncertainty in the transport equation is the velocity due to uncertain hydraulic conductivity values; that is the porosity, dispersivity, retardation and chemical reaction are assumed to be deterministic.

### 3.3. Methodology overview

This section provides an overview of the search algorithm methodology and a brief description of the various tools used in this work. The specific mathematical tools will be described in detail in a following section. The proposed algorithm includes the following steps:

1. Assembly of all available hydrogeological field information: The proposed strategy depends on the construction of a groundwater flow and transport model that exhibits the degree of heterogeneity and parameter uncertainty known or estimated to exist at the target site location. Boring logs, slug tests, cone penetrometers measurements and pumping test information, from which one can derive permeability estimates, constitute the necessary data base for generating the hydraulic conductivity field required by the model.
2. Approximate source location estimation: Based upon available field information, an approximate location of the DNAPL source is assumed and a probability of

- occurrence is associated with it. The methodology for deriving the distribution function representing the source involves the use of fuzzy logic. Subjective and objective information is combined to create a membership function that describes the degree of truth regarding the location of the source at a particular geographical point. Various physical attributes, such as the distance of the potential source locations to a waste-water lagoon, are quantified using expert opinion and fuzzy logic. The combined effect of each attribute in establishing the initial representation of the location of the approximate source target is obtained using a variant on the Choquet integral.
3. Hydraulic conductivity field generation: To model this system, a Monte-Carlo technique is used wherein realizations of the random hydraulic conductivity field are required. While there are several techniques available to generate realizations from random field statistics we are using a Latin hypercube sampling strategy that accommodates correlated random fields (see Zhang and Pinder, 2003).
  4. Construction of a groundwater flow and transport model of the site: Using the available hydrogeological information, a groundwater flow and transport model code that utilizes a random field representation of hydraulic conductivity and an uncertain source location and strength is created. The flow and transport model we employed for the purpose of this research is the Princeton Transport Code (PTC) which describes three-dimensional saturated flow and mass transport in the presence of a water table.
  5. Concentration plume statistics calculation: A Monte Carlo approach is used to produce the concentration distribution in this system. The Monte Carlo approach involves the creation of a set of realizations of the concentration field, each generated by a hydraulic conductivity realization and source location. The process involves, for each realization, the solution of the groundwater flow and transport equations. The concentration results for each realization and each nodal location are recorded and one can calculate the statistics for each nodal location (that is the mean and variance of the specified species concentration). We will call the resulting mean concentration field the 'composite plume'. One can also use the concentration values at all model nodes to obtain the spatial covariance or correlation matrix.
  6. Sampling location selection: Given the modeled concentration statistics, which are dependent upon field and possibly anthropogenic information regarding the source location, we are now at the point of incorporating any water quality data. There are two important factors that affect the decision on where to collect a concentration sample. The first factor is the reduction in the overall uncertainty that results from taking a sample at a particular location. A Kalman filter is the tool used to determine the impact of sampling at a particular location on the overall uncertainty of the concentration field. It uses the fact that the uncertainty at any point where a sample is taken reduces to the sampling error. The second important factor is the distance of the sampling well from the source location. It is in our interest to choose sampling locations that are closer to the source areas. These two important features are combined using a Choquet integral (as noted above, this is a kind of a distorted weighted average) to produce a score for each

- potential sampling location. The location with the largest score is selected as the optimal sampling point.
7. Source strength determination: A linear optimization problem is solved that seeks to find the set of source strengths that minimizes the summation of the absolute differences between modeled concentration values and measured concentration values at the sampling locations. The flow and transport simulator is coupled with the optimizer by a response matrix that contains the information of how the concentration values at the sampling locations change with unit changes of the magnitudes at the potential sampling locations. After the optimal values for the source magnitudes have been selected, the simulated concentration field (composite plume) is modified to reflect the change in source strength.
  8. Updating the simulated concentration field using real data: After a sample is taken the Kalman filter is used again to update the concentration mean and variance-covariance matrix using the real data.
  9. Comparison of composite with individual plumes: We return now to the source location alternatives. A concentration random field that considers the updated source magnitudes is produced for each different source alternative using the Monte Carlo approach and the field statistics are calculated. Each individual source location plume is compared to the updated composite plume using the method that involves the use of fuzzy sets and their  $\alpha$ -cuts. This strategy finds the degree of similarity between each individual potential source location plume and the composite plume by calculating a measure of the common area between the two plumes weighted by the value of their  $\alpha$ -cut. In other words, the greater the membership value (see below) of a plume at a point, the more weight that is given to the degree of overlap at that point. The larger the common area between the two plumes, the larger the degree of similarity. This degree of similarity is normalized and assigned as a new weight to each potential source location.
  10. Repetition of steps 5-9: The procedure of obtaining the concentration field is followed using the new weights and then a second sample is taken (after the mean and variance-covariance matrix of the plume have been updated with the first sample using the Kalman Filter) at a location that will reduce the new total uncertainty the most while taking into account the proximity of the sampling point to the potential source locations. The process is repeated until convergence on an optimal location and source strength is achieved.

The methodology described above is summarized in the flow diagram presented in Figure 1.

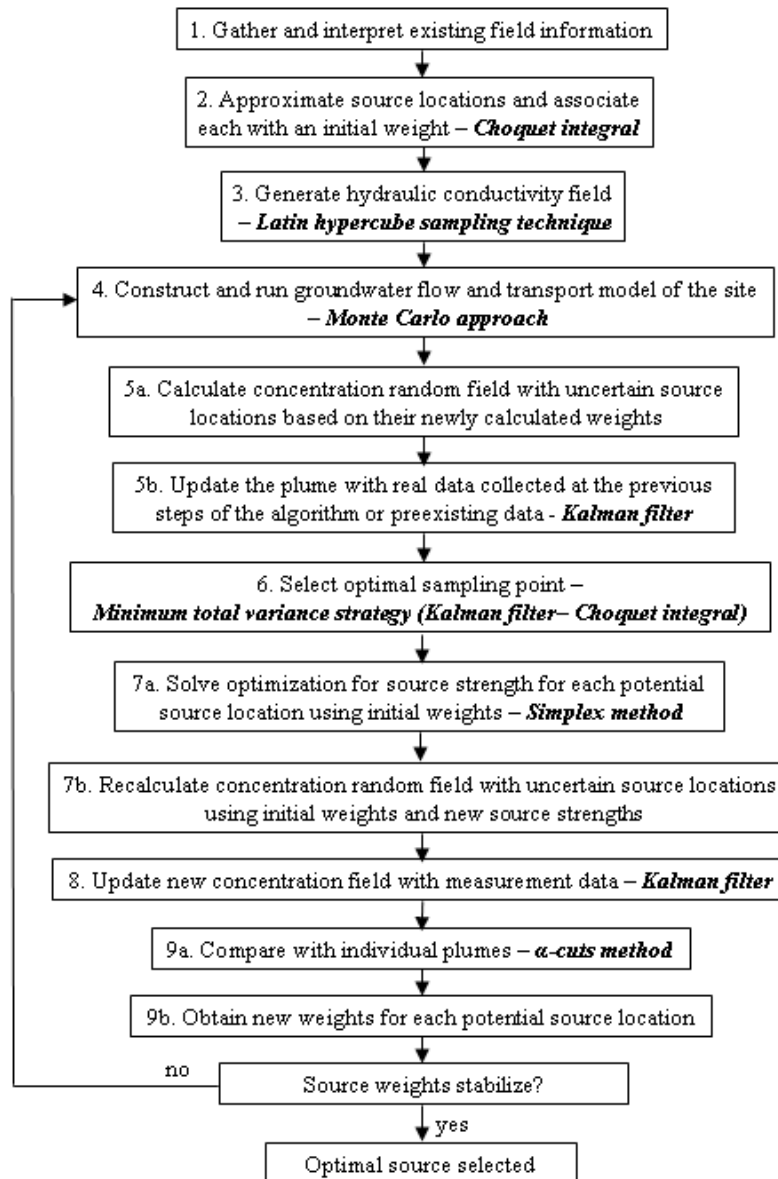


Figure 1. Flow chart of the source search algorithm

### 3.4. Mathematical toolbox

This section provides a detailed description of the various tools introduced in the algorithm steps presented in the previous section. It also explains how these tools were incorporated into the search strategy.

#### 3.4.1. Initial weighting of potential source locations - Choquet integral

As mentioned in step 2, a number of potential DNAPL source locations are identified and each is associated with an initial weight that reflects our confidence that it is the true



source location. These initial weights are determined using a variant on the Choquet integral.

The most commonly used operator to aggregate criteria in decision-making problems is the traditional weighted arithmetic mean. In many cases however, the considered criteria interact. The Choquet integral provides a flexible way to extend the weighted arithmetic mean for the aggregation of interacting and uncertain criteria. To calculate the Choquet integral, we need to define some measure of the importance of each criterion we are considering (Marichal, 2000). A formal way of capturing that importance is the use of ‘monotone measures’.

Let us now provide some important definitions:

**Definition 1.** Let  $A$  be a *fuzzy set* of some set of the universe  $X$ .  $A$  is defined as a function such that for any  $x \in X$ , it assigns a degree of membership,  $m$ ,  $A(x) = m_A(x)$ , where  $m_A(x) \rightarrow [0, 1]$ .

**Definition 2.** Let us denote by  $X = \{x_1, \dots, x_n\}$  the set of elements and  $\mathcal{P}(X)$  the power set of  $X$ , that is the set of all subsets of  $X$ . A *monotone measure*, on  $X$  is a set function  $\mu: \mathcal{P}(X) \rightarrow [0, 1]$ , satisfying the following axioms:

- (i)  $\mu(\emptyset) = 0$ ,  $\mu(X) = 1$  ( $\emptyset$  : empty set)
- (ii)  $A \subset B \subset X$  implies  $\mu(A) \leq \mu(B)$

In this context,  $\mu(A)$  represents the importance of the feature (or group of features)  $A$ . Thus, in addition to the usual weights on criteria taken separately, weights on any combination of criteria need to be defined as well.

Monotone measures can be:

- 1) *additive*, if  $\mu(A \cup B) = \mu(A) + \mu(B)$  whenever  $A \cap B = \emptyset$ ,
- 2) *superadditive*, if  $\mu(A \cup B) \geq \mu(A) + \mu(B)$  whenever  $A \cap B = \emptyset$ ,
- 3) *subadditive*, if  $\mu(A \cup B) \leq \mu(A) + \mu(B)$  whenever  $A \cap B = \emptyset$ .

Note that in the case of an additive measure, it suffices to define the  $n$  weights:  $\mu(\{x_1\}), \dots, \mu(\{x_n\})$  to define the measure entirely, but in general, one needs to define the  $2^n$  coefficients corresponding to the  $2^n$  subsets of  $X$ .

We introduce now the concept of a discrete Choquet integral.

**Definition 3.** Let  $\mu$  be a monotone measure on  $X$ . The *discrete Choquet integral* of a function  $f: X \rightarrow \mathcal{R}$  with respect to  $\mu$  is defined by:

$$\int f d\mu := \sum_{i=1}^n (f(x_{(i)}) - f(x_{(i-1)})) \mu(A_{(i)})$$

where  $*(i)$  indicates that the indices have been permuted so that:

$$0 \leq f(x_{(1)}) \leq \dots \leq f(x_{(n)}) \leq 1 \quad \text{and} \quad A_{(i)} := \{x_{(i)}, \dots, x_{(n)}\}.$$

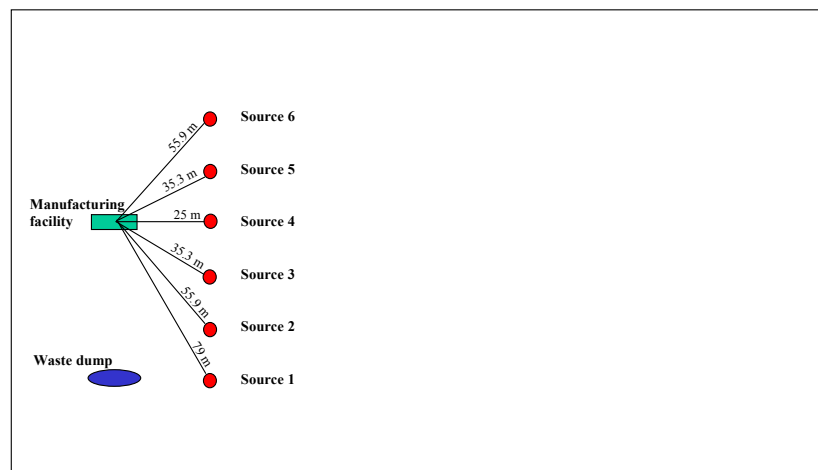
In our framework, the set  $X$  of elements is the set of identifying features of the source: a monotone measure  $\mu$  on  $X$  will represent the importance of each feature or of every group of features, and the Choquet integral will perform a kind of average of all partial scores, taking into account the importance of all groups of features.

The definitions presented above can be found in Dubois and Prade (2004). For more information on the information fusion technique, fuzzy sets, monotone measures and the Choquet integral, the reader is directed to Klir et al. (1997), Klir and Yuan (1995), Grabisch (1996).

### 3.4.1.1. Application for synthetic examples

We will now present an example of how the initial weights for each potential source location are obtained for the synthetic example problems presented later in this work. There are six potential source locations considered in the synthetic examples. Each possible source location is described by a three-dimensional vector, whose coordinates are values of the identifying features of the source. For the synthetic examples presented in this work, those features include: the source location proximity to a manufacturing facility (A), the proximity to a waste dump (B), and the distance of the water table from the ground surface (C).

In Figure 2, one can see the model domain with the locations of the manufacturing facility (green rectangle), the waste dump (blue oval shape) and the potential source locations (red circles). The distances of all the potential sources to the manufacturing facility are also shown.



**Figure 2. Location of manufacturing facility, waste dump and potential source locations (not to scale).**

All the features mentioned above describe a measure of distance. Thus for each feature, a membership function capturing the meaning of ‘near’ is provided by an expert and it is used to obtain the membership degree of each feature value for the particular site. The membership functions for each of the three features used in this example are presented in Figure 3 and Figure 4.

The distances from the manufacturing facility (shown in Figure 2), from the waste dump and to the water table are measured for each of the six potential source locations. Given the distance measurements and using the membership functions provided by the site expert one can now calculate the membership degrees (scores) that correspond to each feature and each source location. For example, if the distance to a manufacturing facility is 79 m, the corresponding membership degree is 0.61 (Figure 3.3). Table 1 summarizes these results.

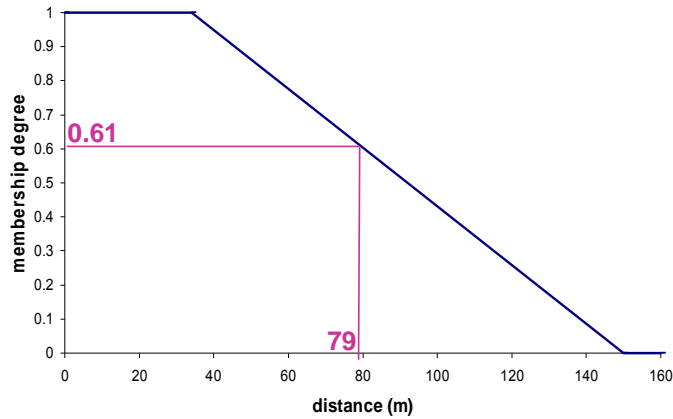


Figure 3. Membership function representing the meaning of ‘near the manufacturing facility or waste dump.’

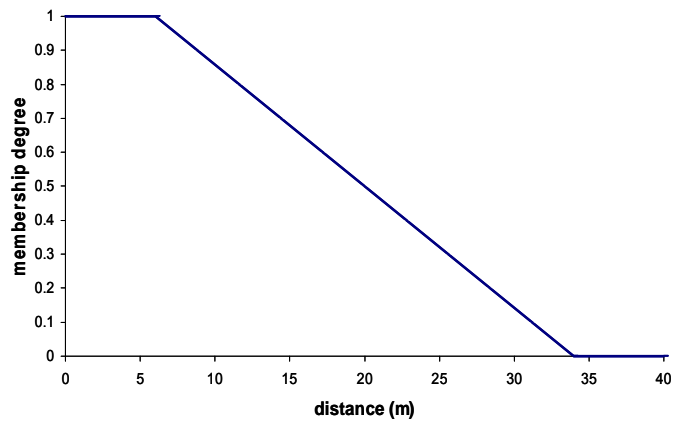


Figure 4. Membership function representing the meaning of ‘near’ water table.

Table 1. Distances and corresponding membership degrees for all potential source locations and all features.

	A - Distance to facility (m)	B - Distance to waste dump (m)	C - Distance to water table (m)	Membership degree (score)		
				$\sigma(A)$	$\sigma(B)$	$\sigma(C)$
Source 1	79	25	10.75	0.61	1	0.84
Source 2	55.9	35.3	10.75	0.81	0.99	0.84
Source 3	35.3	55.9	10.75	0.99	0.81	0.84
Source 4	25	79	10.75	1	0.61	0.84
Source 5	35.3	103	10.75	0.99	0.40	0.84
Source 6	55.9	127.5	10.75	0.81	0.19	0.84

In addition, the expert provides monotone measures that contain all the information about the importance of each individual feature and all groups of features for

identifying the true source. In our case the expert defined the six monotone measures needed as follows:

$$\mu(A) = 0.3, \mu(B) = 0.5, \mu(C) = 0.2, \mu(A, B) = 0.7, \mu(A, C) = 0.7, \mu(B, C) = 0.8$$

It is evident from the values defined above that there is significant interaction between the criteria (features). For example, the importance of the proximity to a waste dump is 0.5 and the importance of the depth to the water table at any of the potential source locations is 0.2. The combined importance of these features though is 0.8. This means that when a location is close to a waste dump and at the same time the water table is close to the ground surface, the possibility of that location being the true source location is greatly increased. If the water table at the potential source location is close to the surface, but it is far from the waste dump, then the importance of this fact is low.

Various questions that the expert can take into consideration when defining the relative importance of the features include but are not limited to:

- Did the manufacturing operation use DNAPL and in what quantities?
- Did the facility have floor drains that carried DNAPL?
- Was DNAPL discarded on the land surface?
- Is there residual DNAPL on the soil surface?
- Has a soil boring showed the existence of DNAPL?
- Have soil gas investigations found high soil gas readings?
- Is there testimony of workers disposing of DNAPL inappropriately?
- Did the facility have any underground storage tanks?
- Did the waste dump receive any DNAPL and in what quantities?

The discrete Choquet integral can now be used to combine all the individual scores to provide a global degree of confidence of the statement ‘source location  $i$  belongs to the group of true source locations’ for each possible source location. The advantage of using the Choquet integral instead of a weighted average is that it provides a flexible way aggregate interacting and uncertain criteria.

We will now go through an example to illustrate how the discrete Choquet integral is calculated. Let’s choose source location 1 for illustration purposes. The membership degrees (scores) for this source location are:  $\sigma(A) = 0.61$ ,  $\sigma(B) = 1$  and  $\sigma(C) = 0.84$ . We have to order them and index them accordingly:  $\sigma_1(A) = 0.61 < \sigma_2(C) = 0.84 < \sigma_3(B) = 1$ . The formula for the Choquet integral (denoted here as  $h$ ) is as follows:

$$h(\sigma_1, \sigma_2, \sigma_3) = \sum_{i=1}^3 (\sigma_i - \sigma_{i-1}) \mu(\{x_i, \dots, x_3\})$$

$$h(\sigma_1, \sigma_2, \sigma_3) = \sigma_1 \mu(A, C, B) + (\sigma_2 - \sigma_1) \mu(C, B) + (\sigma_3 - \sigma_2) \mu(B)$$

$$h(0.61, 0.84, 1) = 0.61 \cdot 1 + (0.84 - 0.61) \cdot 0.8 + (1 - 0.84) \cdot 0.5$$

$$h(0.61, 0.84, 1) = 0.874$$

The global scores, calculated using the Choquet integral, are presented in Table 2. All scores were divided by the larger score value in order to normalize them. The higher the score, the larger our confidence that the particular source location is the true one. The normalized scores represent the initial weights used by the algorithm and they reflect the number of times each source will be considered when calculating the concentration realizations.

**Table 2. Partial and global scores for each potential source location.**

	Score for facility	Score for waste dump	Score for water table	Global weight	Standardized global weight
<b>Source 1</b>	<b>0.61</b>	<b>1</b>	<b>0.84</b>	<b>0.874</b>	<b>0.96</b>
<b>Source 2</b>	0.81	0.99	0.84	0.915	1
<b>Source 3</b>	0.99	0.81	0.84	0.876	0.96
<b>Source 4</b>	1	0.61	0.84	0.819	0.89
<b>Source 5</b>	0.99	0.40	0.84	0.753	0.82
<b>Source 6</b>	0.81	0.19	0.84	0.630	0.69

### 3.4.2. Flow and transport equations

In this work we are using a finite element numerical model called PTC (Princeton Transport Code) to solve the flow and transport partial differential equations. The theory and use of PTC is described in detail by Babu et al (1997). In our application we assume a steady state flow equation and a conservative convection-dispersion transport equation coupled with Darcy's law as described by the following equations:

$$\nabla \cdot (\mathbf{K} \cdot \nabla h) = 0 \quad (1)$$

$$\frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D} \cdot \nabla c) - \nabla \cdot (\mathbf{v} c) = 0 \quad (2)$$

$$\mathbf{v} = -\frac{\mathbf{K}}{n} \nabla h \quad (3)$$

where h: hydraulic head, K: hydraulic conductivity, D: hydrodynamic dispersion, c: solute concentration, n: effective porosity, v: pore velocity.

Equation 1 describes the steady state flow of water through a porous medium. The hydraulic conductivity is a property of the medium that describes its capacity to transmit flow of a specific fluid. Equation 2 is the transport equation that describes how the contaminant concentration changes with time. Equation 3 is called Darcy's Law and is a constitutive equation that relates groundwater pore velocity with the hydraulic head information from the flow equation and hydraulic conductivity (Herrera, 1998).

Among all the input parameters of a groundwater flow and transport model the most uncertain is hydraulic conductivity. Hydraulic conductivity values can vary significantly in locations that are separated only by a few meters. Since it is not possible to directly measure hydraulic conductivity in every location where a hydraulic conductivity value is needed, these values need to be estimated using hydraulic conductivity measurements taken at different locations. This process generates additional uncertainty. Errors in hydraulic conductivity estimates will result in errors in the groundwater velocity calculations, creating errors in the contaminant concentration results. Stochastic modeling provides a way of quantifying the uncertainty in hydraulic conductivity estimates and propagating it to the contaminant concentration output. In this work, we model hydraulic conductivity as a spatially correlated random field.

### 3.4.3. Random hydraulic conductivity field generation – Latin hypercube sampling

As mentioned before, one of the main assumptions of the search algorithm presented here is that the primary source of uncertainty in the transport equation is the velocity due to the uncertainty and heterogeneity in the hydraulic conductivity. Thus hydraulic conductivity is treated as a random variable, while all other model parameters are assumed to be deterministic. In the application of the search algorithm to Hunters Point Shipyard, the uncertainty in hydraulic conductivity is characterized by possibility theory (Zadeh, 1978), a generalization of probability theory. This is discussed further in Chapter 5.

#### 3.4.3.1. Statistical definitions

Let us now provide some useful statistical definitions. Most of the following definitions can be found in Casella and Berger, 2002.

Definition 1. A *random variable* is a function from a sample space  $S$  into the real numbers.

With every random variable  $X$ , we associate a function called the cumulative distribution function of  $X$ .

Definition 2. The *cumulative distribution function* or *cdf* of a random variable  $X$ , denoted  $F_X(x)$ , is defined by:

$$F_X(x) = P_X(X \leq x), \text{ for all } x.$$

The cumulative distribution function describes the probability that the random variable  $X$  is less or equal than a specific value  $x$ .

Definition 3. The *probability density function* or *pdf* of a discrete random variable  $X$  is given by:

$$f_X(x) = P_X(X = x), \text{ for all } x.$$

Random variables are often characterized by their moments. The most often used moments are the first moment, which is the expected value or mean and the second moment, known as the variance.

Definition 4. The *expected value or mean* of a random variable  $X$  denoted by  $E(X)$ , is:

$$E(X) = m_x = \int_{-\infty}^{+\infty} xf_X(x)dx .$$

Definition 5. The *variance* of a random variable  $X$  denoted by  $\text{Var}(X)$ , is:

$$\text{Var}(X) = \sigma_x^2 = E[(X - m_x)^2] = \int_{-\infty}^{+\infty} (x - m_x)^2 f_X(x)dx .$$

Definition 6. A random variable is called Gaussian or normal if its pdf is given by:

$$f_Y(y | m, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y-m)^2}{2\sigma^2}}, -\infty < x < \infty .$$

If  $X$  is a random variable whose natural logarithm is normally distributed (that is,  $Y = \ln X \sim N(\mu, \sigma^2)$ ), then  $X$  has a lognormal distribution.

Definition 7. The pdf of a lognormal random variable is given by:

$$f_X(x | m, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \frac{1}{x} e^{-\frac{(\log x - m)^2}{2\sigma^2}}, 0 < x < \infty, -\infty < m < \infty, \sigma > 0 .$$

Definition 8. The *expected value or mean* of a lognormal random variable X denoted by  $E(X)$ , is:

$$E(X) = e^{m+(\sigma^2/2)} .$$

Definition 9. The *variance* of a lognormal random variable X denoted by  $\text{Var}(X)$ , is:

$$\text{Var}(X) = e^{2(m+\sigma^2)} - e^{2m+2\sigma^2} .$$

Usually, the data collected in an experiment consist of several observations on a variable of interest.

Definition 10. The marginal probability density function,  $f_1(x_1)$ , of random variable  $X_1$  is defined by:

$$f_1(x_1) = \int_{-\infty}^{+\infty} f(x_1, x_2) dx_2 .$$

Definition 11. The random variables  $X_1, \dots, X_n$  are called a *random sample* of size n from the population  $f(x)$  if  $X_1, \dots, X_n$  are mutually independent random variables and the marginal pdf of each  $X_i$  is the same function.

Definition 12. The sample mean is the arithmetic average of the values in a random sample. It is usually denoted by:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i .$$

Definition 13. The sample variance is the statistic measure defined by:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 .$$

In an experimental situation, we usually observe values of more than one random variable. Probability models that involve more than one random variable are called multivariate models.

Definition 14. The joint distribution function,  $F(x_1, x_2)$ , of two random variables  $X_1$  and  $X_2$  is defined by:

$$F(x_1, x_2) = P[X_1 < x_1 \text{ and } X_2 < x_2] .$$

There are several numerical measures that define the strength of the relationship between two random variables. The two most often used ones are the covariance and correlation functions.

Definition 15. The covariance of two random variables  $X_1$  and  $X_2$  is defined by:

$$\text{Cov}(X_1, X_2) = E((X_1 - \mu_{X_1})(X_2 - \mu_{X_2})) .$$

Definition 16. The correlation of two random variables  $X_1$  and  $X_2$  is defined by:

$$\rho_{X_1, X_2} = \frac{\text{Cov}(X_1, X_2)}{\sigma_X \sigma_Y} .$$

In this work, we model hydraulic conductivity as a spatially correlated random field with a log-normal probability distribution. To characterize the random field we only need to know its mean and covariance matrix. If hydraulic conductivity measurement data are available one can use geostatistical methods to define the statistical parameters of the random field.

### 3.4.3.2. Variogram analysis

One of the most common techniques used to describe the spatial correlation of a random variable is the semi-variogram (called simply variogram for the rest of this document) analysis. Variogram analysis consists of types of variogram models: 1) the experimental (or empirical) variogram calculated from the data and 2) the model (or theoretical) variogram best fit to the data.

The experimental variogram value,  $\gamma(h)$ , is half the average squared difference of the data values over all pairs of observations whose locations are separated by the same distance (h). The experimental variogram equation is the following:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{(i,j)_{h_{ij}=h}} (u_i - u_j)^2,$$

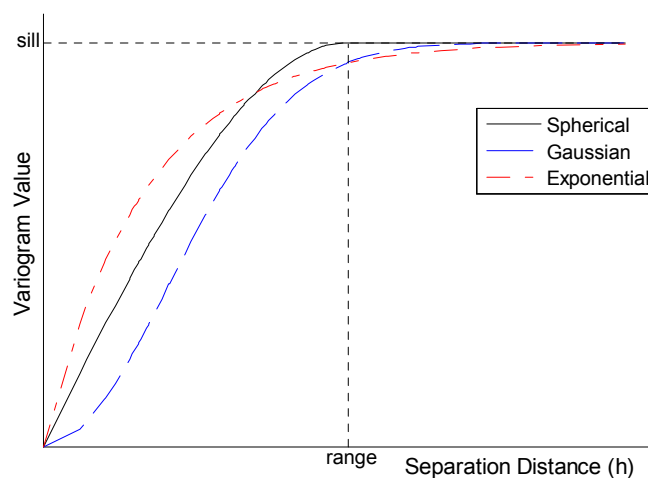
where:

$u_i$ = data values,

$h$ = separation distance, and

$N(h)$ = number of pairs of data whose locations are separated by a distance  $h$ .

The model variogram is a predefined mathematical function that describes spatial continuity. The appropriate model is chosen by fitting the model variogram to the experimental variogram. A very important restriction on the model variogram is that it has to provide a positive definite covariance matrix. A way to satisfy the positive definiteness condition is to choose mathematical functions that are known to be positive definite (Isaaks and Srivastava, 1989). The three most commonly used positive definite variogram models are: the spherical, exponential and Gaussian models (Figure 5).



**Figure 5. Three important model variogram types: spherical, Gaussian and exponential.**



The major features of a variogram model are the *range*, the *sill* and the *nugget effect*. Theoretically, as the separation distance ( $h$ ) between points increases, the corresponding variogram values should also increase until they reach a plateau (where they remain relatively constant). The separation distance at which variogram values stop increasing is called the *range*. Shorter ranges signify less similarity in data values throughout the domain, whereas larger ranges imply that data values are significantly similar over the domain. The *sill* is the plateau the variogram reaches at the range. Theoretically, at a zero separation distance, the variogram value is zero (no local variance), but it is very usual in reality to have a sharp increase in variogram values for some very small separation distance. This phenomenon is called the *nugget effect*. The nugget effect is caused by various factors, such as sampling errors and small scale variability (Isaaks and Srivastava, 1989).

The variogram model used in the synthetic examples presented in Chapter 4 as well as the field applications presented in Chapter 5 is the exponential model. The choice of the variogram model was arbitrary in the case of the synthetic examples since there were no real hydraulic conductivity data to fit to. For the field applications the choice of variogram model was based on the trend of the hydraulic conductivity data. The exponential model variogram equation is given by:

$$\gamma(h) = c_0 + c \left[ 1 - \exp\left(-\frac{3h}{a}\right) \right],$$

where:  $c_0$  is the nugget,  $c$  is the sill,  $a$  is the range and  $h$  the separation distance.

After choosing a model variogram we know the statistics of the hydraulic conductivity field so the next step is to generate a set of realizations that reflect the statistical structure of the measured data. There are many different methods for generating random fields. As mentioned earlier in this work we are using a strategy called Latin hypercube sampling.

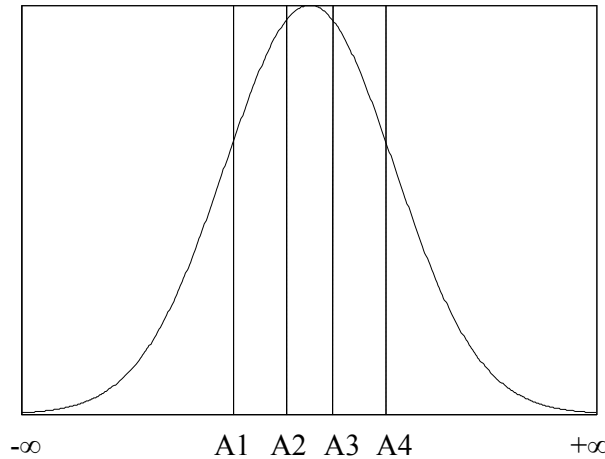
### 3.4.3.3. Latin hypercube sampling

We have already noted that Latin hypercube sampling (Lhs) was first introduced by McKay et al, 1979. In the Lhs process, input variables are treated as random variables having specified probability distribution functions (McWilliams, 1987).

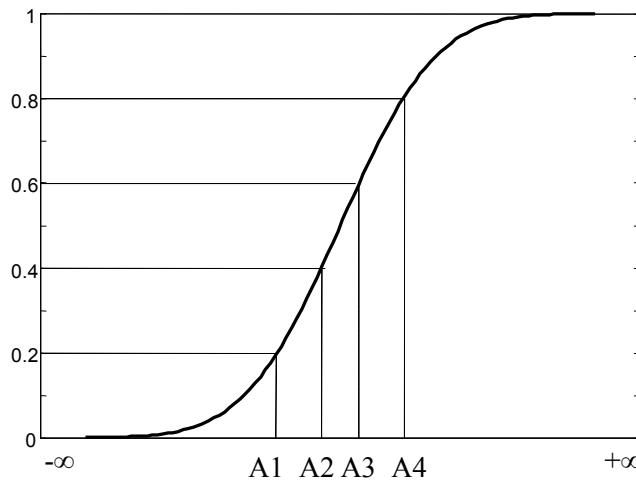
The Latin hypercube sampling strategy is a stratified sampling technique which can produce more precise estimates than random sampling of the distribution function (Iman et al., 1981). The probability density function of the variable of interest is divided into a number of non-overlapping, equal-probability intervals (Figure 6 and Figure 7). Samples are taken, one from each interval, and they are permuted in a way such that the correlation of the field is accurately represented. This is achieved by the use of rank correlation. The main idea of the rank correlation method is to rearrange the samples taken using the Lhs technique in such a way as to create a correlation matrix that is as similar as possible to the target correlation matrix. The set of rearranged values can be used as an input to simulators to produce realizations of output variables (Zhang, 2002).

A more detailed description of Latin hypercube sampling with application to sensitivity analysis techniques can be found in Iman et al. (1981a, b). A tutorial on Latin

hypercube sampling can be found in Iman and Conover (1982). A recent comparison of Latin hypercube sampling with other techniques is provided by Helton and Davis (2001). The effectiveness of Lhs as a hydraulic conductivity random field generator was demonstrated in the work of Zhang (2002) and Zhang and Pinder (2003). For a detailed description of the Latin hypercube sampling technique see Zhang (2002).



**Figure 6. Intervals used with a Latin hypercube sample in terms of a normal probability density function.**



**Figure 7. Intervals used with a Latin hypercube sample in terms of a normal cumulative distribution function.**

In the application of the search algorithm to Hunters Point Shipyard, the uncertain hydraulic conductivity values are represented by possibility distributions instead of probability distributions, due to the type of data employed to hydrogeologically characterize the site and estimate the hydraulic conductivity field. As such, in this case, a modified Lhs technique, called possibilistic Latin hypercube sampling (PLhs) was

employed to generate random fields from the uncertain hydraulic conductivity values. This procedure works very similarly to Lhs, the main difference being that samples are drawn from possibility distributions, which are structurally and theoretically similar to fuzzy sets (Section 3.4). Further discussion on PLhs is provided by Ross et al (*in review*)

#### 3.4.4. Concentration plume statistics calculation

A Monte Carlo approach is used to calculate the concentration distribution in the geologic system studied in this work. The Monte Carlo approach involves the use of the hydraulic conductivity realizations that were previously generated by the Latin hypercube sampling strategy in combination with the potential source locations. The groundwater flow and transport model of the site is run using one hydraulic conductivity realization and one of the potential source locations. The selection of the source location that will be used at each flow and transport simulation depends on their assigned weight. For example, let's assume there are 2 potential source locations and the weight for the first potential source location is double that of the second location. If we create 300 hydraulic conductivity realizations, then the first potential source location will be used for 200 realizations and the second potential source location will be used for the remaining 100 realizations. This way we ensure that the source location with a weight of 1 is used twice as many times as the source location with a weight of 0.5.

The concentration results for each realization at each nodal location are recorded and the concentration statistics for each nodal location (i.e. the mean and variance of the specified species concentration) are calculated. We will call the resulting mean concentration field the 'composite plume'.

The concentration values at all nodal locations are considered in the calculation of the spatial covariance matrix. The calculation of the covariance matrix is very important because it captures the uncertainty of the concentration field. By using the Monte Carlo simulation technique the hydraulic conductivity uncertainty was transferred through the simulator to contaminant concentration uncertainty. The concentration uncertainty provides vital information for the next step of the algorithm, which is the selection of water quality sampling locations.

#### 3.4.5. Water quality sampling location selection

At this point, the water quality data are incorporated into the search strategy. There are two important factors that were considered when selecting a new water quality sampling location. The first factor is the reduction in overall uncertainty of the contaminant concentration field that will result if we take a sample at a particular location. The Kalman filter is used to determine this factor. A significant concept in the Kalman filter is that, although we do not know the concentration value at points where water quality samples have not been taken, we do know that the uncertainty at any point where a sample is taken reduces to the sampling error. Application of this concept allows one to determine the impact of taking a sample at a target sampling location on the overall uncertainty of the concentration field. Thus, by testing the reduction in uncertainty attributable to potentially selecting a sample from each of the target sampling locations, the location providing the greatest reduction in plume uncertainty can be determined.

The second factor that was taken into account when selecting a new sampling location is the distance of this location from the source area. The closer the sampling location to the source area the more information it provides about the exact location of the true source. Thus, it is in our interest to first choose samples that are closer to the source areas.

The two important factors described above are combined using a Choquet integral and a global score is obtained for each sampling location (Figure 8). The higher this score, the better candidate the sampling location. Thus, the sampling location with the highest score is selected as the new sampling location.

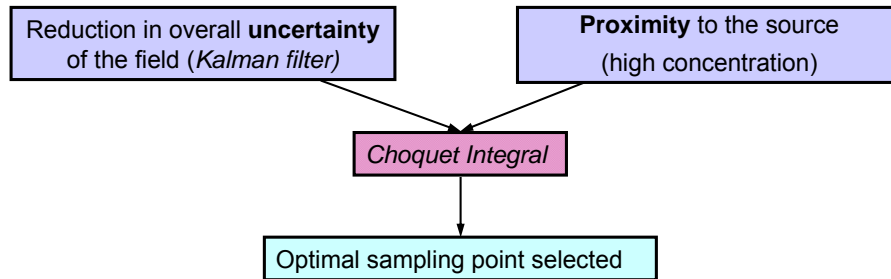


Figure 8. Strategy for the selection of a water quality sampling location.

### 3.4.5.1. Linear Kalman filter

The linear Kalman filter is a Best Linear Unbiased Estimator (BLUE) that combines the available prior information about the system with measurement data to produce estimates that are ‘linear’ (since they are weighted linear combinations of the prior variable values and the measurement values); ‘unbiased’ since both model and observation errors have zero mean; and ‘best’ because the filter seeks to minimize the error variance (Drecourt, 2004).

In Figure 9 we can see a flow chart that describes how the Kalman filter is used as part of the overall strategy employed in this work.

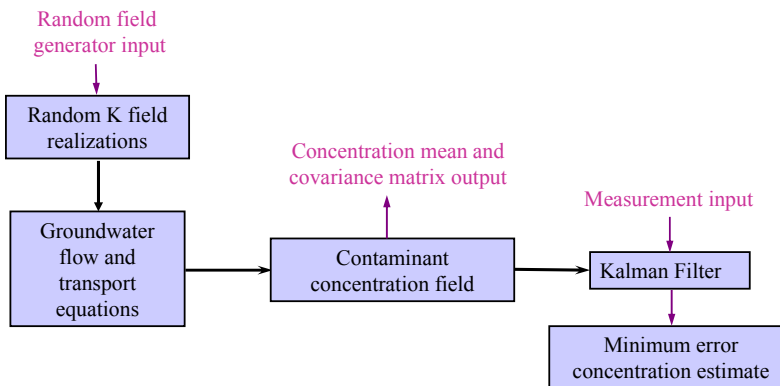


Figure 9. Kalman filter as part of the search algorithm.

The basic concept behind the Kalman filter is that it tries to find an ‘optimal estimate’ of the variables of interest when noisy data are available. If we consider a Bayesian framework, then the filter needs to propagate the conditional probability density of the variables of interest, conditioned on the real data. The ‘optimal estimate’ of the variable of interest is represented by the mean of the conditional distribution. The main assumptions regarding the use of the Kalman filter is that the model and measurement noises need to be Gaussian and white (noise values not correlated in space and time) (Maybeck, 1979).

In this work, we use the discrete static Kalman filter. The choice of a static filter is made because we do not consider time as part of our problem. The flow and transport equations are solved for one time period and sampling is assumed to occur at a short time scale, relative to the dynamics of groundwater flow. In what follows, the equations for the dynamic and static version of the Kalman filter are presented.

#### *Discrete dynamic linear Kalman filter*

The Kalman filter is based on two equations: the state equation, which describes how the state of the system changes over time (in the case of a dynamic system) and the measurement equation, which describes how the system is updated when measurement information becomes available.

State equation :  $x_{n+1} = \Phi x_n + w_n$

where:

$x_n, x_{n+1}$  : estimates at time  $t_n$  and  $t_{n+1}$  respectively

$\Phi_n$  : state transition matrix from time  $t_n$  to  $t_{n+1}$

$w_n$  : system noise,  $w_n \sim N(0, Q_n)$

$Q_n$  : covariance matrix of the system noise  $w_n$

Measurement equation:  $z_n = H_n x_n + u_n$

where:

$z_n$  : vector of  $l$  noise corrupted measurements at time  $t_n$

$H_n$  : measurement matrix of dimension  $l \times q$  at time  $t_n$ , where  $q$  is the number of estimate locations

$x_n$  : vector of dimension  $m$  that is an estimate of the desired quantity based on the measurements

$u_n$  : vector of random measurement noise,  $u_n \sim N(0, R_n)$

$R_n$  : measurement noise covariance matrix

The optimal updated estimate will be a linear function of the a priori estimate and the measurement  $z$ :

$$\hat{x}_n^+ = K_n^1 \hat{x}_n^- + K_n^2 z_n ,$$

where:

$\hat{x}_n^+$  : posterior estimate of the system state at time  $t_n$

$\hat{x}_n^-$  : prior estimate of the system state based on the measurement  $z_n$

$K_n^1$  and  $K_n^2$  are time varying weighting matrices

The matrices  $K_n^1$  and  $K_n^2$  are determined through the derivation of the Kalman filter.

The Kalman filter can be thought of as a predictor-corrector type of estimator. The time update (predictor) equations for the state variable and the error covariance are the following:

$$\hat{x}_{n+1}^- = \Phi \hat{x}_n^+ + w_n$$

$$P_{n+1}^- = \Phi P_n^+ \Phi^T + Q_n,$$

where:

$P_{n+1}^-$  : error covariance estimate

The measurement update (corrector) equations are the following:

1) Compute Kalman gain  $K_n$

$$K_{n+1} = P_{n+1}^- H_{n+1}^T (H_{n+1} P_{n+1}^- H_{n+1}^T + R_{n+1})^{-1}$$

2) Update estimate with measurement  $z$

$$\hat{x}_{n+1}^+ = \hat{x}_{n+1}^- + K_{n+1} (z_{n+1} - H_{n+1} \hat{x}_{n+1}^-)$$

3) Update the error covariance

$$P_{n+1}^+ = (I - K_{n+1} H_{n+1}) P_{n+1}^- ,$$

where:  $-$  denotes prior estimate and  $+$  denotes posterior estimate

#### *Discrete static linear Kalman filter*

In the case of a discrete static filter the state equation is given by:

$$x_{n+1} = x_n \text{ and } P_{n+1} = P_n ,$$

which implies that the variable  $x$  and the error covariance matrix  $P$  do not change over time. Since the estimate is not related to time, all the time subscripts in this section are dropped.

The measurement equation is given by:

$$z = Hx + u .$$

The final equations that are used to update the state variable and the error covariance matrix are the following:

1) Compute Kalman gain:

$$K = P^- H^T (H P^- H^T + R)^{-1} \quad (4)$$

2) Update estimate with measurement  $z$ :

$$\hat{x}^+ = \hat{x}^- + K(z - H\hat{x}^-) \quad (5)$$

3) Update the error covariance:

$$P^+ = (I - KH) P^- . \quad (6)$$

#### *Incorporation of the Kalman filter into the search algorithm*

The approach taken in this work in order to incorporate the Kalman Filter into the search algorithm is similar to that of Herrera (1998) and Zhang (2002).

If we define the vector of concentrations at all nodal locations as the state variable, then the spatial mean concentration vector and covariance matrix calculated from the Monte Carlo simulation would represent prior estimates of the state variable and the error

covariance. Then, we can use the Kalman filter to condition these prior estimates with the measurement data.

In the Kalman filter equations we substitute  $x$  with  $C$ , which represents the contaminant concentration vector that contains concentration values at all nodal locations:

$$C = (c_1, c_2, c_3, \dots, c_m),$$

where  $c_i$  is the concentration at node  $i$  and  $m$  is the total number of nodal locations.

The corresponding covariance matrix has the following format:

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \dots & \dots & \dots & \dots \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{bmatrix}. \quad (7)$$

In this work, we choose one sampling location at a time. If the  $k$ -th sampling location coincides with  $c_j$  then the corresponding sampling matrix  $H$  will have the following format:

$$H = (0, 0, \dots, 0, 1, 0, \dots, 0),$$

where: the number 1 is located at the  $j$ -th position. The sampling error covariance associated with the water quality measurement at the  $j$ -th location is denoted by  $r_j$ .

Using the Kalman gain formula (Equation 4) we can calculate the Kalman gain in two steps. First we calculate the product  $P^- H^T$ , and then the product  $(HP^- H^T + R)^{-1}$ .

$$P^- H^T = (P_{1,j}^-, P_{2,j}^-, \dots, P_{m,j}^-)^T \quad (8)$$

$$(HP^- H^T + R)^{-1} = \frac{1}{P_{j,j}^- + r_j}, \quad (9)$$

where  $r_j$  is the sampling error covariance associated with the water quality measurement at the  $j$ -th location.

The Kalman gain ( $K^G$ ) is now calculated by substituting Equations 8 and 9 into equation 3.4:

$$K^G = \frac{1}{P_{j,j}^- + r_j} (P_{1,j}^-, P_{2,j}^-, \dots, P_{m,j}^-)^T.$$

If we substitute  $K^G$  into equation 3.7, we can calculate the updated covariance matrix:

$$P^+ = (I - K^G H)P^- = \begin{pmatrix} 1 & 0 & 0 & -\frac{P_{1j}^-}{P_{jj}^- + r_j} & 0 & \dots & 0 \\ 0 & 1 & & -\frac{P_{2j}^-}{P_{jj}^- + r_j} & & & \\ \dots & & & \dots & & & \\ & & & 1 - \frac{P_{jj}^-}{P_{jj}^- + r_j} & & & \\ & & & \dots & & & 0 \\ 0 & \dots & \dots & 0 & -\frac{P_{mj}^-}{P_{jj}^- + r_j} & \dots & 1 \end{pmatrix} P^-$$

The diagonal elements of this matrix are given by the following equation:

$$P_{ii}^+ = P_{ii}^- - \frac{(P_{ij}^-)^2}{P_{jj}^- + r_j}.$$

According to the above equation, the change of the estimate variance at a location  $i$  caused by a measurement taken at location  $j$  is represented by the term  $\frac{(P_{ij}^-)^2}{P_{jj}^- + r_j}$ . If we

want to calculate the effect of a measurement taken at location  $i$  on the total variance of the concentration field, then we have to sum the variances at all nodal locations:

$$\sigma_T^2 = \sum_i P_{ii}^+ = \sum_i P_{ii}^- - \frac{1}{P_{jj}^- + r_j} \sum_i (P_{ij}^-)^2.$$

The term  $\sum_i P_{ii}^-$  in the above equation represents the prior total variance. The term

$\frac{1}{P_{jj}^- + r_j} \sum_i (P_{ij}^-)^2$  is always less than or equal to the prior total variance. The total variance

reduction is achieved when  $\frac{1}{P_{jj}^- + r_j} \sum_i (P_{ij}^-)^2$  is a maximum.

### *Practical considerations*

There are some practical considerations that we need to take into account when applying the Kalman filter:

- The covariance matrix update should be symmetric and positive definite. To overcome ill-conditioned problems, an alternative expression for  $P(+)$  can be used (called Joseph form):

$$P_{n+1}^+ = (I - K_{n+1}^G H_{n+1})P_{n+1}^- (I - K_{n+1}^G H_{n+1})^T + K_{n+1}^G R_{n+1} K_{n+1}^{G^T}.$$

The right-hand side of the above equation is the summation of two symmetric matrices, thus the result is a symmetric matrix. The first matrix on the left-hand side of the above equation is positive definite and the second is nonnegative definite, thus the resulting matrix  $P_{n+1}^+$  is a positive definite matrix (Mohinder and Andrews, 2001).



- In many Kalman filtering applications, the individual components of the measurement noise vector are uncorrelated, resulting in a diagonal covariance matrix of the measurement noise. In this case, it is possible to treat the components of the measurement vector  $z$  as independent scalar measurements. There are two main advantages of the scalar implementation: i) Less computation time: The number of computations required for vector implementation is a cubic function of the number of measurements. When the scalar implementation is used the number of computations is greatly reduced since it is a linear function of the number measurements. ii) Enhanced numerical accuracy: The scalar implementation of the filter requires no matrix inversions resulting in a more robust method (Mohinder and Andrews, 2001).

#### *Bias in the Kalman filter estimates*

As mentioned above, the central assumption that makes the Kalman filter an optimal estimator is that both the model and measurement errors have zero-means and are uncorrelated. In real life applications this assumption is often violated resulting in biased estimates (Drecourt, 2004).

The definition of the bias of an estimator is given below (Casella and Berger, 2002):

Definition 1: The bias of a point estimator  $W$  of a parameter  $\theta$  is defined as the difference between the expected value of  $W$  and  $\theta$ , i.e.

$$Bias_{\theta}(W) = E_{\theta}W - \theta.$$

In this work, we assume unbiased observations. Therefore, the model bias is defined as the difference between model estimates and measurement values.

In the context of our work, we are concerned about the possibility that all the modeled and measured values will differ by a constant value (constant bias). If bias indeed exists, then the information provided by samples will severely distort the modeled surface making it difficult to apply the search algorithm. When testing the search algorithm on a field application, we found that models with limited calibration can lead to bias in the sense described above.

Although bias in the groundwater models is a known phenomenon, there is no single, generally accepted strategy to accommodate it. In the case of groundwater transport, bias takes the form of concentration error being correlated in space. Since the concentration at a point is proportional to the magnitude of the contaminant source, an error in the source strength naturally leads to model bias. The approach we used to correct the bias in this work was to modify the source magnitude such that it is consistent with the measurement data.

#### 3.4.6. Optimization problem – solving for the source strength

The strategy we developed to address the bias problem is to use an optimization program that adjusts the source strength while minimizing the summation of the absolute differences between modeled concentration values and measured concentration values at the sampling locations. Whenever a sample becomes available, the optimization model will adjust the source strength, such that the modeled mean value is as consistent as possible with the observed values at those locations where samples have been collected. Because a change in the source concentration leads to a uniform change in the modeled

concentration values, such a strategy appears to be a logical bias correction technique since it corrects for a uniform error over the concentration field. Since the source magnitudes are linearly related to the concentration at all nodes in the model domain the resulting optimization problem is linear.

This optimization technique, described in the next section, was first introduced by Gorelick et al. (1983) to identify unknown pipe leak magnitudes in contaminated aquifers.

### 3.4.6.1. Optimization problem formulation

#### *Decision variables*

The decision variables in this optimization problem are the source magnitudes for each potential source location. The concentration values at the sampling locations are the state variables and they depend on the source magnitudes.

#### *Objective function*

We define the best solution to be the one that minimizes the summation of the absolute differences between modeled concentration values and measured concentration values at the sampling locations.

$$\min \sum_i |c_i - z_i| \quad i = 1, \dots, n$$

where,

$c_i$ : model concentration at sampling location  $i$

$z_i$ : measured concentration at sampling location  $i$

$n$ : total number of locations where samples have been taken

#### *Constraints*

The constraints required for this optimization problem are physical in nature. The source magnitude for each potential source location cannot be negative and cannot exceed the solubility limit of the compound involved:

$$m_j \geq 0 \quad j = 1, \dots, k$$

$$m_j \leq m^* \quad j = 1, \dots, k$$

where,

$m_j$ : source magnitude at potential source location  $j$

$m^*$ : solubility limit for chemical compound of interest

$k$ : total number of potential source locations.

#### Response matrix technique

The response matrix technique was used to generate the constraints that relate the source magnitude with the concentration at the sampling locations.

The function that describes the relationship between the source magnitude and the contaminant concentration at any node in the model domain can be formally described as:

$$c_i = c_i(\mathbf{m})$$

where,

$\mathbf{m}$  : set of all potential source magnitudes

$c_i$  : simulated contaminant concentration that results from the sources described by  $\mathbf{m}$ .

The relationship between  $\mathbf{m}$  and  $c_i$  is linear.

Our algorithm relies on the PTC model to describe the contaminant concentration at all nodal locations in the groundwater model domain. PTC uses the following equation to determine concentration values:

$$\frac{\partial c(\mathbf{m})}{\partial t} - \nabla \cdot (\mathbf{D} \cdot \nabla c(\mathbf{m})) - \nabla \cdot (\mathbf{v} c(\mathbf{m})) = 0$$

where,

$c$  : solute concentration in any of the model domain nodes

$\mathbf{m}$  : set of all potential source magnitudes

$\mathbf{D}$  : hydrodynamic dispersion matrix

$\mathbf{v}$  : pore velocity vector

$t$  : time

The response matrix that relates the source magnitudes and the contaminant concentration at sampling locations is:

$$A = \begin{bmatrix} \frac{\partial c_1}{\partial m_1} & \frac{\partial c_1}{\partial m_2} & \dots & \frac{\partial c_1}{\partial m_k} \\ \frac{\partial c_2}{\partial m_1} & \frac{\partial c_2}{\partial m_2} & \dots & \frac{\partial c_2}{\partial m_k} \\ \dots & \dots & \dots & \dots \\ \frac{\partial c_n}{\partial m_1} & \frac{\partial c_n}{\partial m_2} & \dots & \frac{\partial c_n}{\partial m_k} \end{bmatrix}.$$

The elements of the response matrix A consist of derivatives of concentration with respect to source magnitude and are calculated using the following formula:

$$\frac{\partial c_i}{\partial m_j} \approx \frac{\Delta c_i}{\Delta m_j} = \frac{c_i(\mathbf{m} + \delta) - c_i(\mathbf{m})}{\delta_j}$$

where:

$\delta$  : vector with all elements zero except for the j-th which has the value of  $\delta_j$ . In this case  $\delta_j$  is a unit change in the magnitude of the potential source location j.

The concentration value at each sampling location is calculated as a weighted average using Equation 3.10.

$$c_i = c_{i0} + \sum_{j=1}^k w_j \frac{\partial c_i}{\partial m_j} m_j, \quad (10)$$

where:

$w_j$  : weight associated with potential source location j, and

$c_{i0}$ : concentration at sampling location  $i$  under the base case scenario for the source magnitudes. In our algorithm, we chose a base case scenario that involves zero magnitudes for all potential source locations, thus  $c_{i0}$  is zero.

The weights used in Equation 10 are the initial weights that were calculated using the Choquet integral technique. During the search algorithm iterations these weights are updated and sometimes they take zero values. If these updated weights were used in the optimization problem described here, then the corresponding potential source locations would be eliminated from the set of decision variables. In the case of zero weights, there would be no way to recover those potential source locations in the following algorithm iterations. Thus, the initial non-zero weights are always used at this algorithm step.

#### *Incorporation of absolute values in the objective function as part of a linear problem*

The form of the objective function defined previously as  $\min \sum_i |c_i - z_i|$  is not compatible with linear optimization because it contains absolute values. Fortunately, there is an easy way to overcome this problem.

We need to replace the absolute values in the objective function with the difference between two components  $X_i$  and  $X'_i$ . The objective function of the problem now becomes:

$$\min \sum_i (X_i + X'_i) \quad \text{for } i = 1, \dots, n.$$

The minimization formula presented above ensures that at most one of the two variables  $X_i$  or  $X'_i$  will be in the solution for each pair because the objective function is improved that way (Gorelick et al., 1983).

The final optimization formulation takes the following form:

$$\min \sum_i (X_i + X'_i),$$

such that

$$\begin{aligned} -X'_i + X_i + \sum_{j=1}^k w_j \frac{\partial c_i}{\partial m_j} m_j &= z_i & i = 1, \dots, n \\ m_j &\leq m^* & j = 1, \dots, k \\ X'_i, X_i &\geq 0 & i = 1, \dots, n \\ m_j &\geq 0 & j = 1, \dots, k \end{aligned}$$

The optimization problem defined above has only linear constraints and was solved using the revised simplex routine from the IMSL Fortran numerical library.

#### 3.4.7. Comparison of composite and individual plumes – $\alpha$ -cut method

The last step in the search algorithm's iteration involves the comparison of contaminant concentration plumes. A concentration random field that considers the updated source magnitudes is produced for each individual source alternative using the Monte Carlo

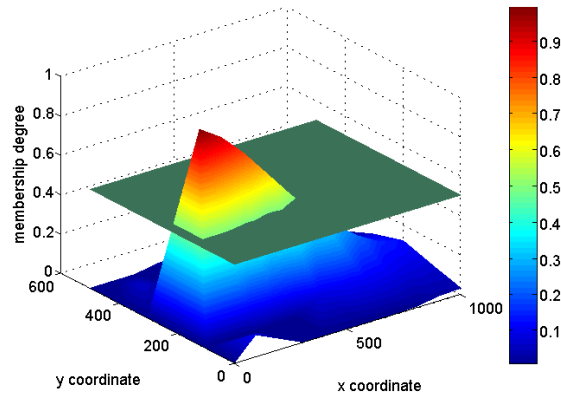
approach and the field statistics that are calculated. For example, if we are considering three potential source locations, we will generate three different concentration fields. Each concentration field is created by considering only one of the three potential source locations at a time. We will call the mean of these concentration fields ‘individual plumes’. Each individual plume is compared to the updated composite plume using the following method:

All plumes (both individual and composite) are represented as fuzzy sets with membership functions defined as normalized concentration values (we divide all concentration values by the maximum concentration value). Several  $\alpha$ -cuts of the fuzzy sets are considered.

**Definition 1.** An  $\alpha$ -cut is a crisp set that contains all the elements of a fuzzy set whose membership degrees are greater or equal to the specified value of  $\alpha$ .

Figure 10 shows a fuzzy set and its 0.5  $\alpha$ -cut. In this work we use 10  $\alpha$ -cuts ( $\alpha_i=0.1, 0.2, \dots, 1$ ). Each  $\alpha$ -cut for the updated plume is compared with the corresponding  $\alpha$ -cut of each individual plume and a measure of the area ( $S$ ) that is common in the 2  $\alpha$ -cuts is recorded. The global degree ( $g$ ) of similarity between the two plumes is obtained by weighting the common area  $S$  by the  $\alpha$  value itself and summing all the products:

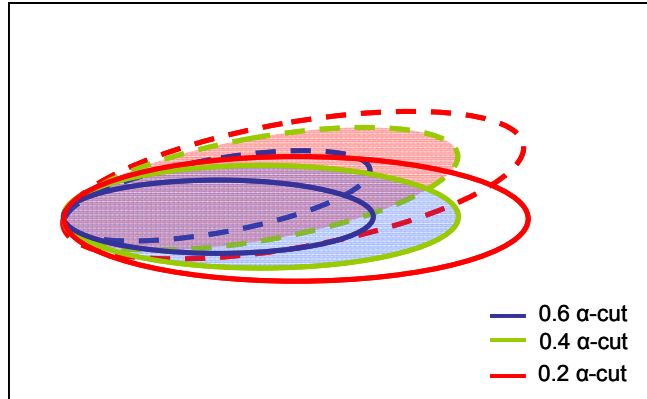
$$g = \sum_i \alpha_i S_i \quad i = 1, 2, \dots, 10.$$



**Figure 10. Normalized concentration plume presented as a fuzzy set and its 0.5  $\alpha$ -cut.**

Figure 11 shows how the  $\alpha$ -cuts are compared. The solid lines represent the composite plume and the dashed lines one of the individual plumes. If we consider the 0.4  $\alpha$ -cut (green lines), then the purple area shown is the common area in which we are interested. This common area provides a measure of how similar the two plumes are. This area is more important when we are considering the higher  $\alpha$ -cuts; this is why we chose to weight the areas according to their corresponding  $\alpha$ -cut value.

The advantage of using this method to compare the plumes is twofold: The intersection of the two plumes is emphasized and the higher concentration values are weighted more. The degree of similarity between each individual source location plume and the updated plume is normalized and assigned as a new weight to each potential source location.



**Figure 11. Comparison of  $\alpha$ -cuts. The common area of the 0.4  $\alpha$ -cuts is shown in purple.**

### 3.4.8. Iteration procedure

After the new weights are obtained, the ‘composite’ plume is recalculated using the newly calculated weights and source strengths. The mean and covariance of the ‘composite’ plume is then updated by the Kalman filter using all the sampling information that is available at that point. A new optimal sampling location is then selected that will reduce the new total uncertainty the most while taking into account the proximity of the sampling point to the potential source locations. A new set of source magnitudes (simplex method) is then selected that reflects the addition of a new water quality sample. The composite plume is then calculated again using the new source magnitudes and initial weights and it is updated using the Kalman filter. The composite plume is then compared to the ‘individual’ plumes and new weights are obtained. The process is repeated until convergence on an optimal location and source strength is achieved. The convergence criterion used for the examples described in this work is the following: the summation of the absolute differences between weights for two consecutive steps should be less than 0.1.

## 4. Results and Discussion

### 4.1 Synthetic example

While the best way to test the proposed DNAPL source search algorithm is to apply it in the field, as a first step we consider a simple synthetic test problem in order to further elucidate the source search algorithm and demonstrate the effectiveness with which the algorithm finds the true source location. The illustrative example problem is a one-dimensional homogeneous aquifer problem.

The advantage of synthetic examples is that the true source location is known, permitting unambiguous verification of the search strategy's efficacy. For this synthetic problem, one of the potential source locations was selected as the true source and a contaminant plume was generated using one realization of the hydraulic conductivity field. When a sample was needed, the calculated concentration at the proposed sampling location was used as the surrogate for the concentration that would be measured in the field.

The hypothetical aquifer system used in this example is shown in Figure 12a. The aquifer is 1000m long and 500m wide, with constant head boundaries along the left side ( $h=1\text{m}$ ) and right ( $h=0\text{m}$ ) side. The mean hydraulic conductivity is 10m/d. There are 6 potential source locations, shown in Figure 12a, and the true one is assumed to be number 1. The number of potential sampling wells is 70 and they are shown in Figure 12b. Figure 13 shows the true plume generated by a single hydraulic conductivity realization assuming that the true source location is number 1.

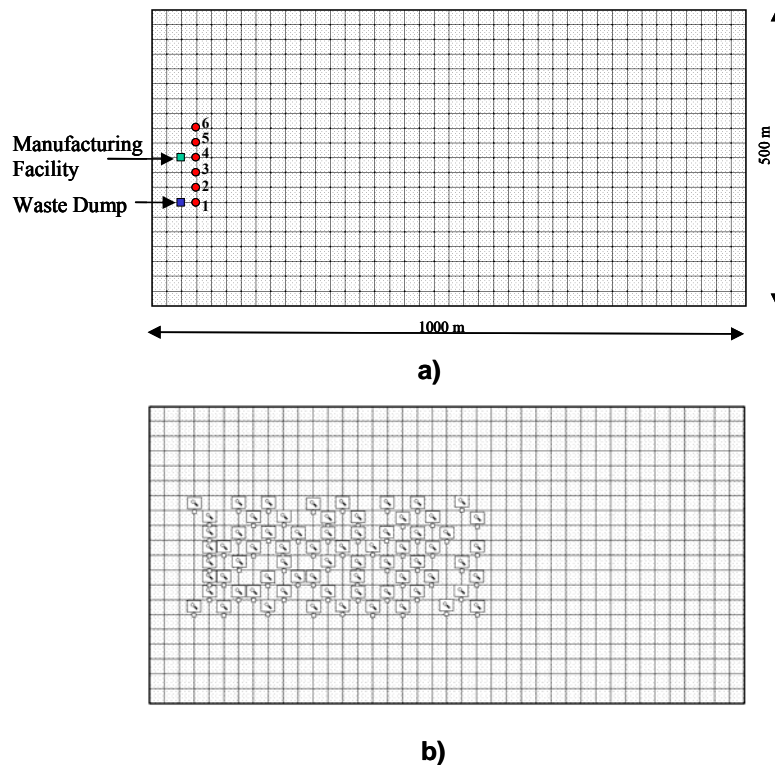
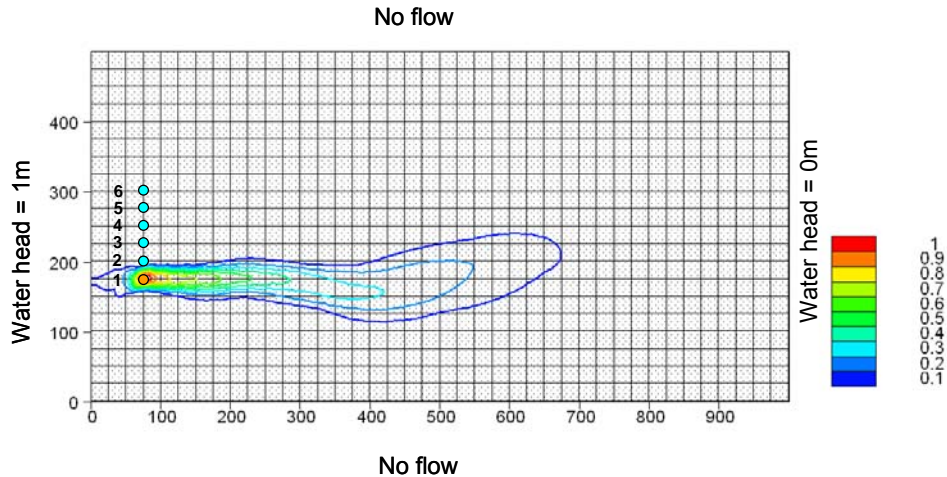


Figure 12. a) Synthetic aquifer for example 1, b) Potential water quality sampling locations.



**Figure 13. True plume generated by a single realization of hydraulic conductivity for single source problem.**

Figure 14 to Figure 21 show the weights and updated plumes that are obtained at each algorithm iteration. The location where a new concentration sample is taken at the current iteration is shown by a red dot. Concentration samples taken at previous steps are shown by black dots.

The initial weights shown in the upper left corner of Figure 14 are the ones calculated using the information fusion technique (Choquet integral) described in the previous chapter. As noted the Choquet integral combines two features: the reduction in the total concentration uncertainty that a sample produces when selected and the proximity of the sampling location to the source location. The proximity to the source locations is not measured as distance but instead it is represented by the actual concentration values; this means that high concentrations are closer to the potential source locations than lower concentrations. The importance (weight) that is given in each of these features is 0.4 for the proximity to the source and 0.6 for the reduction in uncertainty. The same values were used for all 5 examples presented in this section.

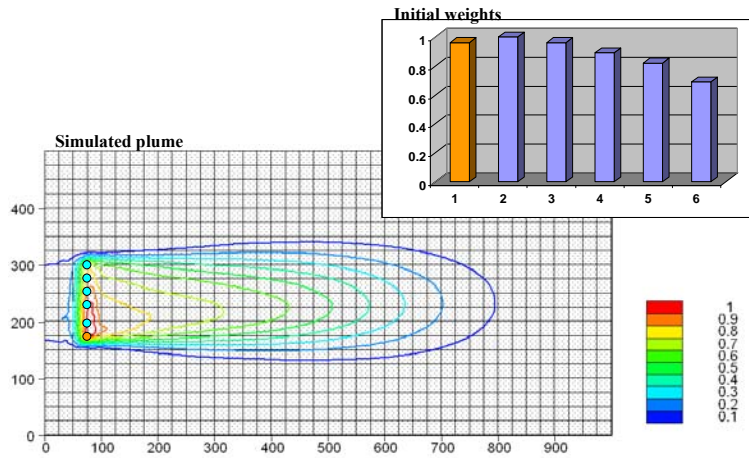
The individual plumes that the composite is compared against are shown in Figure 22. For the examples presented in this section the comparison is made using 10 alpha-cuts (0.1, 0.2, ..., 0.9, 1). Figure 23 shows how the uncertainty of the concentration field is reduced after taking each sample. The uncertainty reduction is calculated using the Kalman filter.

For all the examples presented in this chapter, we assume that we know the source strength and thus we can use normalized concentrations (everything is scaled between zero and one). Therefore, one shouldn't expect any concentration values higher than one. As is evident in some of the updated concentration plumes, there are some areas with unusually high concentrations (that exceed the value of one). This happens during the Kalman filtering step because there is no upper (or lower) limit on the values that can be calculated using the Kalman filter. In these examples when the Kalman filter produces negative values, we set the value equal to zero and when it produces values that exceed one we set them equal to one. It seems that this doesn't affect the convergence of the algorithm, and it happens mostly at the early stages of the process. As more samples are

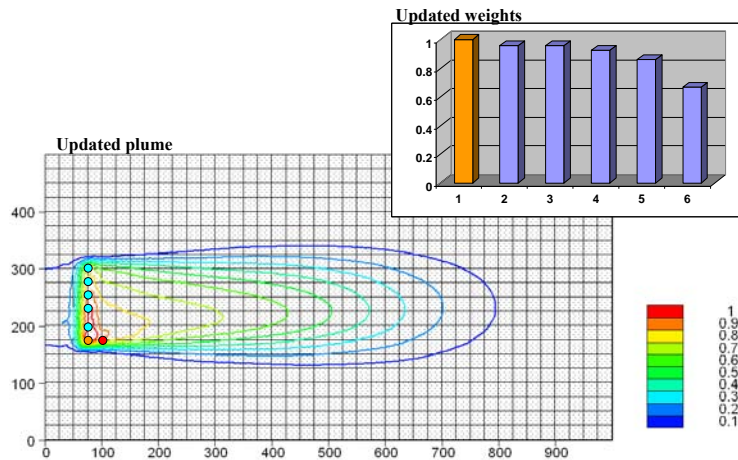


taken a better representation of the true plume is obtained and the unusually high concentration values don't appear anymore.

In the case of the single source problem, the search algorithm converges to the true source location after taking 7 concentration samples. Convergence is assumed to be reached when the weights stabilize. The weights for the rest of the potential source locations are sufficiently small. The weight for sources 3 to 6 is zero and for source 2 is 0.07.



**Figure 14. Simulated plume obtained using the initial source location weights for single source problem.**



**Figure 15. Updated plumes and obtained weights after taking 1 concentration sample for single source problem.**

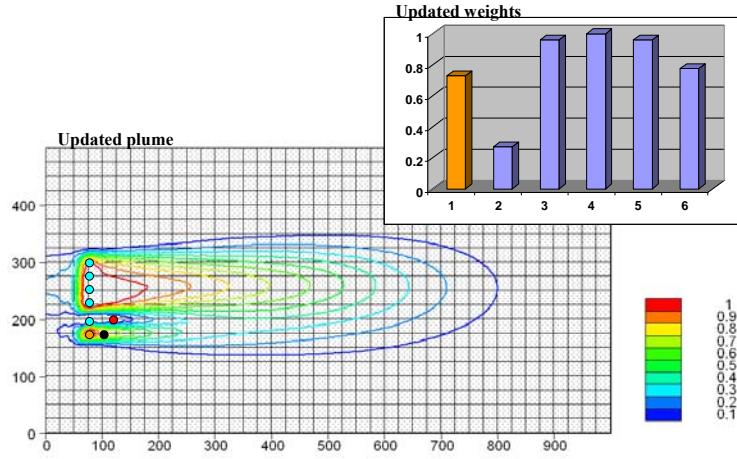


Figure 16. Updated plumes and obtained weights after taking 2 concentration samples for single source problem.

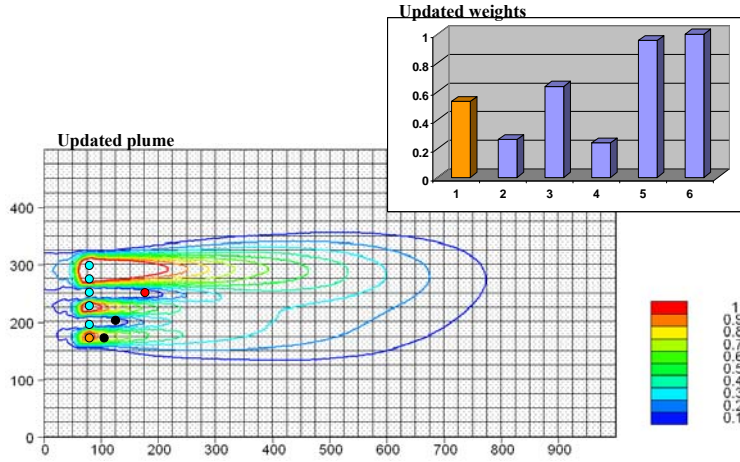


Figure 17. Updated plumes and obtained weights after taking 3 concentration samples for single source problem.

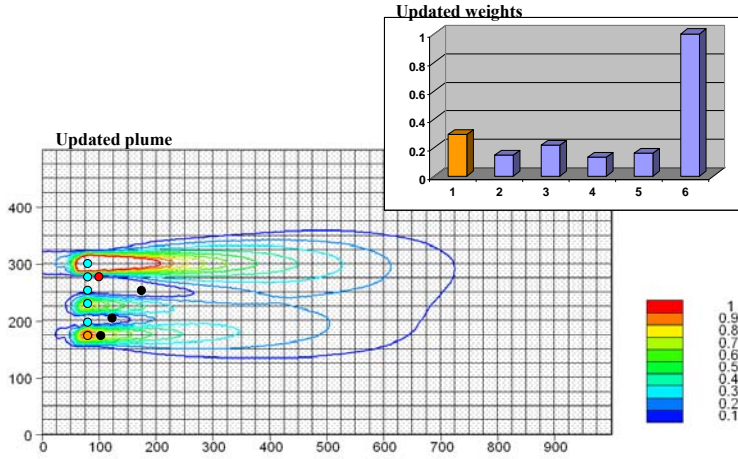


Figure 18. Updated plumes and obtained weights after taking 4 concentration samples for single source problem.

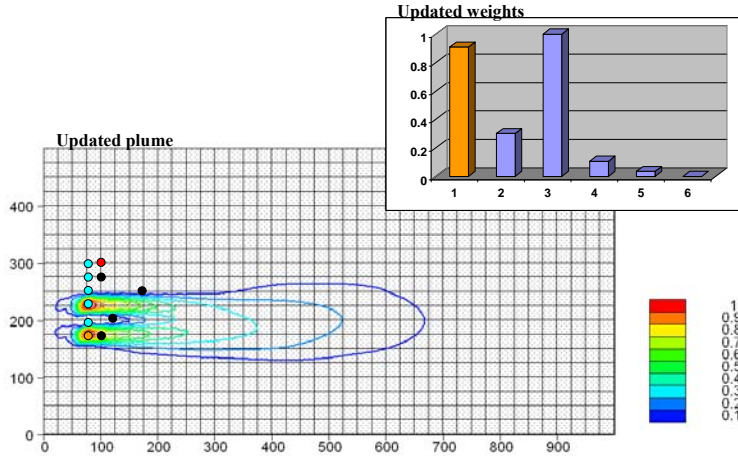


Figure 19. Updated plumes and obtained weights after taking 5 concentration samples for single source problem

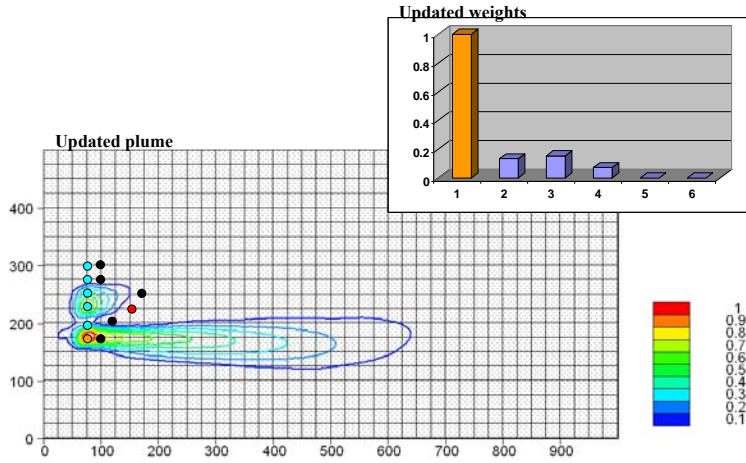


Figure 20. Updated plumes and obtained weights after taking 6 concentration samples for single source problem

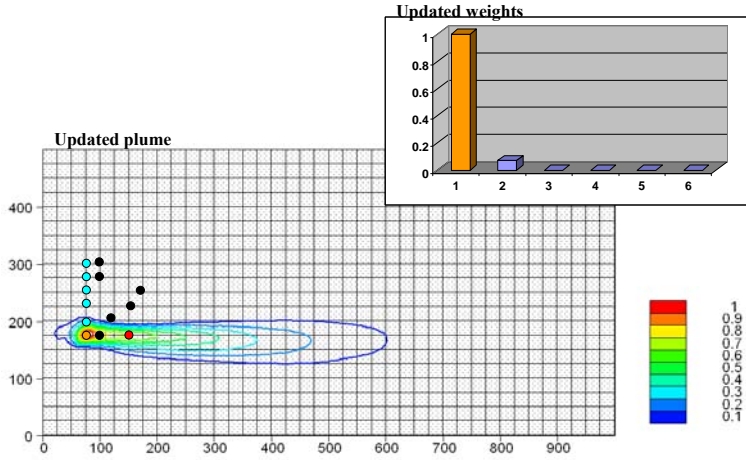
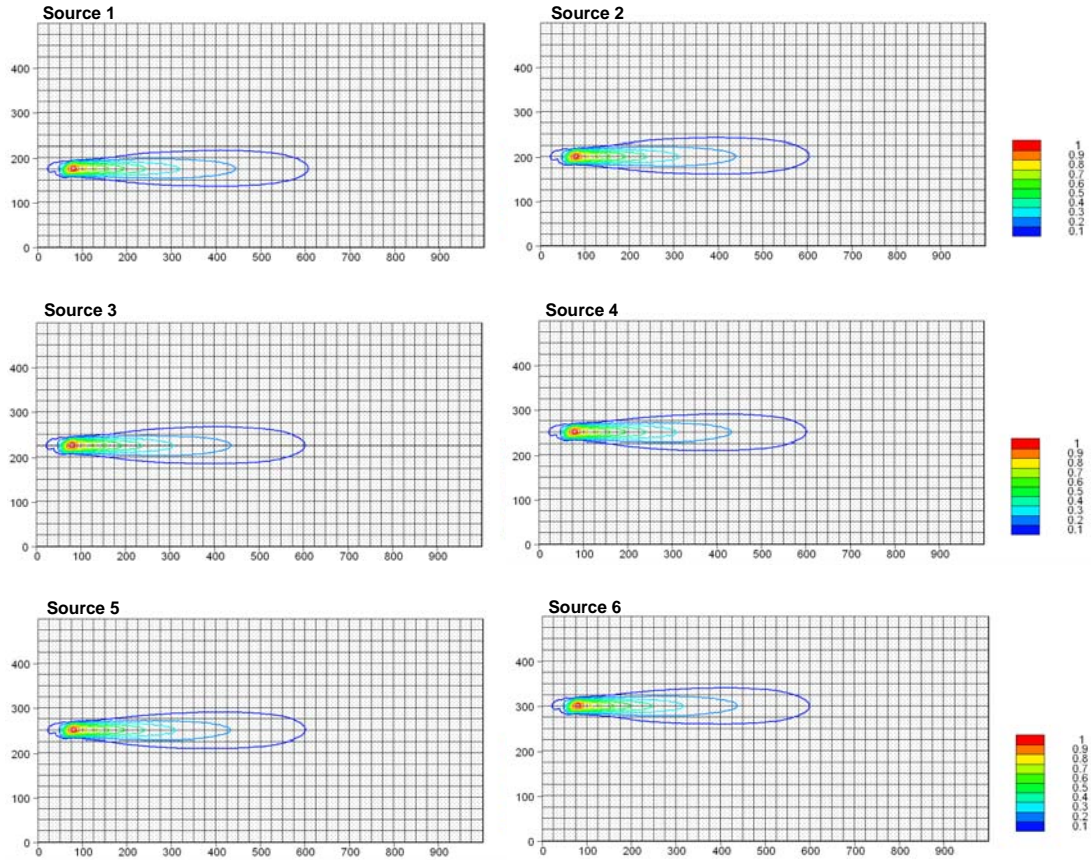


Figure 21. Updated plumes and obtained weights after taking 7 concentration samples for single source problem



**Figure 22. Individual plumes of mean concentration for each potential source location for single source problem**

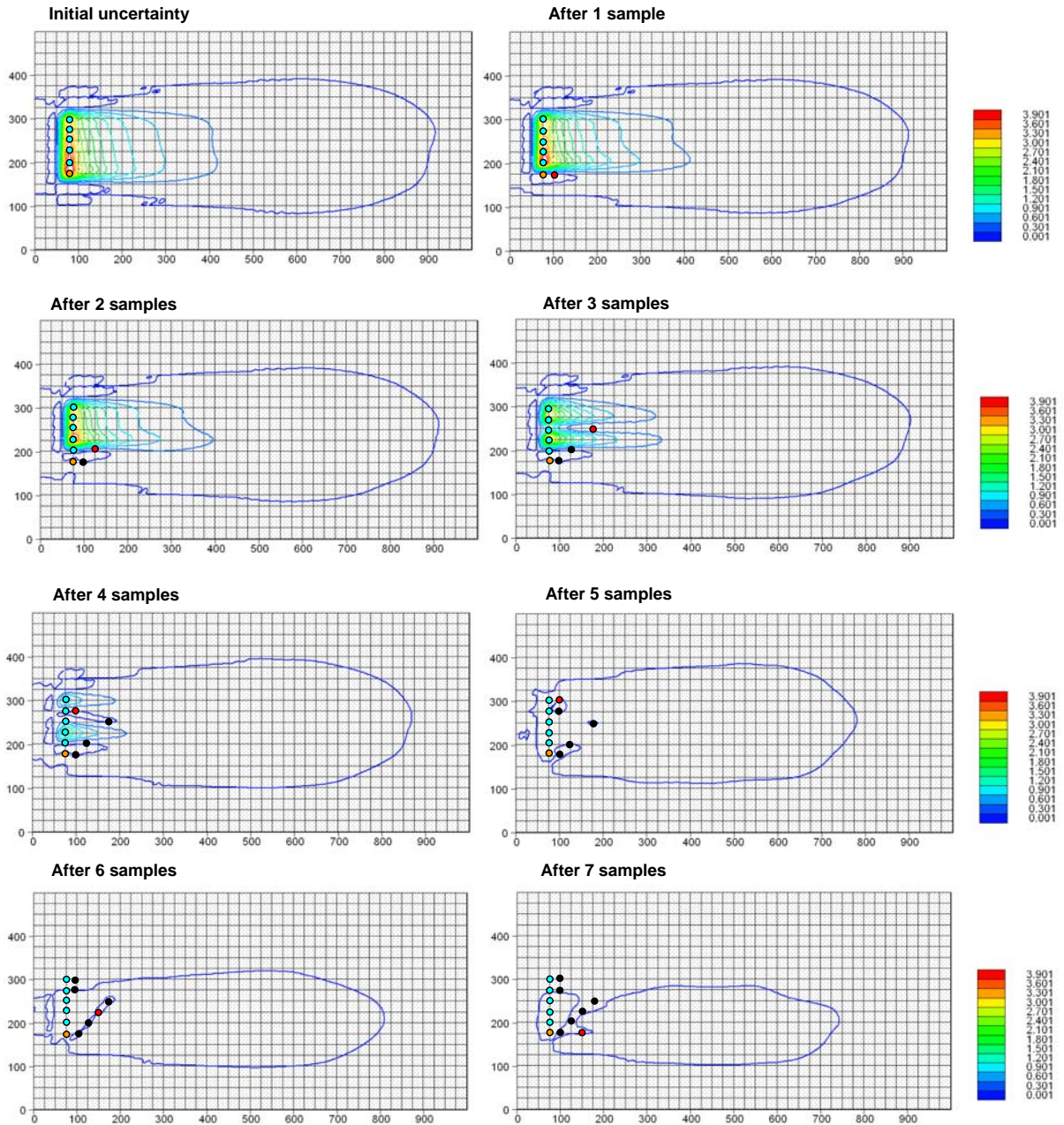


Figure 23. Contaminant concentration uncertainty after taking each sample for single source problem

## 4.2. Sensitivity analysis results

The sensitivity analysis herein shows how variation in input parameters of the algorithm affects its convergence. Investigated are the alpha cuts, initial weights, the true source location and the number of Monte Carlo simulations performed.

Rather than employ a linear scale of  $\alpha$ -cuts (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0), one can consider a logarithmic scale (0.3, 0.47, 0.6, 0.7, 0.78, 0.85, 0.9, 0.95, 1.0). Using this scale, the algorithm converges after taking just 7 samples and also the

weights that correspond to non-true potential source locations are reduced to values quite close to zero. This latter phenomenon implies that higher concentration  $\alpha$ -cuts are more important than lower ones, when comparing contaminant plumes.

Testing the algorithm using uniform initial weights (all equal to unity), we observed that convergence is not very sensitive to initial weights; the algorithm identified the true source after taking 7 samples.

In the synthetic example, the number of Monte Carlo simulations was kept low (100 realizations) due to time constraints. Increasing the realizations to 500, the algorithm reached convergence in 7 iterations, which is the same as the case with 100 realizations. This demonstrates that the algorithm is not significantly affected by the number of realizations employed.

The sensitivity analysis demonstrates that, overall, the algorithm is relatively insensitive to parameter changes. An exception to this is the number and type of  $\alpha$ -cuts, where adjusting the scale of the alpha cuts can positively affect the outcome and definitiveness of the DNAPL source search.

## 5. Field Applications

The final and most important test of the proposed search algorithm was its application to real world problems. Two field sites, where DNAPL contamination has been detected and the approximate location and depth of the DNAPL source has been identified by site experts, were selected in order to test the DNAPL source search algorithm. In one case (Anniston Army Depot), the DNAPL source location was withheld from algorithm testing and revealed to us after the algorithm identified a source ('blind test'). In the second case (Hunters Point Shipyard), the source was only approximately known during algorithm testing and a precise location was ultimately suggested by the source finder.

### 5.1. Anniston Army Depot

#### 5.1.1. Site description

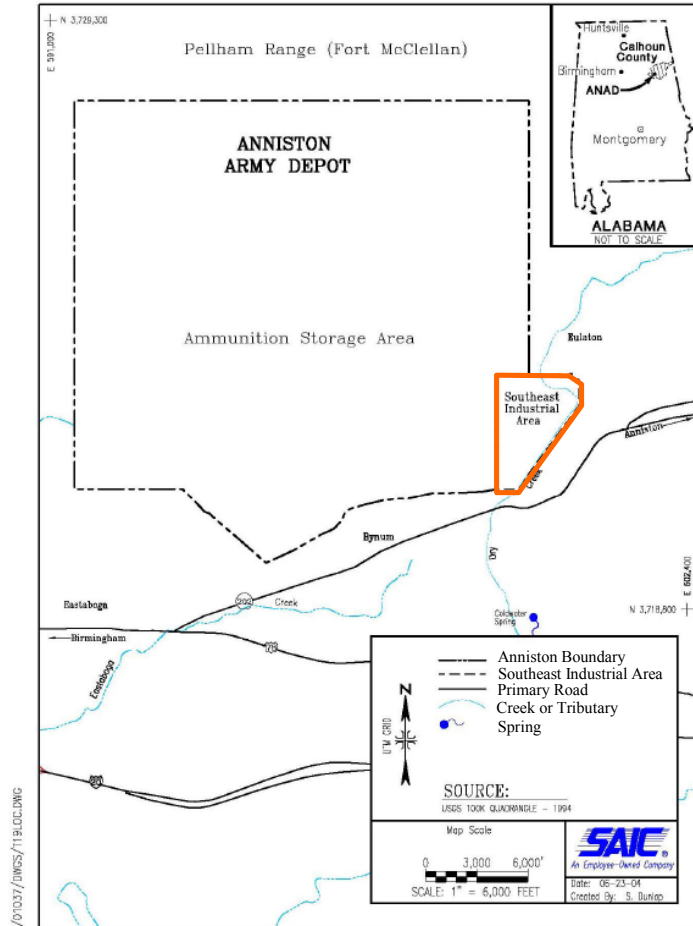
This section describes the site that was chosen as a 'blind test' site for the DNAPL source search algorithm. The site is located at the Anniston Army Depot (ANAD) in Anniston, Alabama. All site information summarized herein was taken from the feasibility study and remedial investigative reports (SAIC, 2005, 2006). The ANAD was used as a munitions storage facility since its construction in 1941 and consists of the Southeast Industrial Area (SIA) and Ammunition Storage Area (ASA) amidst office buildings and warehouses (Figure 24)

Activities performed at ANAD included overhauling, testing and storage of combat vehicles and munitions. ANAD also performed maintenance on weapons, ammunition, missiles and chemical munitions. Historically, these activities have resulted in the production of hazardous solid and liquid wastes, such as metals, cyanide, phenols, pesticides, herbicides, chlorinated hydrocarbons, petroleum hydrocarbons, solvents, acids alkalis, chelating agents, asbestos and creosote. These wastes were disposed of on-site in trenches, lagoons, and landfills from the 1940s through the late 1970s.

Investigations regarding the quality of groundwater at ANAD have confirmed contaminant migration to the groundwater. Thus, ANAD entered an Environmental Protection Agency (EPA) program to develop and implement a remedial strategy to address the groundwater quality problem.

ANAD has a total of 47 Solid Waste Management Units (SWMUs) considered by the Army as requiring restoration; 29 are located in the SIA. Previous investigations indicate that 9 of these 27 are possible source areas for DNAPL contamination of groundwater. In this study we will focus on SWMU 12 (Facility 414 Old Lagoons).





**Figure 24. ANAD location (After SAIC, 2006).**



**Figure 25. SWMU 12 location (black rectangle) and model domain (red boundary) (After SAIC, 2006).**

SWMU 12 is located on the southwest part of the SIA (Figure 25). It is currently covered with fill, is heavily vegetated and has an overall level elevation. SWMU 12 consisted of three lagoons used for the disposal of abrasive dust wastes containing cadmium and possibly lead, metal plating, cleaning solutions, fuels, oils and solvents and was used from 1960 to 1978. In 1978 the liquid from the lagoons was pumped to the A-Block Lagoon, a lined surface impoundment. The lagoon remnants were piled until November 1982, when the pile was excavated and 9,594 tons of materials were transported to an authorized hazardous waste landfill facility. In 1996 and 1997, in-situ chemical oxidation via hydrogen peroxide injection was applied to the SWMU 12 soils and groundwater.

The geologic profiles from the surface to depth below the SIA consists of a clay residuum, weathered bedrock and unweathered bedrock that collectively comprise an interconnected aquifer. The average depth to the water table was 27.4 feet below the top of casing. The flow of groundwater is illustrated by a groundwater potentiometric map (Figure 26) that was later employed to calibrate a groundwater flow and transport model. More site information is provided by SAIC (2005, 2006) and Dokou (2008).

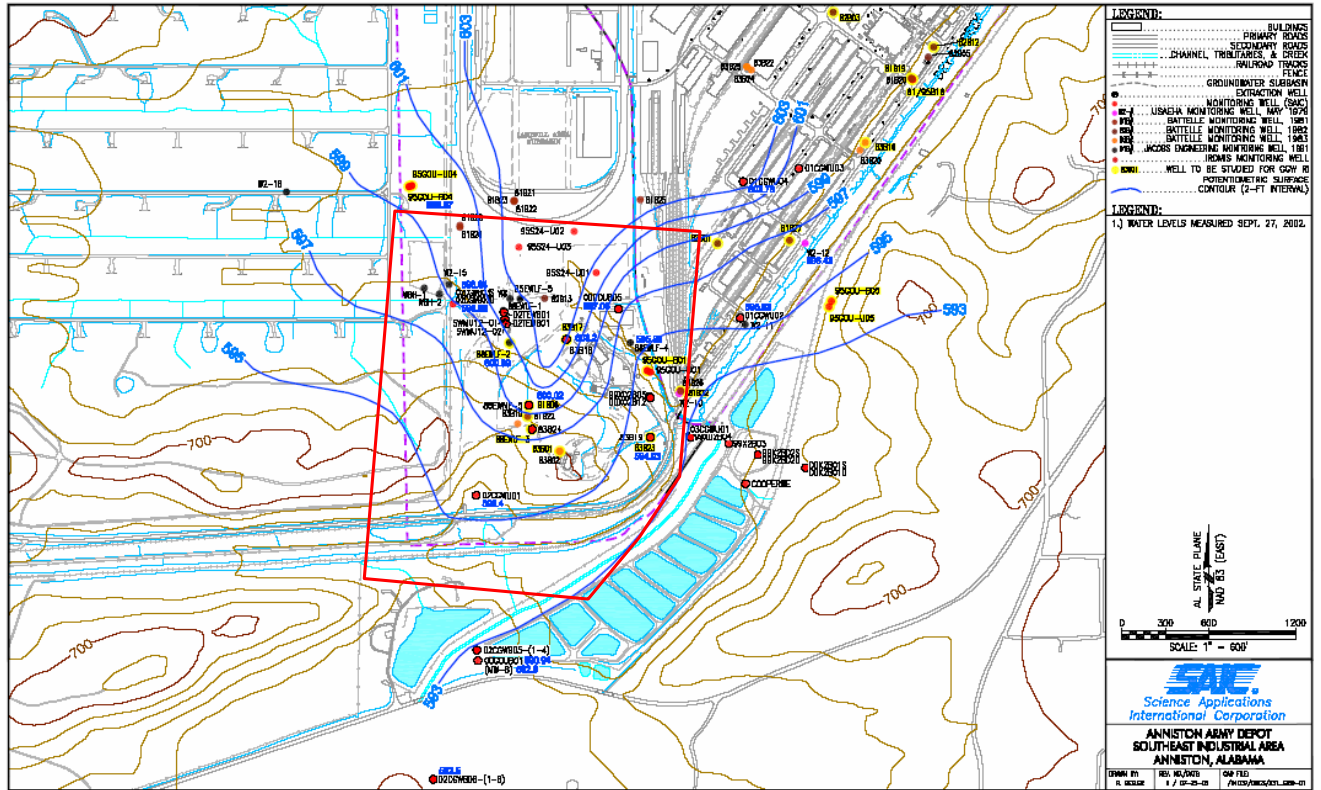


Figure 26. Potentiometric map (After SAIC, 2006).

### 5.1.2. Groundwater flow and transport model

A model was developed for the DNAPL source search algorithm that was based upon a more extensive MODFLOW model constructed by SAIC. The model presented herein consists of 6 layers (Figure 27) and a mesh with 964 nodes (Figure 28), and was solved by the Princeton Transport Code (PTC).

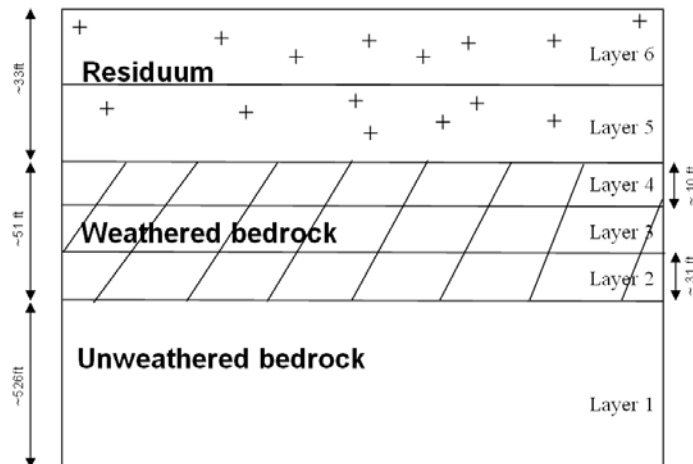
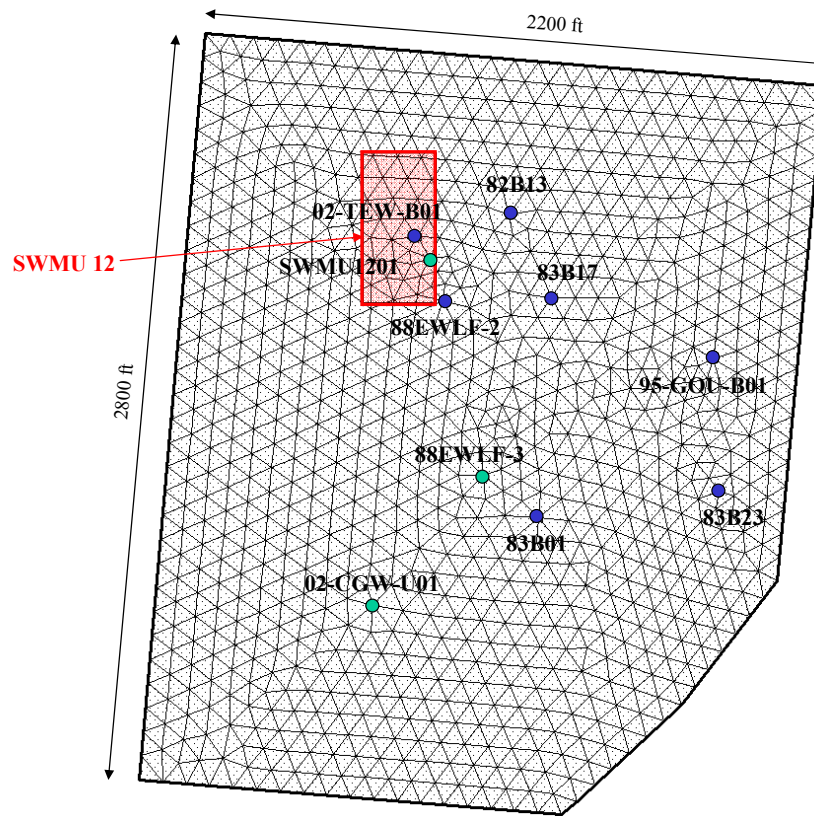


Figure 27. Vertical discretization of the model domain.



**Figure 28. Finite element grid and location of monitoring wells. Green circles represent wells screened in the residuum interval and blue circles wells screened at the weathered bedrock interval.**

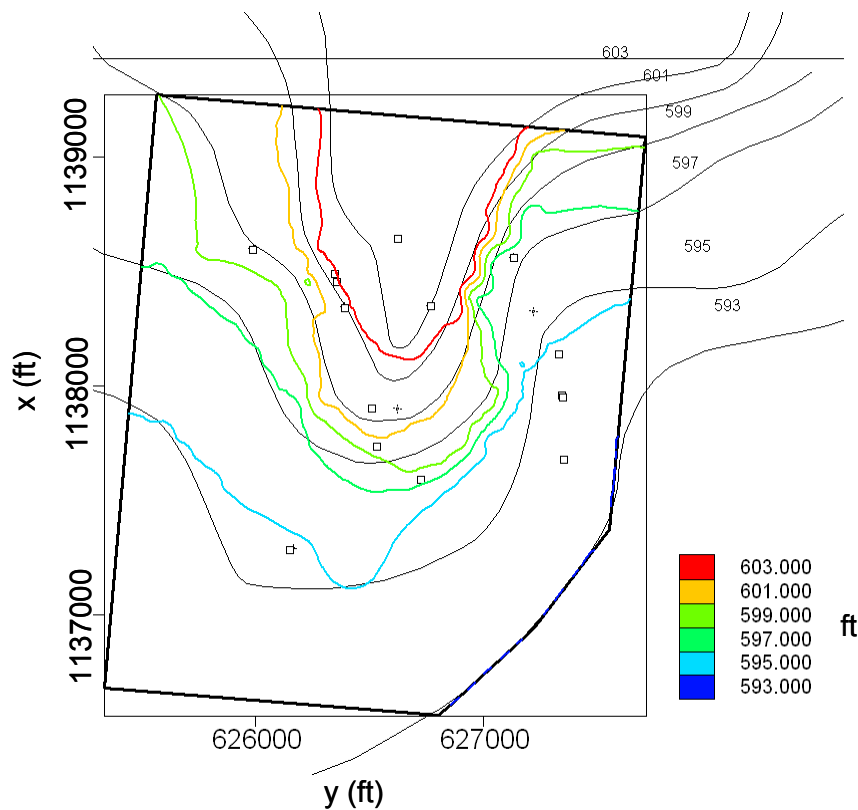
Boundary conditions are artificial constant head values specified in order to calibrate the flow and transport model. Each geological layer was represented by a mean hydraulic conductivity ( $K$ ) value and specified variance. However, the type of data available dictated different methods by which the hydraulic conductivity statistics were determined. The resulting random variable hydraulic conductivity values, however, we all sampled using Monte Carlo techniques. The mean hydraulic conductivity was specified for model calibration purposes by SAIC in the residuum and in the unweathered bedrock (0.028 ft/day and 6 ft/day, respectively). In both cases, the hydraulic conductivity is assumed isotropic. Since there were no actual hydraulic conductivity measurements available to facilitate the creation of a variogram for these two geological layers, an expert-provided correlation length permitted the construction of a variogram model. With this measure of uncertainty, 200 realizations of hydraulic conductivity for these layers were generated using Lhs.

Calibrated values of hydraulic conductivity for the weathered bedrock, as determined by SAIC modelers, ranged from 0.15 ft/day to 850 ft/day for horizontal hydraulic conductivity and from 0.015 ft/day to 30 ft/day for vertical hydraulic conductivity. The mean of these calibrated values was used as the mean of this layer's random variable. Uncertainty for hydraulic conductivity in this layer was determined

with variogram analysis and used to facilitate the creation of 200 hydraulic conductivity realizations for the eventual Monte Carlo simulations.

Water quality monitoring wells whose observations we used to run the search algorithm number ten and are located in the residuum (3 wells) and the weathered bedrock (7 wells). Their locations are plotted in Figure 28. More detailed model information is provided by Dokou (2008).

The calibrated flow field simulated by the stochastic model is shown in Figure 29. The colored contours represent calibrated Monte Carlo simulated hydraulic head results, and the black contours represent the hydrogeologists interpretation of the well water level measurements. This is considered to be a fairly reasonable match, given that hydraulic conductivity values were provided by SAIC and, as such, were not varied in the calibration process.

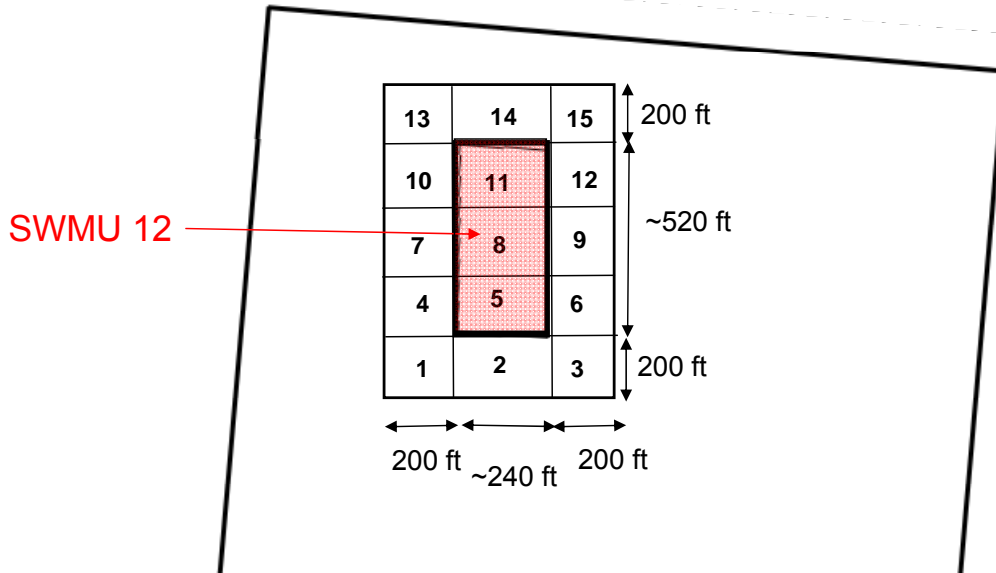


**Figure 29. Flow field results for stochastic model (colored contours) and potentiometric map created by hydrogeologist using well water level measurements (black contours).**

### 5.1.3. Source search algorithm

Figure 30 shows the 15 preliminary potential source location choices. Each block represents a potential source location. The SWMU 12 area was divided in three areas (northern part - block 5), middle part (block 8) and southern part (block 11). An area of 200 ft around SWMU 12 was also considered and divided into 12 blocks in a similar

way. Potential source locations (blocks) are defined in both the residuum and weathered bedrock layers. Each source block includes several finite element nodes.

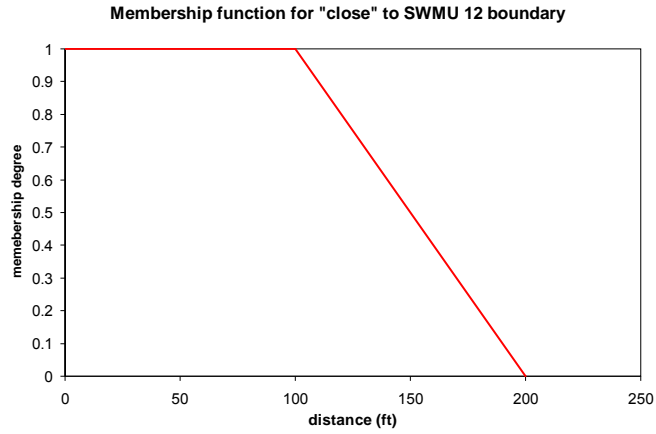


**Figure 30. Preliminary potential source locations.**

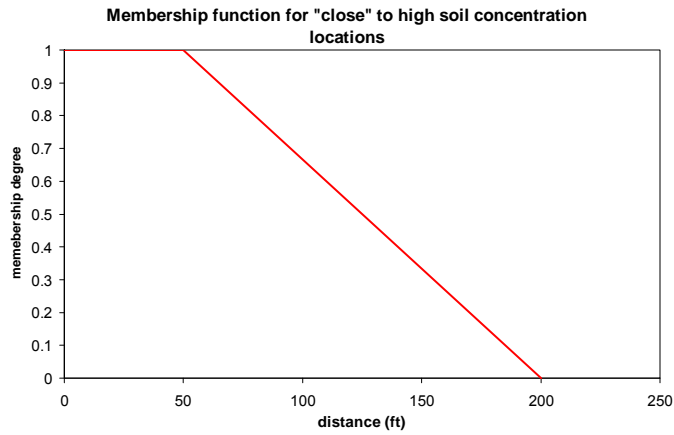
The Choquet integral was used to calculate the initial weights for each source location that represent our confidence that the source belongs to the group of true source locations. Three important features were considered:

1. Distance of source block center from the borders of SWMU 12
2. Distance of source block center from locations with high soil concentrations
3. Distance from the average TCE contour that is greater than 10,000  $\mu\text{g/L}$

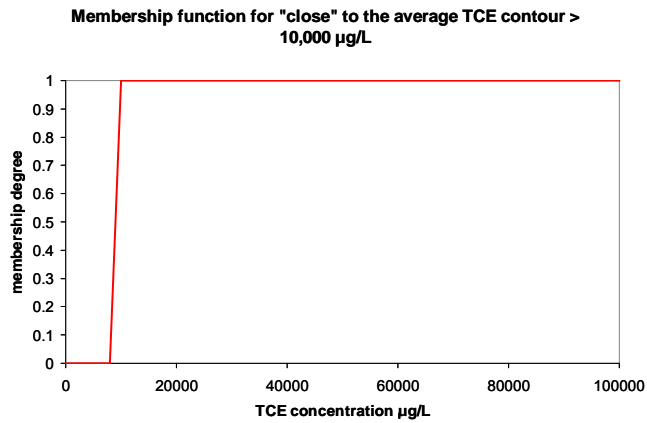
For each feature a membership function was created by the site expert. Figure 31 shows the membership function that represents the meaning of ‘close’ to the SWMU 12 borders. If the distance of the source block center from the SWMU12 border is less or equal to 100 feet then it is assigned a membership degree of 1. For distances greater than 100 feet the membership degree is a linear function of distance. In Figure 32 the membership function that explains the meaning of ‘close’ to the high soil concentration locations is presented. The membership function for the last feature, i.e. the meaning of ‘close’ to the average TCE contour greater than 10,000  $\mu\text{g/L}$  is shown in Figure 33.



**Figure 31. Membership function for 'close' to the SWMU 12 boundary.**



**Figure 32. Membership function for 'close' to the high soil concentration locations.**



**Figure 33. Membership function for 'close' to the average TCE contour greater than 10,000 µg/L.**

Figure 34 shows the locations where high soil concentrations were found. Table 3 summarizes all the information needed and the final results for the Choquet integral technique.

The source blocks that have the highest global score are numbers 5, 6, 8 and 9. We chose to use only two of these potential source locations (numbers 8 and 9) and include number 7 in the final set of potential source locations that were used as input to the search algorithm. This was done because these three locations are located adjacent to each other horizontally and they would provide a good test for the algorithm. Furthermore, location 7 has the fifth largest global weight. Sources 5 and 6 are located directly below locations 8 and 9 and they would potentially ‘block’ locations 8 and 9. It would be interesting to include locations 5 and 6 in future work as part of the potential source locations set. For now, we chose three locations (7, 8 and 9) with corresponding initial weights 0.41, 1 and 0.9.

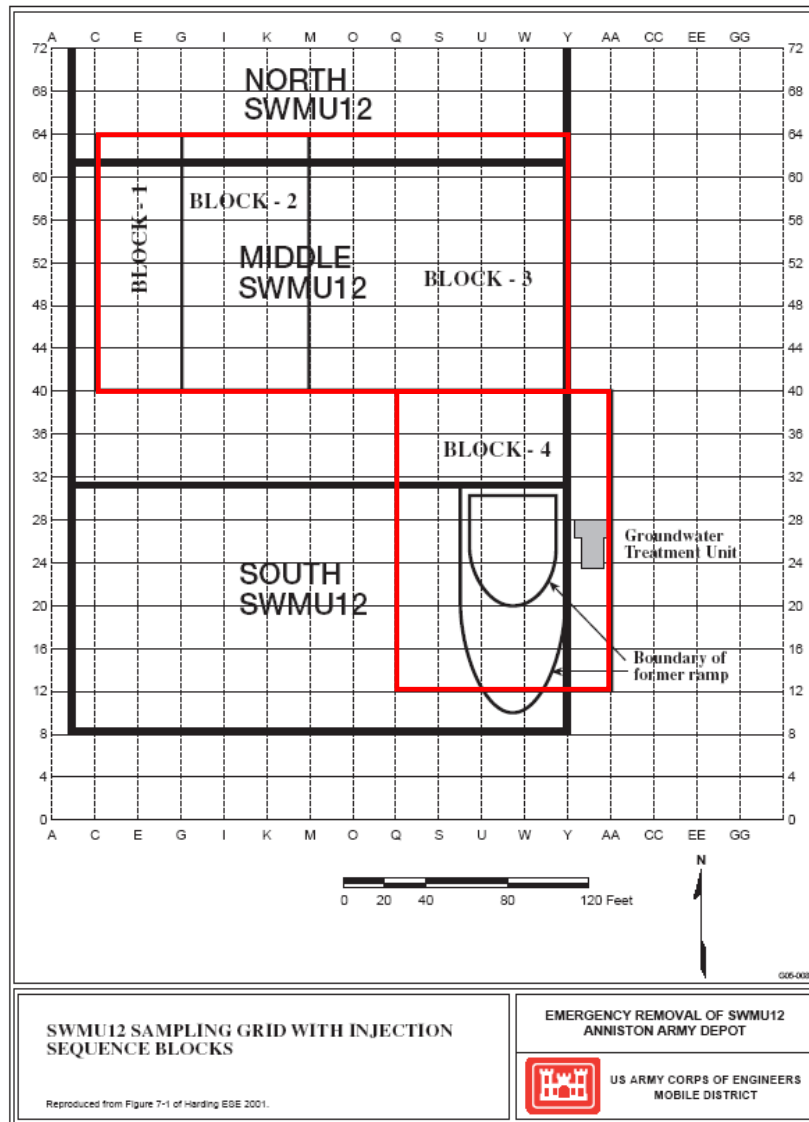


Figure 34. Locations with high soil concentrations (red blocks).



**Table 3. Choquet integral results for 15 preliminary potential source locations**

	SWMU 12		High soil concentrations		TCE contour	Choquet Integral
	distance (ft)	membership degree	distance (ft)	membership degree	membership degree	Global score
1	138.6	0.614	283.8	0	0	0.0921
2	100	1	118.8	0.4684	0	0.3936
3	138.6	0.614	138.6	0.3298	0	0.2763
4	100	1	138.6	0.3298	0	0.3342
5	0	1	26.4	1	1	1
6	100	1	79.2	0.7456	1	0.9416
7	100	1	112.2	0.5146	0	0.4134
8	0	1	0	1	1	1
9	100	1	99	0.607	1	0.902
10	100	1	171.6	0.0988	0	0.2352
11	0	1	125.4	0.4222	0	0.3738
12	100	1	158.4	0.1912	0	0.2748
13	138.6	0.614	330	0	0	0.0921
14	100	1	303.6	0	0	0.15
15	138.6	0.614	330	0	0	0.0921

5.1.4. Test results

In running the search algorithm, we discovered that well 02-CGWU01 was producing an infeasible solution to the optimization formulation, thus it was removed from the data set and the algorithm was run with the remaining 9 real field samples. The results from this run are shown in Figure 36 to Figure 45. The sequence of water quality samples selected by the algorithm is presented in Table 4.

From the results it is evident that sample 2 (well SWMU 1201) is very important and it dictates the results and algorithm convergence. This sample has the maximum concentration value in the data set and it is close to the potential source locations. The samples taken after sample 2 are not very informative. We observe that the first 3 samples are the ones that are inside the plume boundary. The rest are located west of the plume boundary.

**Table 4. Sampling sequence information**

	Well	PTC Layer	Geologic Layer	Average TCE concentration (µg/L)
1	02-TEW-B01	2	Weathered Bedrock	820
2	SWMU 1201	5	Residuum	129,580
3	88EWLF-2	3	Weathered Bedrock	155
4	82B13	4	Weathered Bedrock	2.6
5	83B01	2	Weathered Bedrock	3.83
6	95-GOU-B01	4	Weathered Bedrock	0.66
7	83B17	4	Weathered Bedrock	1.47
8	83B23	2	Weathered Bedrock	0.59
9	88EWLF-3	5	Residuum	0.53

The algorithm assigns a value of 1,100,000  $\mu\text{g/L}$  for each of locations 2 and 3 in the residuum interval. This magnitude equals the solubility limit of TCE and is the upper bound constraint of the optimization problem. Source location 3 in the residuum interval has the highest weight. The weight assigned to source location 2 is not very high (0.28) but it is still considered part of the true source zone area because it has a high magnitude and because the weight of 0.28 is not reduced at all even after taking all the 9 samples. We observe that the weights do not always follow the distribution of the magnitudes.

We conclude that both source locations 2 and 3 are probable true source locations, source location 3 having a higher probability of being the true source area. The depth of the DNAPL source area was identified by the algorithm as the residuum interval.

We need to note here that usually when the magnitude is high for a particular source location the corresponding weight is likely to be high and vice versa, but they are not always proportional. This is because the geometries of the plumes might fit differently regardless of the actual magnitudes of the data, since at the comparison step all plumes are normalized and moreover, because the composite plume is being updated by the Kalman filter after the choice of magnitudes is made.

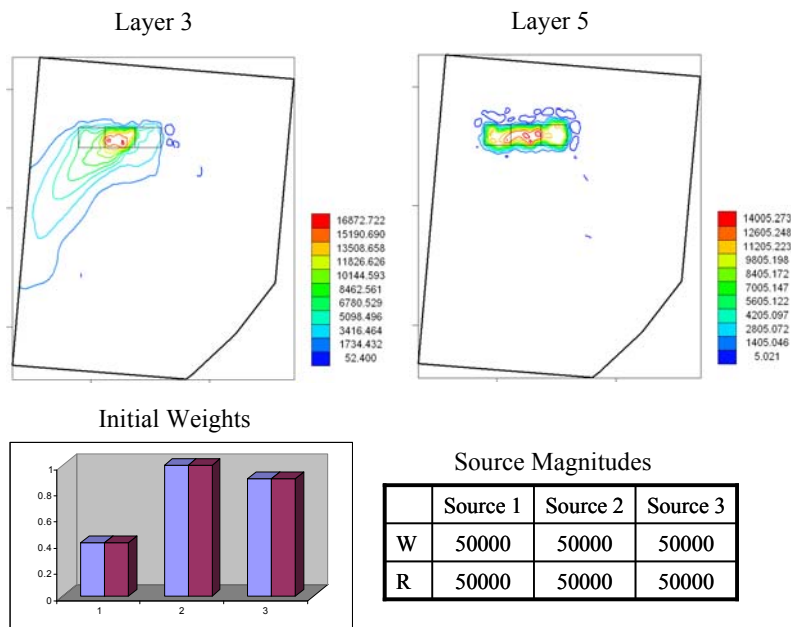


Figure 35. Search algorithm results for case 2 – real data before taking any samples.

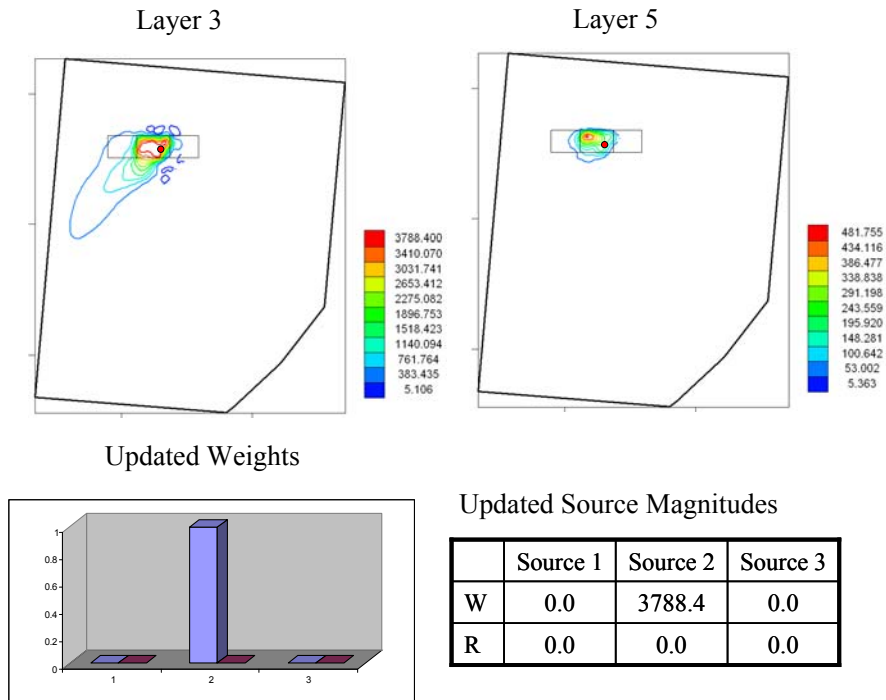


Figure 36. Search algorithm results for case 2 – real data after taking 1 sample.

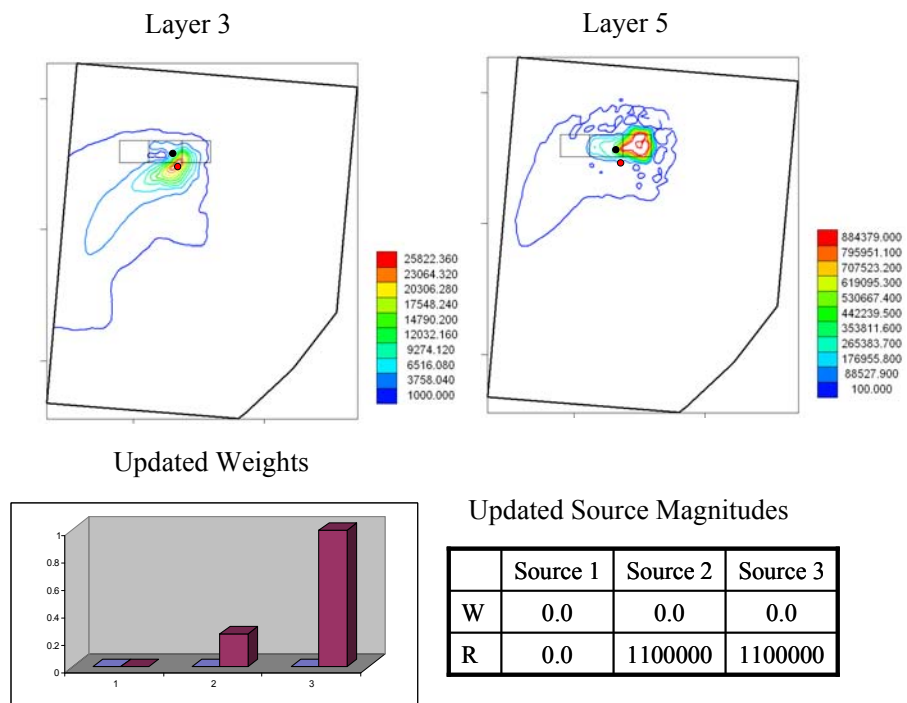


Figure 37. Search algorithm results for case 2 – real data after taking 2 samples.

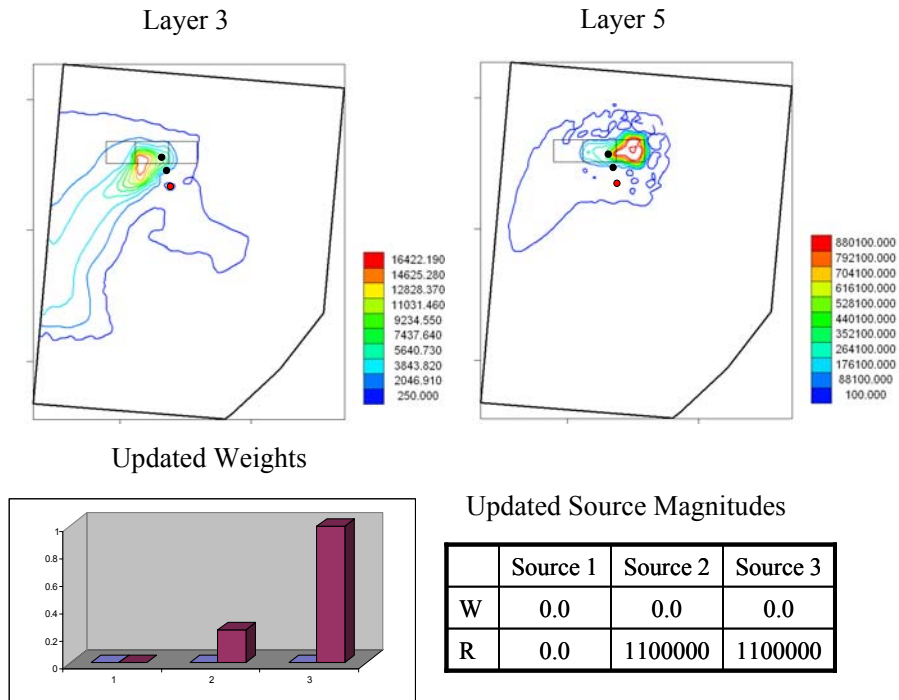


Figure 38. Search algorithm results for case 2 – real data after taking 3 samples.

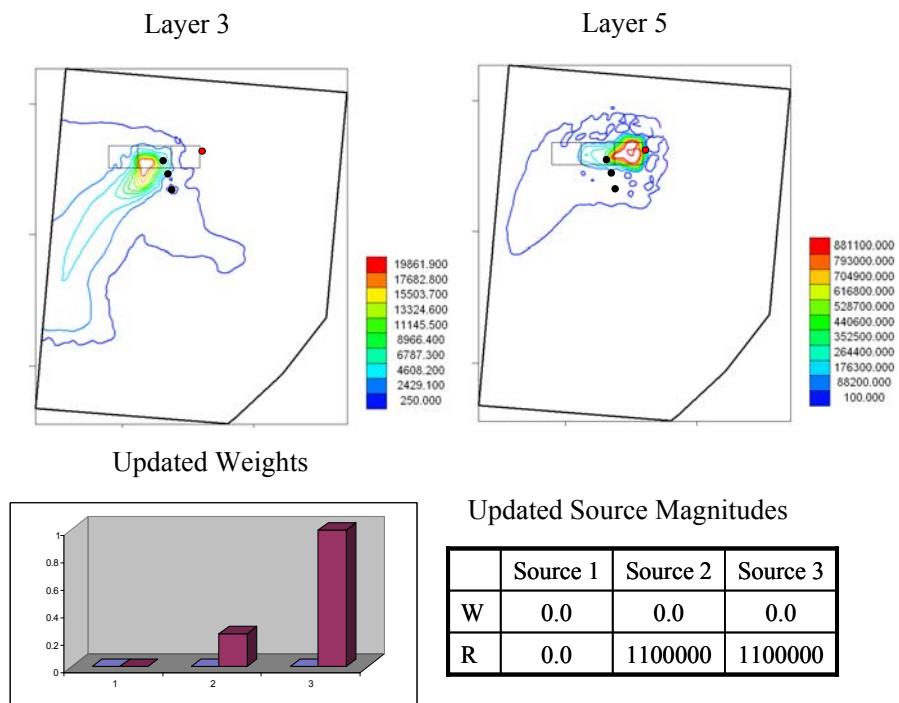


Figure 39. Search algorithm results for case 2 – real data after taking 4 samples.

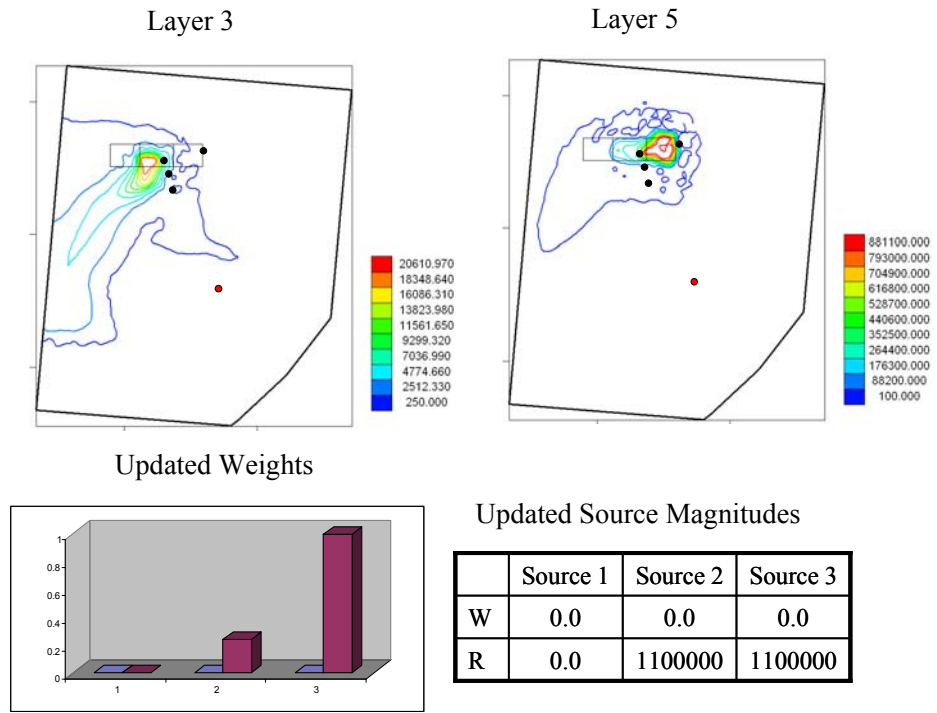


Figure 40. Search algorithm results for case 2 – real data after taking 5 samples.

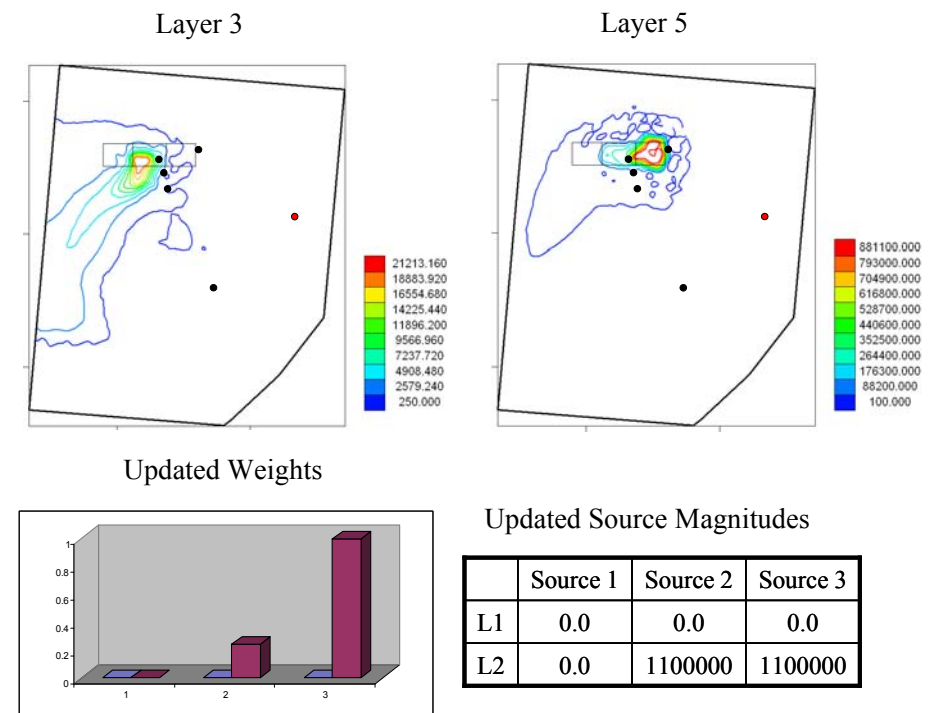


Figure 41. Search algorithm results for case 2 – real data after taking 6 samples.

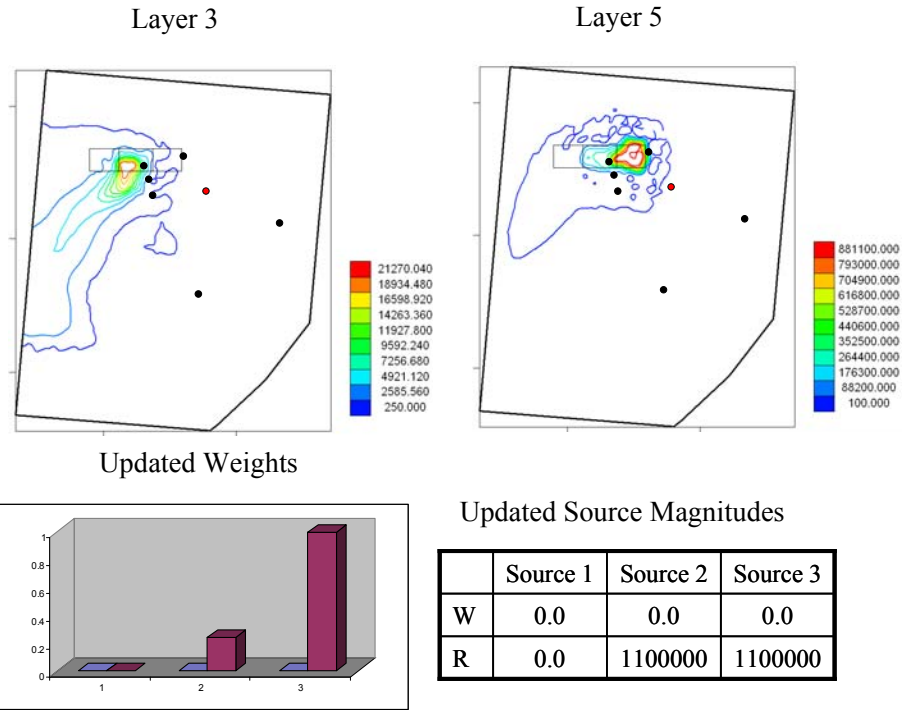


Figure 42. Search algorithm results for case 2 – real data after taking 7 samples.

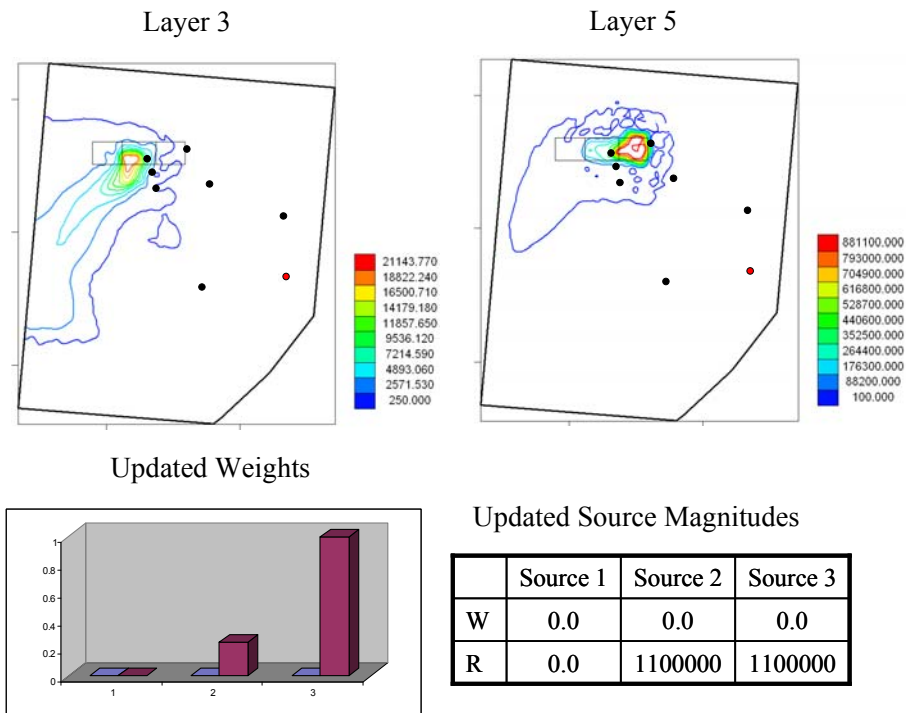
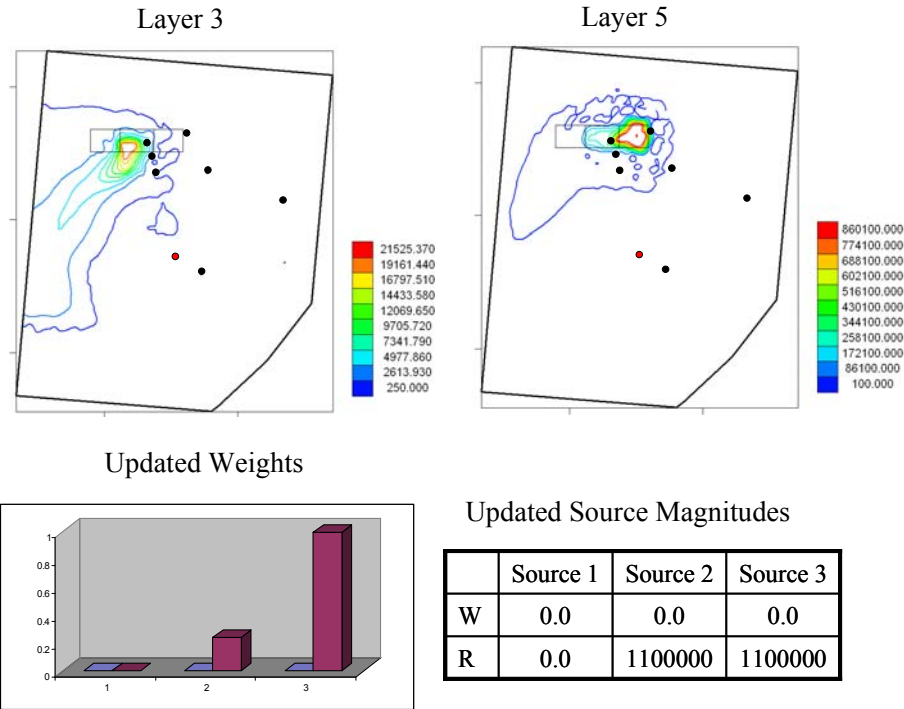


Figure 43. Search algorithm results for case 2 – real data after taking 8 samples.



**Figure 44. Search algorithm results for case 2 – real data after taking 9 samples.**

After the algorithm tests were completed the true source location was revealed to us. The best estimate of the site experts regarding the true source location is that the whole SWMU 12 is a source area (see Figure 30). They are also suspecting that the DNAPL source has moved to the west of the SWMU 12 area. Regarding the DNAPL depth, their belief is that most of the DNAPL is still in the residuum interval, as indicated by our algorithm. But, they have evidence that some of the DNAPL source has moved to the weathered bedrock interval as well. The most challenging question they still need to answer is what depth the DNAPL has reached.

In light of this evidence, we can conclude that the choice of the upper bound limits as the values for the magnitudes for both source locations 2 and 3 can be a result of the fact that the area of the potential source location chosen is much smaller than the actual source area. The algorithm is trying to compensate for the smaller area by choosing the largest possible source magnitude.

The tests performed here included three potential source locations. Two of them were outside the SWMU 12 area (source locations 1 and 3) and one was in the SWMU 12 area (source location 2). Our algorithm was successful in including source location 2 in the true source location selected and also confirmed the suspicion that the DNAPL source has moved to the west of the SWMU 12 (source location 3). The estimated DNAPL source area though is larger than the one identified by the algorithm. The DNAPL depth chosen by the algorithm is the same as the one suspected by the site experts (residuum interval).

It is very important to note that the biggest challenge in this work was to identify the correct source by using water quality data that were collected from the wells in less than optimal locations. As can be seen in the figures presented above the direction of the

plumes that emanate from the source areas is southwest, due to the curved flow field created by the complex hydraulic conductivity field. All of the available data are either in the middle of the model domain or to the east of the source locations.

Another important factor that might have affected the algorithm performance is the existence of a pumping well on the southeast of the model domain. This well is outside the model domain but it has been verified by the site experts that it affects the hydraulic heads and consequently the contaminant concentration field inside the model domain. The drawdown area of the pumping well is shown in Figure 46. The effect of the pumping well would be to pull the plumes towards the east.

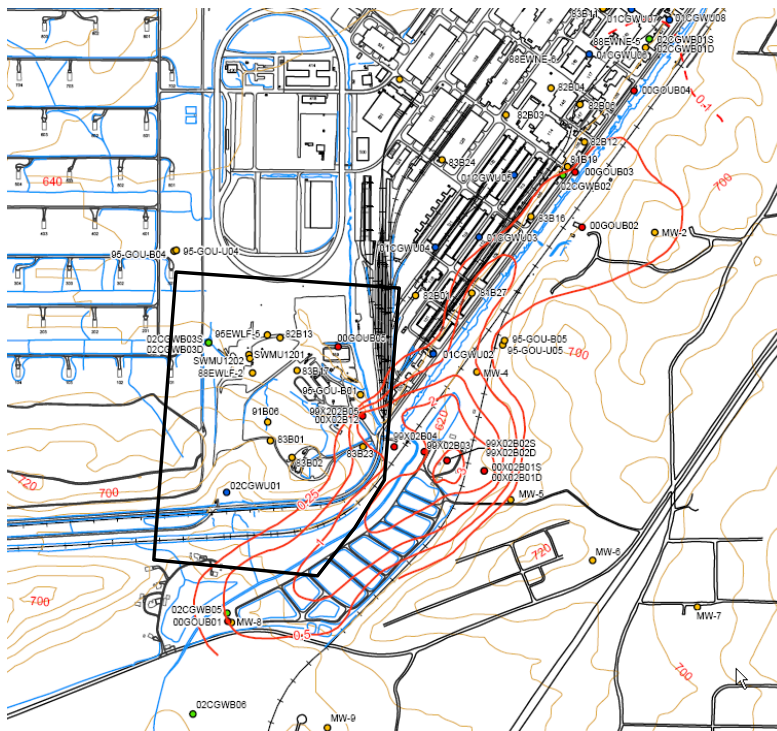


Figure 45. Pumping well drawdown area (After SAIC, 2006)

## 5.2. Hunters Point Shipyard

### 5.2.1. Site description

This section describes the site that served as a second test of the DNAPL source search algorithm. The site is located at Hunters Point Shipyard (HPS) in San Francisco, California (Figure 47). This site is divided into five remedial units and the remedial unit (RU) of particular relevance and interest to this investigation is named RU-C5. All information pertinent to RU-C5 and HPS overall, as summarized in this report, is provided in detail in a feasibility study report (SulTech, 2008) and a technical memo (CE2, 2006).

HPS was used generally for ship repair and maintenance by the Navy from 1940 until its deactivation in 1974. The Navy leased the land to a private ship repair company and resumed occupancy of the land in 1987. Due to the aforementioned activities



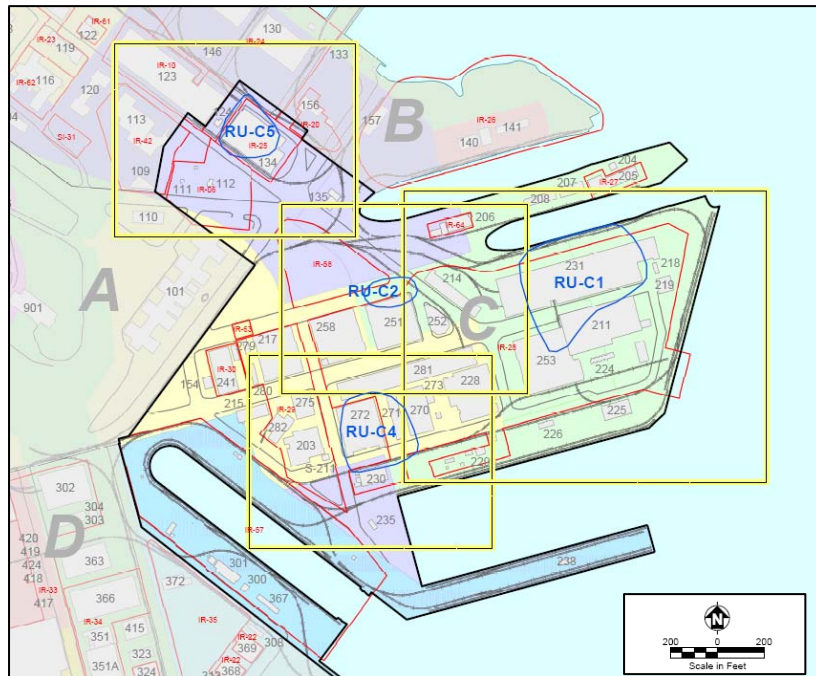
historically performed on site, certain chemicals of concern reside in the soil and groundwater, potentially posing risk to human populations. These chemicals include metals such as antimony, arsenic, cadmium and copper and organic compounds such as trichloroethene (TCE), tetrachloroethene (PCE) and vinyl chloride. Because of the presence of such hazardous materials at HPS the entire site was placed on the National Priorities List in 1989 as a Superfund site. HPS was then designated for closure in 1991.



**Figure 46. Hunters Point Shipyard is located on San Francisco Bay in southern San Francisco; image courtesy of (SulTech, 2008)**

RU-C5 is located in the northwestern part of Parcel C at HPS (Figure 48). Building 134, used for parts cleaning by the Navy from 1940 to 1974, is located in RU-C5 and contains a concrete dip tank that drained into a below-grade sump that resides partly in the building. It is this sump and dip tank that is suspected to be the source of a DNAPL plume that generally emanates from the building's footprint. A second DNAPL plume containing much lower concentrations originates at the location of a former fuel tank farm located southwest of Building 134. This second plume and associated water quality samples are not considered in the ensuing analysis, as the Building 134 DNAPL plume is of primary concern.

The general geologic profile is comprised of five geologic units from the surface downward: 1) Artificial Fill, 2) Undifferentiated Upper Sand, 3) Bay Mud, 4) Undifferentiated Sediments, and 5) Bedrock. A number of borehole investigations and hydraulic conductivity measurements for all units permit a fairly comprehensive hydrogeological characterization of the RU-C5 subsurface. This characterization is discussed next.



**Figure 47. RU-C5 is the most northwestern remedial unit at Hunters Point Shipyard; Building 134 is located in the center of RU-C5; image courtesy of TetraTech (TetraTech, 2004)**

### 5.2.2. Hydrogeologic characterization

Unlike the field test at ANAD, the site investigation at HPS provided a wealth of data that contributed to a detailed characterization of the site's hydrogeology. As a first step in such a characterization, borehole logs from RU-C5 were transformed from the traditional qualitative descriptions into quantitative soil contribution percentages. This was a particularly straightforward task for most boreholes, as soil percentages accompanied the qualitative soil descriptions. Consider, as an example, the qualitative soil description of a soil sample at a depth of 5 feet below ground surface at well IR06MW32A:

*Grayish Green Gravelly Silt with Sand.*

The corresponding quantitative soil contribution percentages, as provided by the geologist are:

*20% gravel, 15% sand, 5% clay (60% silt is implied).*

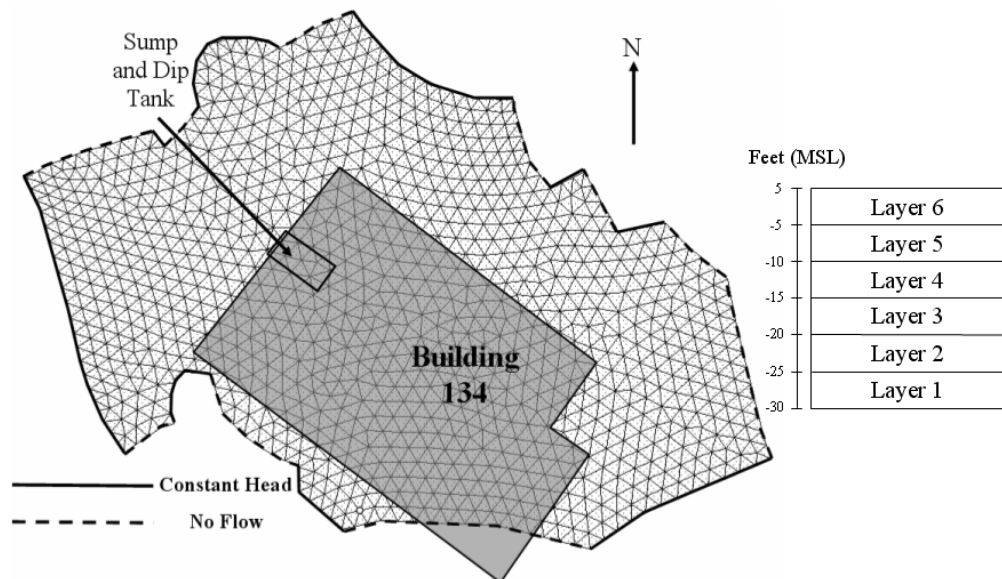
For those boreholes without an accompanying quantitative form, the Burmeister soil classification was applied as in Ross et al (2007). What results after the quantification step is a quantified representation of the soil along vertical profiles at 100 borehole locations that are located more uniformly throughout RU-C5 than are the hydraulic conductivity measurements.

It is well documented that a relationship exists between soil composition and hydraulic conductivity. Ross et al (2007) developed a fuzzy logic-based pedotransfer

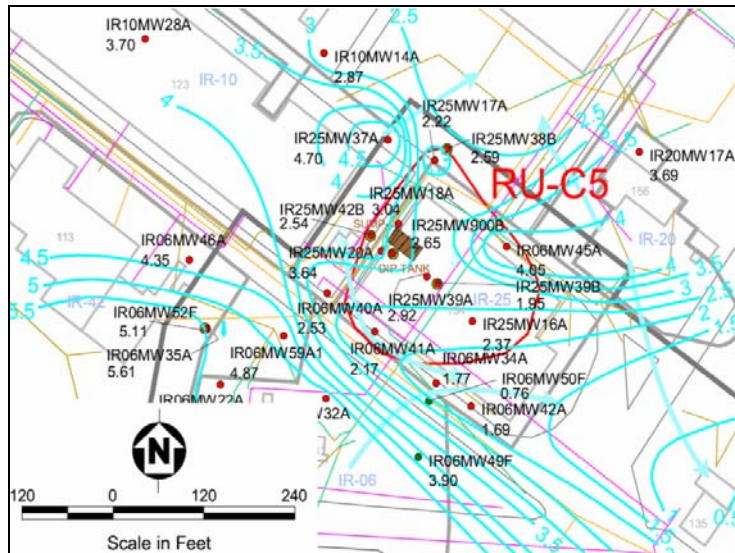
function to relate soil type to hydraulic conductivity. A similar model was developed herein in order to predict hydraulic conductivity along the 100 vertical profiles where the borehole logs were previously qualified. The resulting possibilistic hydraulic conductivity predictions were then spatially estimated using fuzzy kriging (Bardossy et al, 1990a,b) and updated with available physical measurements of hydraulic conductivity (from slug tests) using possibilistic Kalman filtering (Ross et al, 2006, 2008). The resulting hydraulic conductivity field comprises information from both soil analyses and hydraulic conductivity measurements, making it more informed than traditionally derived hydraulic conductivity fields.

### 5.2.3. Groundwater flow and transport model

A groundwater flow and transport model was developed for RU-C5 at Hunters Point Shipyard so that the DNAPL source search algorithm could be tested on the site. The model is comprised of 6 layers and a 1054 node per layer mesh (Figure 49). The flow and transport equations were solved by PTC. The model boundary was drawn to maximize the number of constant head boundary conditions (Figure 49). Where the boundary did not overlap with head contours on the potentiometric map (Figure 50), the boundary was drawn perpendicular to adjacent equi-potential lines and specified to be “no flow.” In this field test, because of the manner in which the hydraulic conductivities are specified, mathematical layers do not correspond to distinct hydrogeologic units. Rather the number of mathematical layers was selected to maximize computational accuracy and reduce computation time.



**Figure 48. The flow and transport model of Hunters Point Shipyard was comprised of 6 mathematical layers and 1054 nodes; boundary conditions were specified to be either constant head or no flow.**



**Figure 49. A potentiometric map drawn from 2002 measurements reveals unique head contours (blue lines) and suggested groundwater flow directions (blue arrows).**

The model was calibrated to match head levels contoured by hand on a 2002 potentiometric map (Figure 50). Though the contours on this map differ from those plotted in later maps, the site was fairly hydrologically inactive before 2002 (Kilduff, 2008). In order to calibrate the model, certain phenomena had to be inferred. The potential high surrounding well IR25MW37A was assumed to be the result of infiltration along rail lines and through clean fill on the surface in the area surrounding this well (Hall, 2008). The hydraulic potential map identified a downward gradient in the center of the model that model calibration revealed to be responsible for drawing head contours toward Building 136.

The calibrated flow field simulated by the stochastic model is shown in Figure 51. The colored contours represent calibrated Monte Carlo simulated hydraulic head results and the black contours represent the hydrogeologist's interpretation of the well water level measurements. This is considered to be a very good calibration. Further, it corroborated the quality of the estimated hydraulic conductivity field.

#### 5.2.4. Source search algorithm application

The water quality samples used in the DNAPL source search algorithm were all selected based upon both the date on which the sample was taken and the magnitudes of the concentration of DNAPL of interest. Water quality samples from February 2001 were considered because, though water quality had been measured around Building 134 as early as June 1994, the greatest number of simultaneous water quality measurements were made in February 2001. This date is also significant, as remedial activities commenced later in the month. The DNAPL of concern was TCE and 10 wells measured TCE in February 2001 (Table 5).

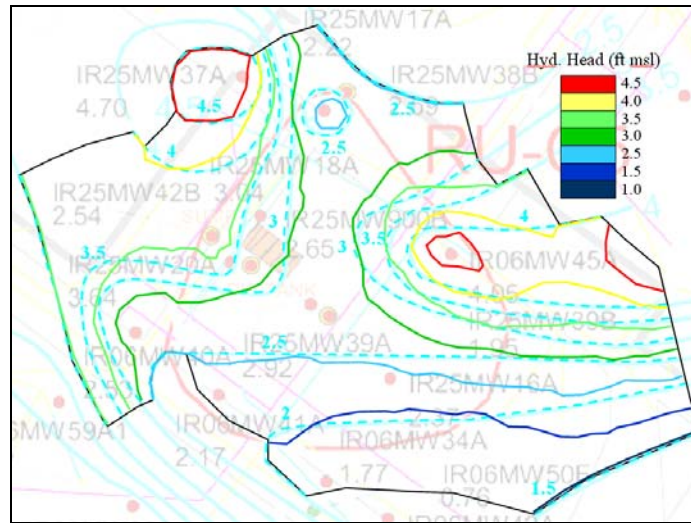


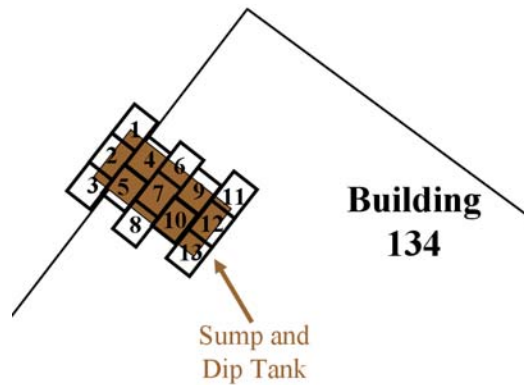
Figure 50. Calibrated model hydraulic heads correspond to measurement-based head contours very well.

Table 5. Available water quality measurements and their locations in the vicinity of Building 134; greyed out wells provided infeasible solutions and were eliminated from consideration.

Well	TCE ( $\mu\text{g/L}$ )	Date Measured	Measurement Layer
IR25MW900B	1100	02/01/01	3, 4
IR25MW19A	1600	02/05/01	3, 4
IR25MW903B	350	02/01/01	3, 4
IR25MW18A	890	02/01/01	6
IR25MW15A1	5000	02/01/01	4, 5, 6
IR25MW15A2	1100	02/01/01	2, 3, 4, 5
IR25MW905B	4.6	02/01/01	3, 4
IR25MW902B	1600	02/01/01	3, 4
IR25MW16A	6	02/26/01	5, 6
IR25MW901B	1200	02/01/01	3, 4, 5

A preliminary execution of the DNAPL source search algorithm assumed 13 potential source locations in the vicinity of the sump and dip tank (Figure 52) in the top-most mathematical layer. This execution identified the northernmost potential source as the true source, yet failed to estimate a reasonable source magnitude. As a result, a more focused source search was attempted that focused upon the northwest area of the sump and dip tank (locations 1, 2 and 3).

Because of the prior information regarding the approximate location of the Building 134 source and because a sensitivity analysis performed on the source search algorithm revealed little sensitivity to initial weights (Ch. 4), the choquet integral was not employed to calculate initial source weights for the three potential sources. Each source was initially assumed to be equally possible.



**Figure 51. Originally, 13 small areas around the sump and dip tank were considered as possible locations for the true TCE source.**

In this field test, it is possible for the source to exist in any of the geological units, and likewise it is possible to specify the source in any one of the model’s mathematical layers. However, the shallow depths of the sump and dip tank, in conjunction with available water quality samples, suggest that the true source is quite shallow and likely resides in the topmost mathematical layer. As such we conservatively specified the three potential sources to be located in the topmost mathematical layer. Considering the geographical positions of these three potential sources, they are located in a low permeability zone comprised of sandy clay.

The use of all 10 water quality measurements proved impractical during executions of the DNAPL source locator algorithm. Measurements from wells IR25MW19A, IR25MW18A, IR25MW15A1, IR25MW15A2 and IR25MW905B produced infeasible solutions to the optimization formulation and were removed from the data set. Measurements from wells IR25MW900B (layers 3,4), IR25MW903B (layers 3,4), IR25MW902B (layers 3, 4), IR25MW16A (layers 5, 6), IR25MW901B (layers 3, 4, 5) were ultimately used to help locate the TCE source, resulting in 11 data points. Such infeasibilities can actually be used as red flags. This is explained in Chapter 6.

#### 5.2.5. Test results

Execution of the DNAPL source locator resulted in the selection of source location 1 as the true source (Figure 52). The source was identified immediately and with certainty after a single data point (well IR25MW902B, layer 4). In order to ensure that further water quality measurements would not vary the selected source, the algorithm was run for the remaining water quality measurements (Figure 53 through Figure 55), listed in Table 6. After the sixth sample was taken, the size, shape and orientation of the TCE plume remained relatively unchanged.

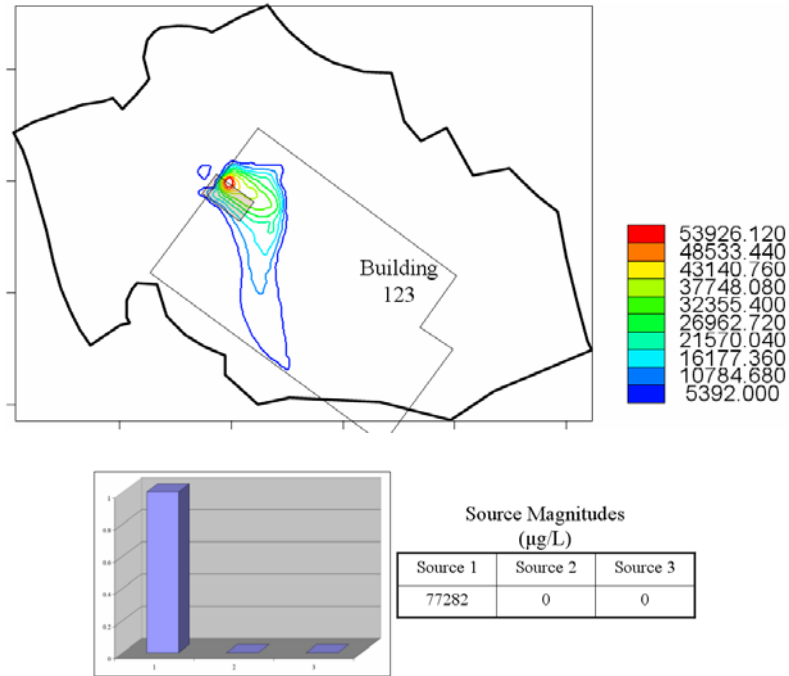


Figure 52. Search algorithm results – after taking one sample; concentration in  $\mu\text{g/L}$

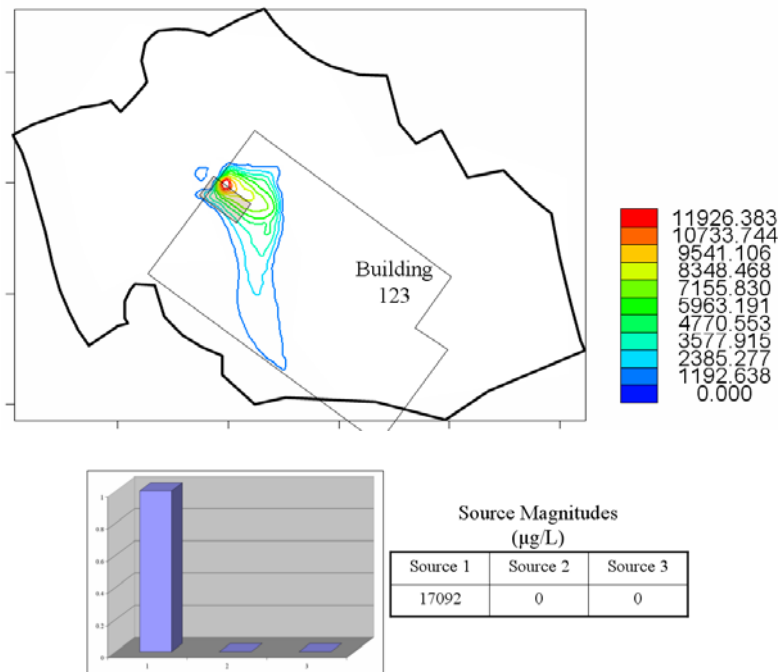


Figure 53. Search algorithm results –after taking two samples (same results after taking 3 through 5 samples) ; concentration in  $\mu\text{g/L}$

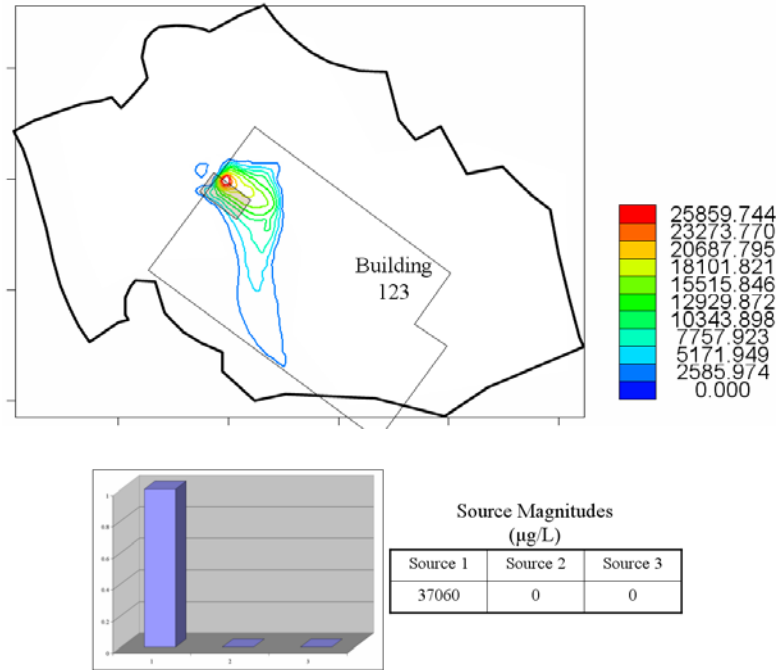


Figure 54. Search algorithm results – after taking six samples (remains unchanged for samples 7 through 10) ; concentration in µg/L

Table 6. The order in which water quality data were selected reveals a proclivity of the source finder to select water quality samples nearer to potential sources.

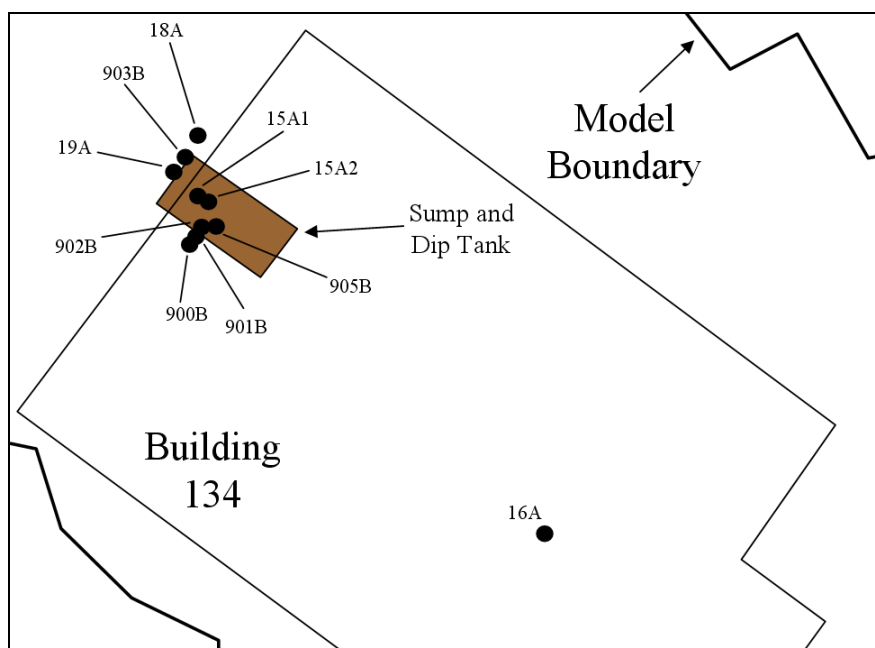
Order Selected	Well	Layer	TCE (µg/L)
1	902B	4	1100
2	903B	4	350
3	902B	3	1100
4	903B	3	350
5	901B	5	1200
6	900B	4	1100
7	901B	4	1200
8	900B	3	1100
9	901B	3	1200
10	16A	5	6
11	16A	6	6

Though the true source did not vary throughout the execution of the DNAPL source locator, the magnitude of the source did. After the fifth water quality sample, the calculated magnitude of the source increased from 17092 µg/L to 37060 µg/L. It can be concluded from this that the water quality measurement at well IR25MW901B had greater influence than the other wells' measurements upon the identification of the true



source and its magnitude. This is interesting to note since the measurements at wells IR25MW15A1, IR25MW902B and IR25MW19A were all of greater magnitude and similar proximity to the identified source. Nevertheless, except for well IR25MW15A1, the water quality at these wells is tested at greater depths than at well IR25MW901B, as the latter is measured in the same mathematical layer as the identified source.

The shape and size of the generated plume are inconsistent with the approximate plume illustrated in site's feasibility study (Sultech, 2008) and the volatile organic carbon measurements plotted in the technical memo (CE2, 2006) regarding RU-C5 where the interpolated plume (calculated from passive soil gas measurements in 2006) extends into the remedial unit that neighbors RU-C5 to the north. However, water quality measurements made in the same year reveal only TCE concentrations north of the sump and dip tank that fall below the practical quantification limit. In addition, the southward-trending path of the plume is entirely consistent with the directions of groundwater flow illustrated on the potentiometric map as well as with the locations of TCE detections (Figure 56).



**Figure 55. Measurements of TCE in groundwater are predominantly located below and around the sump and dip tank.**

The conclusion regarding the true source is further corroborated by that fact that in the area of the sump and dip tank, the potentiometric map reveals a hydraulic high. As such, minor changes in position of the suspected source result in significant changes in the shape and direction of the emanating plume. Were the source slightly to the north of location 1, associated contaminants would be transported to the north/northwest of the sump and dip tank by groundwater flow. A plume emanating from a spot just south of location 1 would flow more directly toward the south, rather than the southeast direction taken by the plume in Figure 55.

## 6. Conclusions

### 6.1. Summary

In summary, an optimal DNAPL source search algorithm was developed, tested and evaluated. The purpose of the proposed algorithm is to help groundwater professionals in their attempt to identify, at least cost, the location, magnitude and geometry of a DNAPL source.

First, the various tools employed in the search algorithm are presented and the modifications needed for incorporation into the proposed methodology discussed. The search strategy is then tested using synthetic example problems of increasing complexity. The algorithm is successful in locating the true source location in all cases. A sensitivity analysis of the input parameters of the algorithm was performed to investigate its robustness. The final and most challenging test of the algorithm was its application to field sites.

### 6.2. Conclusions

The proposed search algorithm performed very well when tested using a synthetic example that represents a situation that can be encountered in the field. The results of a sensitivity analysis provide useful insight into various aspects of the search algorithm. The most important parameters that affect convergence include the type of  $\alpha$ -cuts used at the plume comparison. The results are not very sensitive to the initial source weights, although in a different problem, they might be able to speed the algorithm's convergence. Model parameters such as the number of Monte Carlo simulations did not affect the results in the one-dimensional example, as long as the hydraulic conductivity field is accurately represented.

When the algorithm was field tested at ANAD it was successful in choosing the best possible locations among the potential source location alternatives although the actual true source location has a larger aerial extent. The search algorithm predicted that the source is located in the upper layer of the aquifer. The site experts estimate that most of the DNAPL is indeed located in the upper layer (residuum). At HPS, the algorithm identified a most likely source. Because the true source location is unknown, model results cannot be properly corroborated. Nevertheless, all other source locations around the sump and dip tank were essentially dismissed by the search algorithm, accentuating the certainty with which source location 1 was identified as the true source.

Two major challenges face the field testing of the algorithm. First, the accurate identification of a source location is predicated upon the construction of an accurate and calibrated groundwater flow model. Though this task was successfully completed in the field tests presented within this report, a dearth of hydrogeologic data from a site investigation can preclude a modeler from properly calibrating a flow model. A second challenge is the discovery of water quality measurements that lead to infeasible optimization results. This occurred in both field tests presented herein. Though this is a challenge, where information regarding the DNAPL source is lacking, such a phenomenon can actually assist a modeler in revising their assumptions regarding the timing and magnitude of the source in the flow and transport model. For instance, the

water quality measurements that produced infeasible solutions in the HPS field test may in fact indicate that assumptions regarding the source timing may need to be refined.

### **6.3. Contributions to the field**

The main contribution of this work is the development of a new enabling technology for the identification of a DNAPL contaminant source by combining water quality information (hard data) with expert knowledge (soft data).

An advantage of the search algorithm is that it combines expert knowledge and computer simulations into an integrated optimal predictor of the DNAPL source location. The Choquet integral is the tool that enables the expert opinion regarding the source locations to be quantified and used as input in the search algorithm. Expert insight is also utilized in the process of water quality samples selection. A new method for comparing contaminant concentration plumes is introduced in this work. This technique is rooted in fuzzy set theory. The benefit of using this technique lies in the fact that it not only calculates the intersection of the plumes that are being compared but it also weights more the higher concentration zone common in the plumes.

Another advantage of the proposed algorithm is that water quality information can be treated in two ways: firstly, the algorithm can utilize existing water quality information and provide the best estimate of the true source location given this existing information. But, secondly, it also has the capability to identify where the best sampling location is if the groundwater professionals decide to take new water quality samples. The information of each newly selected sample can be used in real time to update the model and to select the next optimal sampling location until the algorithm converges to a solution.

The search strategy described in this dissertation seeks to identify the location, magnitude and depth of the DNAPL source zone. In contrast to the majority of previous works, the model presented here is a three-dimensional, stochastic model that was successfully applied not only in a variety of synthetic examples but in two real world problems.

### **6.4. Future work**

Future improvements regarding the general formulation of the search algorithm could involve a non-linear optimization problem formulation that will solve for source strength and weights simultaneously. Another suggestion is to include the Kalman filter equations as constraints in the source strength optimization problem. When the flow and transport equations are solved conservation of mass is ensured. The process of updating the concentration values with measurement data using the Kalman filter does not provide any mass balance calculation. Thus, by incorporating the Kalman filter equations into a source strength optimization technique appropriate constraints can be imposed that ensure mass conservation.

## References

Alabert, F., “The practice of fast conditional simulations through the LU decomposition of the covariance matrix”, *Mathematical Geology*, 19(5), 369-386, 1987.

Alapati, S., and Z. J. Kabala, “Recovering the release history of a groundwater contaminant using a non-linear least-squares method”, *Hydrological Processes*, 14(6), 1003-1016, 2000.

Andricevic, R., “Coupled withdrawal and sampling designs for groundwater supply models”, *Water Resources Research*, 29(1), 5–16, 1993.

Aral, M. M., J. B. Guan, and M. L. Maslia, “Identification of contaminant source location and release history in aquifers”, *Journal of Hydrologic Engineering*, 6(3), 225-234, 2001.

Atmadja, J., and A. C. Bagtzoglou, “Pollution source identification in heterogeneous porous media”, *Water Resources Research*, (37), 2113-2125, 2001.

Atmadja, J., and A. C. Bagtzoglou, “State of the Art Report on Mathematical Methods for Groundwater Pollution Source Identification”, *Environmental Forensics*, (2), 205-214, doi:10.1006/enfo.2001.0055, 2001.

Babu D. K., G. F. Pinder, A. Niemi, D. P. Ahlfeld, S. A. Stothoff, “Chemical transport by three-dimensional flows“, Manual, 1997.

Bagtzoglou, A. C., A. F. B. Tompson, and D. E. Dougherty, “Probabilistic simulation for reliable solute source identification in heterogeneous porous media”, *Water Resources Engineering Risk Assessment*, NATO ASI Series, (G29), 189-201, 1991.

Bagtzoglou, A. C., A. F. B. Tompson, and D. E. Dougherty, "Application of particle methods to reliable identification of groundwater pollution sources". *Water Resources Management*, (6), 15-23, 1992.

Bardossy, A., I. Bogardi, and W.E. Kelly, "Kriging with imprecise (fuzzy) variograms I: Theory", *Mathematical Geology*, 22(1), 63-79, 1990.

Bardossy, A., I. Bogardi, and W.E. Kelly, "Kriging with imprecise (fuzzy) variograms II: Application", *Mathematical Geology*, 22(1), 81-94, 1990.

Baun, S. A., and A. C. Bagtzoglou, "A computationally attractive approach for near real-time contamination source identification", *International Conference, Computational Methods in Water Resources*, Chapel Hill, North Carolina, 2004.

Bras R. L., "Sampling network design in hydrology and water quality sampling: A review of linear estimation theory applications", *Applications of Kalman Filter to Hydrology, Hydraulics and Water resources*, Chiu, C.L. (ed.) 155-200, AGU Chapman conference, Pittsburgh, USA, 1978.

Casella, G., and R. L. Berger, "Statistical inference", Wadsworth Group, 2002.

CE2 Corporation, "Technical memorandum for contamination delineation at remedial unit C5", 2006.

Datta, B., J. E. Beegle, M. L. Kavvas, and G. T. Orlob, "Development of an expert system embedding pattern recognition techniques for pollution source identification", *National Technical Information Service*, Springfield, Virginia, U.S.A.

Davis, M. W., "Production of conditional simulations via the LU triangular decomposition of the covariance matrix", *Mathematical Geology*, 19(2), 91-98, 1987.

Domenico P. A., and F. W. Schwartz, “Physical and chemical hydrogeology”, John Wiley and Sons, 1990.

Dokou, Z., “Optimal Search Strategy for the Definition of a DNAPL Source”, PhD dissertation, Civil and Environmental Engineering, University of Vermont, 2008.

Drecourt, J-P., “Data assimilation in hydrogeological modeling”, PhD dissertation, DHI Water and Environment, Denmark, 2004.

Dubois D., H. Prade, “On the use of aggregation operations in information fusion processes”, *Fuzzy Sets and Systems*, 142, 143-161, 2004.

Dubois D., H. Prade, and R. R. Yager, “Fuzzy information engineering. A guided tour of applications”, John Wiley and Sons, New York, 1996.

Eppstein, M. J. and D. E. Dougherty, “Simultaneous estimation of transmissivity values and zonation”, *Water Resources Research*, (32)11, 3321-3336, 1996.

Ferraresi, M., E. Todini, and R. Vognoli, “A solution to the inverse problem in groundwater hydrology based on Kalman filtering”, *Journal of Hydrology*, (175), 567-581, 1996.

van Geer F. C., “Application of Kalman filtering in the analysis and design of groundwater monitoring networks”, *TNO Institute of Applied Geoscience*, report PN 87-05, 1987.

Gooaverts, P., “Applied geostatistical series: Geostatistics for nature resources evaluation”, Oxford University Press, 1997.

Graham, W., and D. B. McLaughlin, “Stochastic analysis of nonstationary subsurface transport, 2, Conditional moments”, *Water Resources Research*, 25(11), 2331–2355, 1989.

Graham, W. D., and C. D. Tankersley, “Forecasting piezometric head levels in the Floridian aquifer: A Kalman filtering approach”, *Water Resources Research*, 29(11), 3791–3800, 1993.

Gutjahr, A. L., L. W. Gelhar, A. A. Bakr, and J. R. MacMillan, “Stochastic analysis of spatial variability in subsurface flows, 2, Evaluation and application”, *Water Resources Research*, 14(5), 953–959, 1978.

Gorelick, S. M., B. Evans, and I. Remson, “Identifying source of groundwater pollution: An optimization approach”, *Water Resources Research*, 19(3), 779-790, 1983.

Grabisch, M., “A graphical interpretation of the Choquet integral”, *European Journal of Operational Research*, (89), 445-456, 1996.

Hayden, N. J., X. Wei, Z. Li, J. Doris, G. F. Pinder, D. M. Rizzo, and A. J. Rossman, “Groundwater flow and transport studies in a large-scale physical model with media layering”, *Internal Report, UVM Department of Civil and Environmental Engineering*, 2007.

Helton, J. C., and F. J. Davis, “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems”, *Reliability Engineering and System Safety*, Elsevier, 2003.

Herrera, G., “Cost effective groundwater quality sampling network design”, PhD dissertation, Civil and Environmental Engineering, University of Vermont, 1998.

Iman R. L., J. E. Campbell, J. C. Helton, “An approach to sensitivity analysis of computer models: Part I – Introduction, input, variable selection and preliminary variable assessment”, *Journal of Quality Technology*, 13(3), 174-183, 1981a.

Iman R. L., J. E. Campbell, J. C. Helton, “An Approach to Sensitivity Analysis of Computer Models: Part II - Ranking of Input Variables, Response Surface Validation, Distribution Effect and Technique Synopsis”, *Journal of Quality Technology*, 13(4), 1981b.

Iman, R. L., and W. J. Conover, “A distribution free approach to inducing rank correlation among input variables”, *Communications in Statistics*, 11(3), 311-344, 1982.

Isaaks, E. H., and R. M. Srivastava, “An introduction to applied geostatistics”, Oxford University Press, 1989.

Jazwinski A. H., “Stochastic processes and filtering theory”, Academic Press, San Diego, California, 1970.

Journel, A. B., and C. J. Huijbregts, “Mining geostatistics”, Academic Press, New York, 1978.

Kalman R. E., “A new approach to linear filtering and prediction problems”, *Journal of Basic Engineering*, (82), 35-45, 1960.

Kilduf, E., *personal communication*, June, 2009.

Klir G. J., Z. Wang, and D. Harmanec, “Constructing fuzzy measures in expert systems”, *Fuzzy Sets and Systems*, 92, 251-264, 1997.

Klir G. J., and B. Yuan, “Fuzzy sets and Fuzzy logic: Theory and applications”, Prentice Hall, 1995.



Kunstmann, H., W. Kinzelbach and T. Siegfried, “Conditional first-order second-moment method and its application to the quantification of uncertainty in groundwater modeling”, *Water Resources Research*, 38(4), doi: 10.1029/2000WR000022, 2002.

Li, Z., “Applications using the ordinary and extended Kalman filter to characterize groundwater contaminant sources”, PhD dissertation, Civil and Environmental Engineering, University of Vermont, 2007.

Li, Z., D. Rizzo, and N. Hayden, “Utilizing artificial neural networks to backtrack source location”, *Summit on environmental modeling and software, 3rd biennial meeting of the international modeling and software society*, Burlington, Vermont, 2006.

Liu, C., and W. P. Ball, “Application of inverse methods to contaminant source identification from aquitard diffusion profiles”, *Water Resources Research*, 35(7), 1975-1985, 1999.

Mahar, P. S. and B. Datta, “Optimal monitoring network and ground-water-pollution source identification”, *Journal of Water Resources Planning and Management – ASCE*, 123(4), 199-207, 1997.

Mahar, P. S. and B. Datta, “Identification of pollution sources in transient groundwater systems”, *Water Resources Management*, 14(3), 209-227, 2000.

Mahinthakumar, G. K. and M. Sayeed, “Hybrid genetic algorithm - Local search methods for solving groundwater source identification inverse problems”, *Journal of Water Resources Planning and Management-ASCE*, 131(1), 45-57, 2005.

Marichal, J., “An Axiomatic Approach of the Discrete Choquet Integral as a Tool to Aggregate Interacting Criteria”, *IEE Transactions on Fuzzy Systems*, (8)6, 800-807, 2000.

Matheron, G., “The intrinsic random functions and their applications”, *Advances in Applied Probability*, (5), 493-468, 1973.

McKay, M. D., W. J. Conover, and R. J. Beckman, “A comparison of three methods in for selecting values of input variables in the analysis of output from a computer code”, *Technometrics*, 21(2), 239-245, 1979.

McLaughlin, D. B., “Application of Kalman filtering to groundwater basin modeling and prediction”, *Real-Time Forecasting/Control of Water Resource Systems*, edited by E. F. Wood, Pergamon, New York, 109–123, 1976.

McWilliams, T. P., “Sensitivity analysis of geologic computer models: a formal procedure based on Latin hypercube sampling”, *Mathematical Geology*, 19(2): 81-90, 1987.

Mejia, J. and I. Rodriguez-Iturbe, “On the synthesis of random fields from the spectrum: an application to the generation of hydrologic spatial process”, *Water Resources Research*, (30)4, 705-711, 1974.

Mirghani, B., M. Tryby, R. Ranjithan, and K. Mahinthakumar, “Grid-enabled simulation-optimization approach for solving groundwater characterization problems”, *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montreal, 2006.

Mohinder S. G., and A. P. Andrews, “Kalman filtering: Theory and practice using MATLAB”, John Wiley and Sons, 2001.

Morrison, R. D., “Application of forensic techniques for age dating and source identification in environmental litigation”, *Environmental Forensics*, (1), 131-153, 2000.

Neupauer, R.M., B. Borchers, and J. L. Wilson, “Comparison of inverse methods for reconstructing the release history of a groundwater contamination source”, *Water Resources Research*, 36(9), 2469-2475, 2000.

Neupauer, R. M., and R. H. Lin, “Identifying sources of a conservative groundwater contaminant using backward probabilities conditioned on measured concentrations”, *Water Resources Research*, 42, W03424, doi: 10.1029/2005WR004115, 2006.

Neupauer, R. M., and J. L. Wilson, “Adjoint method for obtaining backward –in-time location and travel time probabilities of a conservative groundwater contamination”, *Water Resources Research*, 35(11), 3389-3398, 1999.

Neupauer, R. M., and J. L. Wilson, “Adjoint-derived location and travel time probabilities in a multi-dimensional groundwater system”, *Water Resources Research*, 37(5), 1657-1668, 2001.

Neupauer, R. M., and J. L. Wilson, “Backward probability model using multiple observations of contamination to identify groundwater contamination sources at the Massachusetts Military Reservation”, *Water Resources Research*, 41, W02015, doi: 10.1029/2003WR002974, 2005.

Rizzo, D. M, D. E. Dougherty, and M. Yu, “An adaptive long-term monitoring and operations system (aLTMOs) for optimization in environmental management”, *Joint Conference on Water Resource Engineering and Water Resources Planning and Management*, Minneapolis, Minnesota, USA, 2000.

Robin, M.J., A. L. Gutjahr, E. A. Sudicky, and J. L. Wilson, “Cross-correlated random field generation with the direct Fourier transform method”, *Water resources Research*, (29)7, 2385-2397, 1993.

Ross, J., M. Ozbek, and G. F. Pinder, "Fuzzy kalman filtering of hydraulic conductivity", *Computational Methods in Water Resources 16<sup>th</sup> International Conference*, 2006.

Ross, J., M. Ozbek, and G. F. Pinder, "Fuzzy inference of hydraulic conductivity from soil grain data and field observations", *Mathematical Geology* 39(8), 765-780, 2007.

Ross, J., M. Ozbek, and G. F. Pinder, "Kalman filtering of possibilistic hydraulic conductivity", *Journal of Hydrology* 354(1-4), 149-159, 2008.

Ross, J., M. M. Ozbek, and G. F. Pinder, "Groundwater flow and transport modeling with correlated possibilistic data", *Advances in Water Resources*, submitted.

SAIC (Science Applications International Corporation), "Anniston Army Depot southeast industrial area comprehensive groundwater remedial investigation, Phase III at Anniston Army Depot, Anniston, Alabama", Draft, 2005.

SAIC (Science Applications International Corporation), "Anniston Army Depot southeast industrial area comprehensive groundwater feasibility study for operable unit 1 at Anniston Army Depot, Anniston, Alabama", Draft, 2006.

Singh, R. M., B. Datta, and A. Jain, "Identification of unknown groundwater pollution sources using artificial neural networks", *Journal of Water Resources Planning and Management-ASCE*, 130(6), 506-514, 2004.

Skaggs, T. H., and Z. J. Kabala, "Recovering the release history of a groundwater contaminant", *Water Resources Research*, 30(1), 71-79, 1994.

Skaggs, T. H., and Z. J. Kabala, "Recovering the history of a groundwater contaminant plume: Method of quasi-reversibility", *Water Resources Research*, 31(11), 2669-2673, 1995.

Skaggs, T. H., and Z. J. Kabala, "Limitations in recovering the history of a groundwater contaminant plume", *Journal of Contaminant Hydrology*, 33, 347-359, 1998.

Snodgrass, M. F. and P. K. Kitanidis, "A geostatistical approach to contaminant source identification", *Water Resources Research*, 33(4), 537-546, 1997.

Sugeno M., "Theory of fuzzy integrals and its applications", PhD thesis, Tokyo Institute of Technology, 1974.

SulTech (Sullivan Consulting Group and Tetra Tech), "Feasibility study report for Parcel C", Final Report, 2008.

Tetra Tech, "Parcel C groundwater summary report: Phase III groundwater data gaps investigation", Revised Final Report, 2004.

Wagner, B. J., "Simultaneous Parameter - Estimation and Contaminant Source Characterization for Coupled Groundwater - Flow and Contaminant Transport Modeling", *Journal of Hydrology*, 135(1-4), 275-303, 1992.

Wilson, J. L. and J. Liu, "Backward tracking to find the source of pollution", *Waste Management Risk Remediation*, (1), 181-199, 1994.

Woodbury, A. D., and T. J. Ulrych, "Minimum relative entropy inversion: Theory and application to recovering the release history of a groundwater contaminant", *Water Resources Research*, 32(9), 2671-2681, 1996.

Wu J., and C. Zheng, "Contaminant monitoring network design: recent advances and future directions", *Advances in Earth Science*, 19, 429-436, 2004.

Yu, Y.-S., M. Heidari, and Wang Guang-Te, "Optimal estimation of contaminant transport in ground water", *Water Resources Bulletin*, 25(2), 295-300, 1989.

Zadeh, L., “Fuzzy sets as a basis for a theory of possibility”, *Fuzzy Sets and Systems*, 1(1), 3-28, 1978.

Zhang, Y., “Optimal design of groundwater-quality monitoring networks”, PhD dissertation, Civil and Environmental Engineering, University of Vermont, 2002.

Zhang, Y., and G. F. Pinder, “Latin hypercube lattice sample selection strategy for correlated random hydraulic conductivity fields, *Water Resources Research*, (39)8, 1226, doi:10.1029/2002WR001822, 2003.

Zhou Y, C. B. M Te Stroet, F. C., Van Geer, “Using Kalman filtering to improve and quantify the uncertainty of numerical groundwater simulations, 2, Application to monitoring network design”, *Water Resources Research*, 27(8), 1995-2006, 1991.

Zou, S., and A. Parr, “Optimal estimation of two-dimensional contaminant transport”, *Ground Water*, 33(2), 319-325, 1995.

# **Appendix A**

## List of Publications

Ross, J., M. M. Ozbek, and G. F. Pinder. “Groundwater flow and transport modeling with correlated possibilistic data”, *Advances in Water Resources*, submitted.

Ross, J., M. Ozbek, and G. F. Pinder. “Aleatoric and epistemic uncertainty in groundwater flow and transport simulation”, *Water Resources Research*, 45, W00B15, doi: 10.1029/2007WR006799, 2009.

Dokou, Z. and G. F. Pinder. “Optimal search strategy for the definition of a DNAPL source”, *Journal of Hydrolog* , doi:10.1016/j.jhydrol.2009.07.062, 2009, *in press*.



## **Appendix B**

# **DNAPL Source Finder Code Documentation**

**Zoe Dokou, James Ross, George Pinder**

University of Vermont  
College of Engineering and Mathematical Sciences  
August 2009

Prepared for SERDP  
(Strategic Environmental Research and Development Program)

## **Preface**

The computer program documented in what follows was created as part of the research project entitled Optimal Search Strategy for the Definition of a DNAPL Source funded by SERDP. The documentation should be used in concert with the report provided for that project. The concepts and code embodied in this computer program were developed over approximately two decades at the University of Vermont. Individuals who contributed to the development of elements of the program presented herein, prior to the authors of this document, are William A. McGrath, Graciela Herrera de Olivares, Yingqi Zhang, Xinyu Wei and we wish to acknowledge their contributions.

## **Scope of Document**

The purpose of this document is to describe in detail each component to the DNAPL source locator algorithm (Dokou, 2008). In addition, an example problem is provided to illustrate the practical use of the source search algorithm.

## **Technical objective**

To develop, test and evaluate a computer assisted analysis algorithm to help the groundwater professional identify, at least cost, the location and geometry of a DNAPL source.

## **Technical approach**

The DNAPL source location is generally too small and filamentous to identify via borings or geophysical methods.

The plume emanating from a DNAPL source is typically quite large and easily discovered, although identification of its extent may require the collection of considerable data.

The strategy for defining the DNAPL source presented here, exploits these facts.

## **Assumptions**

- 1 . A groundwater plume has been identified and a preliminary field investigation has been conducted.
2. There is reason to believe that the plume is generated by the suspected DNAPL source.
3. Enough hydrological information on the site exists to construct a groundwater flow and transport model, assuming the hydraulic conductivity is known with uncertainty.
4. The primary source of uncertainty in the transport equation is the velocity due to the uncertainty and heterogeneity in the hydraulic conductivity, that is the porosity, dispersivity, retardation and chemical reaction are assumed to be deterministic.

## **Explanation of Document**

The following sections clarify how different parts of the DNAPL source locator algorithm work. Contained in each section is a brief explanation of what each part of the code is intended to do, a list of the fortran and data files necessary for each part of the code and examples of the various input files to each part of the code. The numbers in the examples of input data files correspond to the example presented in Section 10.

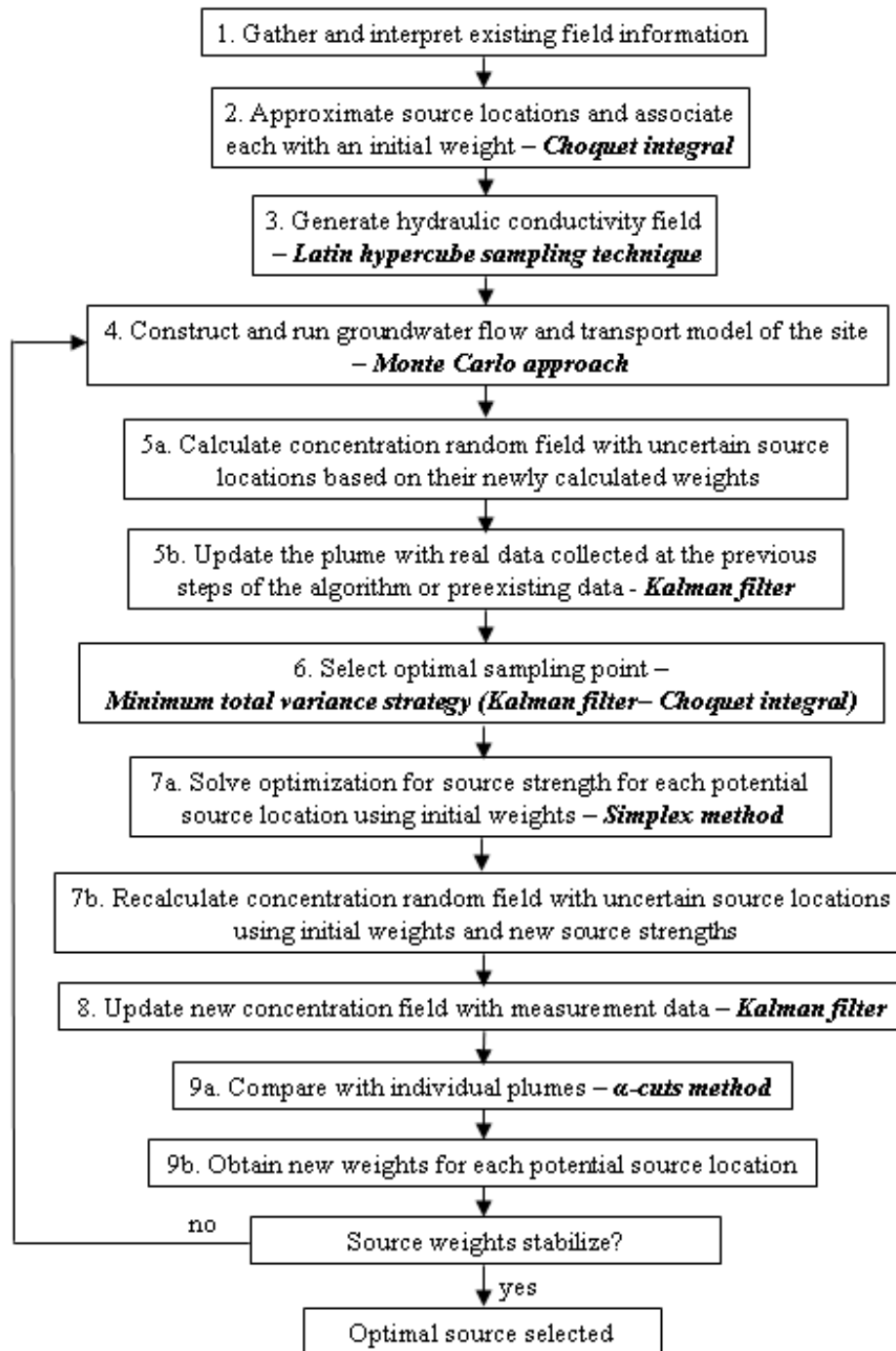


Figure 1: Flow diagram of the search algorithm

## Table of Contents

1. Lhs Code Overview .....	104
1.1 Main file: main_lhs.for .....	104
1.2 Subroutines .....	105
1.3 Modules .....	106
1.4 IMSL subroutines .....	106
1.5 Input .....	106
1.6 Include files.....	109
1.7 Output .....	109
1.8 Example input files .....	109
2. Individual Source Simulations Code Overview.....	111
2.1 Main file: simmain.for .....	111
2.2 Subroutines .....	111
2.3 IMSL subroutines .....	112
2.4 Input .....	112
2.5 Include files.....	113
2.6 Output .....	113
2.7 Example input files .....	114
3. Initial Weights Code Overview .....	116
3.1 Main file: weights_choquet_main.for.....	116
3.2 Subroutines .....	116
3.3 IMSL subroutines .....	117
3.4 Input.....	117
3.5 Include files.....	118
3.6 Output .....	118
3.7 Example input files .....	118
4. Simulation Code Overview.....	120
4.1 Main file: simmain.for .....	120
4.2 Subroutines .....	121

4.3 IMSL subroutines .....	121
4.4 Input .....	121
4.5 Include files.....	123
4.6 Output .....	123
4.7 Example input files .....	123
5. Kalman Code Overview.....	124
5.1 Main program: kalman.for .....	124
5.2 Subroutines .....	124
5.3 Input .....	125
5.4 Include files.....	126
5.5 Output .....	126
5.6 Example input files .....	126
6. Choquet code overview.....	128
6.1 Main file: choquet_main.for .....	128
6.2 Subroutines .....	128
6.3 Input .....	129
6.4 Include files.....	130
6.5 Output .....	130
6.6 Examples of input files .....	131
7. Optimization Code Overview .....	132
7.1 Main program: optimization.for .....	132
7.2 Subroutines .....	132
7.3 Input .....	133
7.4 Include files.....	133
7.5 Output .....	134
7.6 Examples of input files .....	134
8. Alpha Cut Code Overview.....	135
8.1 Main program: a_cut_main.for .....	135
8.2 Subroutines .....	136
8.3 Input .....	137

8.4 Include files.....	138
8.5 Output .....	138
8.6 Examples of input files .....	138
9. Run code overview .....	140
9.1 Main program: main_run.for .....	140
9.2 Subroutines .....	141
9.3 Include files.....	141
9.4 Input files .....	141
10. Example Problem.....	142
10.1 Problem statement.....	142
10.2 Choquet integral code and initial source weights .....	143
10.3 Latin hypercube sampling.....	145
10.4 Simulation code .....	146
10.5 Choquet code .....	146
10.6 Kalman filter code .....	147
10.7 Optimization code.....	147
10.8 Alpha cut code .....	147
10.9 Ensuing iterations .....	148
10.10 Algorithm driver execution.....	150



## **List of Tables**

Table 1. PTC model parameters.....	143
Table 2. Distances and corresponding membership degrees .....	145
Table 3. Membership degrees and Choquet integral-calculated global weights for potential source locations.....	145
Table 4: Final weights for the potential source locations .....	148

## List of Figures

Figure 2. Sample problem setup .....	142
Figure 3. Membership functions for (a) “Near to the Manufacturing Facility, “Near to the Waste Dump” and (b) “Near to the Water Table” .....	144
Figure 4. info.dat.....	146
Figure 5. Plume before taking no samples and initial weights for the potential source locations .....	149
Figure 6. Plume after taking 1 sample and resulting weights for the potential source locations .....	149
Figure 7. Plume after taking 2 samples and resulting weights for the potential source locations .....	150

## 1. Lhs Code Overview

Lhs is a stratified sampling technique where the probability density function defining a random variable (i.e. uncertain hydraulic conductivity) is divided into a number of non-overlapping intervals, which have equal probability. Samples are taken from those intervals and are permuted in a way such that the correlation of the field is accurately represented. The Lhs generates the hydraulic conductivity realizations that will be used in the Monte Carlo simulations.

This part of the DNAPL source locator code is run separately from the ‘driver’ described in Section 9. As such, the main file `main_lhs.for` and associated subroutines (described below) are compiled and run with the necessary input data files (`info.dat` and `mesh.dat`). When this is accomplished, output files (`conk.dat`, `kerror.dat` and `sub-err.dat`) are created. Unlike the output files `kerror.dat` and `sub-err.dat`, `conk.dat` is used as input to the individual source simulations (Section 2). There is one `conk.dat` file for every numerical layer in the groundwater flow and transport model. As such, this code must be executed once for each layer. The user must ensure that the name of the output file written in line 237 of `main_lhs.for` (in the case where the number of simulations is greater than the number of nodes in the model’s mesh) or line 83 of the `sub_sample.for` (in the case where the number of simulations is less than the number of nodes in the model’s mesh) is changed to `conk(i).dat`, where `i` is the layer number for which the hydraulic conductivity realizations are being generated.

The information presented below regarding the various fortran and data files are meant to clarify how each is relevant to the appropriate execution of the Lhs code. The example input data files presented at the end of this section are intended for Layer 1 of the model presented in Section 10.

### 1.1 Main file: `main_lhs.for`

The main file program does the following:

- Calls subroutine "readinput" to read the input parameters.
- Calls subroutine `allocate_cor` that allocates some variables
- Selects a variogram model
- Writes the target correlation matrix into a file called `target.dat`, for later use.

- If the number of realizations is less than the number of nodes, this program generates the (nodes+1) realizations first.
- Generates the inverse of the normal distribution.
- Calls subroutine 'permt' that puts the samples of each block into one matrix with a random order.
- Calculates the covariance and correlation matrix of k\_sam.
- Factorizes the correlation matrix, forms the lower triangular matrix.
- Transforms the k matrix to the kstar matrix that has the desired correlation matrix, does the matrix multiplication.
- Re-arranges the elements in the k\_sam matrix, so it has the same rank correlation matrix as matrix kstar. Rank1 is the subroutine that performs the ranking.
- Calculates the mean, variance, covariance and correlation matrix of the re-arranged matrix k.
- Outputs the realizations for the big blocks into file k0.dat
- If the number of nodes is greater or equal to the number of realizations then calls subroutine sub\_sample. This subroutine randomly selects the required sample size from the nodes+1 realizations that have already been generated.
- Outputs the statistics of the samples as well as the deviation of the statistics.
- Renames k0.dat to conk.dat

## 1.2 Subroutines

- **readinput.for:** This subroutine opens the "info.dat" file and reads the input parameters.
- **allocate\_cor.for:** This subroutine allocates space for the correlation variables.
- **allocate\_k.for:** This subroutine allocates space for the hydraulic conductivity variables.
- **allocate\_mean.for:** This subroutine allocates space for the mean variable.
- **allocate\_tem.for:** This subroutine allocates space for some temporal variables.

- **model\_Exponential.for, model\_Gaussian.for, model\_Power.for, model\_Spherical.for, model\_User.for:** These subroutines fit the variogram to one of the 4 models or allow the user to use a different model variogram.
- **permt.for:** This subroutine creates random permutations of the Lhs samples and puts them in a matrix.
- **rank1.for:** This program first determines the rank of a matrix A (sub1.for) and then re-arranges the elements in matrix B according to this rank matrix, so A and B will have the same rank correlation matrix.
- **sub1.for:** Sub1 is the subroutine that calculates the rank of a matrix
- **sub\_sample.for:** This subroutine randomly selects the required sample size of realizations from the nodes+1 realizations that have already been generated.

### 1.3 Modules

- **Dim1\_data.for, Dim2\_data.for, Dim2\_data.for:** These modules declare some allocatable variables.

### 1.4 IMSL subroutines

- **DCORVC:** computes the variance-covariance matrix.
- **DLFTDS:** Computes the transpose(R)\*R Cholesky factorization of a real symmetric positive definite matrix.
- **DLINRT:** Computes the inverse of a real triangular matrix.
- **DMRRRR:** Multiplies two real rectangular matrices, A\*B.
- **DMXYTF:** Multiplies a matrix A by the transpose of a matrix B, A\*transpose(B).

### 1.5 Input

- **info.dat:** Input data are located in the file: info.dat. There are two cases.

#### Case 1

**Imesh:** =1 (an existing mesh will be used).  
 Line #1: Imesh ( Imesh = 1 in this case )

Line #2: Nodes: the number of nodes in the mesh;

Line #3: Nsim

Nsim: number of simulations;

Line #4: Mtype

Mtype: which variogram is used?

= 1, Spherical model is used

= 2, Exponential model is used

= 3, Gaussian model is used

= 4, Power model is used

= 5, User defined model

Line #5: ax ay

ax: correlation length in X direction;

ay: correlation length in Y direction;

Line #6: cvar cnugget

cvar: variance;

cnugget: nugget effect;

Line #7: cmean

cmean: mean.

## Case 2

**Imesh:** =2 (a square mesh or other regularly shaped mesh will be generated).

Line #1: Imesh ( Imesh = 2 in this case )

Line #2: nx ny

nx: number of nodes in x direction;

ny: number of nodes in y direction;

Line #3: dx dy

dx: distance between the adjacent two nodes in x direction;

dy: distance between the adjacent two nodes in y direction;

Nodes=nx\*ny: number of nodes in the mesh;

Line #4: Nsim

Nsim: number of simulations;

Line #5: Mtype

Mtype: which variogram is used?

Mtype = 1, spherical model:

$$\gamma(h) = \omega \left[ \frac{3}{2} \left( \frac{h}{a} \right) - \frac{1}{2} \left( \frac{h}{a} \right)^3 \right]$$

Mtype = 2, exponential model:

$$\gamma(h) = \omega \left[ 1 - \exp \left( - \frac{h}{a} \right) \right]$$

Mtype = 3, Gaussian model:

$$\gamma(h) = \omega \left\{ 1 - \exp \left[ - \left( \frac{h}{a} \right)^2 \right] \right\}$$

Mtype = 4, exponential model:

$$\gamma(h) = \omega h^\lambda$$

Mtype = 5, user defined model, the code has to be written by the user.

Line #6: ax ay

ax: correlation length in X direction;

ay: correlation length in Y direction;

Line #7: cvar cnugget

cvar: variance;

cnugget: nugget effect;

Line #8: cmean

cmean: mean.

- **mesh.dat:** In case1, a mesh already exists. There is no need to generate a new one. So the mesh will be read from the file “mesh.dat”. The number of lines in this file is the number of nodes in the mesh. For each line, it has the format:

j xn(j) yn(j)

where

j: node number;

xn(j): x axis of j node;

yn(j): y axis of j node.

## 1.6 Include files

- **lhs.inc:** This file specifies the parameters that are used in (almost) every subroutine.

## 1.7 Output

- **conk.dat:** This file is copied to the flow and transport simulation directory and used as the input file to flow and transport simulation. It is an unformatted file.
- **kerror.dat, sub-err.dat:** These two output files contain information about the deviation of the statistics of the samples from the real statistics. The numbers are the sum of squares of the deviation: Means, variances, and the covariances. Reference only.

## 1.8 Example input files

- **info.dat:**

```
1
861
100
2
200      200
1  1
1
```



- **mesh.dat:**

1	820.471	123.496
2	804.527	121.116
3	808.848	97.111
4	832.763	106.297
5	834.144	125.537
6	813.117	73.392

... (lines corresponding to nodes 7 through 860 omitted here for brevity)

861	924.432	395.342
-----	---------	---------

## 2. Individual Source Simulations Code Overview

This part of the code performs the Monte Carlo simulations for each individual source location (using the conk.dat files as input) and outputs all the realizations to be used in other parts of the program. In addition, this part of the code creates the file 'sources.dat' that is used in the alpha cuts part of the code (Section 8). Each potential source location is used separately in a Monte Carlo simulation using ptc, and the mean concentration for each source is calculated. The source code is similar to the simulation code (Section 4), but this time only one source location is considered at a time.

Like the Lhs code (Section 1), this portion of the code is executed independently of the driver (Section 9). Files necessary for the driver, however, are output by this part of the code. These files, denoted

### 2.1 Main file: simmain.for

The main file program performs the following tasks:

- Calls subroutine "readinput" to read the input parameters.
- Reads the sampling well locations (samloc.dat) and the locations where a concentration estimate is needed (varloc.dat).
- Reads K mean from ptc files.
- Runs ptc, Nsim (number of realizations) times. Before each ptc run, the subroutine nrm2log is called to convert the normal realizations of hydraulic conductivity to lognormal.
- Calculates the statistics of the set of concentration realizations.
- Calls IMSL subroutine DCORVC (calculates the variance-covariance matrix).
- Outputs mean concentration of in file cmean.
- Outputs the concentration variance layer of interest in ccov.dat.

### 2.2 Subroutines

- **readinput.for:** This subroutine opens the "simu.dat" file and reads the input parameters.
- **nrm2log.for:** This subroutine converts normal realizations of a field to lognormal realizations.

- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse:** This subroutine returns the position of the first non-blank character. It is located in the noutfile.for file.

### 2.3 IMSL subroutines

- **SVRGP:** sorts an array by algebraic values and returns a pointer array.
- **DCORVC:** computes the variance covariance matrix.

### 2.4 Input

The input files for the simulation part of the code are:

- **Simu.dat:**

Line 1: Node	total number of nodes in the numerical mesh;
Line 2: Layer	number of layers in the simulation model;
Line 3: Ntot	total number of simulations;
Line 4: Nloc	number of locations where estimates are needed;
Line 5: Nsam	number of potential sampling locations;
Line 6: sigmax, sigmay, sigmaz	standard deviation of Kx, Ky and Kz
- **Location Files:** There are two files: *samloc.dat* and *varloc.dat*.  
*Samloc.dat:* contains the node information of the potential sampling locations.  
*Varloc.dat:* contains the node information of the locations where estimates are needed.
  - **PTC Input Files:** All the input files to PTC should be prepared and put into this directory. The hydraulic conductivity files now should correspond to the average hydraulic conductivity at each node. The format is the same as before. The files should be renamed as: *condx\_m.dat*, *condy\_m.dat* and *condz\_m.dat*.
    - Run file name is: *PTC\_mesh.run*.

- Change name of ptc file 'bctran.dat' to 'bctran\_init.dat'
- Additional ptc files needed: adsor.dat, bcflow.dat, bcleak.dat, dispX.dat, dispY.dat, dispZ.dat, elev.dat, ini\_c.dat, ini\_h.dat, poros.dat, PTC\_Mesh\_Mesh.inc, rain.dat, stor.dat.
- **Conk(i).dat (i=1, # layers):** This is an unformatted file from the output of Lhs. It contains the realizations of hydraulic conductivity, assuming a zero mean and unit variance of logK. The actual mean and variance of K are input from the general input file: simu.dat. The number of files should correspond to the number of layers, e.g. if we have 3 layers with random K then we will have 3 files, conk1.dat, conk2.dat and conk3.dat.

## 2.5 Include files

- **simpara.inc:** This file contains common variable definitions

## 2.6 Output

- **iteration\_num.dat:** Contains the number of the current iteration.
- **cmean(i).dat (unformatted):** Contains the mean concentration at the nodes where estimates are needed. There is one file for each layer. i=1, # Layers
- **ccov(i).dat (unformatted):** Contains the concentration covariance-variance between the nodes where estimates are needed. There is one file for each layer. i=1, # Layers.
- **fconc.dat:** Intermediate file containing concentration data.
- **src\_orig(i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source i, geologic layer j and numerical layer k.
- **c\_all\_orig(i)\_GL(j)\_L(k).dat:** Contains all the concentration realizations for source i, geologic layer j and numerical layer k.
- **hmean(i).dat (formatted):** Contains the hydraulic head mean for each layer i.

## 2.7 Example input files

Snippets of input files not automatically created by ptc or by another part of the DNAPL source locator code are provided below.

- **simu.dat:**

```
861
3
100
861
30
1 1 1
```

- **samloc.dat:**

```
513
34
227
718
... (not all 30 sampling location node numbers are listed here)
124
```

- **varloc.dat:**

```
1
2
3
4
... (nodes 5 through 860 are omitted here for brevity)
861
```

- **condx.dat, condy.dat, condz.dat:**

```
1 4.81310350994244
2 5.48903759167282
3 3.0263477166513
4 6.42986495982318
5 4.78709539411632
6 4.27818709620933
... (nodes 7 through 860 are omitted here for brevity)
7 5.18462964936329
```

- **bctran.dat, bctran.init** (shown for source 1):

```
0/  
0/  
0/  
0/  
0/  
153 1          1.000000e+00  * Layer 1 Stress 1  
0/
```

### 3. Initial Weights Code Overview

The target locations of the possible sources are identified and given initial weights using information fusion. In this approach each possible source location is described by an 3-dimensional vector, whose coordinates are values of identifying features of the source, such as its proximity to a manufacturing facility (A) and a waste dump (B), and the distance to the water table from the ground surface (C).

For each feature, a membership function capturing the meaning of “near” is provided by an expert and it is used to obtain the membership degree (score) of each feature value for a particular site. In addition, the expert provides monotone measures which contain all the information about the importance of each individual feature and all groups of features for identifying the true source. Using the discrete Choquet integral the individual scores are combined and a global degree of confidence of the statement “source location  $i$  belongs to the group of true source locations” is assigned to each possible source location.

Output from this part of the code (`initial_weights.dat`) is used as input to the driver and contains the initial weights of the sources. In cases where the user does not wish to use the Choquet integral to calculate initial weights, the user may create a data file containing uniform weights for all sources and call it `initial_weights.dat`.

#### 3.1 Main file: `weights_choquet_main.for`

The main file program performs the following actions:

- Opens the input file and reads in the number of potential source locations
- Calls subroutine `weights_readin` to read in the membership function input, the monotone measures assigned by the expert and the actual distances of the manufacturing facility, the waste dump and the distance to the water table.
- Calls subroutine `weights_calc` that calculates the global scores using the Choquet integral.

#### 3.2 Subroutines

- **`weights_readin.for`:** This subroutine opens the “`input_weights.dat`” file and reads the input parameters.

- **weights\_calc.for:** This subroutine calculates the standardized global weights for each potential source location. For each potential source location the following procedure is followed: First, the subroutine calculates the membership degrees that correspond to each feature. Then, it ranks the scores from smaller to bigger, and finally applies the discrete Choquet integral formula in order to calculate the global score. The global scores are then standardized with the highest score obtaining a value of 1.

### 3.3 IMSL subroutines

- **SVRGP:** sorts an array by algebraic values and returns a pointer array.

### 3.4 Input

- **input\_weights.dat:** Contains the following information:  
n\_sources: Number of potential source locations.
- **membership.dat:** Contains membership function 'breakpoints'. The first point is where the membership degree is 1 and starts decreasing. The second point is where the membership degree is 0 and remains 0. The membership function's shape is assumed a half trapezoid, since this is typical for distances. The order is:  
Line 1: first point for membership function for feature A (facility).  
Line 2: second point for membership function for feature A (facility).  
Line 3: first point for membership function for feature B (waste dump).  
Line 4: second point for membership function for feature B (waste dump).  
Line 5: first point for membership function for feature C (water table).  
Line 6: second point for membership function for feature C (water table).
- **measures.dat:** Contains the monotone measures (importance) of each feature and of all the combinations of the features. The order is:  
Line 1: monotone measure for A (facility).  
Line 2: monotone measure for B (waste dump).  
Line 3: monotone measure for C (water table).  
Line 4: monotone measure for A and B.



Line 5: monotone measure for A and C.

Line 6: monotone measure for B and C.

- **distances.dat:** Contains the real distances of each source location from the facility and waste dump, and the distance of the ground surface to the water table at that location. The order is:

Lines 1-n: Distance from facility of source locations 1 through n.

Lines n+1-2n: Distance from waste dump for source locations 1 through n.

Lines 2n+1-3n: Distance of ground surface from water table at source locations 1 through n assuming that there are n potential source locations.

### 3.5 Include files

- **weights.inc:** This file contains common variables definitions.

### 3.6 Output

- **initial\_weights.dat:** Contains the standardized initial weights for each potential source location that were calculated using the Choquet integral and will be used as input weights in the simulation part of the code.

### 3.7 Example input files

- **input\_weights.dat:**

12

- **membership.dat:**

34  
150  
34  
150  
6  
34

- **measures.dat:**

0.3  
0.5  
0.2  
0.7  
0.7  
0.8

- **distances.dat:**

25  
35.3  
55.9  
55.9  
35.3  
25  
7  
7  
11

## 4. Simulation Code Overview

This part of the code calculates the statistics of the concentration realizations. For each hydraulic conductivity realization and each potential source location the flow and transport model was previously run using PTC (Princeton Transport Code) and a concentration realization set was obtained for each potential source location. This part of the code selects a subset of the concentration realizations for each source according to their weights. After obtaining the desired number of contaminant concentration realizations they are combined and the statistics (mean and variance-covariance matrix) of this set of realizations is calculated.

### 4.1 Main file: `simmain.for`

The main file program includes the following:

- Calls subroutine "readinput" to read the input parameters.
- Reads all the concentration realizations for each source and each layer and for each node that was previously calculated using the Monte Carlo simulation method and updated by the Kalman filter. These are stored in files with names such as: 'c\_all1\_GL2\_L3.dat for source 1, geologic layer 2 and numerical layer 3. If this is the first iteration, then the file that is read has a name such as : 'c\_all\_orig1\_GL2\_L3.dat which means that it is the original file (not updated by the Kalman filter). After reading in all the concentration realization information the order of the realizations is randomly permuted. The composite plume is calculated using a subset of the set of realizations for each potential source location according to the source's weight. The larger the weight the more realizations of that particular potential source are included in the calculation of the composite plume.
- Calls IMSL subroutine DCORVC (calculates the variance-covariance matrix) to calculate the statistics of the subset of concentration realizations that were selected.
- Outputs the mean concentration in files `cmean (i).dat` (unformatted) and `comp_in_weights(i).dat` (formatted), and the concentration variance in `ccov (i).dat`, where `i` is the numerical layer number.

- Calculates the vertical covariance between layers for the nodes that correspond to the potential sampling locations and outputs it in the file `cover_z.dat`.

## 4.2 Subroutines

- **readinput.for:** This subroutine opens the “simu.dat” file and reads the input parameters.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse:** This subroutine returns the position of the first non-blank character. It is located in the `noutfile.for` file.

## 4.3 IMSL subroutines

- **RNPER:** Generates a pseudorandom permutation.
- **DPERMA:** Permutes the rows or columns of a matrix.
- **DSVRGP:** Sorts an array by algebraic values and returns a pointer array.
- **DCORVC:** Computes the variance covariance matrix.

## 4.4 Input

The input files for the simulation part of the code are:

- **Simu.dat:**

Line 1: n_sources	number of potential source locations.
Line 2: nodes	number of nodes in the numerical mesh.
Line 3: n_layers	number of numerical layers in the simulation model.
Line 4: g_layers	number of geological layers.
Line 5: Nsim	number of simulations.
Line 6: Nsam	number of potential sampling locations.

- **samloc.dat:** Contains the node information of the potential sampling locations.
- **flag\_sim.dat:**
  - flag1= 1: initial simulations (iteration1).
  - flag1 = 2,3,4 etc: subsequent iterations.
  - flag2= 1: input file used is 'initial\_weights.dat'.
  - flag2 = 2: input file used is 'new\_sources.dat'.
- **diff.dat:** If the total number of simulations is not the same as the original then we have to find the difference and allocate it accordingly. If the difference is positive we add it to the source with the highest weight, if it is negative we subtract it from the source with the smallest weight. The difference is stored in this file.
- **initial\_weights.dat:** Contains the initial weights for each potential source location.
- **new\_sources.dat:** Contains the updated weights for each potential source location.
- **src\_orig(i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source i, geologic layer j and numerical layer k that were originally calculated for each individual source at step 2.
- **c\_all\_orig(i)\_GL(j)\_L(k).dat:** Contains all the concentration realizations for source i, geologic layer j and numerical layer k that were originally calculated for each individual source at step 2.
- **src (i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source i, geologic layer j and numerical layer k that were updated with the new source strength calculated by the optimization part of the code.

- **c\_all (i)\_GL(j)\_L(k).dat:** Contains all the concentration realizations for source i, geologic layer j and numerical layer k that were updated with the new source strength calculated by the optimization part of the code.

#### 4.5 Include files

- **simpara.inc:** This file contains common variable definitions

#### 4.6 Output

- **cmean(i).dat (unformatted):** Contains the mean concentration at the nodes where estimates are needed. There is one file for each layer.  $i=1, \# \text{ Layers}$ .
  - **comp\_in\_weights(i).dat (formatted):** Contains the mean concentration at the nodes where estimates are needed. There is one file for each layer,  $i=1, \# \text{ Layers}$ .
  - **ccov(i).dat (unformatted):** Contains the concentration covariance-variance between the nodes where estimates are needed. There is one file for each layer,  $i=1, \# \text{ Layers}$ .
1. **covar\_z.dat:** Contains the vertical (between layers) concentration covariance matrix only for the sampling location nodes.

#### 4.7 Example input files

Only the files simu.dat and samloc.dat are created by hand for input to this part of the code. Samloc.dat was described in Section 2.

- **simu.dat:**

```
12
861
6
2
100
30
```

## 5. Kalman Code Overview

The Kalman Filter code is used from the second iteration onwards for updating the composite plume with the true concentration values for the samples that were obtained in previous iterations. This part of the code is executed automatically when the driver (Section 9) is used.

### 5.1 Main program: kalman.for

This program calls subroutine readinput that reads in the input variables and subroutine readvar that reads in the prior mean and variance, and the locations of the samples that were selected at previous iterations. Subroutine readvar calls subroutine calculations where the updating of all the previously selected samples is performed.

### 5.2 Subroutines

- **readinput.for:** This subroutine opens the “input\_kalman.dat” file and reads the input parameters.
- **readvar.for:** This subroutine reads the prior concentration mean and variance and the location where the samples were taken at the previous iterations. It also calls subroutine calculations where the updating is performed.
- **calculations.for:** This subroutine performs the updating of the mean and variance-covariance matrix of concentration with the real data.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse.for:** This subroutine returns the position of the first non-blank character. It is located in the noutfile.for file.
- **update\_z.for:** This subroutine uses the vertical covariance matrix calculated in the simulation part of the code and the real data collected at a specific node and a specific layer. It calculates new concentration values that will be used as “artificial data” for the same nodes where samples were taken but for the numerical layers that belong to the same geologic layer as the numerical layer where the sample was taken.

### 5.3 Input

- **input\_kalman.dat:** Contains the following information. As created by the user, the first line of this code will have Nsam\_all equal to 1. The driver will amend this line as more samples are taken.

Nsam_all:	Number of samples taken so far at all layers.
n_layers:	Number of numerical layers.
g_layers:	Number of geological layers.
Nloc:	Number of locations where estimates are needed.
Nsam:	Number of potential sampling locations.
r:	Sampling error.

- **ccov(i).dat:** Contains the concentration covariance-variance between the nodes where estimates are needed. It is the output from the simulation code,  $i=1, 2, \dots, n$ , where  $n$ =number of numerical layers where sources are located.
- **cmean(i).dat:** Contains the mean concentration at the nodes where estimates are needed. It contains the output from the simulation code,  $i=1, 2, \dots, n$ , where  $n$ =number of numerical layers where sources are located.
- **max\_sample\_all.dat:** Contains the node numbers of all the previously selected samples (with different indexing than the ptc mesh).
- **real\_sample\_node\_all.dat:** Contains the ptc indexing of the nodes where samples have been taken.
- **layers\_all.dat:** Contains the layer info for all samples taken.
- **g\_layer(j).dat:** The first number in this file is the number of numerical layers that correspond to this geologic layer ( $j$  is the geologic layer). The rest of the entries are the specific layer numbers that correspond to the geologic layer.
- **g\_layer.dat:** This file contains an index that defines which geologic layer each numerical layer belongs to.
- **plumereal.dat:** Contains the real concentration data at the potential sampling locations.
- **covar\_z.dat:** Contains the vertical (between layers) concentration covariance matrix only for the sampling location nodes.



## 5.4 Include files

- **gaparam.inc:** This file contains common variable definitions.

## 5.5 Output

- **creal\_new.dat:** This file contains the new ‘artificial data’ calculated for the layers that belong to the same geologic layer using the vertical covariance info.
- **mean\_combined\_all.dat:** Contains the updated mean concentration at the nodes where estimates are needed (formatted).
- **ccov.dat:** Contains the updated concentration covariance-variance between the nodes where estimates are needed (unformatted).
- **cmean.dat:** Contains the updated mean concentration at the nodes where estimates are needed (unformatted).

## 5.6 Example input files

The only input files that must be created by the user for this part of the code are `input_kalman.dat`, `g_layer(j).dat`, `g_layer.dat` and `plumereal.dat`. The remaining input files are created automatically as output from other parts of the code.

- **input\_kalman.dat:**

```
0
6
3
861
30
1.0000000E-12
```

- **g\_layer1.dat:** (corresponds to geological layer 1)

```
1
1
```

- **g\_layer2.dat:** (corresponds to geological layer 2)

```
3
2
3
4
```

- **g\_layer3.dat:** (corresponds to geological layer 3)

2  
5  
6

- **g\_layer.dat:**

1  
2  
2  
2  
3  
3

- **plumereal.dat:** (In the case where there are 30 sampling locations and 6 numerical layers, there are  $6 \times 30 = 180$  entries in this data file. The first 30 are the concentration measurements for layer 1, the second 30 for layer 2, etc. For any particular sampling location where a water quality measurement was not taken in a given layer, that entry should be specified as some very small number (i.e. 0.0001).

19.222  
0.0001  
7.607  
10.684  
1.608  
8.536  
0.0001  
7.173  
1.375  
... (entries 10 through 179 are omitted here for brevity)  
0.0001

## 6. Choquet code overview

This part of the code performs the selection of the optimal sampling location for the current iteration. Two features are taken into consideration: the reduction in uncertainty and the proximity to the potential source locations (resulting in higher concentration values). These features are combined using a discrete Choquet integral (a kind of distorted weighted average) based on the importance (monotone measures) of the two features. This part of the code is executed automatically when one runs the driver (Section 9)

### 6.1 Main file: `choquet_main.for`

The main file program performs the following tasks:

- Calls subroutine `choquet_read_in` that reads in the input parameters from file `input_choquet.dat`.
- Calls subroutine `choquet_calc` that performs the choquet integral calculations.

### 6.2 Subroutines

- **`choquet_read_in.for`:** This subroutine opens the “`input_choquet.dat`” file and reads the input parameters.
- **`choquet_calc.for`:**

This subroutine performs the Choquet integral calculations:

- First it reads in the covariance matrix and the mean calculated from the simulation code (or Kalman filter after the second step). Then it calculates the total variance before taking the sample (that is the summation of all the diagonal elements of the covariance matrix). Then it tries all the possible sampling locations, each one at a time, and calculates the total variance reduction that each one will provide using the equation provided in Herrera, 1998.
- Then it maps the uncertainty reduction to the corresponding membership degree using the following equation (membership function):  $y=x/\text{max\_ratio}$  for  $0<x<\text{max\_ratio}$  and  $y=1$  for  $\text{max\_ratio}<x$ , where `max_ratio` is the maximum value of the reductions in uncertainty produced

by each potential sampling well. This means that the maximum reduction in uncertainty is assigned a membership degree of 1 and the rest is scaled by dividing by the `max_ratio`.

- Finds the highest alpha-cut in which the concentration of the sampling location (prior) belongs to and stores it.
- Finds the membership degree of the `a_cut` using the following equation (membership function):  $y=x$  if  $0 < x < 1$  and  $y=1$  if  $x \geq 1$  (linear membership function).
- Uses the Choquet integral to calculate the final score.

*The user must ensure that the new sample has not been taken before. So, first we open the file `max_sample_all.for` and read the samples that we have already taken.*

- Finds the maximum of the scores, and the potential sampling location that corresponds to that score is chosen as the next sampling location. It is stored in the file `max_sample.dat`.
- Finds the new (reduced) variance that would result after taking each of the potential samples.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse.for:** This subroutine returns the position of the first non-blank character. It is located in the `noutfile.for` file.

### 6.3 Input

- **input\_choquet.dat:** Contains the following information:
  - `n_old_samples:` Number of samples already taken in previous iterations.
  - `ns_layers:` Number of numerical layers with samples.
  - `Nloc :` Number of locations where estimates are needed.
  - `Nsam:` Number of potential sampling locations.
  - `meas_cut:` Monotone measure for nodes with high concentration.
  - `meas_unc:` Monotone measure for uncertainty.
  - `r:` Sampling error.

- **ccov(i).dat:** Contains the concentration covariance-variance between the nodes where estimates are needed. It is the output from the simulation or the Kalman filter code. i: layer number
- **cmean(i).dat:** Contains the mean concentration at the nodes where estimates are needed. It is the output from the simulation or the Kalman filter code, i: layer number.
- **max\_sample\_all.dat:** Contains the node numbers of the samples that have already been taken (with different indexing than the ptc mesh). It is both input and output for this part of the code. It is being updated each time to include the newly selected sampling node number.
- **samloc.dat:** Contains the nodes of the potential sampling locations.

#### 6.4 Include files

- **choquet.inc:** This file contains common variable definitions.

#### 6.5 Output

- **count\_samples.dat:** Contains a variable that counts how many locations have the same score as the maximum score. If there is more than one location with the same score one of them is chosen randomly.
- **max\_sample.dat:** Contains the node number of the newly selected sample (with different indexing than the ptc mesh).
- **layer.dat:** Contains the layer numbers of the newly selected sample.
- **layer\_all.dat:** Contains the layer number of all selected samples.
- **max\_sample\_all.dat:** Contains the node number of the newly selected sample and all the previous ones (with different indexing than the ptc mesh).
- **real\_sample\_node\_all.dat:** Contains the node number of the newly selected sample and all the previous ones (using the ptc mesh index).
- **max\_sample\_L(i).dat:** Contains the node number of the selected samples for each layer i (with different indexing than the ptc mesh).

## 6.6 Examples of input files

The only input file that needs to be created by the user for this part of the code is `input_choquet.dat`. The remaining input files are output from other parts of the code.

- **input\_choquet.dat:**

```
1
6
861
30
0.5
0.5
1.0000000E-12
```

## 7. Optimization Code Overview

This part of the code is a linear optimization program that seeks to find the set of source strengths that minimize the summation of the absolute differences between modeled concentration values and measured concentration values at the sampling locations. The flow and transport simulator is coupled with the optimizer by a response matrix that contains the information on how the concentration values at the sampling locations change with unit changes in the magnitudes at the potential sampling locations. After the optimal values for the source magnitudes have been selected, the simulated concentration field (composite plume) is modified to reflect the change in source strength.

### 7.1 Main program: **optimization.for**

This program calls subroutine `readinput.for` to read the input parameters and calls subroutine `simplex.for`, which performs the source strength optimization.

### 7.2 Subroutines

- **readinput.for:** This subroutine opens the “opt.dat” file and reads the input parameters.
- **simplex.for:** This subroutine performs the source strength optimization. It does the following:
  - Reads sampling location nodes, mean concentration for all sources, all layers and all the concentration realizations for all sources.
  - Reads initial weights for each potential source location.
  - Calculates the Jacobian matrix for all sampling locations. Each element of the Jacobian matrix represents the change in concentration that will occur at each sampling location if a unit change in source strength for each potential source location occurs.
  - Constructs the linear optimization problem and solves it using an IMSL subroutine `DDLPRS` using the revised simplex method.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.

- **cparse.for:** This subroutine returns the position of the first non-blank character. It is located in the noutfile.for file.

### 7.3 Input

- **opt.dat:** Contains the following information:
  - n\_sources: Number of potential source locations.
  - nodes: Number of ptc nodes.
  - n\_layers: Number of numerical layers with sources.
  - g\_layers: Number of geologic layers with sources.
  - Nsim: Number of simulations.
  - Nsam: Number of potential sampling locations.
  - flag: flag = 0:ordinary residuals 1:normalized residuals.
- **samlocdat:** Contains the nodes of the potential sampling locations.
- **src\_orig(i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source i, geologic layer j and numerical layer k that were originally calculated for each individual source at step 2.
- **c\_all\_orig(i)\_GL(j)\_L(k).dat:** Contains all the concentration realizations for source i, geologic layer j and numerical layer k that were originally calculated for each individual source at step 2.
- **initial\_weights.dat:** Contains the initial weights for each potential source location.
- **input\_kalman.dat:** This is the input file for the Kalman filter part of the code. Only the current iteration number (first variable in the file) is read.
- **max\_sample\_all.dat:** Contains the node numbers of all the previously selected samples (with different indexing than the ptc mesh).
- **layer\_all.dat:** Contains the layer number of all selected samples.
- **plumereal.dat:** Contains the real concentration data for all sampling locations.

### 7.4 Include files

- **optpara.inc:** This file contains common variable definitions.



## 7.5 Output

- **jacobian.dat:** Contains the Jacobian matrix.
- **cnew\_opt.dat:** Contains three values for each sampling location. The first is the newly calculated concentration value, the second is the true data value and the third is their difference.
- **magnitude.dat:** Contains the source strength (magnitude) for each potential source location at the current iteration.
- **magnitude\_all.dat:** Contains the source strength (magnitude) for each potential source location for all iterations.
- **obj\_value.dat:** Contains the optimal objective value for all iterations.
- **src (i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source i, geologic layer j and numerical layer k that were updated with the new source strength calculated by the optimization part of the code.
- **c\_all (i)\_GL(j)\_L(k).dat:** Contains all the concentration realizations for source i, geologic layer j and numerical layer k that were updated with the new source strength calculated by the optimization part of the code.

## 7.6 Examples of input files

The only input data file that needs to be created by the user for this part of the code is opt.dat.

- **opt.dat:**

```
12
861
6
3
100
30
0
```

## 8. Alpha Cut Code Overview

This part of the code is used for comparison of the updated composite plume and the individual potential source plumes. The plumes are represented as fuzzy sets with membership functions defined as normalized concentration values. Several  $\alpha$ -cuts of the fuzzy sets are considered. Each  $\alpha$ -cut for the updated plume is compared with the corresponding  $\alpha$ -cut of the individual plume and the number of nodes (N) that are common in the 2  $\alpha$ -cuts is recorded. The global degree (g) of similarity between the two plumes is obtained by weighting the number of nodes present in the two  $\alpha$ -cuts by the  $\alpha$  value itself. Based upon the outcome of this comparison, the weights of the sources are ultimately amended. This part of the code is run automatically when the driver (Section 9) is executed.

### 8.1 Main program: a\_cut\_main.for

This program calls subroutine a\_cut\_readin to read the input parameters.

- Calls subroutine read\_conc, which reads the mean concentration for the composite plume and individual sources.
- If the mesh is triangular then the program:
- Calls subroutine a\_cut\_calc to calculate an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut) for the combined plume.
- Calls subroutine a\_cut\_calc to calculate an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut) for the individual plumes.
- Calls subroutine intersect to find the number of intersecting nodes between the combined plume and the individual plumes.

If the mesh is quadrilateral then the program:

- Calls subroutine a\_cut\_calc\_quad to calculate an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut) for the combined plume.

- Calls subroutine `a_cut_calc_quad` to calculate an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut) for the individual plumes.
- Calls subroutine `intersect_quad` to find the number of intersecting nodes between the combined plume and the individual plumes.
- Calls subroutine `mult_weight` to multiply the number of intersecting nodes (or elements for the triangular case) for each alpha cut and for each potential source by their corresponding weight.
- Finds the maximum of the scores (weights).
- Calculates the weights of the other sources and normalizes them by assuming that the weight for the most probable source is 1.
- Outputs the new weights in files `new_sources.dat` and `new_sources_count.dat`

## 8.2 Subroutines

- **a\_cut\_readin.for:** This subroutine opens the “`a_cut_input.dat`” file and reads the input parameters.
- **read\_conc.for:** This subroutine reads in the concentration of the composite plume and the concentration of the individual plumes for each potential source location.
- **a\_cut\_calc.for:** Only for triangular mesh: This subroutine creates an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut).
- **a\_cut\_calc1.for:** Only for triangular mesh: This subroutine creates an index of zeros and ones (1: node belongs to alpha cut, 0: node doesn't belong to alpha cut).
- **intersect.for:** Only for triangular mesh: This subroutine is used to find the number of intersecting nodes between the combined plume and the individual plumes.
- **a\_cut\_calc\_quad.for:** Only for quad mesh: This subroutine creates an index of zeros and ones (1: element belongs to alpha cut, 0: node doesn't belong to alpha cut).

- **a\_cut\_calc1\_quad.for:** Only for quad mesh: This subroutine creates an index of zeros and ones (1: element belongs to alpha cut, 0: node doesn't belong to alpha cut).
- **intersect\_quad.for:** Only for quad mesh: This subroutine is used in order to find the number of intersecting nodes between the combined plume and the individual plumes.
- **mult\_weight.for:** This subroutine multiplies the number of intersecting nodes for each alpha cut and for each potential source by their corresponding alpha cut weight. The resulting score is stored in variable 'final'.
- **area\_calc.for:** Only for triangular mesh: It calculates the area of each triangular element. The larger the area the more weight is given to the particular element if it is common in the 2 plumes that are being compared.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse.for:** This subroutine returns the position of the first non-blank character. It is located in the noutfile.for file.

### 8.3 Input

- **a\_cut\_input:** Contains info about the input parameters:
 

n_nodes	Number of nodes.
n_elem	Number of elements.
n_layers	Number of layers.
g_layers	Number of geological layers where sources are located.
nsam	Number of sampling locations.
n_sources	Number of sources.
n_acuts	Number alpha cuts.
- **alpha\_cuts.dat:** Contains the desired alpha cut levels.
- **flag\_acut.dat**
  - flag1= flag to indicate if this is the first iteration. If it is then we need to calculate the area. 0: calculate area, 1: read area from file
  - flag2=flag to indicate triangular or quadratic mesh

0: triangular mesh, 1: quad mesh

- **elements.dat:** Contains the element information i.e. which nodes each element contains.
- **coordinates.dat:** Contains the coordinates for each node.
- **cmean(i).dat:** Contains the mean concentration at the nodes where estimates are needed. There is one file for each layer,  $i=1, \# \text{ Layers}$ .
- **src(i)\_GL(j)\_L(k).dat:** Contains the mean concentration for source  $i$ , geologic layer  $j$  and numerical layer  $k$  that were updated with the new source strength calculated by the optimization part of the code.

#### 8.4 Include files

- **a\_cut.inc:** Includes the definition of the common variables.

#### 8.5 Output

- **new\_sources.dat:** This file contains the new weights that are calculated using the `a_cuts` code at each iteration.
- **new\_sources\_count.dat:** This file contains a record of the new and old weights that are calculated using the `a_cuts` code at each iteration.
- **a\_cut\_result.dat:** This file contains the raw (not normalized) scores obtained using the alpha cuts method.
- **a\_cut\_result.dat:** Contains the area for each element.
- **a\_cut\_result.dat:** Contains the area for each element but normalized (all the values are divided by the max area).

#### 8.6 Examples of input files

- **a\_cut\_input:**

```
861
800
6
3
30
12
10
```

- **alphat\_cuts.dat:**

0.1  
0.2  
0.3  
0.4  
0.5  
0.6  
0.7  
0.8  
0.9  
1.0

- **flag\_acut.dat:**

0  
1

- **elements.dat:** information for this data file is contained in ptc.out, an out file created when ptc is run
- **coordinates.dat:** information for this data file is contained in ptc.out, and out file created when ptc is run

## 9. Run code overview

This part of the code is used as a “driver”. It connects other parts of the code (including simulation, Kalman filtering, Choquet integral, optimization and alpha cuts) and runs them in the correct order. The outputs of the various parts of the code are automatically used as input to other parts of the code when this driver is executed.

### 9.1 Main program: `main_run.for`

- The variable `diff_all` that checks the convergence of the algorithm is initialized to a value of 100 (arbitrary large value).
- Then the simulation part of the code is run.
- If this is not the first iteration and there were previous samples taken the code `kalman` is called to update the composite plume.
- Output files for plotting are written.
- Then the file `input_choquet.dat` is updated to reflect the current iteration number.
- The code `Choquet_Integral` integral is run to choose a sampling location.
- The code `Optimization` is run to calculate new source strengths.
- Then the simulation part of the code is run again with the new source strengths and the initial weights.
- The Kalman filter code is run again to update the newly calculated composite plume using the sampling info for the sample.
- The code `alpha cuts` is called to determine the new potential source weights.
- The subroutine `convergence` is called to calculate the variable `diff_all`.
- This procedure continues until the convergence criterion is reached. i.e. until the summation of the differences between new and old weights for each source is less than a predetermined tolerance.

## 9.2 Subroutines

- **convergence.for:** In this subroutine the convergence of the algorithm is checked. The summation of the differences between new and old weights for each source is calculated and compared to a predetermined tolerance.
- **noutfile.for:** This subroutine creates the name of an output file using the counter as part of the name.
- **cparse:** This subroutine returns the position of the first non-blank character. It is located in the noutfile.for file.

## 9.3 Include files

- **run.inc:** This file contains common variable definitions.

## 9.4 Input files

- **c\_all\_orig(i)\_GL(j)\_L(k).dat** (Section 4)
- **src\_orig(i)\_GL(j)\_L(k).dat** (Section 4)
- **coordinates.dat** (Section 8)
- **elements.dat** (Section 8)
- **glayer.dat** (Section 5)
- **glayer(j).dat** (Section 5)
- **plumereal.dat** (Section 5)
- **samloc.dat** (Section 2)
- **initial\_weights.dat** (Section 3)
- **input\_choquet.dat** (Section 6)
- **input\_kalman.dat** (Section 5)
- **opt.dat** (Section 7)
- **simu.dat** (Section 4)
- **flag\_acut.dat** (Section 8)
- **a\_cut\_input.dat** (Section 8)
- **alpha\_cuts.dat** (Section 8)



## 10. Example Problem

### 10.1 Problem statement

Appropriate execution of the DNAPL source locator begins with a candidate groundwater remediation site. A suitable site is one for which a site investigation has produced hydrogeological data to permit the construction of a groundwater flow and transport model and water quality data that confirms the presence of a DNAPL source on site. Once such a site has been identified, a groundwater flow and transport model of the site must be developed.

In this example, the imaginary site we have “selected” is presented in Figure 2. An investigation of this site has revealed 3 geologic layers of uniform conductivity (Figure 2), 30 potential water quality sampling locations (Figure 2, lower right) and 12 potential source locations (4 in each of the three geological layers). At this site there exists a manufacturing facility and a waste dump. A third feature relevant to calculation of initial source weights is the distance of the potential source locations from the water table.

The boundary of the groundwater flow model of the site (Figure 2, plan view) is comprised of two ‘no flow’ conditions (in the north and south) and ‘constant head’ conditions in the west (90m of head) and the east (85m of head). Groundwater flows toward the east. The true (unknown) source is located in numerical layer 2 at 100ft easting and 200ft northing. The plume that emanates from this source is presented in Figure 2, lower right. This calibrated model will later be run many times in order to create the number of realizations necessary to properly run the DNAPL source locator and provide a measure of the uncertainty in the resulting estimated contaminant plume.

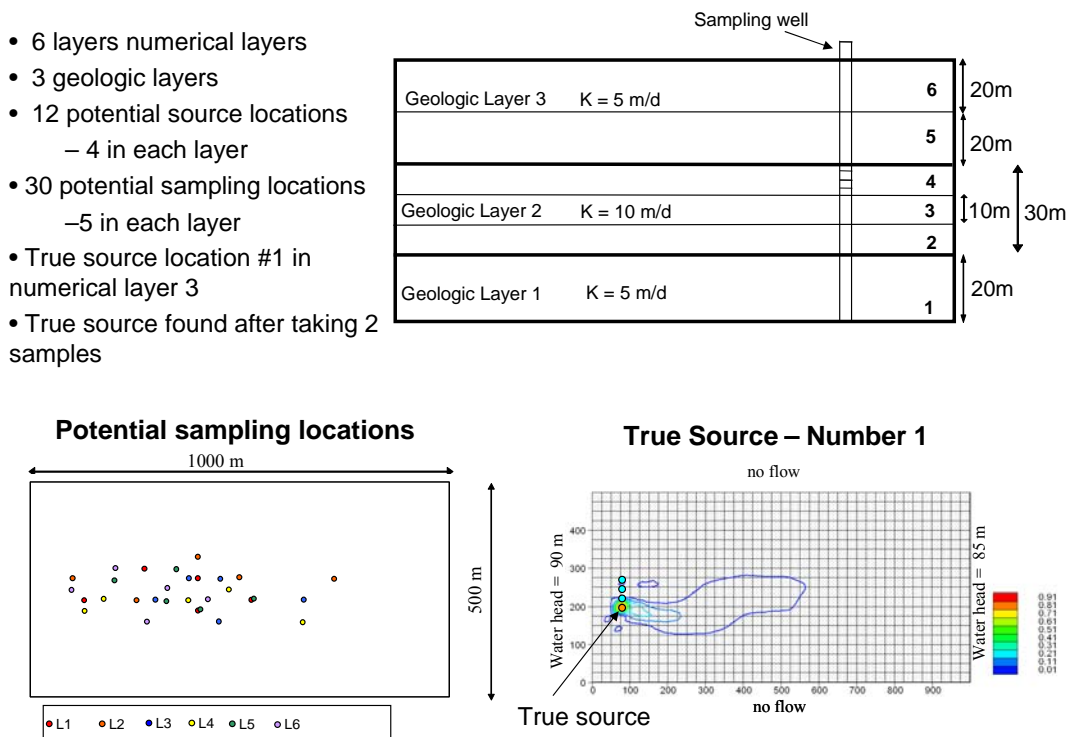


Figure 2. Sample problem setup

The ptc project for this example has 1 layer, 1 stress period (3650 days), a constant mean K and a rectangular mesh of 861 nodes.

The project domain is rectangular with dimensions 1000m × 500m. There are 4 potential source locations in each geologic layer at nodes: 332, 373, 414, 455. The flow and transport parameters for the first layer used in this project are shown in Table 3. Similar parameters were used for the set of the layers (see Argus One .mmb file).

**Table 1.** PTC model parameters

Bottom elevation	0
Elevation L1	20 m
xConductivity (mean)	5 m/d
yConductivity (mean)	5 m/d
zConductivity (mean)	5 m/d
Initial Heads L1	90 m
Storativity	0.001
xDispersivity	1 m
yDispersivity	1 m
zDispersivity	1 m
Porosity	0.3

After running the ptc model, the required ptc files are used as input for the ‘simulation’ part of the code (see Section 2).

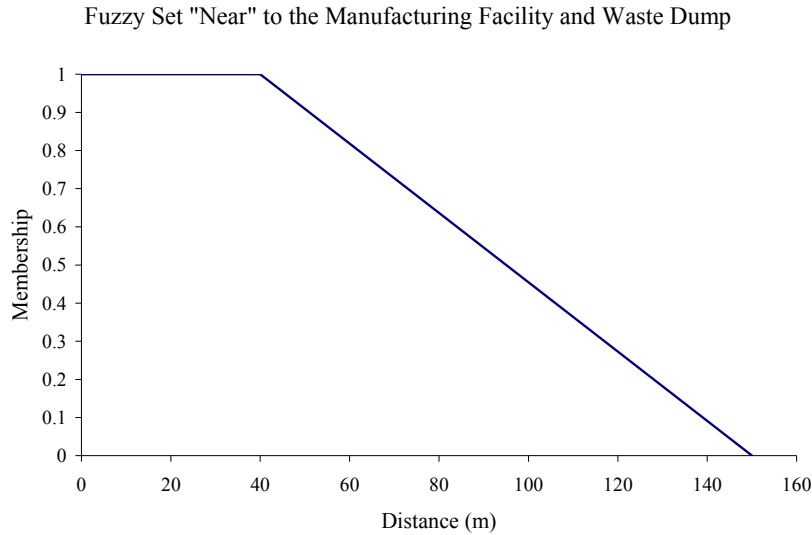
## 10.2 Choquet integral code and initial source weights

After the flow and transport model for this example site was built and calibrated, an expert hydrogeologist familiar with the site provided monotone measures for each of the features (and combinations thereof) that are relevant to calculating initial weights for each of the sources that signify the possibility that each source is the true source. These six specified monotone measures are:

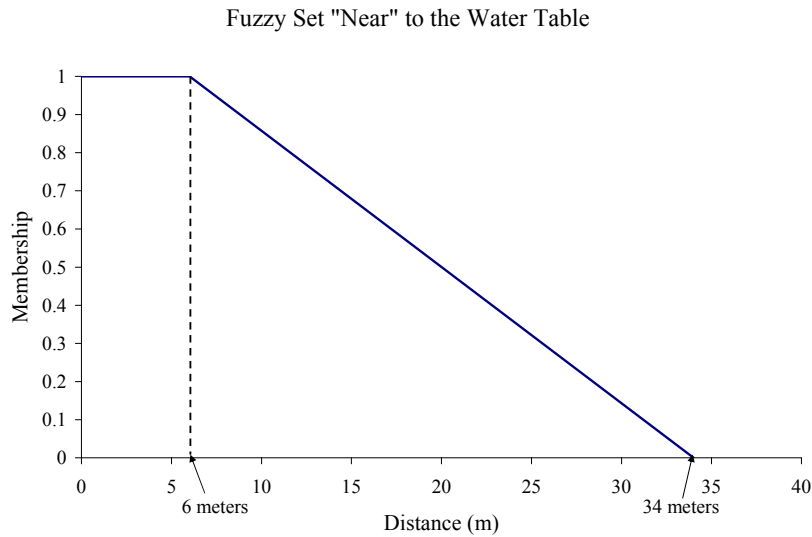
$$\begin{aligned} \mu(A) &= 0.3 & \mu(B) &= 0.5 & \mu(C) &= 0.2 \\ \mu(A, B) &= 0.7 & \mu(A, C) &= 0.3 & \mu(B, C) &= 0.3 \\ \mu(A, B, C) &= 1 & \mu(\emptyset) &= 0.3 \end{aligned}$$

where A represents the manufacturing facility, B represents the waste dump and C represents distance of the source to the water table. Clearly these measures are not additive. Most important are the boundary conditions for the monotone measures: the measure for the full set must equal unity and the measure for the null set must equal zero.

In addition to these monotone measures, the expert provided fuzzy set membership functions to quantify the abstract notions “near” in reference to distances of a potential source to the manufacturing facility (Figure 3a), waste dump (Figure 3a) and water table (Figure 3b).



**a**



**b**

**Figure 3.** Membership functions for (a) “Near to the Manufacturing Facility, “Near to the Waste Dump” and (b) “Near to the Water Table”

The interpretation of these membership functions is as follows: in Figure 3a, the distance 80m belongs to the notion “near to the manufacturing facility/waste dump” with 0.6 membership. In other words 80m represents the notion “near” with 60% strength. On the other hand, 40m represents this notion completely (membership = 1.0). The actual distance measurements and corresponding membership degrees are given in Table 2. The DNAPL source locator employs the Choquet integral to transform the measurements, the fuzzy sets and the monotone measures into the initial weights for the potential sources (Table 3). The calculation of the initial weights is not crucial to successful DNAPL source finding, though informed initial weights have the potential to affect the order in which water quality samples are taken, and ultimately reduce the number of iterations the algorithm needs to arrive at a conclusion regarding the true DNAPL source.

**Table 2.** Distances and corresponding membership degrees

	Distance from facility (m)	Distance from dump (m)	Distance to water table (m)	Membership Degrees		
				$\mu(A)$	$\mu(B)$	$\mu(C)$
Source 1	25.0	55.9	7.0	1.0	0.81	0.96
Source 2	35.3	35.3	7.0	0.99	0.99	0.96
Source 3	55.9	25.0	11.0	0.81	1.0	0.82
Source 4	79.1	35.3	11.0	0.61	0.99	0.82

**Table 3.** Membership degrees and Choquet integral-calculated global weights for potential source locations

	Membership for facility	Membership for waste dump	Membership for water table	Global weight	Standardized Global Weight
Source 1	1.0	0.81	0.96	0.93	0.96
Source 2	0.99	0.99	0.96	0.98	1.0
Source 3	0.81	1.0	0.82	0.91	0.93
Source 4	0.61	0.99	0.82	0.86	0.88

With each water quality sample and corresponding iteration, these initial weights change. Though with the source weights in Table 3 it appears that Source 2 is considered the most possible source location, as water quality samples are considered, ideally Source 1 will appear to be the most possible source.

### 10.3 Latin hypercube sampling

After the source weights are initialized, Latin hypercube sampling (LHS) is performed in order to obtain the hydraulic conductivity realizations for each layer. After the code is executed, the realizations are stored in files conk1.dat through conk6.dat and are necessary input to the DNAPL source locator algorithm's simulation step. For this example, we used a rectangular mesh with 861 nodes (Nodes = 861) and generated 100 realizations (Nsim = 100). When running the LHS code, an input file is required and defines the following (values for this example's layer 1 are provided in parentheses): whether an existing mesh will be used {1 if yes, 0 is no} (Imesh = 1), the number of nodes in the mesh (Nodes = 861), the number of simulations (Nsim = 100), which variogram model is used (Mtype = 2, exponential), correlation lengths in the x and y directions (ax = 200, ay = 200), the variance (cvar = 1), nugget (cnugget = 1) and mean (cmean = 0). For this case, the data file 'info.dat' looks like the following for layer 1 (Figure 4):

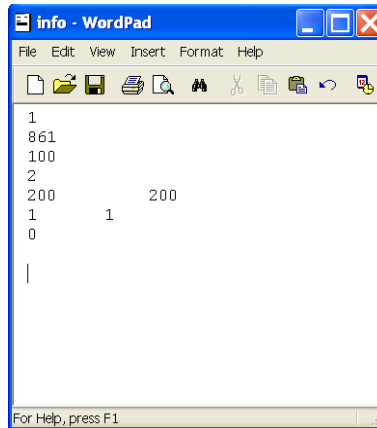


Figure 4. info.dat

In layer 2, the correlation lengths in the x and y directions change to 100m. These realizations are input into the 100 executions of the groundwater flow and transport model.

#### 10.4 Simulation code

This part of the code considers the weights of the potential sources and uses these weights to dictate how many realizations are executed for each potential source. For instance, in Table 3, the weight for source 1 is 0.96. This suggests that 96 realizations are run with source 1 as the true source. Likewise, source 2 would be the true source for the full 100 realizations. Once all realizations are run for all sources, they are combined and the aggregate concentration statistics are calculated. What results are a mean and variance for each node in every layer of the numerical mesh. The means are stored in 'cmean(i).dat' and the variances and covariances in 'ccov(i).dat', where  $i = 1, \dots, \#$  layers. The input information file for the simulation is called 'simu.dat,' which contains the following variables:

n_sources = 4	number of potential source locations in each layer.
nodes = 861	total number of nodes in the numerical mesh.
n_layers = 6	number of numerical layers in the simulation model.
g_layers = 3	number of geologic layers in the simulation model.
Nsim = 100	total number of simulations.
Nsam = 30	total number of sampling locations.

#### 10.5 Choquet code

The next step is Choquet code and the input parameters (in input\_choquet.dat) that are used in this example are:

n_old_samples:0	Number of samples taken in previous iterations.
n_old_samples: 0	Number of previously taken samples.
ns_layers: 6	Number of numerical layers in the model.
Nloc : 861	Number of locations where estimates are needed.
Nsam: 30	Number of potential sampling locations.

meas_cut: 0.5	Monotone measure for nodes with high concentration.
meas_unc: 0.5	Monotone measure for uncertainty.
r: 1E-02	Sampling error.

The Choquet code finds the optimal sampling location and stores it in max\_sample.dat using a different node index than ptc. The node number that ptc is using is stored in real\_sample\_node\_all.dat.

### 10.6 Kalman filter code

The next step is the Kalman Filter 1 updating of the plume's mean and variance-covariance matrix. The input parameters (stored in input\_kalman1.dat) are:

Nsam_all: 1	Number of samples at all layers.
n_layers=6	number of numerical layers in the simulation model.
g_layers=3	number of geologic layers in the simulation model.
Nloc : 861	Number of locations where estimates are needed.
Nsam: 70	Number of potential sampling locations
r: 1E-02	Sampling error.

For this part of the code the file "plumereal.dat" has to be prepared. A hydraulic conductivity realization is selected randomly and the corresponding concentration realization is treated as the "true" plume. The real concentration values at the potential sampling locations come from this "true" plume. After the updating is performed, the new mean is stored in cmean.dat and the new variance-covariance matrix is stored in ccov.dat.

### 10.7 Optimization code

Now the optimization part of the code is run to calculate new source magnitudes. The input file (opt.dat) contains the following information:

n_sources: 4	Number of sources in each layer.
nodes: 861	Number of nodes.
n_layers: 6	Number of numerical layers.
g_layers: 3	Number of geologic layers.
Nsim: 100	Number of realizations.
Nsam: 30	Number of sampling locations.
Flag:1	Flag indicating use of ordinary or normalized residuals.

### 10.8 Alpha cut code

Now that the plume is updated with new magnitudes it has to be compared with the individual plumes that emanate from each source separately to calculate the new weights. We have to run the simulation code 6 times using only one source location at a time. The resulting concentration mean at each node is stored in the file "sources.dat". The order is the same as the order of the 6 sources in the ptc "bctran.dat" file.

The comparison is performed by using the alpha\_cuts part of the code.

The input parameters (in input\_a\_cut.dat) are:

n_nodes: 861	Number of nodes.
n_elem: 800	Number of elements.
n_layers: 6	Number of layers.
g_layers:3	Number of geological layers with sources.
nsam: 30	Number of sampling locations.
n_sources: 4	Number of sources.
n_acuts:10	Number alpha cuts.

In this example there are 10 alpha cut levels used (0.1, 0.2, ... , 0.9, 1). The new weights replace the initial weights in “new\_sources.dat”.

### 10.9 Ensuing iterations

Then a second iteration begins and the simulation part of the code is run again with the new weights resulting in a new concentration mean and variance-covariance matrix. At this iteration and all the ones that follow, before proceeding to the Choquet code, a Kalman Filter is used to update the mean and variance-covariance matrix of the simulated plume with the real data on the samples that were taken at previous iterations. Then the Choquet code is run and the algorithm continues as described above until a convergence is reached. In this case the algorithm converges when the sum of the differences between the new and old weights is less than 0.1.

In this example the algorithm reaches convergence after only 2 iterations (after taking 2 water quality samples). The final weights that correspond to the confidence that we have that each potential source location is the true source location are shown in Table 4.

**Table 4:** Final weights for the potential source locations

	<b>Final Weight</b>
Source 1 – GL1	0
Source 2 – GL1	0
Source 3 – GL1	0
Source 4 – GL1	0
<b>Source 1 – GL2</b>	<b>1.0</b>
Source 3 – GL2	0
Source 4 – GL2	0
Source 1 – GL3	0
Source 2 – GL3	0
Source 3 – GL3	0
Source 4 – GL3	0
Source 2 – GL3	0

As the results suggest the algorithm identified the true source location correctly. It is source number 1 located in the 2<sup>nd</sup> geologic layer. Figures 5 through 7 show the composite plume before taking no samples, after taking one sample and after taking two samples. The sample selected at the current iteration is represented by a red dot, the

samples taken at previous iterations are represented by black dots. The true source location is represented by an orange dot, the other potential source locations by light blue dots. We chose to show results only in one of the layers. The results for the rest are similar.

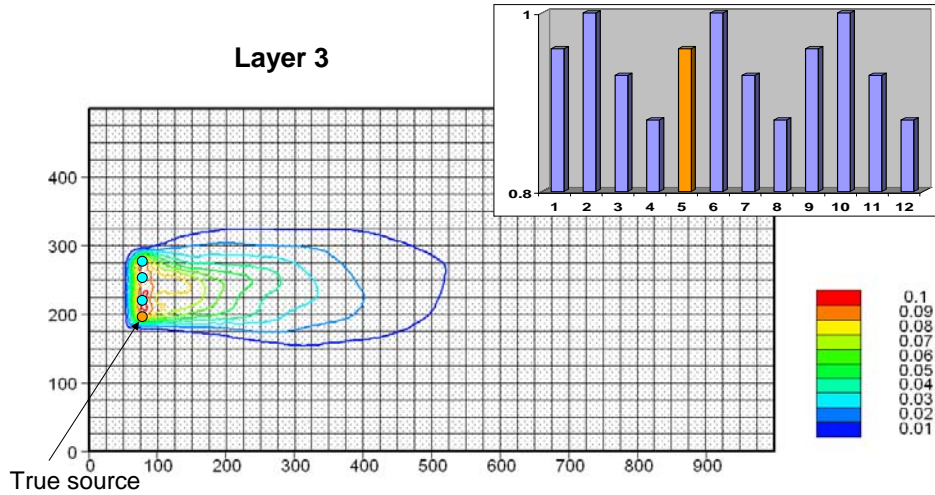
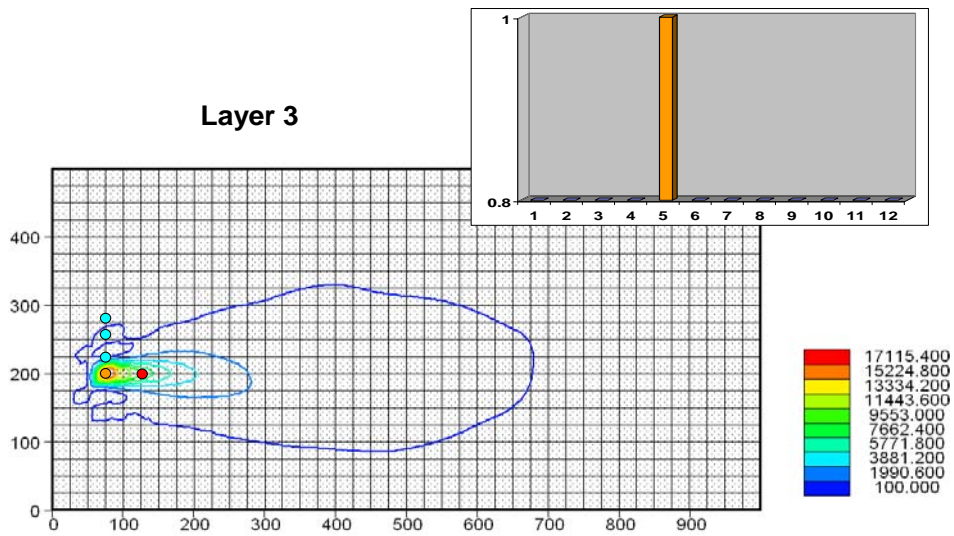


Figure 5. Plume before taking no samples and initial weights for the potential source locations

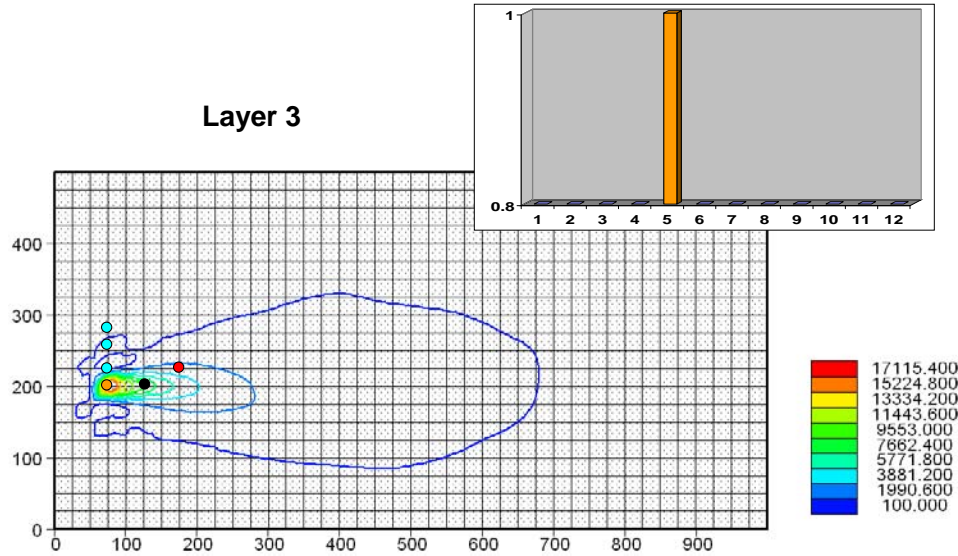


Red dot: current sample

Black dots: previous samples

Figure 6. Plume after taking 1 sample and resulting weights for the potential source locations





Red dot: current sample

Black dots: previous samples

Figure 7. Plume after taking 2 samples and resulting weights for the potential source locations

### 10.10 Algorithm driver execution

Once the hydraulic conductivity realizations are created, the individual concentration fields estimated by the Individual Simulations part of the algorithm and the initial weights specified, the entire remaining algorithm can actually be run with little effort. This is accomplished by using the driver called 'main\_run.dsw.' In the same folder where this driver resides must also reside 5 applications (a\_cut, choquet\_integral, kalman, optimization, and Simulation). Each of these applications require their own input data files. The complete list of input files (each of which is explained earlier in this document) that also need to be saved in the same folder as the driver is provided below.

#### Individual Simulation Concentration Statistics

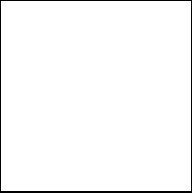
c\_all\_orig(i)\_GL(j)\_L(k).dat (all concentration realizations for source i, geologic layer j, and numerical layer k)\*

src\_orig(i)\_GL(j)\_L(k).dat (mean concentrations)

\*These two types of files are created by running the individual simulation code for each source and geologic layer by hand. The source number and geologic layer number in the file name must be changed by hand in the fortran code itself, to reflect the changing input data. For instance, if data for source 3 in geologic layer 2 is presented to the individual simulation code, the fortran code must be amended such that the output files are renamed 'c\_all\_orig3\_GL2\_L.dat' and 'src\_orig3\_GL2\_L.dat' on lines 328 and 371, respectively of the fortran code. Files are automatically created for all numerical layers. A model with 3 potential sources, 4 geological layers and 6 numerical layers will have  $3 \times 4 \times 6 = 72$  'c\_all' files and 'src\_orig' files.

#### Model-Related Data

coordinates.dat (define the x and y coordinates for each node)



elements.dat (defines which nodes belong to each model element)  
glayer.dat (defines which geological layer each numerical layer belongs to)  
glayer(j).dat (specifies the number of numerical layers belonging to geological layer j)

Water Quality Data

plumereal.dat

samloc.dat

Application Input Files

initial\_weights.dat (lists the initial source weights)

input\_choquet.dat

input\_kalman.dat

opt.dat

simu.dat

flag\_acut.dat

a\_cut\_input.dat

alpha\_cuts.dat