1-30-2017

# Comparing Groundwater Model Calibration Approaches: Does an Optimized Model Better Reflect Reality When Both Flow and Transport Observations are Applied?

Vivian E. Sovinsky
*University of Connecticut*, vivian.sovinsky@uconn.edu

# Comparing Groundwater Model Calibration Approaches: Does an Optimized Model Better Reflect Reality When Both Flow and Transport Observations are Applied?

Vivian E. Sovinsky

B.S., University of Connecticut, 1981

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2017

**APPROVAL PAGE**

Masters of Science Thesis

# Comparing Groundwater Model Calibration Approaches: Does an Optimized Model Better Reflect Reality When Both Flow and Transport Observations are Applied?

Presented by

Vivian E. Sovinsky, B.S.

Major Advisor_____

Amvrossios C. Bagtzoglou

Associate Advisor_____

Jeffrey Starn

Associate Advisor_____

Gary A. Robbins

University of Connecticut

2017

# ACKNOWLEDGEMENT

First, I would like to thank my major advisor Dr. Amvrossios Bagtzoglou for his encouragement, insights and ongoing support over the past three years. I sincerely appreciate his skill in always finding time in his challenging schedule to keep me pointed toward my goal. I learned a great deal from his enjoyable geostats presentations during my independent study. I am also very grateful for the steady support from my committee member Dr. Jeffrey Starn of USGS who guided me during my first exposure to automated groundwater modeling and through the details of applying formal calibration methods. My other committee member, Dr. Gary Robbins drew me initially to the field of groundwater sustainability during stimulating lectures and labs and encouraged me to explore groundwater modeling. All provided essential training, support and encouragement needed for me to complete this work.

I would also like to thank Dr. Zoi Dokou, who joined the department recently and immediately began supporting my work with careful reviews of my briefings and computations. Many fellow students and professors in the department have encouraged me and given me pointers on how to deal with obstacles encountered in research.

Department support and the Graduate School staff have been essential in keeping me on track with the master's process. All my friends and family have supported me throughout my thesis work. Special thanks to several close supporters: my sister Rene, my daughter Marcella and long-time friend Tavita, who heard the most about my challenges and believed in me. My grandchildren Ellie, Lily, Louella, and Scarlett inspired me to study, in order that I might help in some way to create a better future for the next generations.

# Table of Contents

# Abstract

This research compares two approaches to the application of formal calibration methods to groundwater modeling: sequential and combined. There exists research, both theoretical work and field studies, that point to improved estimation of hydraulic parameters when multiple types of observations, flow and transport, are applied simultaneously. There also exist theories about why combined calibration might not be compatible with the differing mathematical basis of flow compared to transport. This research has taken a closer look at the mechanisms at work by comparing these approaches when the input parameters are known. Using stochastic methods instead of field data, the original input parameters are specified. A synthetic heterogeneous K field is simulated using SGeMS Sequential Gaussian Simulation. This K-field, derived zones of porosity and defined boundary conditions comprise the simulation of a large confined aquifer. Two sets of synthetic observations obtained from forward models MODFLOW and MODPATH, heads and travel times, guide the calibration. Applying PEST++, the sequential approach performs two calibrations: a flow calibration using heads followed by a transport calibration using travel times. Both sets of observations are applied simultaneously in a single calibration run in the combined approach. Comparing final parameter estimates of each approach to the initial synthetic reality values shows that better results were achieved for both hydraulic conductivity and porosity with the combined approach. However, the sequential approach performed well with results falling within one standard deviation of the true values.

## Figures

## Tables

# 1 INTRODUCTION

## 1.1 Background

### 1.1.1 Importance of Groundwater Sustainability

It is essential that the quantity and quality of our fresh water resources be conserved, both in the US and across the globe. A large portion of the fresh water needed for public drinking supplies and agriculture in the United States is supplied by groundwater. Groundwater in the US in 2005 supplied 33% of public water supplies, 98% of domestic (well) supplies; 42% of irrigation needs, and 60% of livestock requirements (U.S. Geographical Survey, Groundwater Use in the United States). Groundwater aquifers reside at various depths beneath the surface and may consist of a wide variety of porous materials such as clay, silt, sand, gravel, sandstone, crystalline rock and limestone, or various combinations of porous materials (Bear, 1972).

These naturally occurring geologic structures have void spaces (pore space) that allow water to be stored and to flow. The natural water cycle replenishes aquifers with recharge from rainfall, thus aquifers serve as underground fresh water reservoirs. Formally, the term aquifer implies the structure can supply a useful amount of water under natural conditions (Bear, 1972). Aquifers are not isolated from the surrounding geology; the water flow may be interconnected in various ways: flowing to/from surface water, from soil in the unsaturated zone, to/from intermediate confining units, and withdrawn by man-made features such as wells (Figures 1.1 and 1.2). Historically, groundwater has been considered water of the highest quality as it was protected from pollution that could more easily reach surface waters. However

with increasing population and the resultant expansion of farming and industry, increased contamination travels along many paths and is endangering groundwater supplies.

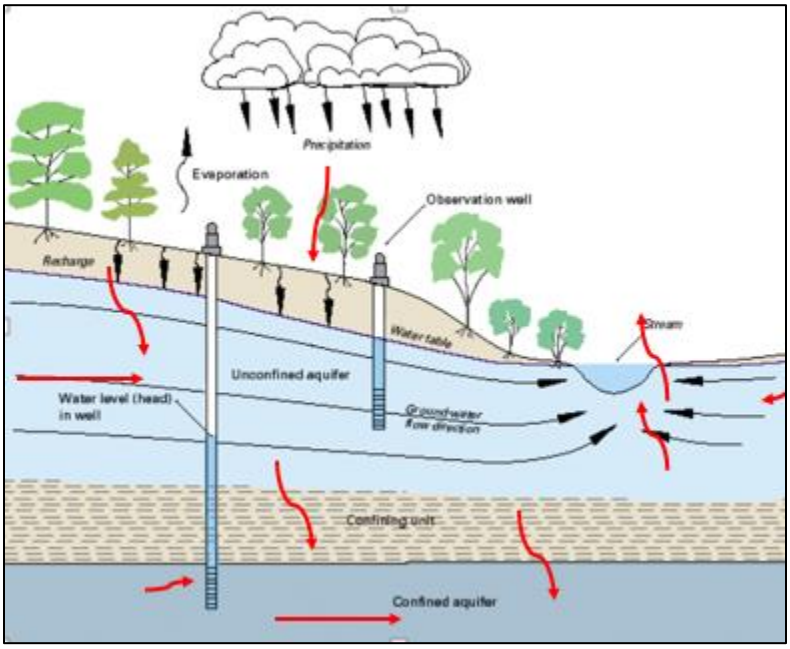



Figure 1.1: Diagram demonstrating the interconnection of groundwater aquifers with surrounding environment. Red arrows indicate possible paths of water and contaminants (modified U.S. Geological Survey, Groundwater Basics).

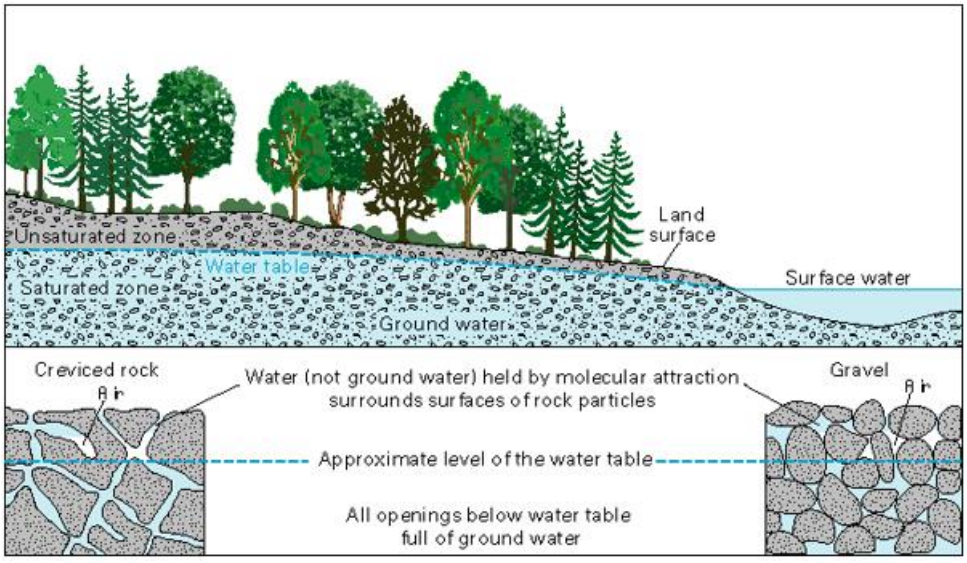


Figure 1.2: Diagram showing makeup of two typical aquifer types. Aquifer boundary/water table is identified by the saturated zone beneath: all void space contains water (U.S. Geological Survey, Groundwater Basics).

One classification of aquifers is based upon the way they interconnect. For example, unconfined aquifers can receive significant flow from several different directions; they are recharged by rain water from above that seeps through the soil, leaching soil chemicals into the unconfined groundwater below, recharging the aquifer but potentially carrying inorganic or organic substances that may cause contamination (Figure 1.1). Confined aquifers have limited flow directly above and below, so somewhat resemble an underground pipe, but one consisting of porous material (Figure 1.1). Confined aquifers may be connected to the environment at intermittent locations or either end; again, potentially receiving needed fresh water recharge, as well as liquid or solid inputs that can cause contamination. Water can also flow to/from aquifers with an apparently solid rock boundary through fractures in the rocks.

Groundwater can be utilized by public or private entities in a variety of ways, both natural and man-made. Well water withdrawal is the method most understood. The well driller creates a hole deep enough to tap into an underground aquifer with sufficient flow to meet a specific pumping requirement (e.g. 5 gallons per minute for a domestic well). Other pathways include the baseflow of groundwater to streams, rivers or surface reservoirs which are then tapped for drinking, irrigation, industrial use or power. Also, the potentiometric surface, the virtual level reflecting the pressure of a confined aquifer, can meet the earth's surface and through an opening to the surface, the hydraulic pressure can produce a spring of water above the ground. However groundwater is obtained, societies need to monitor this valuable resource,

so use does not become misuse or “mining”, withdrawing more water than the natural water cycle can replenish.

Significant reduction in groundwater reserves is already occurring across the U.S. and the globe. Increasing populations and extreme climate variations are major contributing factors. Some areas facing severe droughts or high irrigation demands show reserves down to 40% their prior averages. A huge high plains aquifer, the Ogallala is considered by scientists to be at risk of running dry (A Vanishing Aquifer, 2016--National Geographic Society). This aquifer supplies 30% of the total U.S. demand for irrigation. Even if pumping is reduced, or recharge from precipitation increases, it is not always possible to remedy all the damage that occurs to the underground basin; sometimes the structure begins to collapse due to the decrease of the water pressure that normally maintains it.

Thus, whether the focus is on quantity or quality of groundwater, there are increasing threats to our groundwater reserves. This is why hydrologists and scientists have studied these geologic formations so closely, and are constantly developing new ways to understand, monitor and remediate aquifers, building upon a wide range of field measurement methods and the extensive use of analytics, modeling and statistics.

### 1.1.2 Hydraulic Properties of Aquifers

Specific geologic properties have been identified by scientists as essential for predicting how water will flow through the underground system. This research focuses upon two key properties: hydraulic conductivity K, which can be defined as the ease by which water can flow through the aquifer due to variations in hydraulic pressure, and porosity, the ratio of void space to total space. Within a single geographic area of interest, even within a single aquifer, the geologic composition may vary greatly, causing the values of these properties to change

significantly across the 3D space. If not for this kind of variation or heterogeneity, one could use a uniform field as an approximation for these values. In the typical field study, it is difficult and expensive to obtain values for these properties at a desired sampling density. However, measurements of hydraulic behavior tend to be more readily available, for example, hydraulic heads (i.e., water level in wells) and tracer concentration.

To conserve the quality and quantity of aquifers, it is important to know how much water is flowing (volumetric flow) and the velocity at which the water travels through the interconnected pore space (effective pore velocity). These values assist regional and national planners in a variety of groundwater related tasks, such as approving the placement of new wells, evaluating the path of a contaminant, and planning pollution remediation strategies. But analytically solving for either flow volume or velocity requires knowledge of the underground hydraulic properties mentioned above (hydraulic conductivity and porosity), which are often not available. One could compare trying to understand the internals of an aquifer to gathering data about the internal biological details of an individual human being. Until the advent of modern medical equipment that can scan the body's internals, doctors had to diagnose internal disorders mainly by using external observations. In the field of hydrology, even with the advent of more sophisticated geophysics methods, the large spatial extent of aquifers presents a great challenge.

Therefore, to apply mathematics to analyzing groundwater, a number of workarounds to compensate for the lack of input properties values have been developed. As will be shown in the Methods Section, the geologic properties are inputs required by the equations needed to model an aquifer, while typical outputs of these equations are variables that reflect field measurements. This data availability issue will lead us to the importance of inverse modeling

or calibration, the central tool for this research: the ability to use output measurements to determine input geologic properties.

#### *1.1.2.1 Input parameters*

**Hydraulic Conductivity**

The first property of great interest is hydraulic conductivity (K) which provides an indication of how much water (volume per time) will pass through a unit area of the aquifer, for each unit difference in hydraulic head. K is considered a composite parameter as the value is effected by properties of the fluid and the geology. K can vary greatly with the properties of the porous media, with measured values spanning several orders of magnitude (Bear, 1972).

**Porosity**

Porosity is an additional property required for solving transport equations. Without knowing how much space is available for water to flow, velocities and travel times cannot be computed and the transport of contaminants cannot be analyzed. Also, there is pore/void space within the media which does not allow for flow, i.e. it is not interconnected with other pores. In hydrology, one therefore requires "effective porosity" for transport analysis. Effective porosity is the ratio of interconnected pores space to total space. All future references to porosity in this paper imply effective porosity.

#### *1.1.2.2 Output/Measurements*

**Hydraulic Heads**

These measurements indicate the pressure exerted by the water to create a height of unconfined water above a specified datum, for example, the level of water in a well (Bear, 1972). In confined aquifers, this is also referred to as the piezometric head or surface. In this

case it is a virtual water level representing where the water would rise to if given an outlet; thus the piezometer is a tool used to measure this pressure head.

**Travel Times**

Travel times (also referred to as residence time) are measures of how long it takes a particle to travel from one point to another in the groundwater environment. The term residence time emphasizes that the particle resides in the specified aquifer (or a specified space) for a specified length of time. One could measure the time from its entry into the aquifer to it reaching a particular boundary. In this work we use a travel time that indicates travel from one boundary (the first column) of the aquifer to our observation points.

### 1.1.3 Groundwater Modeling

#### *1.1.3.1 Flow Modeling*

For decades hydrologists have been able to model the flow of groundwater using computer models that divide the underground volume of interest into cells, where each cell is assumed to have constant property values, such as hydraulic conductivity and porosity. Finite difference and finite element methods are typically applied, as these allow for the solution of large numbers of simultaneous equations that are created by mathematically describing the flow to and from each cell. Equations for each cell describe the volume of water passed between cells as a function of hydraulic head; these equations are solved for the head value at each cell. Once heads are determined, cell flows can then be computed. The equations utilized (See Methods Section 2.3.1) require that the hydraulic conductivity K for each cell be specified. MODFLOW is a 3D Groundwater Model; it provides for finite difference models for groundwater flow and includes numerous optional modules, such as modeling interactions with surface water, transport modeling, and groundwater management (Harbaugh, 2005).

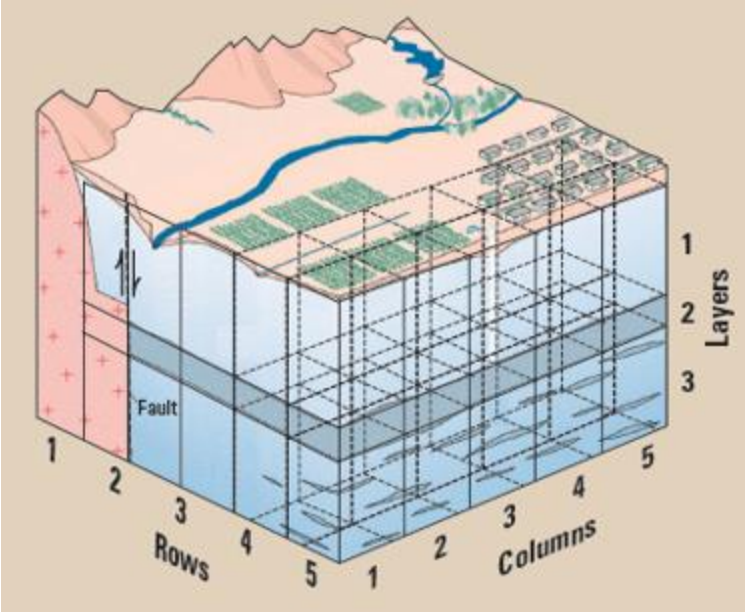



Figure 1.3: Flow modeling is performed by dividing aquifer into layers and 3D cells. Flow equations are applied to each cell creating numerous simultaneous equations to solve, producing results of head gradients between cells and flows across cell faces. Properties are assumed to be constant across each cell. K for each cell must be specified for the equations to be solved.

Various work-arounds have been developed to enable running this model when few K field values are available. Three common approaches are (1) estimating an average K for the entire volume, (2) estimating zones of homogeneous K values using expert knowledge of the field site, and (3) creating statistical simulations to provide a statistical distribution (or combination of distributions) of K. Some of the statistical approaches, like the Sequential Gaussian Simulation used in this work use known spatial relationships of K, and can produce a heterogeneous K field that reflects known spatial continuity (Isaacs and Srivastava, 1989). Another statistical approach, transition probability indicator simulation (TPROGS) has been shown to more closely simulate the geology of high-K facies (Lee et al., 2007). Eggleston et al.

compared multiple stochastic simulations: Sequential Gaussian, simulated annealing, and kriging (Eggleston et al., 1996).

#### *1.1.3.2 Transport Modeling*

Transport modeling can really be thought of as post-processing of the flow model results. These models build upon the output heads and cell budgets (flows) produced by the flow simulation. By adding effective porosity, flows can be transformed into effective pore velocities, and used to determine both pathways and travel times. MODPATH is the particle tracking model designed to work with MODFLOW outputs (Pollock, 2012).For this work we applied a backward simulation of the motion of particles from the observation locations back to the first column of the model. This backward time will be equal to the time for the particle to travel forward with the flow of groundwater (advective transport) from the starting boundary of the model to the center of each observation square. Dispersion was not included in the transport model in order to isolate the effects of porosity; multiple transport processes would have complicated the influence of porosity values.

### 1.1.4 Model Calibration

Model calibration methods were developed to allow the modeler to guess or estimate values of input parameters such as K and porosity, as well as boundary conditions. The estimates are used to run the groundwater model to obtain the model outputs (e.g., heads, and travel times). The model outputs are then compared to actual field measurements. Depending upon the results, the inputs may be further adjusted, to attempt a better match to field observations. This is continued until the closeness of model outputs to field observations satisfies some criteria established by the modeler.

Calibration methods include both manual and automated approaches. Often environmental consulting firms prefer to use manual approaches due to their simplicity. However, the manual method can be time-consuming and may introduce significant error due to the subjectivity of the approach (Hill and Tiedeman, 2007).

Governmental agencies and some firms prefer to use more formal calibration methods. PEST++ is an automated approach to calibration developed by John Doherty at Watermark Numerical Computing (Doherty 2014; Doherty, 2015). PEST++ is related to the original program PEST; PEST++ was designed to be easier to apply than PEST and to address issues with highly parameterized inverse modeling. Automated statistical methods estimate values of input parameters that provide a good match to field observations. Among these methods, non-linear regression, a means to approximate non-linear behavior using derivatives, has become a leading approach and is implemented by a variety of calibration tools.

If the values of the resultant estimates more closely reflect the true aquifer geology that is relevant to the water dynamics, models can become better predictors of future outputs (Hill and Tiedeman, 2007; Doherty, 2015). Some typical outputs of great use to environmental planners are hydraulic heads and travel time or velocity. Hydraulic heads are indicators of levels of water in wells and storage. This value can serve as a key indicator of stress on an aquifer from excessive pumping. Paths and travel times can provide valuable information about the transport of contaminants and the potential sources of those contaminants. Thus, for our models to serve as better predictors of future hydraulic behavior, the calibration process is an important way to refine our models (Hill and Tiedeman, 2007).

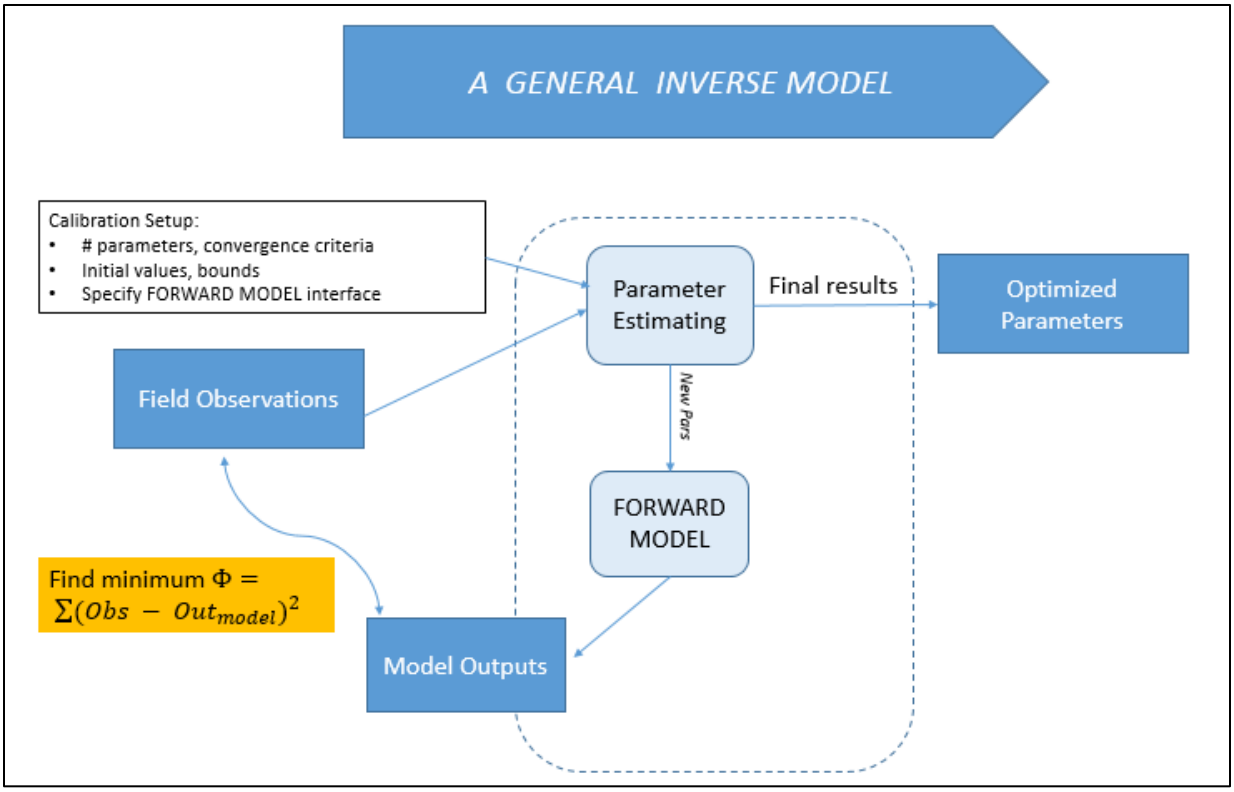



Figure 1.4: Summary of the general automated calibration process showing how outputs (the actual field measurements and the modeled outputs) are compared and guide the estimation of the inputs (geologic parameters) until an optimized set of parameters is found. The forward model is run over and over until the estimated inputs create outputs close to the field observations.

## 1.2 Objective

This study poses the following question: does the optimized reality more closely resemble the true reality when multiple types of observations are applied? To answer this question, this work focuses on comparing two key approaches to automated calibration: (1) the sequential approach: the sequential use of a single type of observation for driving the estimates of each input parameter one at a time and (2) the combined approach: using multiple types of observations together to drive the calibration process. The thesis statement becomes:

**The combined approach to model calibration, applying both flow and transport observations, produces optimized results that better reflect reality.**

This is the hypothesis to potentially either disprove or support with this study's results.

There has been a significant body of research supporting combining observations (Gorelick et al., 1987; Gailey et al., 1991; Sonnenborg et al., 1996; Anderman and Hill, 1999). A groundwater quality management project used a homogeneous-K synthetic reality and a finite element model along with non-linear regression to estimate flow and transport parameters, simultaneously considering deviations of modeled and observed concentrations and hydraulic heads (Gorelick, 1987). Gorelick's approach is then successfully applied to a Gloucester Landfill study in Ottawa, Canada (Gailey, 1991). Both of these studies supported their conclusions with statistical analysis. Gorelick and Gailey both perform a statistical analysis of the parameter estimates. Another field study, this one at a waste residue site, also estimates flow and transport parameters in a combined approach (Sonnenborg, 1996). An early effort by Strecker and Chu raised issues about the sequential approach, stating that the uncertainty caused by solving for K based solely on head data, and then using those K parameters in the subsequent transport model, amplifies the error in transport model outputs (Strecker and Chu, 1986). Many researchers believe that the combined approach is more likely to produce a closer approximation of the hydraulic reality, and therefore a more accurate model, leading to better model predictions.

The above research and additional field studies have influenced the way modeling is taught and is incorporated in a set of guidelines provided by Hill and Tiedeman in their comprehensive text book, Effective Groundwater Model Calibration:

[Guideline] 4. Include many kinds of data as observations (hard data) in the regression.

> Add different kinds of observations; this can be critical to obtaining a reasonably accurate model. In groundwater flow model calibration, it is very important to include information about flows (Hill and Tiedeman, 2007 p. 261).

Applying the sequential approach to modeling flow does not utilize additional information about transport which could potentially improve the model's knowledge of the site's overall hydraulic behavior. With the combined approach both types of observations are included in the estimation process. The objective function, Phi, used to quantify the closeness of model results to observations, becomes a combined weighted sum of the squared residuals of both heads and travel times. All observations are combined into a single estimation process.

In order to add another perspective to these investigations, this research uses a stochastic K input reality as a foundation. This work simulates an aquifer with a heterogeneous two dimensional K-field and a large spatial expanse, typical of groundwater modeling projects. Through simulating the original reality, the project has access to the actual K and porosity values – enabling a comparison of the resulting estimates of the two calibration approaches to the true "reality" values. If instead a field study were chosen there would be no way to know the actual input parameters to this level of detail. So use of a simulated reality facilitates quantitative comparison of the sequential and combined approaches.

While there are both field research and simulation studies to support the thesis statement, questions have been raised (C. Voss, Editor's message, Hydrogeology Journal, 2011). Voss states that the differences in the analytical basis of groundwater flow and groundwater transport make the combining of these sets of observations questionable. Voss believes a flow calibration using only head measurements will consider or query more of the aquifer than the transport calibration. Two factors contribute to this: (1) the observations of flow tend to be distributed in both low K and high K areas, and (2) the flow equation resembles

the diffusion equation. Models addressing transport calibration generally obtain most of the observations in high K channels (Voss, 2011a).

## 1.3 Scope of Work

This work sets up a clearly specified initial "synthetic reality" using geostatistics to produce a heterogeneous K field and using a relation of porosity to K to create realistic porosity values for the transport model (See equations in Methods). There are two major sets of calibration tests performed (Table 1.2). One estimates a total of four parameters (3 K and 1 porosity) and one estimates 7 parameters (3K and 4 porosities). In order to create the observations for the calibration tests, forward model runs of the flow and transport models are performed (Table 1.1).

Table 1.1: Forward Model Runs to Create Observations (MODFLOW, MODPATH).

| MODFLOW | MODPATH |
|---|---|
| Create Head Observations for K Reality A | Create travel time observations for K Reality A and single porosity reality |
| | Create travel time observations for K Reality A and 4 porosity reality |

Table 1.2: Reverse Model Runs to Estimate Input Parameters (PEST++, MODFLOW & MODPATH).

| Model Set #1: 4 Estimated Parameters | Model Set #2: 7 Estimated Parameters |
|---|---|
| Sequential Runs | |
| • Sequential Flow (3 Ks) | • Sequential Flow (3 Ks) |
| • Sequential Transport ( 1 P) | • Sequential Transport ( 4 Ps) |
| Combined Runs | |
| • Combined Flow & Transport (3Ks and 1P) | • Combined Flow & Transport (3Ks and 4Ps) |

## 1.4 Organization

Section 1 Introduction provides some basic background on the importance of groundwater sustainability and an introduction to groundwater modeling. It also provides a research context for the thesis. The methods utilized to accomplish the study are detailed in Section 2 Methodology. Since field data were not utilized, Section 2 includes an explanation of the creation of the synthetic reality that is used to drive the initial forward model runs of MODFLOW and MODPATH, generating simulated field observations. This section also includes the governing equations for the various models and methods applied in the course of this work. The way the computer runs were organized and the various dependencies among the runs are explained both in diagrams and text. Some interim data results are included in the Methods section to aid in envisioning the process. The final results of the various runs are presented in Section 4 Results. Section 5 Conclusion compares the results to the thesis statement and provides possible rationale for them. It also discusses proposals for extending this work.

# 2 METHODOLOGY

## 2.1 Modeling Software and Hardware

This section lists the main software components and computing resources employed to perform this research. The basic description and version are provided here. More details of download links and dates may be found in Appendix B.

Table 2.1: Software applications utilized for research. All provided in the public domain.

| Application | Description | Version |
|---|---|---|
| SGeMS | Stanford Geostatistical Modeling Software | V2.5b |
| MODFLOW | USGS's 3-D Finite Difference Groundwater Model | 1.0.9 |
| MODPATH | USGS's particle-tracking post-processing program that uses MODFLOW output files to perform transport calculations. | Modpath.6_0 |
| Pest++ | Watermark Numerical Computing's Parameter Estimation (Calibration Program) – author John Doherty | 3.5+fixes |

Table 2.2: Scientific computing language software and libraries utilized for research. Most are provided in the public domain. Enthought provides a free academic license.

| Programming Languages & Libraries | Description | Version |
|---|---|---|
| Python | Anaconda for Jupyter notebooks. Also includes a script editor named Spyder, which would remove the need for Enthought. | Version 3.5.1 |
| Python | Enthought Canopy for viewing/editing python scripts | Version 1.5.1 |
| FloPy | Python library to create, run, and post-process MODFLOW-based models using a programming interface. | Version 3.2.5 |
| NumPy | Python Library for array operations and manipulation | Version 1.11 |
| Panda | Python Library for data analysis | ~Version 0.19.0 |

Table 2.3: Hardware/OS Utilized for Research.

| Hardware/OS | Description | Version |
|---|---|---|
| Prostar Laptop High Performance Computer | Gaming-style computer custom made by Prostar: PRP150<br>16GB RAM<br>Solid state C Drive<br>Separate E Drive | PRP150SM-A-R-B<br>P150SM-A, 15.6”<br>FHD/MATT |
| Intel Core i7-4810MQ CPU | 4-processor CPU | 2.8 |
| Windows 7.0 Professional OS | 64-bit | Service Pack 1 |

The multi-processor CPU was used to accelerate the execution of calibration software. Using a master and 3-slave setup, the execution time was cut into one third. PEST and PEST++ Version 3 both provide a feature called “Yet Another Run Manager” (YAMR), which can be invoked through a windows command. A short windows command file is required to specify directory names and files that need to be copied into each of the four directories. The final results of the calibration run are found in the master file. The main PEST++ process runs in the master directory which delegates runs of the models (e.g., MODFLOW) in each of the slave directories, so there can be three simultaneous executions of MODFLOW—each of which passes their results back to the main PEST++ process running in the master directory

## 2.2 Creating Synthetic Reality

### 2.2.1 Create Synthetic K-field

The Stanford Geostatistical Modeling Software (SGeMs) was applied to utilize geostatistical principles in generating the synthetic reality for a 2D hydraulic conductivity (K) field. K values in the field often demonstrate a natural log normal (ln) distribution across a

spatial domain. SGeMS offers a number of geostatistical simulations. The simulation chosen for this work was the Sequential Gaussian Simulation (SGS), as it creates a log normal distribution while maximizing entropy and conforming to a pre-selected variogram. LnK is considered to vary spatially, that is, the lnK of locations near to each other are considered to be closer in value than those that are farther apart. A maximum is reached at a particular distance (range). To capture the spatial variation of lnK, a variogram was created, which presents variance as a function of lag distance, or distance between pairs of lnK values.

The following were chosen to represent the geostatistical parameters that would specify the variogram and additional statistics required to complete the simulation. A Gaussian distribution is considered a good match to hydraulic conductivity spatial behavior in the field, as is a nugget of zero. The geostatistical simulation produced a standard normal distribution which was then back-transformed using a mean of 5 (lnK-K in m/d) and a variance of 4 lnk-K in m/d). These values were selected to create a wide range of K values, and to ensure some of the Ks were high. The mean converted to K is $e^5$ or 148 m/d.

Table 2.4: Variogram and Simulation Parameters.

| Parameter name and symbol | Description | Value |
|---|---|---|
| Range $a$ | distance where variogram is 95% of sill value | 200m |
| Variance(sill s) or $\sigma^2$ | variance at lags > range | 4 (lnK-K in m/d) |
| Mean μ | mean of simulated lnK's | 5 (lnK-K in m/d) |
| Nugget n or $C_o$ | variance at lag of 0 | 0 |
| Variogram Curve | Choices: Exponential, Gaussian, Spherical | Gaussian |

The Gaussian variogram therefore follows this equation (Isaaks and Srivastava, 1989):

$$\gamma(h) = C_o + \sigma^2 \left(1 - \exp\left(-\frac{3h^2}{a^2}\right)\right) \quad (1)$$

Where $h$ is the lag distance, $a$ is the practical range, $\sigma^2$is the variance and $C_o$is the nugget.

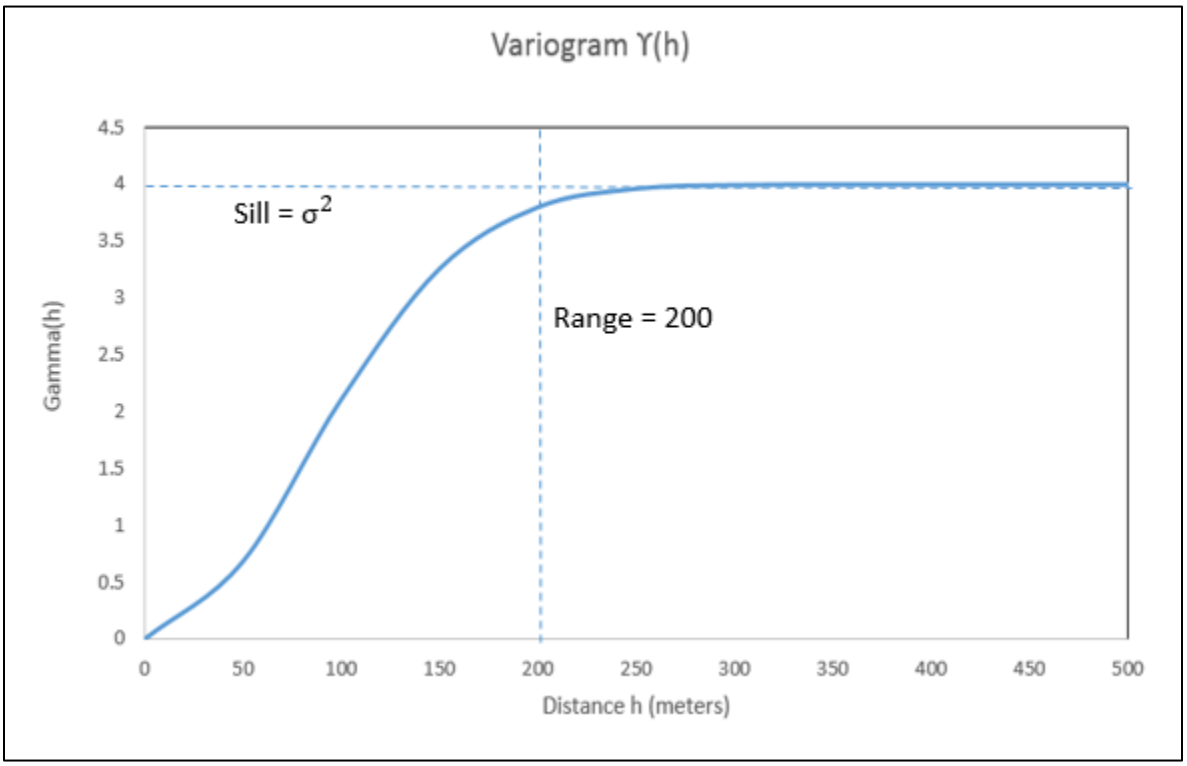



Figure 2.1: Variogram for spatial relationship among ln K—hydraulic conductivity values.

The Sequential Gaussian Simulation (SGS) generated a 500m by 1000m field consisting of 1m by 1m cells. The stochastic K-field captures the spatial relationship among lnK values by accepting the variogram parameters described above as input. The size of the field was chosen in order to resemble the extent of an aquifer/aquifer portion that is typically modeled in practice.

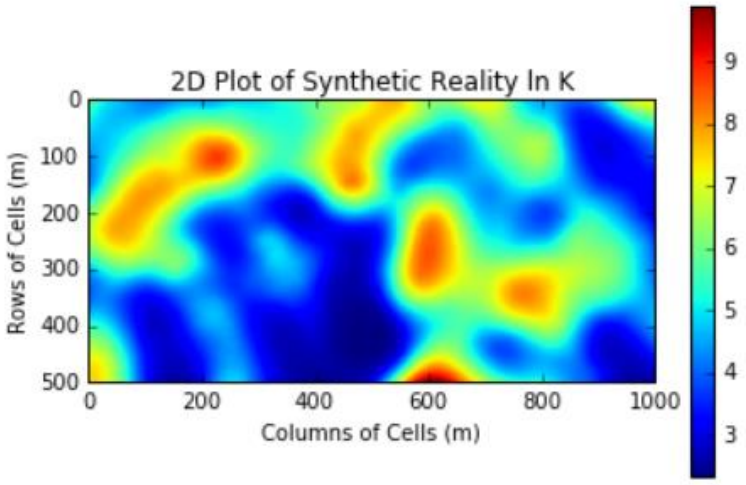



Figure 2.2: Output of SGeMS Sequential Gaussian Simulation for Reality A.

The simulation of K within SGeMS produces multiple equi-probable realizations. For this work, 10 realizations were specified and one was chosen as the lnK distribution for the synthetic reality. The initial output created by SGeMS was a set of 10 standard normal distribution realizations. This was achieved by setting the sill of the variogram to the value of 1(Bohling, 2007). This approach allows more flexible post-processing of the simulation outputs; the outputs can then be back-transformed to a chosen mean and standard deviation by a custom python script executed within SGeMS. The mean used was lnK=5 and the standard deviation was lnK = 2. This script also converted lnK values back to K. The final Ks were then exported and named Sim A.

### 2.2.2 Establish Boundary Conditions

To fully describe our synthetic aquifer, boundary conditions are required to satisfy the governing equations which are the basis of MODFLOW (Harbaugh, 2005). MODFLOW requires, in addition to values of K for each cell, any boundary conditions such as no-flow boundaries, specified heads or flows, and any fluxes into or out of cells. For this model a

confined aquifer with a depth of 30 meters was specified. To the left and right constant head boundaries were defined, in order to create flow from left to right. The upper and lower boundaries are defined as no-flow boundaries and the top and bottom of the single layer model is automatically no-flow due to being defined as confined.

The exact specifications used for the MODFLOW model are provided in Appendix A.

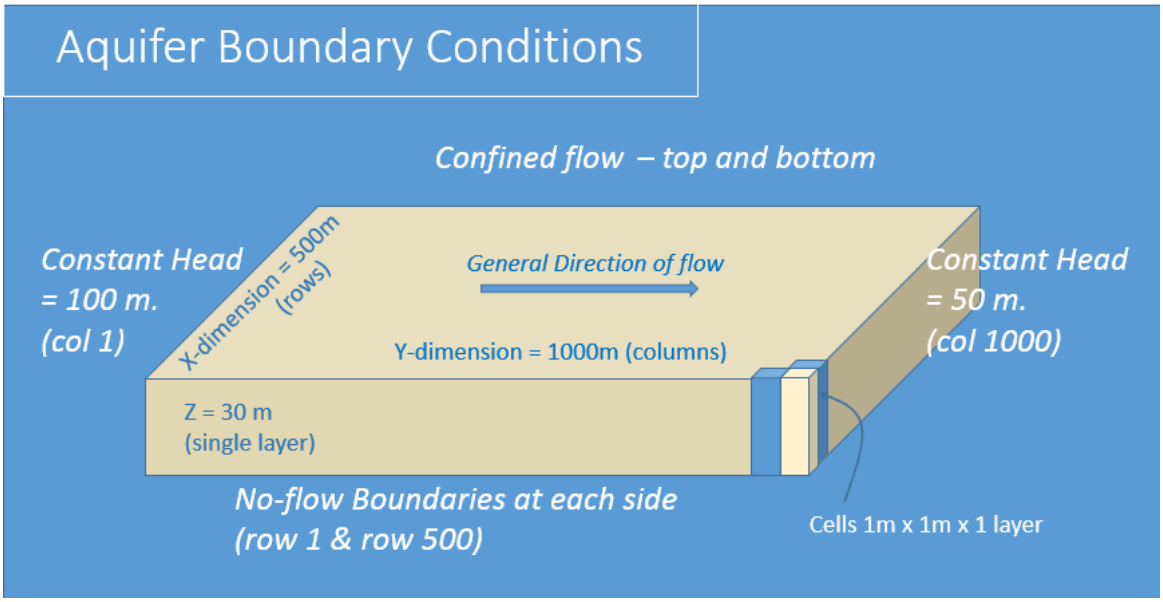



Figure 2.3: Boundary Conditions for Aquifer Simulation.

### 2.2.3 Assumptions and Limitations

This section will mention some inherent assumptions or limitations to the chosen research approach, some resulting from creating a fictional model to represent a field situation and some resulting from the limited number of tests performed:

1) Firm boundary conditions are specified. These boundary conditions often do not exist, or are not precisely known at actual sites. The lack of knowing exact fluxes of water at boundaries in field situations often complicates the estimation of input parameters,

since modelers often still simulate the boundaries as either no-flow or an estimated specified head or flow.

2) The granularity of the cells is sufficient to capture the aquifer behavior without introducing an amount of model structural error (due to assuming constancy across cells) that would yield invalid results.
3) A single simulated reality was designed – so results could be specific to this simulation.
4) There are 200 equally spaced observations. Sampling density = 1 sample per 2500 $m^2$. Often in field situations there exist areas that are difficult to sample thus providing an uneven sampling, and typically less samples are provided.
5) The number of observations outnumbers the number of parameters that are estimated. It is not uncommon to need to estimate more parameters than observations. This is referred to in the literature as a "highly parameterized" inverse problem (Hill and Tiedeman, 2007). This calibration may behave better due to estimating fewer parameters.
6) Some research (Lee et al., 2007; Eggleston et al., 1996) considers the Sequential Gaussian Simulation for a K field to be an unlikely representation of a truly existing heterogeneous K-reality. Lee et al. suggest that the simulation does not produce sufficient connection among high-K values which are typically found in heterogeneous aquifers. Thus, there are significant efforts being pursued to create more geologically accurate models of K (Lee et al., 2007; Eggleston et al., 1996).

### 2.2.4 Create Synthetic Porosity field

The effective porosity was simulated in two ways. For the first set of tests, a single porosity value of 0.2 was assumed to be valid for the entire grid. For the second set of tests,

four zones of porosity were created. These four zones were derived by first assuming the porosity was proportional to lnK. A realistic range of porosities (0.05 to 0.30) was mapped to the range of lnK that existed in the data (Figure 2.4), resulting in the following equation for a line:

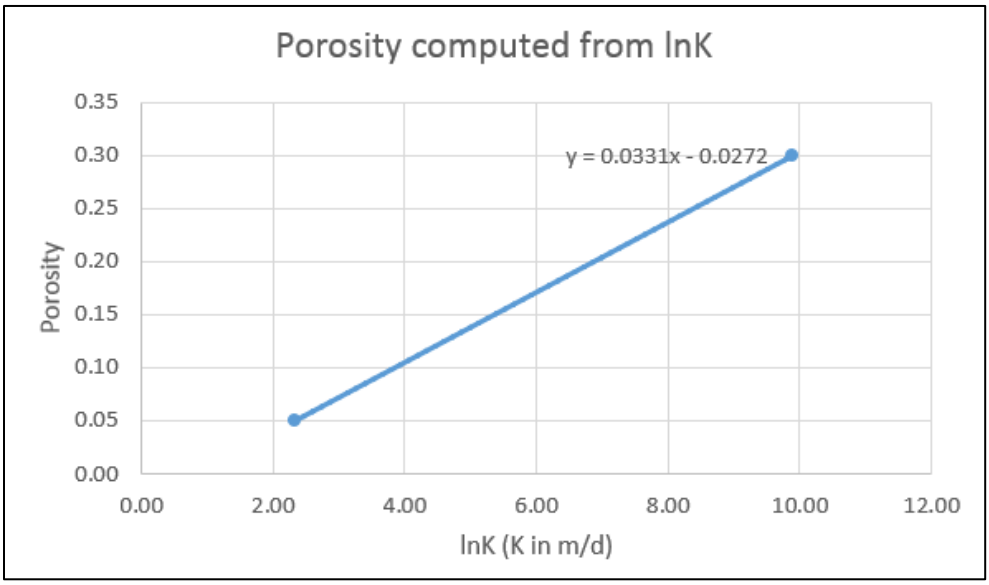



Figure 2.4: Porosity is assumed to be proportional to lnK and a chosen range of porosities is mapped to the range of lnK values in the synthetic reality.

After converting the K field to a porosity field using this equation; the same four zones representing ranges of K, were then used to divide the porosity field into four zones. The average porosity of each zone was computed and then plugged back into all the cells of the zone. The resulting porosity field therefore had four values: P1 = 0.0730, P2 = 0.1170, P3 = 0.1600. P4 = 0.2120 (Results Table 3.3) – each zone average filling one of the four zones which are based on ranges of K values.

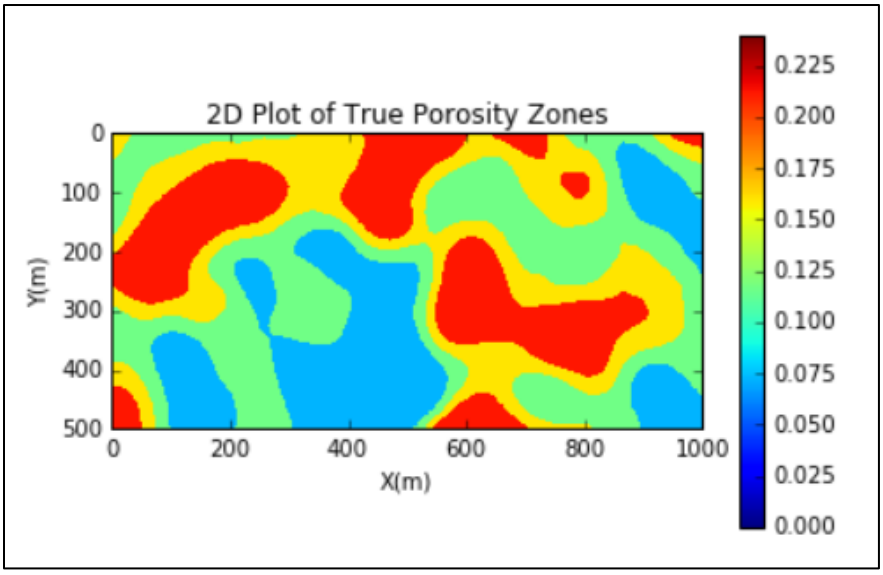



Figure 2.5: Four uniform porosity zones populated by average porosity values.

## 2.3 Creating Observations (Forward Modeling)

### 2.3.1 Governing Equations for Groundwater Flow

To describe the flow of groundwater in 3-D space analytically, the mass balance equation is combined with Darcy's Law. The groundwater flow caused by hydraulic head gradients is described by convection; water moves in a smooth path from higher to lower hydraulic head.

$$Q = -KA\frac{\partial h}{\partial l} \qquad q = \frac{Q}{A} = -K\frac{\partial h}{\partial l} \tag{2}$$

$$\frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y) + \frac{\partial}{\partial z}(q_z) = W \tag{3}$$

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial x}\left(K_{zz}\frac{\partial h}{\partial z}\right) = W \tag{4}$$

where $K_{xx}$, $K_{yy}$, $K_{zz}$ are hydraulic conductivity along x,y,z axis (assumed parallel to the principal axes of the K tensor) [L/T];$h$ is the potentiometric head [L]; $W$ is volumetric flux per unit volume of sources/sinks of water [$T^{-1}$]; $W$ consists of any recharge, withdrawals, flows into or out of an aquifer.

Assumptions:

1) No sources or sinks. $W = 0$
2) Steady-state flow.
3) Coordinate axes are aligned to K principal axes
4) 2-D Model: No gradient for head in z-direction. $\frac{\partial h}{\partial z} = 0$
5) $K_{yy} = K_{xx}$

The simplified equation for 2-D steady-state flow is:

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right) = 0 \qquad (5)$$

And since $K_{yy} = K_{xx}$ this becomes

$$\frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial h}{\partial y}\right) = 0 \qquad (6)$$

The finite difference approach used by MODFLOW looks at the incremental flows across model cells. The 3D model space, in this work a single layer with finite thickness, is divided into multiple rectangular cells and finite-difference methods are used to solve this equation in MODFLOW (Harbaugh, 2005). For each cell, discrete versions of this equation are formed to compute the delta head between cells. All the equations are then solved simultaneously by MODFLOW. Once heads (*h* in Equation 6) are determined, the cell budgets or flows at each cell face can be computed using Darcy's Law.

As discussed in the Introduction, hydraulic parameters are properties that effect transport in groundwater. In particular, hydraulic conductivity K, may vary significantly (many orders of magnitude) with the varying geologic composition of the aquifer (heterogeneity). Thus, for some studies, capturing this variation will be needed for sufficiently accurate results versus making simplifying assumptions.

### 2.3.2 Creating Flow Model Observations: Heads

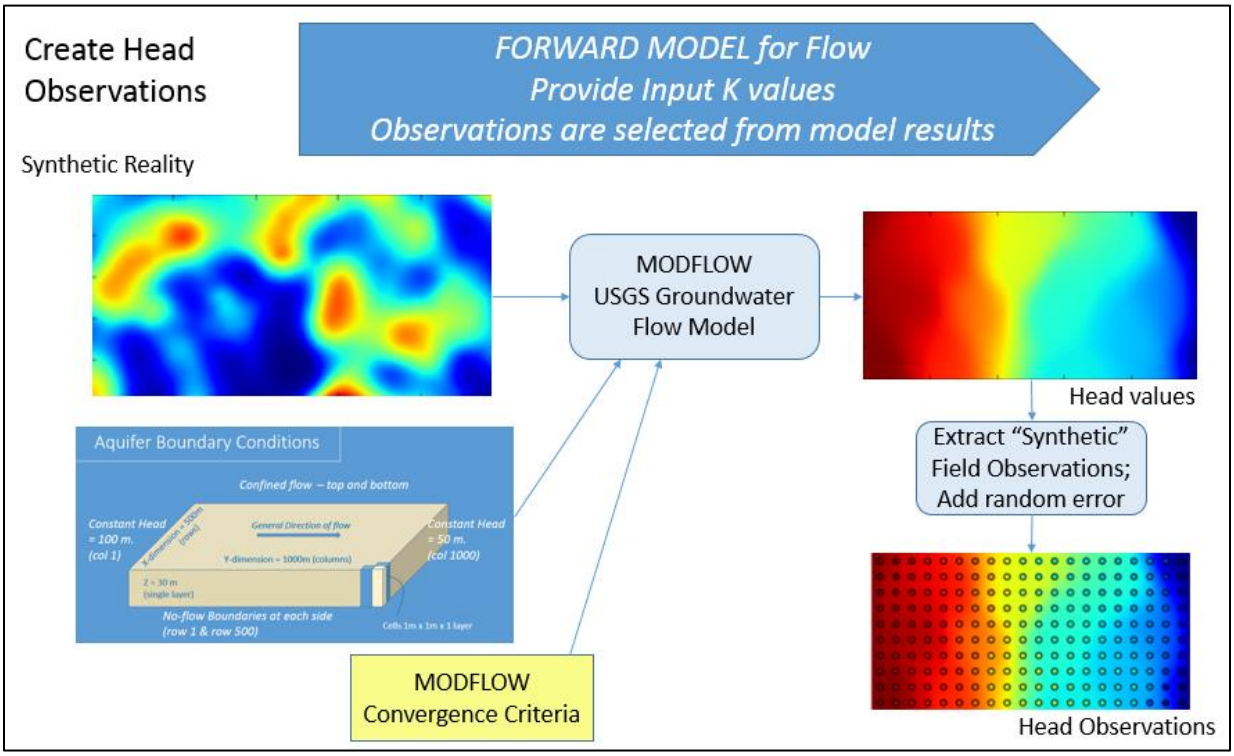



Figure 2.6: Creating synthetic head observations using synthetic reality and MODFLOW; python code (using *FloPy* subroutines) is used to configure and execute the MODFLOW model and to extract the head observations from the results.

The key process steps followed to create observations representing the synthetic reality are shown below. All these steps are performed using Python code within a single Jupyter notebook:

1) Import K-field from SGeMS export.
2) Describe the synthetic model and MODFLOW run parameters by making calls to various *FloPy* routines, a library for interfacing with MODFLOW via python. See Modflow *FloPy* calls in Appendix A. Specify boundary conditions, model dimensions, K values for cells, and convergence criteria for MODFLOW.

3) Run MODFLOW. Output is the complete set of 500 by 1000 hydraulic heads – the head for each cell and the cell budgets (flows through cell faces).

   NOTE: It was preferable that MODFLOW execution time be reasonable (1-2 minutes). Otherwise the time to run the inverse model could be very long and expand the time to complete this work.

4) Select equally spaced observations, in the center of each 50m by 50m square (2500$m^2$ sampling density), creating 200 observations.

5) Add random error to the observations to simulate measurement error. Used a normal distribution with stddev of 0.5 m, mean of 0 m. Use of higher stddev values for heads caused a lack of convergence later in the inverse modeling process.

> NOTE: Since the range of head values is 50 m, 0.5 m represents 1% of the total range. Using a normal distribution, most values will fall within 3 stddevs of the mean, so errors will vary from -3% to +3% of the range. Since field measurements for heads do not tend to have errors more than 0.3 m, this choice of error magnitude was considered more than sufficient.

6) Write these final observation values to a text output file so they can be easily copied into PEST++ control file.

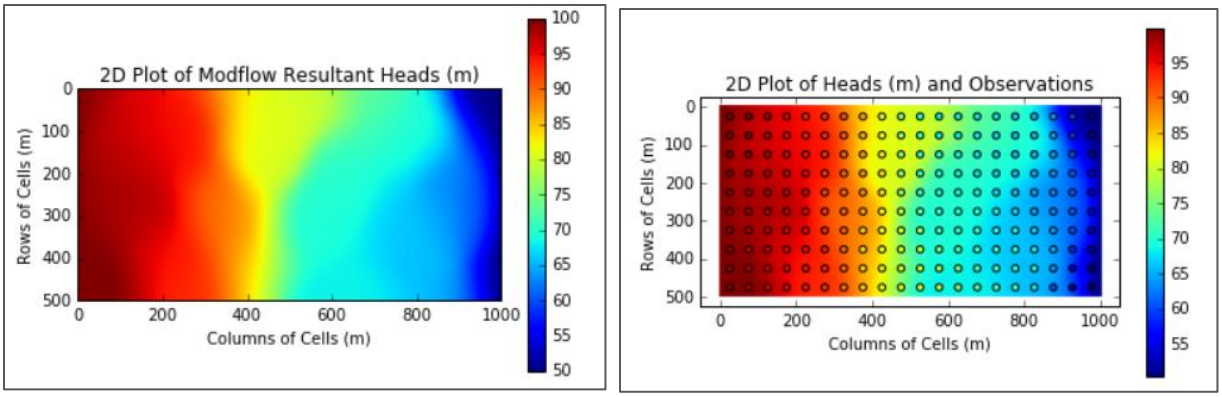



Figure 2.7: Resultant head field & synthetic observations created by the MODFLOW forward model run executed and plotted within a python notebook.

### 2.3.3 Governing Equations for Groundwater Advective Transport

The analytical basis for advective transport uses the same starting point as the governing equations for flow: mass balance and Darcy's law. See Equations 2 and 3 above. The flow/area is replaced with a velocity expression that includes porosity.

Effective porosity or $n$ = (interconnected pore volume)/ (total volume) (7)

Porosity is needed to compute velocity as water can only travel through available pore space. The higher the porosity, the lower the velocity as more volumetric flow can travel through a cell when more space is available.

$$v = \frac{Q}{nA} = \frac{q}{n} \text{ and } q = nv \qquad (8) \text{ \& } (9)$$

The governing equation is therefore:

$$\frac{\partial}{\partial x}(nv_x) + \frac{\partial}{\partial y}(nv_y) + \frac{\partial}{\partial z}(nv_z) = W \qquad (10)$$

1) No sources or sinks. $W = 0$
2) Steady-state flow.
3) No retardation.
4) No dispersion.
5) Advective transport only.

6) 2D model; no gradient of velocity in z direction

So the equation used is:

$$\frac{\partial}{\partial x}(nv_x) + \frac{\partial}{\partial y}(nv_y) = 0 \quad (11)$$

### 2.3.4 Creating Transport Model Observations: Travel Times

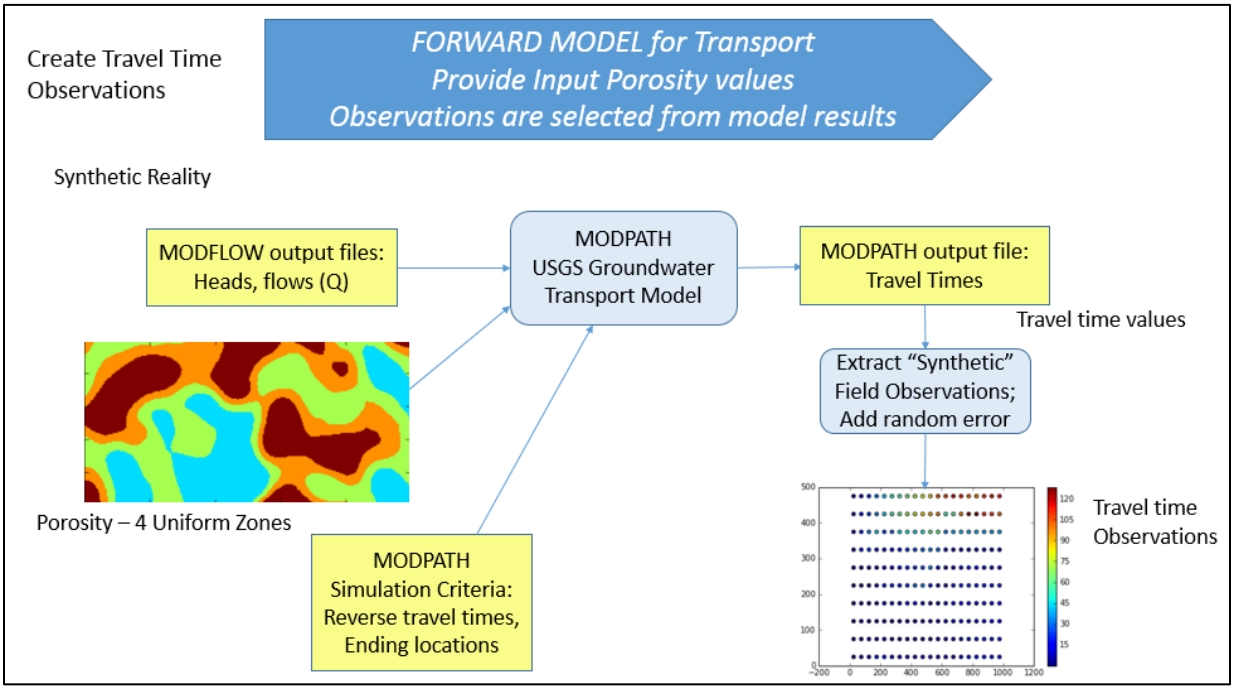



Figure 2.8: Create synthetic travel time observations using outputs of MODFLOW, synthetic porosity reality and MODPATH; python code (using *FloPy* subroutines) is used to create the inputs for MODPATH and to specify what travel times to compute.

The key process steps followed to create transport observations representing the synthetic porosity reality are shown below. All these steps are performed within a single Jupyter notebook:

NOTE: Steps 1-3 are repeated from flow model runs – due to some issues around separating MODPATH from MODFLOW for this run. Ideally one should not need to repeat MODFLOW run here. Using the output files from MODFLOW should be sufficient.

1) Import 2D K-field from SGeMS export.
2) Describe the synthetic model and MODFLOW run parameters by making calls to various *FloPy* subroutines. See Appendix A.
3) Run MODFLOW using "reality" parameters. Output is the complete set of 500 by 1000 hydraulic heads – ending head for each cell, and the cell budgets (volumetric flow through each face).
4) Describe the desired simulation for MODPATH: using *FloPy* subroutines for interfacing to MODPATH, the transport simulation is setup. For this study the endpoint simulation was chosen. MODPATH was directed to track the backward travel of 200 particles, one particle from each location where a head observation was created. Since particles would be traveling from the leftmost (high head) boundary to each observation location it was more convenient to specify the ending points (observation coordinates) than the starting locations. The porosity values were specified by modifying the *filename.mpbas* MODPATH file. Additional MODPATH settings provide names of the following output files from MODFLOW:
    a. Cell budgets
    b. Heads
    c. Discretization: numbers of layers, rows and columns (1 layer, 500 rows, 1000 columns)

5) Run MODPATH. Output will be a set data for 200 particles that travel from the high head boundary through the cells and end up at each observation location.
6) Extract the 200 travel times from the data using subroutines from the *Pandas* library (See Appendix B).
7) Add random error to the observations to simulate measurement error. Used normal distribution with stddev of 2 days, mean of 0 days.

> NOTE: Since the range of travel time values is about 100 days, 2 days represents 2% of the total range. Using a normal distribution, most values will fall within 3 stddevs of the mean, so errors will vary from -6% to +6% of the range. While field measurements for travel times are known to have errors up to several years, due to the range of the data (a result of using the high end of K values), this choice of error magnitude was considered more than sufficient. Additional runs initiated recently using lower K's have travel time ranges up to 120 years and therefore an error of 2% representing 2.62 years is used.

8) Write these travel times to an output file so they can be easily copied into a PEST++ control file.

## 2.4 Calibration: Estimating Input Parameters (Inverse Modeling)

### 2.4.1 Calibration Goals, Techniques and Equations

The calibration runs form the central focus of this research work. The earlier processes are needed to establish the inputs to the calibration runs, most importantly, the observations. In

typical field work you would not have many measurements for hydrologic properties such as K or porosity. You might have some approximation of boundary conditions. And you may have other types of expert knowledge of the site such as likely upper and lower boundaries of various properties like K, and porosity, heads, residence/travel times, etc. However, the gathering of more accessible measurements such as heads, travel times, and contaminant concentrations is far less complex and less expensive than gathering hydraulic properties based upon highly-varying geology. Calibration tools enable scientists to work backward from these observations to estimate the values of the input hydraulic properties.

The tool PEST++ applies non-linear regression and supporting algorithms, for example, Gauss-Marquardt-Levenberg (GML) capabilities. Marquardt-Levenberg refers to the Marquardt Lambda approach described below in "Basic steps followed by PEST++". Regularization methods are provided to make models exhibit better behavior. The PEST++ tool repeatedly estimates new parameter values and judges the closeness of the fit of model outputs to the observations (Doherty, 2015). PEST++ provides the mathematical tools to create the next "educated guess" for input parameters. In order to quantitatively judge how decent the parameter estimates are, the forward groundwater or transport model (or both) must be re-run with each new set of model inputs to obtain model results to compare to the real observation measurements – in this study, the simulated "true observations".

As PEST++ reruns the algorithms and the appropriate forward model(s), it is looking for the minimum value of a specific objective function Phi that compares the model results to true observations: this function consists of a sum of the squared residuals between modeled results and "real" observations. For example, for the calibration of the flow model, the following equation represents the objective function:

$$\Phi = \sum(h_{orig} - h_{model})^2 \qquad (12)$$

Where Φ represents the objective function Phi, $h_{orig}$ is the original head observation, and $h_{model}$ is the model's head output.

The following summarizes the basic steps followed by PEST++ to preform non-linear regression:

1) Prior to starting any iterations: Computes Jacobian Matrix: This is a matrix that captures how model results will change as parameters are changed, for every observation and every parameter combination. The Jacobian contains the sensitivities of each model output to changes in each parameter (Doherty 2015). In order to compute this matrix many runs of the associated groundwater model are performed.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial z_1}{\partial k_1} & \frac{\partial z_1}{\partial k_2} & \cdot & \frac{\partial z_1}{\partial k_m} \\ \frac{\partial z_2}{\partial k_1} & \frac{\partial z_2}{\partial k_2} & \cdot & \frac{\partial z_2}{\partial k_m} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial z_n}{\partial k_1} & \frac{\partial z_n}{\partial k_2} & \cdot & \frac{\partial z_n}{\partial k_m} \end{bmatrix}$$

Where:

Each model output $o_i = z_i[k]$; i from 1 to n for n outputs

And each parameter $k_{j;}$ j from 1 to m for m parameters (Doherty, 2015)

PEST++ applies this multi-dimensional information to determine how the objective function will change as the parameters change. See Figure 2.9 for a 2 parameter (2D) example.

2) At each iteration: Computes vectors to traverse the multi-dimensional objective function using the Marquardt Lambda algorithm which adds numerical efficiency to earlier approaches (Doherty, 2015). In a single PEST++ iteration, multiple vectors are computed based on varying the parameter lambda ($\lambda$); each vector represents a different set of parameter values. PEST++ invokes the appropriate groundwater model(s) to run for each set of parameters and then chooses the result which has the lowest Phi. Figure 2.9 provides a visual showing the single vector chosen with the first iteration. The runs of the groundwater model required for a single iteration is equal to the number of parameters plus 1, thus if estimating 7 parameters, 8 runs are required. Therefore, the execution time of the models needs to be kept reasonable.
3) At each iteration: this best next set of values (determined by identifying the best vector) and the corresponding Phi are compared with the results of earlier iterations to see if convergence has occurred. In this work two main approaches to convergence were used:
    a) Phi convergence – the objective function no longer changes beyond a specified interval.
    b) Parameter Convergence – the parameters no longer change beyond a specified interval.

    When the chosen convergence criteria have been met, the calibration is complete.

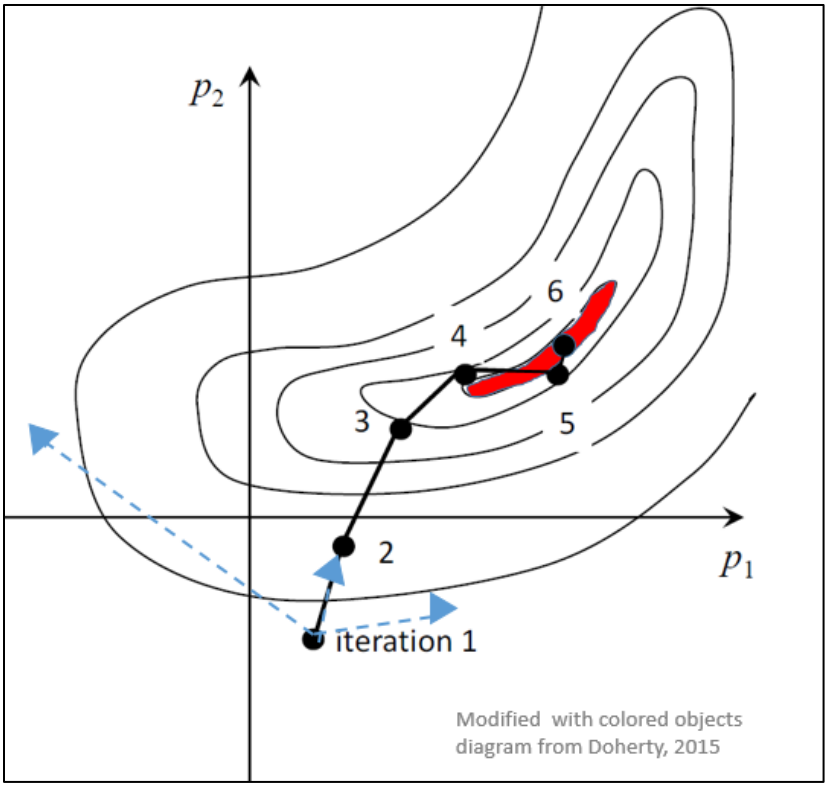



Figure 2.9: Contour diagram of objective function as a function of 2 parameters of the same type, showing vectors computed automatically using Marquardt Lambda, and how PEST++ progresses at each iteration until parameters result in the minimum objective function value.

This calibration approach was applied in two different ways: the sequential approach and the combined approach, which are introduced below:

#### *2.4.1.1 Introduction to the Sequential Calibration Approach*

In the first method, the sequential approach, the calibration is performed first on the flow model, using only head observations to estimate values of K. Then the outputs of that model are used to set up the transport calibration (Figure 2.11). The outputs from step 1 needed to run the transport model include:

- MODFLOW heads (*filename*.hds)

- MODFLOW volumetric flows or cell budgets (*filename*.cbc)

The transport model calibration is then run using only the travel/residence time observations. It estimates porosity values.

Each of these two calibrations uses the appropriate objective function to know when convergence is occurring: Equation 13 for the flow calibration and Equation 14 for the transport calibration. The first compares head observations to modeled heads, and the second compares travel time observations to modeled travel times.

$$\Phi_h = \sum(h_{orig} - h_{model})^2 \quad (13)$$

$$\Phi_t = \sum(t_{orig} - t_{model})^2 \quad (14)$$

The alternative (combined) approach discussed below uses a single combined calibration.

#### *2.4.1.2 Introduction to the Combined Calibration Approach*

This approach combined the application of multiple sets of different types of observations; in this work that means using both heads and travel times observations. The combined approach is achieved by creating a multi-objective function according to Equation 15 (Hill and Tiedeman, 2007). The contributions to the objective functions are weighted, so that when the calibration is run, the contributions to the objective function will be nearly equal at the end of the calibration. This generally involves trial and error. It helps to start with weighting values that make the objective function contributions nearly equal at the start of calibration. Then one keeps adjusting the weights until the final objective function has nearly equal contributions from heads and travel times.

$$\Phi = S(b) = \sum_{i=1}^{NH} \omega_h \ [h_{orig} - h_{model}(b)]^2 + \sum_{j=1}^{NT} \omega_t [t_{orig} - t_{model}(b)]^2 \quad (15)$$

Where b represents a vector of the independent variables – the parameters which determine the model results; $\omega_h$ is the weight applied to head observations, $\omega_t$ is the weight applied to travel time observations, and NH and NT are the number of head and travel time observations, respectively.

NOTE: Weight can be made to vary with each residuals term, but in this work we used a single weight for heads and a single weight for travel_times. This is because we are using the weights to balance the overall influence of head errors and travel time errors. Therefore the weight term could be shown outside both of the summation terms

See the following sections: 2.4.2- 2.4.3 regarding how these two approaches were performed.

### 2.4.2 Sequential Flow and Transport Calibration

The sequential calibration approach was accomplished by performing two calibration runs in a row: (1) a calibration to estimate K using head observations, and then (2) a calibration to estimate porosity using travel time observations.

#### *2.4.2.1 Sequential Flow Calibration*

The first calibration uses PEST++ integrated with the MODFLOW model as shown in Figure 2.11. The "true" head observations created by the forward flow model (Section 2.3.2) are specified for PEST ++ in the PEST++ control file. An output file is specified to hold the model heads computed by each model run and a model input file is specified to hold the estimated K parameters for each run; thus the PEST++ program knows where to find the

values it needs during the multiple model runs within each iteration to (1) compute the new objective function and (2) run MODFLOW with the new estimates. At the start of calibration run, a new Jacobian matrix is computed. The control file also allows one to specify the command needed on the computer system to execute the MODFLOW model. See Appendix C for an example pest control file.

To maintain simplicity, a small number of K parameters, four, was chosen to be estimated. These four parameters were assumed to be the values of K in each of the four zones of uniform K. See zones design in Figure 2.10 below.

NOTE: Due to a lack of sensitivity of the first K parameter in comparison with the sensitivity of the other 3Ks, the first K parameter was set to fixed (Hill and Tiedeman, 2007). So 3 K parameters were specified to be estimated, while K1 was set to a constant of 20 m/day for all iterations (the average of the K values in zone 1). See below for more

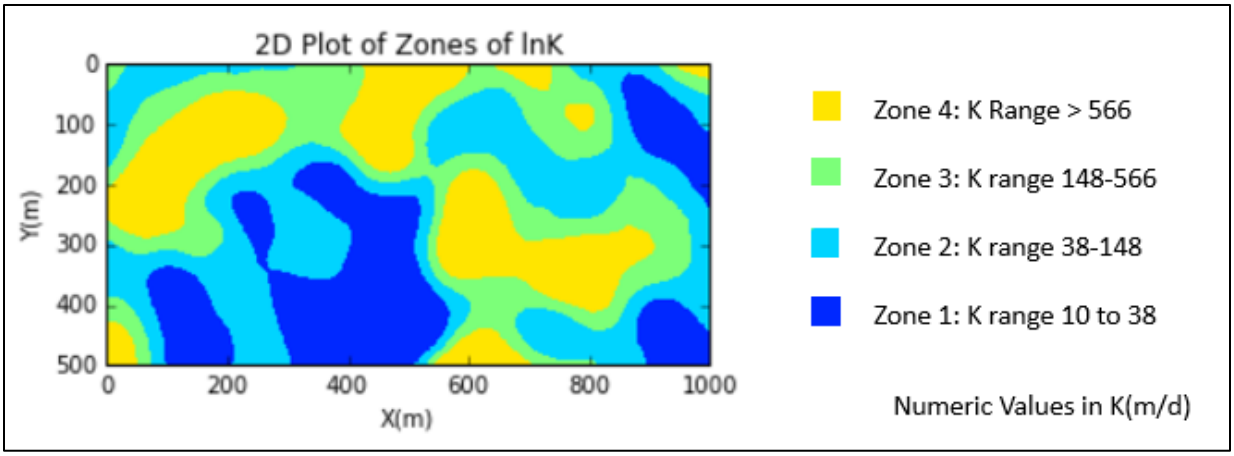



Figure 2.10: Four zones of uniform lnK created for calibration of the flow model.

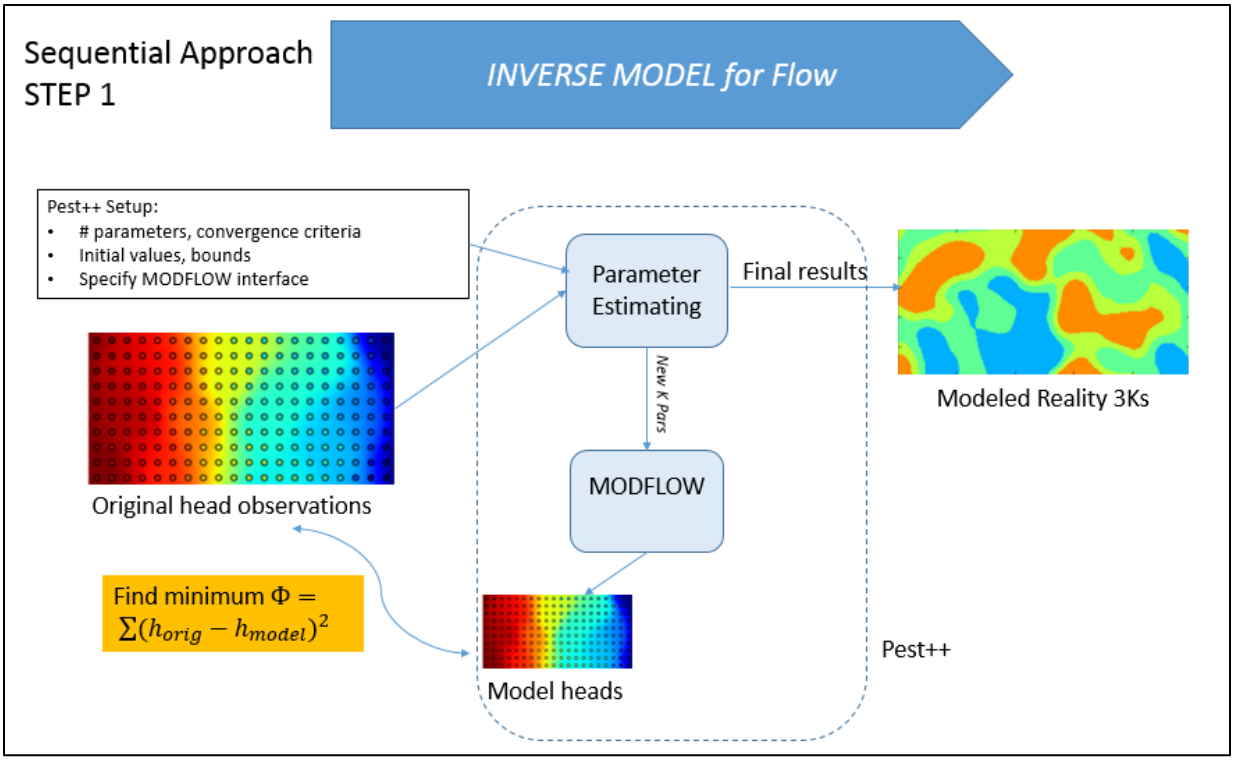



Figure 2.11: Sequential Approach Step 1: Overview of the calibration of the flow model to generate the optimized/modeled K reality. The head observations created by earlier forward model runs are used for comparison to the modeled head results. Parameters are modified, as a minimum value for the objective function (the sum of the squared residuals) is sought.

Pest++ requires the specification of a number of run-time variables (See Appendix C), including initial values and bounds for each of the parameters to be estimated. Averages of each zone provided a rough guideline for the choice of initial values.

Additionally, to ensure uniqueness (Hill and Tiedeman, 2007), the calibration needed to be run multiple times with different initial values. Values were selected that were both higher and lower than the first chosen initial values. If the calibration does not converge to nearly equivalent values during this process, then it means there are local minima which are

preventing convergence to the global minimum value. This occurred during the first attempts at the calibration of flow, and three changes were applied:

1) Low K values were removed from the original MODFLOW model by adding 10 m/day to all values of K. This meant new observations had to be created (i.e., the Forward Model had to be rerun).
2) Pest++ specification was changed to keep K1 (value for zone 1) fixed to 20 m/day.
3) The convergence criterion was changed to allow more iterations: it was changed from converging on Phi to convergence on parameter values. This was achieved by tightening the convergence criteria for Phi. The Pest++ parameter PHIREDSTP which provides the relative objective function reduction that will trigger termination was set to a very small value 0.005 so that it would prevent termination based on Phi and allow calibration to continue until parameters converged.

The third change did not result in improved convergence, but the first two did. This could be attributed to the fact that low conductivity values mean low flow and changes in head would be less sensitive to changes in K when K and flow is low. The composite scaled sensitivities (css) are provided for each PEST++ run. The css value for K1 was much less than the values for K2-K4, by several orders of magnitude. Hill and Tiedeman state that composite scaled sensitivities that are more than two orders of magnitude (or 1%) less than the largest value may interfere with convergence (Hill and Tiedeman, 2007).

During each iteration PEST++ executed MODFLOW with each new set of three K estimates plugged into the zones 2-4, to obtain the next set of modeled head values. These were compared to the original "true observations" using the objective function. PEST++ then continued, searching for values of parameters that would create the minimum objective

function. See Tables 3.4 in Results for values of the objective function at both start and end of the sequential flow calibration runs.

#### *2.4.2.2 Sequential Transport Calibration*

The second calibration uses PEST++ integrated with the MODPATH model. Two sets of calibration runs were created: (1) using a single uniform porosity value across all cells and (2) creating four zones each with a constant porosity values. Again a small number of porosity parameters, one/four, was chosen to be estimated to maintain simplicity. For the second set, these four parameters were assumed to be the values of porosity in each of the same four zones chosen for the uniform K zones. See Figure 2.10. Prior to running this calibration, new MODFLOW outputs need to be created by running MODFLOW using the final estimated K parameters from the sequential flow calibration described above. This would create a new set of heads and cell budgets for ever cell in the model. The only K values that would be available to this run would be the Ks estimated by the flow run; the "true" or synthetic reality values would be unknown.

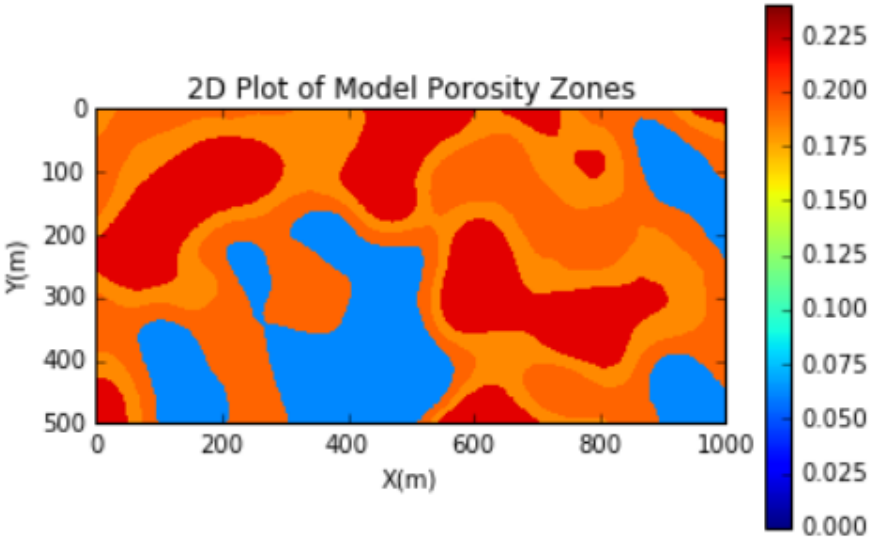


Figure 2.12: Four zones of uniform porosity (designed to match the K zones) were used for calibration of the transport model.

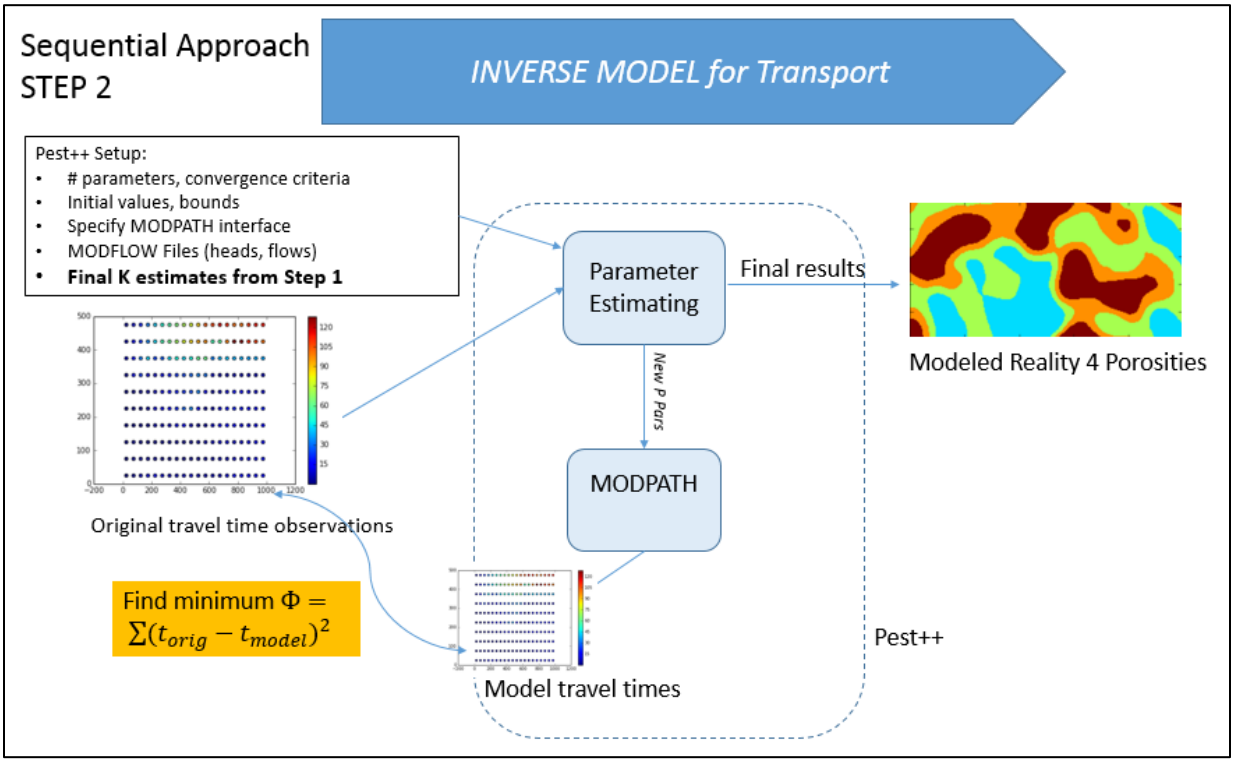



Figure 2.13: Sequential Approach Step 2: Overview of the calibration of the transport model to generate the optimized/modeled porosity reality. The travel time observations created by earlier forward model runs are used for comparison to the modeled travel times. Parameters are modified, as a minimum value for the objective function (the sum of squared residuals) is sought.

Again averages of each zone provided a rough guideline for initial values. To ensure uniqueness (Hill and Tiedeman, 2007), the calibration was run multiple times with different initial values. The transport calibration ran much more smoothly than the flow calibration, so it was not necessary to make any porosities "fixed". This is likely due to the linear relationship between porosity and travel time.

PEST++ executed MODPATH with each new set of porosity estimates plugged into the four zones, to obtain the next set of modeled travel times. These were compared to the original "true observations" using the objective function. Pest++ continued, changing parameter estimates, searching for the minimum objective function. See Tables 3.5 and 3.6 in Results for values of the objective function at both the start and the end of the sequential transport calibration runs.

### 2.4.3 Combined Flow and Transport

Most of the principles explained above for applying PEST++ are still relevant to the combined approach. The key differences are (1) the inclusion of both head and travel time observations in the PEST++ control file, and (2) the need to weight the sum of each set of residuals, so that one set will not exert a greater influence on the resultant objective function than the other set (Hill and Tiedeman, 2007), and (3) the inclusion of two command lines for executing both MODFLOW and MODPATH during each iteration.

The weights that provided a fairly equal contribution to the final objective function for each Combined Set of runs are shown below:

| Combined Run Set | $w_h$ | $w_t$ |
|---|---|---|
| Model Set #1(3Ks and 1 P) | 1.5 | 0.2 |
| Model Set #2 (3Ks and 4 Ps) | 1.5 | 0.3 |

Table 2.5: Weights used for heads and travel times for each combined calibration run set.

Pest++ now minimizes the combined objective function, which is generated according to the observations and weights supplied. The objective function equation for combined Model Set #2 thus becomes:

$$\Phi = S(b) = \sum_{i=1}^{NH} 1.5\ [h_{orig} - h_{model}(b)]^2 + \sum_{j=1}^{NT} 0.3[t_{orig} - t_{model}(b)]^2 \qquad (16)$$

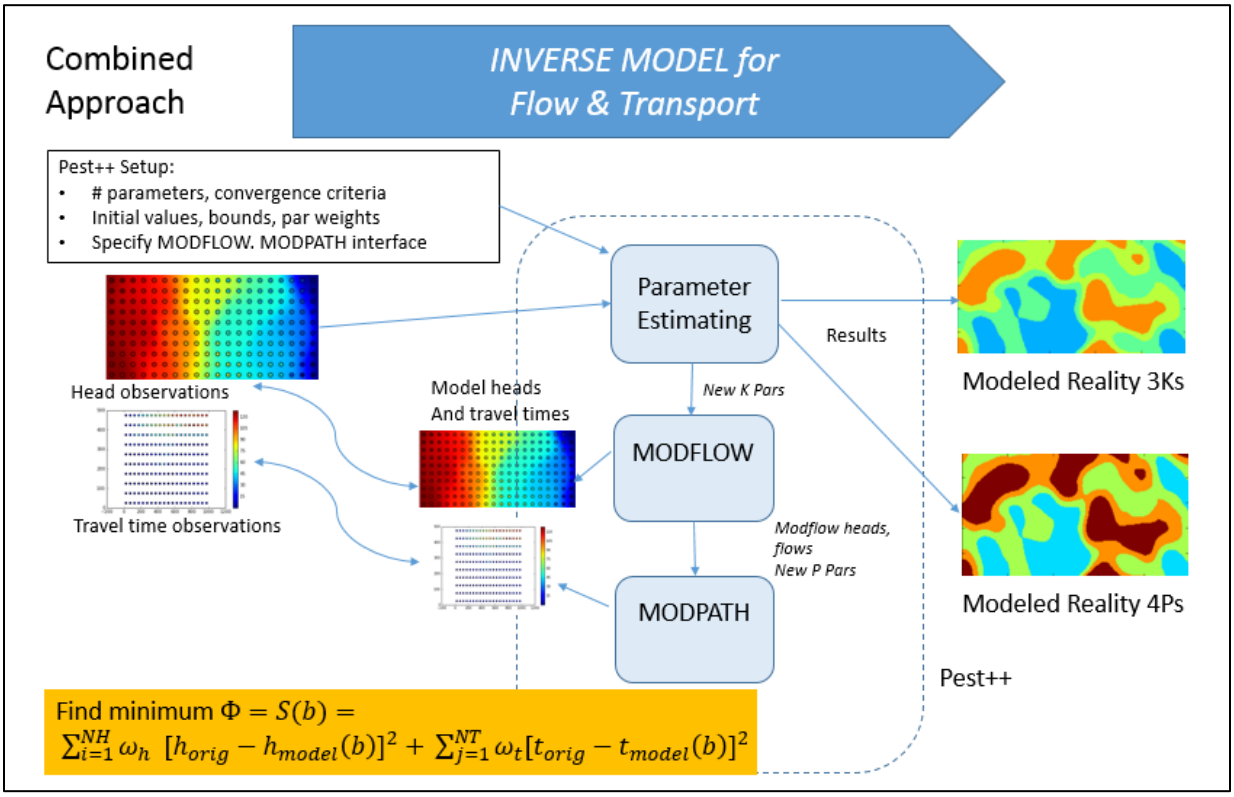



Figure 2.14: Calibrating the combined flow and transport model to obtain optimized K and porosity realities. Both sets of observations created by earlier forward model runs are used for comparison to the modeled heads and travel times. Parameters are modified, as a minimum value for the combined objective function, the weighted sum of the two sets of squared residuals is sought.

As shown in Figure 2.14, Pest++ is integrated with both MODFLOW & MODPATH and runs the models sequentially so that values estimated for K by MODFLOW in each iteration condition the subsequent MODPATH model execution which estimates porosity. Additionally, both estimates influence the combined objective function value at each iteration,

thus exerting influence on the subsequent estimates. Upon completion of the calibration, all optimized parameters are found for (1) all four parameters in the first combined set: 3Ks and one porosity or (2) all seven parameters in the second combined set: 3 Ks and 4 porosities.

Again, it is necessary to rerun each calibration with different sets of initial values; and to check that the resultant estimates are fairly close.

# 3 RESULTS AND DISCUSSION

## 3.1 Comparison of Sequential and Combined Approaches

The final results comparing the different optimized parameters obtained for the Sequential and Combined approaches are presented in Tables 3.1 through 3.4. In both sets of runs the results of the Combined approach are closer to the "true" values of the synthetic reality. Note for both set of runs, sequential percent differences are higher for all parameters compared to percent differences for combined values. While comparing porosity was straight-forward, the process used to compare K estimates consisted of several steps.

| Comparison of True Parameters to Estimated Parameters (Model Set #1- 3Ks, 1 Porosity) | | | | | |
|---|---|---|---|---|---|
| SEQUENTIAL | | | | | |
| Parameter | K Zone Averages: lnK (K in m/d) and Exact Porosity | Zone Std_dev: lnK (K in m/d) | % Relative Standard Deviation | Estimated Values: lnK (K in m/d) | % difference of Estimates to True |
| K2 | 4.350 | 0.3702 | 8.5120 | 4.684 | 7.694 |
| K3 | 5.654 | 0.3877 | 6.8567 | 5.751 | 1.704 |
| K4 | 7.225 | 0.5959 | 8.2480 | 7.614 | 5.376 |
| P1 | 0.2000 | 0.0000 | 0.0000 | 0.2502 | 25.091 |
| COMBINED | | | | | |
| K2 | 4.350 | 0.3702 | 8.5120 | 4.453 | 2.384 |
| K3 | 5.654 | 0.3877 | 6.8567 | 5.625 | -0.522 |
| K4 | 7.225 | 0.5959 | 8.2480 | 7.417 | 2.658 |
| P1 | 0.2000 | 0.0000 | 0.0000 | 0.2320 | 16.017 |

Table 3.1: Comparison of final optimized parameters from sequential and combined approach for Model Set #1 (3 Ks and 1 Porosity).

| Comparison of True to Estimated 3Ks, 1 Porosity | | | | | |
|---|---|---|---|---|---|
| SEQUENTIAL | | | | | |
| Parameter | Zone Averages: K (m/d) and Exact Porosity | K Zone Averages: lnK (K in m/d) and Exact Porosity | Estimated Values: K(m/d) | Estimated Values: lnK (K in m/d) | % difference of Estimates to True |
| K2 | 77.453 | 4.350 | 108.240 | 4.684 | 7.694 |
| K3 | 285.56 | 5.654 | 314.436 | 5.751 | 1.704 |
| K4 | 1373.8 | 7.225 | 2025.922 | 7.614 | 5.376 |
| P1 | 0.20000 | 0.2000 | 0.25018 | 0.2502 | 25.091 |
| COMBINED | | | | | |
| K2 | 77.453 | 4.350 | 85.973 | 4.453 | 2.384 |
| K3 | 285.56 | 5.654 | 277.900 | 5.625 | -0.522 |
| K4 | 1373.8 | 7.225 | 1664.667 | 7.417 | 2.658 |
| P1 | 0.20000 | 0.2000 | 0.2320 | 0.2320 | 16.017 |

Table 3.2: Comparison of final optimized parameters from sequential and combined approach for Model Set #1, displaying equivalent K values in columns 1 and 3.

Note that the column heading for the true values says “K Zone Averages”. There is a difference in granularity between the initial K reality and the estimated K reality. The original K contains a range of K values within each zone – it is heterogeneous, but there is only a single value estimated for each uniform K parameter zone, therefore the average of these values was used for the purposes of comparing the data. Notice also that all values are provided as lnK. This is because all calibration is performed using ln/log of K. Therefore in reviewing any results, they need to be transformed back to lnK. Otherwise using K causes differences between values to be exaggerated.

Additionally, the standard deviation of lnK for each synthetic lnK zone is shown in Table 3.1. While the sequential approach does end up with percent differences greater than the combined approach, the sequential values do fall within a single standard deviation of the average zone value. For this model, the sequential method produces acceptable results for flow estimates, though not as close as the combined estimates.

The approach to porosity analysis differs because both true reality and estimated realities are uniform either across the entire 2D space (for Model Set #1, single porosity run) or across each zone (for Model Set #2, 4 porosity runs). For the "true" values of porosity, there are exact values and a standard deviation of zero for comparison of to the estimates.

The results for Model Set #2 (4 porosities), are consistent with the Model Set #1. See Table 3.3. Combined parameter estimates again show smaller percent differences when compared to the "true" values.

In both sets of results, the fact that generally the values of K parameters are higher for sequential than for combined, can help to explain the generally higher results for porosity estimated in the sequential runs. A higher K will result in a higher Darcy velocity: Q/A. Higher porosity will have a compensating effect; reducing the final pore velocity.

The final estimate of P2 (for the 4 porosity runs) for both the sequential and the combined runs has a noticeably greater percent difference than the other 3 porosity parameters.

| Comparison of True Parameters to Estimated Parameter (Model Set #2- 3Ks, 4 Porosity) | | | | | |
|---|---|---|---|---|---|
| SEQUENTIAL | | | | | |
| Parameter | Zone Averages: lnK (K in m/d) and Exact Porosity | Zone Standard Deviations: lnK (K in m/d) | % Relative Standard Deviation | Estimated Values: lnK (K in m/d) | % difference of Estimates to True |
| K2 | 4.3497 | 0.3702 | 8.5120 | 4.684 | 7.694 |
| K3 | 5.6545 | 0.3877 | 6.8567 | 5.751 | 1.704 |
| K4 | 7.2253 | 0.5959 | 8.2480 | 7.614 | 5.376 |
| P1 | 0.0730 | 0.0000 | 0.0000 | 0.0626 | -14.247 |
| P2 | 0.1170 | 0.0000 | 0.0000 | 0.1922 | 64.245 |
| P3 | 0.1600 | 0.0000 | 0.0000 | 0.1829 | 14.319 |
| P4 | 0.2120 | 0.0000 | 0.0000 | 0.2191 | 3.358 |
| COMBINED | | | | | |
| K2 | 4.3497 | 0.3702 | 8.5120 | 4.509 | 3.668 |
| K3 | 5.6545 | 0.3877 | 6.8567 | 5.612 | -0.750 |
| K4 | 7.2253 | 0.5959 | 8.2480 | 7.483 | 3.566 |
| P1 | 0.0730 | 0.0000 | 0.0000 | 0.0732 | 0.284 |
| P2 | 0.1170 | 0.0000 | 0.0000 | 0.1637 | 39.915 |
| P3 | 0.1600 | 0.0000 | 0.0000 | 0.1601 | 0.078 |
| P4 | 0.2120 | 0.0000 | 0.0000 | 0.2159 | 1.816 |

Table 3.3: Comparison of final optimized parameters from sequential and combined approach for Model Set #2 (3Ks and 4Ps).

| Comparison of True to Estimated 3Ks, 4 Porosity | | | | | |
|---|---|---|---|---|---|
| SEQUENTIAL | | | | | |
| Parameter | Zone Averages: K (m/d) and Exact Porosity | Zone Averages: lnK (K in m/d) and Exact Porosity | Estimated Values: K(m/d) | Estimated Values: lnK (K in m/d) | % difference of Estimates to True |
| K2 | 77.453 | 4.3497 | 108.240 | 4.684 | 7.694 |
| K3 | 285.56 | 5.6545 | 314.436 | 5.751 | 1.704 |
| K4 | 1373.8 | 7.2253 | 2025.922 | 7.614 | 5.376 |
| P1 | 0.0728 | 0.0730 | 0.0626 | 0.0626 | -14.247 |
| P2 | 0.1168 | 0.1170 | 0.1922 | 0.1922 | 64.245 |
| P3 | 0.1600 | 0.1600 | 0.1829 | 0.1829 | 14.319 |
| P4 | 0.2120 | 0.2120 | 0.2191 | 0.2191 | 3.358 |
| COMBINED | | | | | |
| K2 | 77.453 | 4.3497 | 86.939 | 4.509 | 3.668 |
| K3 | 285.56 | 5.6545 | 258.620 | 5.612 | -0.750 |
| K4 | 1373.8 | 7.2253 | 1754.300 | 7.483 | 3.566 |
| P1 | 0.0728 | 0.0730 | 0.0732 | 0.0732 | 0.284 |
| P2 | 0.1168 | 0.1170 | 0.1637 | 0.1637 | 39.915 |
| P3 | 0.1600 | 0.1600 | 0.1601 | 0.1601 | 0.078 |
| P4 | 0.2120 | 0.2120 | 0.2159 | 0.2159 | 1.816 |

Table 3.4: Comparison of final optimized parameters from sequential and combined approach for Model Set #2 (3Ks and 4Ps), showing equivalent Ks in columns 1 and 3.

The bar charts that follow in Figures 3.1 and 3.2 provide an alternative view of the same data shown in the tables. It is clear from these charts that the combined approach is producing more accurate results. Through the simultaneous calibration of K and Porosity, the estimation software is able to consider both flow and transport processes. There are a number of possible explanations for this: (1) when K is computed from only head observations, the error introduced by matching to heads only is fully propagated to the calibration of porosities (Strecker and Chu, 1986), (2) heads often do not constrain the solution as much as transport

values (Hill and Tiedeman, 2007) and (3) simultaneous calibration, through use of a combined objective function, enables a more realistic match to true reality; it allows incremental conditioning of the flow and transport model runs (within each calibration iteration) by the results of the previous iteration.

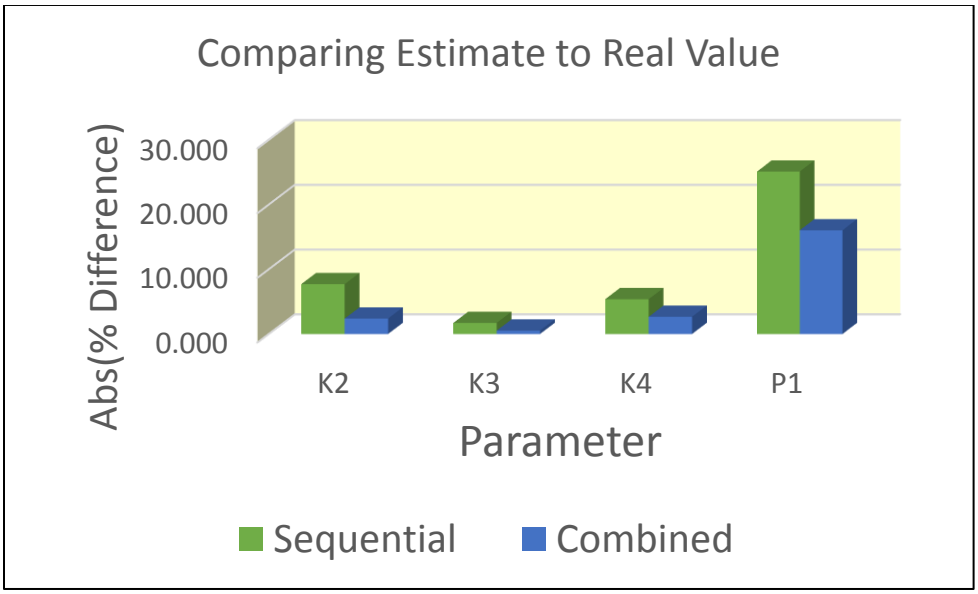



Figure 3.1: Comparison of Final Optimized Parameters from Sequential and Combined Approach for First Parameter Set.

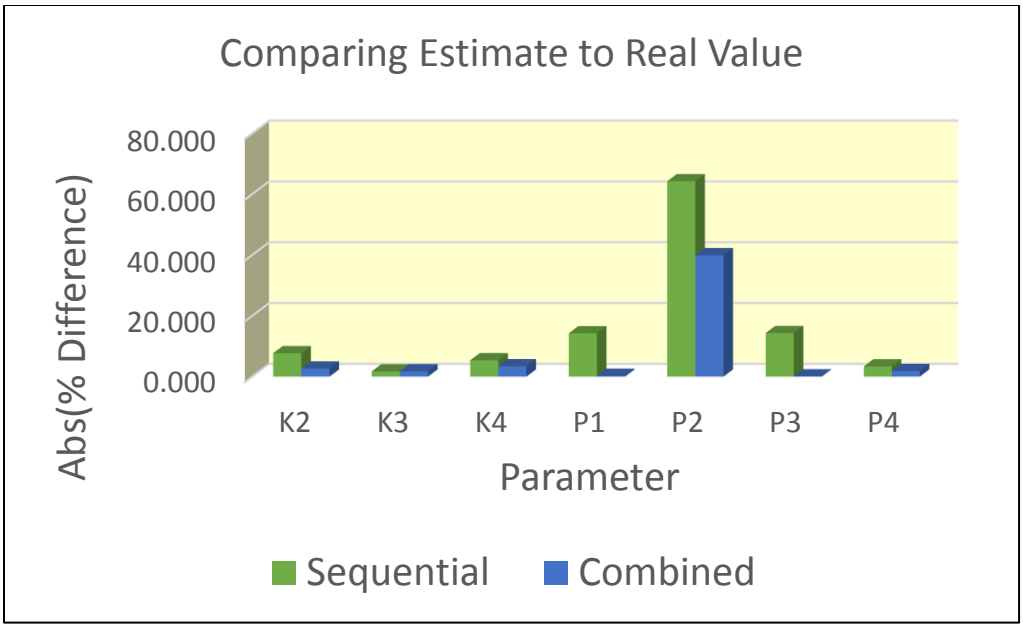



Figure 3.2: Comparison of Final Optimized Parameters from Sequential and Combined Approach for Second Parameter Set.

## 3.2 Sequential Run Spatial Analysis

This section presents some 2D spatial views of the Sequential results as well as analysis that can be gleaned from these spatial views.

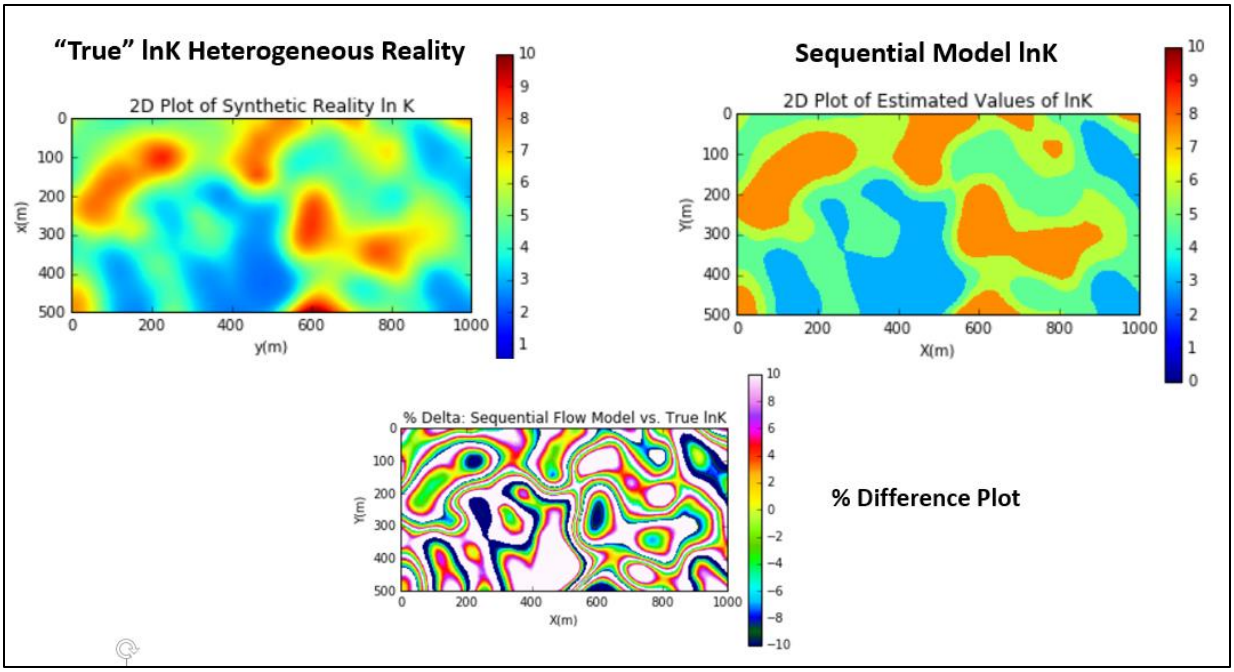



Figure 3.3: Comparing Sequential Flow results to the heterogeneous synthetic K reality.

Looking at the 2D spatial views of lnK in Figure 3.3, it is clear that choosing four zones to correspond to ranges of the synthetic lnK, assists in achieving a result that reflects the spatial differences in lnK, while at a grosser approximation due to the small number of zones. In a field situation, while we would not have the precise ranges of lnK to divide into four zones, we would likely have some measurements as well as some expert geological information to leverage to delineate our zones.

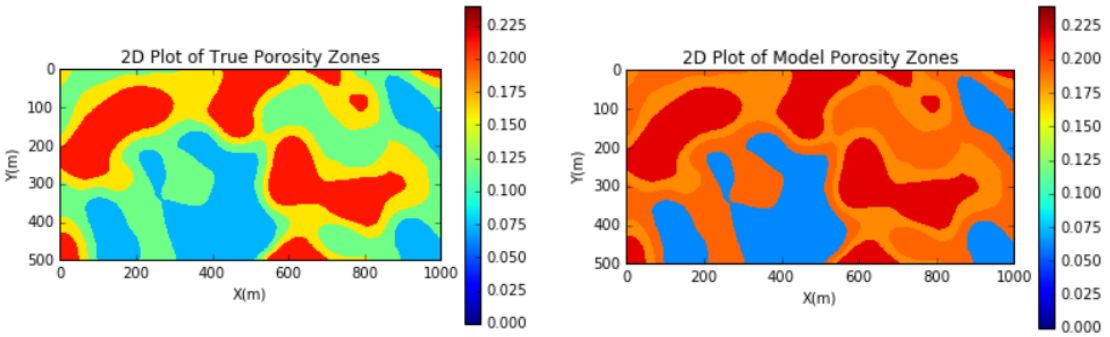



Figure 3.4: Comparing Sequential Transport Model results to the synthetic porosity reality of Model Set #2, 4 uniform porosity zones.

The sequential porosity estimated for four porosity zones shows the greatest color difference in zone 2: green on the left image & orange on the right image. Looking back at our values in Table 3.2, we see the greatest percent difference in zone 2: 64%.

## 3.3 Combined Run Spatial Analysis

This section presents some 2D spatial view of the combined results as well as analysis that can be gleaned from these spatial views.

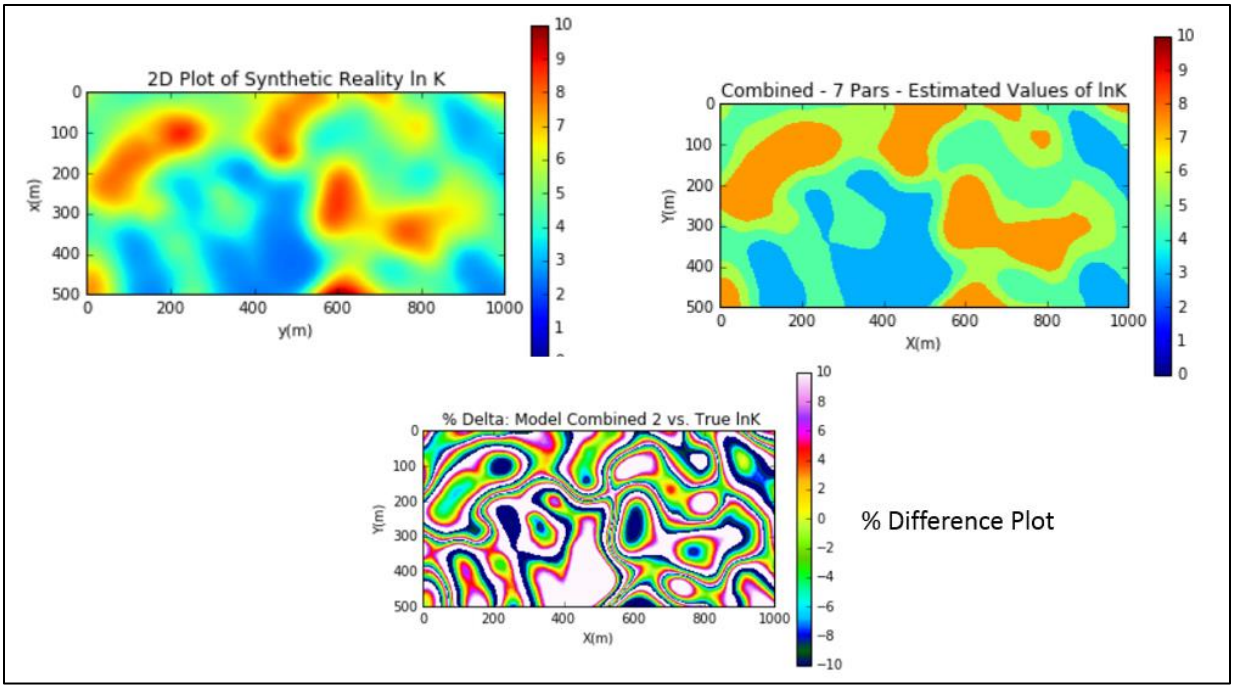



Figure 3.5: Comparing Combined Model Set #2 Flow results to the heterogeneous synthetic K reality.

It is very difficult to visually observe a difference between the above sequential spatial plots and this set of plots. The tables of values provided in the beginning of the Results Section are more useful for analyzing results.

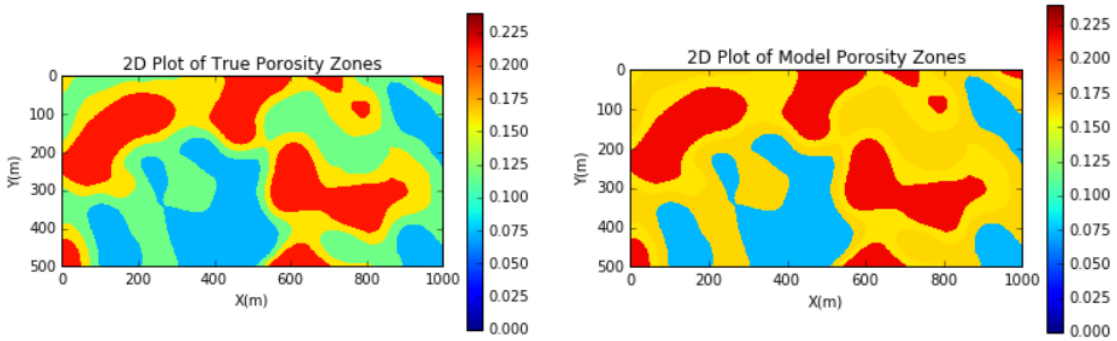



Figure 3.6: Comparing Combined Model Set #2 Transport results to the uniform zones of the synthetic P reality.

The color contrast shown between these two sets of values is less extreme than that seen for the Sequential results, simply re-enforcing the observations of the results tables provided above.

## 3.4 Detailed Results of Multiple Calibration Runs

The following results show some of the details behind the summaries given in Section 3.1. The results of any calibration run could not be considered final, unless a number of runs were made using differing initial parameter values. This would ensure that the objective function phi as a function of the parameters was well-behaved; that is, had one global minimum. Though starting at different initial values, these runs needed to converge to nearly equal parameter values for the results to be accepted.

While initially four runs were often performed per calibration, as the behavior of certain runs was seen to be consistent, especially in the case of the calibration of porosity, less runs were performed later in the research. The mathematics behind the transport runs is linear as opposed to the non-linear relation of parameter K to changes in head (Hill and Tiedeman, 2007). The following tables show the initial values used per run, the initial and final values of phi, and the number of iterations needed for convergence.

Well behaved models often show the majority of convergence in the first few iterations; sometimes even the first one, so parameter estimates do not change a great deal after the first few iterations. The very stringent settings on the convergence criteria used in this research often caused the model to run longer (see Methods Section 2.4.1), and at times the execution could be stopped once the objective function phi was no longer being reduced.

Table 3.5: Sequential Flow Runs with Different Initial Values: Initial & Final Values of lnK (m/d) & Phi, Number of Pest Iterations.

| Parameter | SFlowRun5a | | SFlowRun5b | | SflowRun5c | | SFlowRun5d | |
|---|---|---|---|---|---|---|---|---|
| | Initial | Final | Initial | Final | Initial | Final | Initial | Final |
| K1 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 |
| K2 | 4.094 | 4.687 | 4.605 | 4.684 | 4.317 | 4.682 | 4.942 | 4.684 |
| K3 | 5.298 | 5.755 | 5.704 | 5.751 | 4.828 | 5.747 | 5.011 | 5.750 |
| K4 | 7.601 | 7.615 | 7.313 | 7.615 | 7.244 | 7.614 | 6.685 | 7.611 |
| #iters | | 6 | | 9 | | 14 | | 7 |
| Phi | 207.697 | 83.111 | 118.885 | 83.096 | 375.662 | 83.110 | 1410.210 | 83.046 |

Table 3.6: Sequential Flow Runs with Different Initial Values: Initial & Final Values of K (m/d) & Phi, Number of Pest Iterations

| Parameter | SFlowRun5a | | SFlowRun5b | | SflowRun5c | | SFlowRun5d | |
|---|---|---|---|---|---|---|---|---|
| | Initial | Final | Initial | Final | Initial | Final | Initial | Final |
| K1 | 20 | 20.000 | 20 | 20.000 | 20 | 20.000 | 20 | 20.000 |
| K2 | 60 | 108.567 | 100 | 108.231 | 75 | 107.975 | 140 | 108.187 |
| K3 | 200 | 315.755 | 300 | 314.408 | 125 | 313.285 | 150 | 314.295 |
| K4 | 2000 | 2027.466 | 1500 | 2029.165 | 1400 | 2026.191 | 800 | 2020.869 |
| #iters | | 6 | | 9 | | 14 | | 7 |
| Phi | 207.697 | 83.111 | 118.885 | 83.096 | 375.662 | 83.110 | 1410.210 | 83.046 |

Since K1 was fixed, looking at K2-K4: the final estimates expressed in lnK are nearly equivalent. This is also clear by noticing the final phi, sum of squared residuals of model result to synthetic/"true" observations are also nearly equal– see last row – final values for Phi all are about 83. The Transport runs shown below in both Table 3.7 and Table 3.8 gave identical results; earlier in research, 4 runs were made and the all results agreed; therefore, in later reruns, 2 runs were considered sufficient. Again while initial Phi values differed due to changing initial parameter values, the final Phi values across the runs are identical. The logic applied to

select initial values was as follows: begin with a value near the lnK average of each zone, modify the values in each subsequent run to deviate from the previous set of initial values by a significant amount (but not the extremes of each range), while staying within the zone range. Each zone value was varied to both higher and lower values. Since the PEST++ tool requires input in terms of K, these values were generally round numbers in terms of K as can be seen in the second table (Table 3.6).

Table 3.7: Sequential Transport Run 1 with Different Initial Values: Initial & Final Values of Porosity & Phi, # of Pest Iterations

| Porosity Parameter Final Value , Init and Final Phi, Iterations (identical results) | | |
|---|---|---|
| Parameter | Final Porosity | Final Phi |
| P1 | 0.250200 | 9338.000 |

| | STrans1Run5 | | STran1Run6 | |
|---|---|---|---|---|
| | Initial | Final | Initial | Final |
| P1 | 0.4 | 0.2502 | 0.5 | 0.2502 |
| #iters | | 5 | | 5 |
| Phi | 122205 | 9338 | 323161 | 9338 |

Table 3.8: Sequential Transport Run 2 with Different Initial Values: Initial & Final Values of Porosity & Phi, # of Pest Iterations

| Porosity Paramters Final Values, Init and Final phi, Iteratations (identical results) | |
|---|---|
| Parameter | Final Porosity |
| P1 | 0.0626 |
| P2 | 0.1922 |
| P3 | 0.1829 |
| P4 | 0.2191 |

| | STran2Run10 | | STran2Run11 | |
|---|---|---|---|---|
| | Initial | Final | Initial | Final |
| P1 | 0.06 | see above | 0.08 | see above |
| P2 | 0.1 | | 0.12 | |
| P3 | 0.17 | | 0.15 | |
| P4 | 0.25 | | 0.2 | |
| #iters | | 5 | | 5 |
| Phi | 8836 | 2117 | 5492 | 2117 |

The combined run results showed some variation in the final estimates of the runs with differing initial values. Therefore, some statistics were computed to check the variation among the results. Combined Model Set #1 runs showed very minor variations with relative standard deviations from .001 to .02 (0.1% to 2%). Combined Model Set #2 results showed greater variation with a few higher standard deviation percent's of about 4 to 7%. It is believed that the greater complexity of the latter run contributed to the challenge of obtaining nearly equivalent estimates; the combined runs are more sensitive to the initial values given to the parameters.

The combined run results also show how the final contributions to phi (from heads and travel times) are nearly equal at the end of the runs. This was the result of adjusting the weights for each set of observation until the final phis showed nearly equal contributions.

Table 3.9: Combined Model Set #1 runs with Different Initial Values: Initial & Final Values of lnK (K in m/d), Porosity, and Phi, # of Pest Iterations. Phi Contributions from H – heads and T- travel times as well as Total Phi.

| Parameter | C1Run9 | | C1Run10 | | C1Run11 | | Statistics using lnK | | |
|---|---|---|---|---|---|---|---|---|---|
| | Initial | Final | Initial | Final | Initial | Final | Mean(m/d) | Stddev(m/d) | Rstddev |
| K1 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 2.996 | 0.000 | 0 |
| K2 | 4.094 | 4.442 | 4.605 | 4.503 | 4.317 | 4.415 | 4.453 | 0.045 | 0.010 |
| K3 | 5.298 | 5.607 | 5.704 | 5.716 | 4.828 | 5.552 | 5.625 | 0.083 | 0.015 |
| K4 | 7.601 | 7.419 | 7.313 | 7.410 | 7.244 | 7.423 | 7.417 | 0.006 | 0.001 |
| P1 | 0.3 | 0.2316 | 0.4 | 0.2368 | 0.1 | 0.2277 | 0.2320 | 0.005 | 0.020 |
| #iters | | 11 | | 10 | | 7 | | | |
| Phi Contrib. H | 460 | 210 | 268 | 212 | 4820 | 207 | | | |
| Phi Contrib. T | 3623 | 254 | 5007 | 246 | 846 | 248 | | | |
| Total Phi | 4084 | 464 | 5278 | 458 | 3975 | 456 | | | |

Table 3.10: Combined Model Set #2 runs with Different Initial Values: Initial & Final Values of lnK (K in m/d), Porosity, and Phi, # of Pest Iterations. Phi Contributions from H – heads and T- travel times as well as Total Phi.

| Parameter | C2Run22 | | C2Run23 | | C2Run24 | | C2Run25 | | Statistics using lnK | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Final | Initial | Final | Initial | Final | Initial | Final | Mean(m/d) | Stddev(m/d) | Rstddev |
| K2 | 4.3175 | 4.4652 | 4.6052 | 4.5695 | 4.4427 | 4.5657 | 4.5539 | 4.4364 | 4.509 | 0.068 | 0.0152 |
| K3 | 4.8283 | 5.5554 | 5.7038 | 5.6801 | 5.5215 | 5.6714 | 5.2983 | 5.5413 | 5.612 | 0.074 | 0.0132 |
| K4 | 7.2442 | 7.4698 | 7.3132 | 7.5052 | 7.1701 | 7.5131 | 7.0901 | 7.4438 | 7.483 | 0.032 | 0.0043 |
| P1 | 0.06 | 0.07634 | 0.08 | 0.0692 | 0.07 | 0.06889 | 0.05 | 0.0784 | 0.073 | 0.005 | 0.0667 |
| P2 | 0.1 | 0.1556 | 0.12 | 0.1747 | 0.11 | 0.174 | 0.13 | 0.1505 | 0.164 | 0.012 | 0.0762 |
| P3 | 0.17 | 0.1546 | 0.15 | 0.1666 | 0.16 | 0.1667 | 0.14 | 0.1526 | 0.160 | 0.008 | 0.0473 |
| P4 | 0.25 | 0.2131 | 0.2 | 0.2195 | 0.21 | 0.2189 | 0.18 | 0.2119 | 0.216 | 0.004 | 0.0181 |
| #iters | | 12 | | 9 | | 16 | | 10 | | | |
| Phi Contrib. H | 862 | 202 | 272 | 194 | 353 | 194 | 795 | 205 | | | |
| Phi Contrib. T | 584 | 178 | 427 | 172 | 422 | 172 | 659 | 175 | | | |
| Phi Total | 1447 | 380 | 700 | 365 | 755 | 365 | 1454 | 380 | | | |

## 3.5 Model Bias

Hill and Tiedeman in their book Effective Groundwater Model Calibration, Chapter 6, recommend an analysis of model fit by using a number of approaches to look at potential model bias. Two of these approaches were applied to the results of this work and are shown below. The first method shows final head residuals plotted against the modeled head results. We are able to create a trend line for this first plot and we see it is nearly flat. This tells us that the residual of the Sequential Flow model show little bias with respect to modeled heads, which means that there is an even spread of the residuals across all the possible values of heads (50m to 100m).

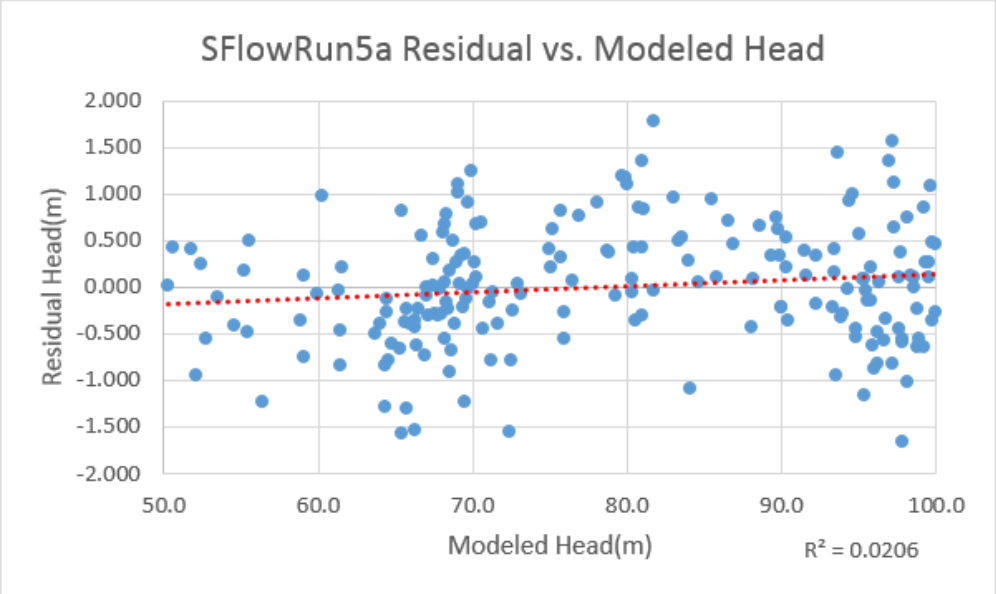



Figure 3.7: Sequential Flow results showing residuals as a function of modeled head.

The second method plots the residuals across the 2D x-y space of the aquifer, that is, looking down on the aquifer. This is a way of checking for bias in a particular spatial extent of the

aquifer. The plot below shows a cluster of negative values on the far right end of the 2D space. This would be an indication that to make the estimates meet the convergence criteria, the heads estimates needed to be made lower (negative residuals) in that portion of the aquifer, perhaps to compensate for the cluster of positive residuals falling in the 400-600 m rectangle.

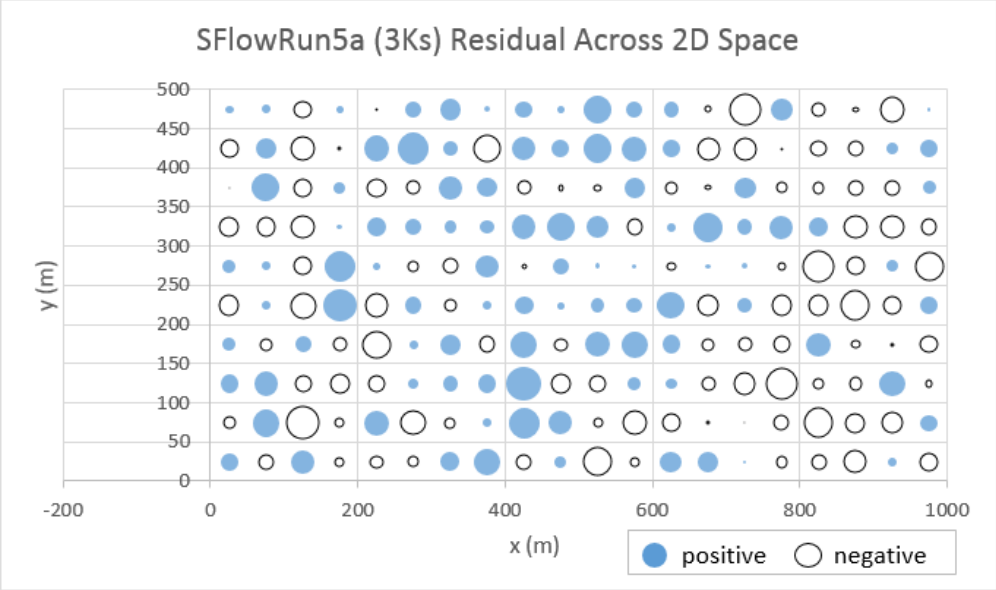



Figure 3.8: Sequential Flow results showing residuals across 2D space.

The combined results shown below are actually quite similar to the sequential results. There is a slightly higher slope of the first plot, but it is still very flat. The second plot shows smaller circle areas due to the lower residuals, but still a cluster of negative residuals at the far end of the aquifer (800m -1000m).

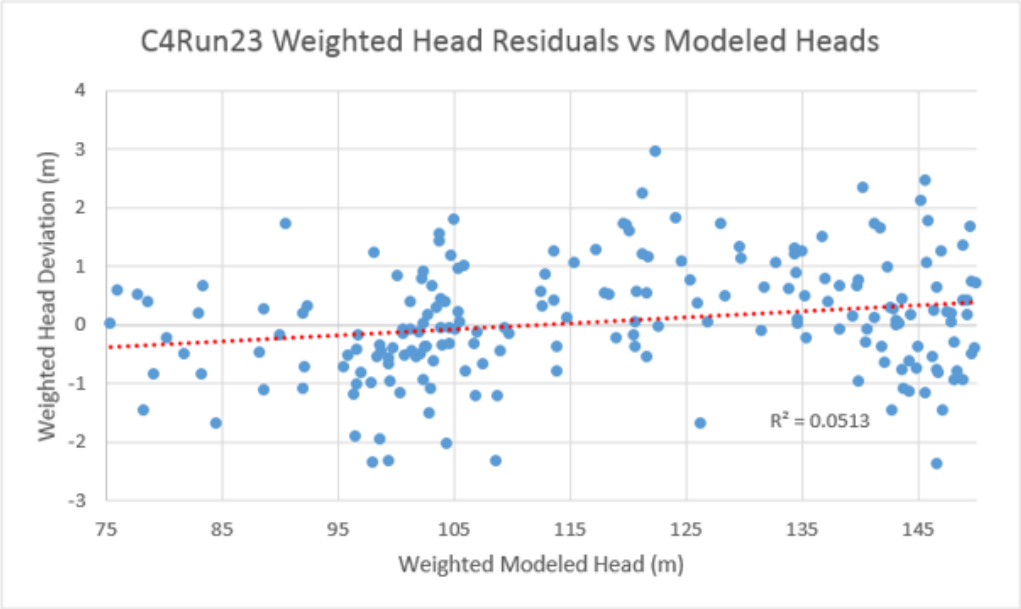



Figure 3.9: Combined Flow results showing residuals as a function of modeled head.

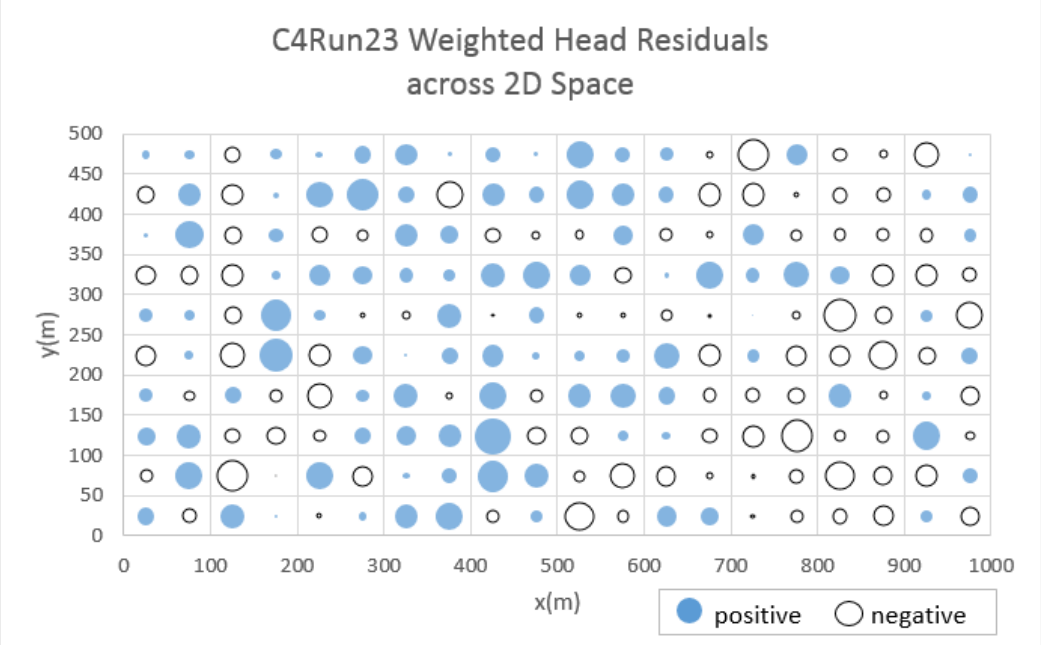



Figure 3.10: Combined Flow results showing residuals across 2D space.

## 3.6 Results General Statements

Taken altogether, the calibration run results show a great deal of consistency. Each combined model performed better than the corresponding sequential model. The detailed run results show that regardless of the initial value, the final estimates agree within a decent tolerance. The exploration of model bias showed little bias overall; however, some did occur at the far end of the aquifer in the spatial bias analysis.

# 4 CONCLUSIONS

## 4.1 Summary

This research was designed to focus solely on the comparison of two different approaches to groundwater model calibration: sequential and combined. Within the specified scope of this research, that is, given the specifics of the synthetic reality, the data values and any assumptions, as well as any limitations to the applied modeling processes; the results of this research work supports the thesis statement: Optimized groundwater models better reflect reality when both flow and transport observations are applied. The results of two sets of calibration runs show consistently closer values (as percent difference from true values) for the combined approach vs. the sequential approach. However, the sequential approach performed well with results falling within one standard deviation of the true values. Numerical modeling of natural processes involves numerous approximations, thus this specific study is subject to a range of conditions and assumptions.

This study is subject to the following major categories of conditions: the chosen synthetic reality, approximations inherent in finite difference modeling, assumptions applied to the flow and transport processes, and the approach to the sampling of the synthetic observations and their measurement error simulation. The synthetic reality brought with it a Sequential Gaussian Simulation of K, distributed log-normal, with a specific means, variance, and range of lnK, and very well-defined boundary conditions. The sampling approach, used for creating synthetic observations, included the following specifications: (1) a single sampling density, (2) the even sampling of the 2D space, (3) the use of the same sampling locations for both the flow model (heads) and transport model (travel times), and (4) a normal distribution used to specify random

measurement errors. Keeping these assumptions in mind, reasoning for the results obtained may be attributed to several factors.

The improvement of calibration results for the combined approach is believed to be directly related to the inclusion of both head and travel time measurements in a simultaneous calibration process. Using the weighted combined objective function, the two sets of measurements were mathematically adjusted together; therefore, each conditioning the other until the combined objective function converges to a minimum. Simultaneously the parameter estimates each converged toward a single value. The adjustment of the weighting of the contributions of each set of observations to create a near-equal contribution to the final objective function value, enforced a normalized influence of each set of observations (Hill and Tiedeman, 2007; Doherty, 2014). The fact that the hydraulic conductivity K affects both flow and transport supports the propagation of interim K estimates in the combined approach (Gailey, 1991). This is the quantitative reasoning not applied to the sequential approach.

The sequential approach assumes the best estimates of K will be found from only head observations and does this by using head observations alone in calibration step 1. Once those Ks are determined, it then uses those final estimates, including their total errors to condition calibration step 2. Earlier researchers have noted a greater error in optimized K using the sequential approach, which then affects the subsequent calibration (Strecker and Chu, 1986; Gailey, 1991). This study does indeed point to a propagation of error caused by using the K results from Step 1 to define the K reality for Step 2.

The concern expressed by Voss in his Hydrogeology Journal Editor's note (Voss, 2011a), regarding the sampling of the aquifer by two separate processes was not thoroughly tested with the specified synthetic reality. It is challenging to simulate realistic high K channels using the

SGS Simulation (Lee et al., 2007; Eggleston et al., 1996). Additionally, due to the uniform sampling of travel time across the aquifer, the transport process sampled the overall aquifer hydrogeology, as opposed to sampling selected subsets of the 2D domain (the high K regions) as Voss states is more likely in field situations. Likely different sampling approaches would be needed to fully test this idea. However, the synthetic hydraulic conductivity was at least heterogeneous, thus potentially contradicting Voss' claim that the combined approach should only be applied to a homogeneous aquifer (Voss, 2011a).

In any case, this kind of questioning of the usefulness of the combined approach in real field situations should be considered in choosing the next steps for future related research.

## 4.2 Future Work

This research could be expanded in order to help to rule out bias that could have been caused by the assumptions made. Additional test cases could change one or more of the following: 1) synthetic reality boundary conditions, 2) statistical parameters of variogram used for specifying hydraulic conductivity in the Sequential Gaussian Simulation, 3) approach to specifying heterogeneous K, 4) sampling density, sampling uniformity, or sampling placement, 5) 2D size and grid resolution, and 6) standard deviation of error added to observations. To create test cases that more closely simulate field studies we could consider lowering the number of observations, creating a non-uniformly placed set of observations, and applying a range of heterogeneous K fields, some geostatistical and some that apply alternate methods to attempt close simulation of natural geologic structures (Lee et al., 2007; Eggleston et al., 1996).

Through applying a statistical range of initial realities, more statistical analysis of the results could be performed. Geostatistical simulations can produce multiple equi-probable realities.

Following fully exercising the software on synthetic test cases and verifying the validity of results, the calibration framework developed by this research could be applied to a field study – replacing the synthetic observations with field observations. While inverse models are certainly applied regularly by modelers to field data, there is often no preliminary exercising of the inverse model with a fully-specified reality.

# 5 Work Cited

A Vanishing Aquifer: What happens when the water runs out? National Geographic Society. August, 2016. URL: http://www.nationalgeographic.com/magazine/2016/08/vanishing-midwest-ogallala-aquifer-drought/, last accessed Dec 1, 1016.

Anderman, E.R., and M.C. Hill. 1999. A new multistage groundwater transport inverse methods: Presentation, evaluation, and implications. *Water Resources Research,* 35 (4), 1053–1063.

Bartolino, J., and S. Vincent. 2013. Groundwater Resources of the Wood River Valley, Idaho. Boise, Idaho: U.S. Geological Survey. URL: http://pubs.usgs.gov/fs/2013/3005/, last accessed Dec 1, 2016.

Bohling, G. 2007. S-GeMS Tutorial Notes presented in Hydro-geophysics: Theory, Methods, and Modeling, Boise State University. Kansas Geological Survey. URL: http://people.ku.edu/~gbohling/BoiseGeostat, last access October 18, 2014.

Bear, J. 1972. *Dynamics of Fluids in Porous Media*. Mineola, N.Y.: Dover Publications, Inc.

Doherty, J. 2014. *PEST, Model-independent parameter estimation-User manual* (5th ed., with slight additions): Brisbane, Australia.: Watermark Numerical Computing.

Doherty, J. 2015. *Calibration and Uncertainty Analysis for Complex Environmental Models: Pest: Complete theory and what it means for modelling the real world.* Brisbane, Australia.: Watermark Numerical Computing.

Eggleston, J.R., S.A. Rojstaczer and J.J. Peirce. 1996. Identification of hydraulic conductivity structure in sand and gravel aquifers: Cape Cod data set. *Water Resources Research,* 32 (5), 1209–1222.

Gailey, R.M., A.S. Crowe and S.M. Gorelick. 1991. Coupled process parameter estimation and prediction uncertainty using hydraulic head and concentration data. Advanced Groundwater Resources, 14 (5), 301-314.

Harbaugh, A.W. 2005, MODFLOW-2005, The U.S. Geological Survey modular ground-water model—the Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods 6-A16, variously p.

Hill, M.C., and C.R. Tiedeman. 2007. *Effective Groundwater Model Calibration*. Hoboken, N.J.: John Wiley & Sons, Inc.

Isaaks, E.H., and R.M. Srivastava. 1989. *An Introduction to Applied Geostatistics*. New York, N.Y.: Oxford University Press

Lee, S-Y, S.F. Carle, and G.E. Fogg. 2007. Geologic heterogeneity and a comparison of two models: Sequential Gaussian and transition probability-based geostatistical simulation. *Advances in Water Resources*, 30, 1914–1932.

Pollock, D.W. 2012. *User Guide for MODPATH Version 6—A Particle-Tracking Model for MODFLOW*: U.S. Geological Survey Techniques and Methods 6–A41, 58 p.

Remy, N., A. Boucher, and J. Wu. 2009. *Applied Geostatistics with SGeMS: A User's Guide*. New York, N.Y: Cambridge University Press.

Sonnenborg, T.O., P. Engesgaard and D. Rosbjerg. 1996. Contaminant transport at a waste residue deposit 1. Inverse flow and nonreactive transport modeling. *Water Resources Research*, 32 (4), 925–938.

Strecker, E.W., and W. Chu. 1986. Parameter Identification of a Groundwater Contaminant Transport Model. Groundwater, 24 (1), 56-62.

Wagner, B.J., and S.M. Gorelick. 1987. Optimal Groundwater Quality Management Under Parameter Uncertainty. *Water Resources Research,* 23 (7), 1162–1174.

U.S. Geological Survey. The USGS Water Science School, Water Science Photo Gallery. URL: https://www.eeducation.psu.edu/earth111/node/911, last accessed December 1, 2016.

U.S. Geological Survey. The USGS Water Science School, Groundwater Basics. Aquifers and Groundwater. URL: http://water.usgs.gov/edu/earthgwaquifer.html, last accessed December 1, 2016.

U.S. Geological Survey. Pest++, Wisconsin. Pest++ Version 3.5 Input Instructions. URL: http://wi.water.usgs.gov/models/pestplusplus/, last accessed Nov 30, 2016.

U.S. Geographical Survey. The USGS Water Science School, Groundwater Use in the United States. URL: http://water.usgs.gov/edu/wugw.html, last accessed December 1, 2016.

Voss, C.I. 2011. Editor's message: Groundwater modeling fantasies –part1, adrift in the details. *Hydrogeological Journal*, 19, 1281–1284.

Voss, C.I. 2011. Editor's message: Groundwater modeling fantasies –part2, down to earth. *Hydrogeological Journal*, 19, 1455–1458.

# 6 Appendix A Example Flopy Routine Calls for MODFLOW and MODPATH

```
# Modflow-NWT set up and run - change modelname for new runs
# Create model files and run modflow
# July 11 changed version to 'mfnwt' for upgrade to python 3.5 & new flopy
ml = fp.modflow.Modflow(modelname='Flow1',version = 'mfnwt', exe_name ='C:/WRDAPP/MODFLOW-NWT_1.0.9/bin/MODFLOW-NWT_64.
exe')
name = 'Flow1'
# Inititialize variables
h = 1
k = 1
h1 = 100
bot = -30
delrow = 1
delcol = 1
hk = data
# Set head for start of flow
head_left = 100.
# Set head for end of flow
head_right = 50.
# Set iniital head for all cells
head_initial_all = 75.

# Feb 16 Create high K at left and right boundaries - 2 orders of mag > maximum input K
maxdata = data.max()
hk[:, 0] = maxdata*100
hk[:, -1]= maxdata*100

# Call Modflow dis routine to create .dis file
dis = fp.modflow.ModflowDis(ml, nlay = 1, nrow=nrow, ncol = ncol, delr = delrow, delc=delcol, top = 0, botm = bot, layc
bd=0)
#
ibound = np.ones((nlay, nrow, ncol), dtype=np.int32)
ibound[:, :, 0] = -1
ibound[:, :, -1] = -1

# Create array for starting head values
strt = np.ones((nlay, nrow, ncol), dtype=np.float32)
# Feb 7 - changed initial to 75
# Change starting head for all cells to initial value
strt[:,:,:] = head_initial_all
# Then change starting head for column 0 to head_head
strt[:, :, 0] = head_left
# Then change starting head for last column to head_right
strt[:, :, -1] = head_right
```

Figure A.1: Example of setting up variables for Flopy MODFLOW calls and Flopy call to setup flow Discretization Package (Dis).

```
# Call Modflow Basic routine to create .bas file
bas = fp.modflow.ModflowBas(model=ml,ibound=ibound, strt=strt,ixsec=False,ichflg=True, stoper=None, hnoflo=-999.99,exte
nsion = 'bas', unitnumber=13)

# July 11 - replaced with Jeff's command but kept my format for chedfm
# July 16 - changed chedfm from formatted to None to hook up with new approach to getting obs
oc = fp.modflow.ModflowOc(ml, ihedfm=0, iddnfm=0, chedfm=None, cddnfm=None, cboufm=None,
                          compact=True, stress_period_data={(0, 0): ['save head', 'save budget']},
                          extension=['oc', 'hds', 'ddn', 'cbc'], unitnumber=[14, 51, 52, 53])
oc
# Dec 14 Changed head tolerance to .001

# Call Modflow Nwt routine to create .nwt file
nwt = fp.modflow.ModflowNwt(ml, headtol=0.001, fluxtol=500, maxiterout=100,
                            thickfact=1e-05, linmeth=2, iprnwt=1, ibotav=0, options='COMPLEX')

# Dec 14 Changed layer type to 0

# July 11 changed 'iupwcb=53' to 'ipakcb=53'
# Call Modflow upw routine to create .upw file
upw = fp.modflow.ModflowUpw(ml, laytyp=0, layavg=0, chani=1.0, layvka=1, laywet=0, ipakcb=53,
                            hdry=1, iphdry=1, hk=hk, hani=1.0, vka=1.0,
                            noparcheck=False, extension='upw',
                            unitnumber=31)
#
ml.write_input()

# Dec 14 changed run_model2 to run_model
# Run Modflow with files created above
success,output = ml.run_model()
```

Figure A.2: Example of MODFLOW Flopy calls to create remaining flow packages: Basic (Bas), Output Control (Oc), Newton Solver (Nwt), and Upstream Weighting (Upw) and to run MODFLOW (run_model).

```
warning: assuming SpatialReference units are meters
FloPy is using the following executable to run the model: C:/WRDAPP/MODFLOW-NWT_1.0.9/bin/MODFLOW-NWT_64.exe

                                MODFLOW-NWT-SWR1
    U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUNDWATER-FLOW MODEL
                            WITH NEWTON FORMULATION
                            Version 1.0.9 07/01/2014
                   BASED ON MODFLOW-2005 Version 1.11.0 08/08/2013

                    SWR1 Version 1.03.0 08/30/2013

 Using NAME file: Flow1.nam
 Run start date and time (yyyy/mm/dd hh:mm:ss): 2016/08/08 19:16:58

 Solving:  Stress period:     1    Time step:     1    Groundwater-Flow Eqn.
 Run end date and time (yyyy/mm/dd hh:mm:ss): 2016/08/08 19:17:54
 Elapsed run time: 56.660 Seconds

  Normal termination of simulation
```

```
print("ml_run_model = %s" % success)
```

```
ml_run_model = True
```

Figure A.3: Output after above code is run inside Jupyter Notebook (see complete Jupyter Notebook: RunModflowGetHeadObsErr.ipynb for more details).

```
# prepare Modpath files
SimulationType = 1 # 1 endpoint; 2 pathline; 3 timeseries
TrackingDirection = 2 # 1 forward; 2 backward
WeakSinkOption = 1 # 1 pass; 2 stop
WeakSourceOption = 1 # 1 pass; 2 stop
ReferemceTimeOption = 2 # 1 time value; 2 stress period, time step, relative offset
StopOption = 2 # 1 stop with simulation 2; extend if steady state 3; specify time
ParticleGenerationOption = 2 # 1 automatic; 2 external file
TimePointOption = 1 # 1 none; 2 number at fixed intervals; 3 array
BudgetOutputOption = 2 # 1 none; 2 summary; 3 list of cells; 4 trace mode
ZoneArrayOption = 1 # 1 none; 2 read zone array(s)
RetardationOption = 1 # 1 none; 2 read array(s)
AdvectiveObservationsOption = 1 # 1 none; 2 saved for all time pts 3; saved for final time pt

options = [SimulationType, TrackingDirection, WeakSinkOption, WeakSourceOption, ReferemceTimeOption,
           StopOption, ParticleGenerationOption, TimePointOption, BudgetOutputOption, ZoneArrayOption,
           RetardationOption, AdvectiveObservationsOption]
#July 19 change approach to getting dis file name so maybe can get rid of the call to dis
#dis_file = dis.file_name[0]
dis_file = '{}.dis'.format(ml.name)
head_file = '{}.hds'.format(ml.name)
bud_file = '{}.cbc'.format(ml.name)
```

Figure A.4: Example of setting up variables for Flopy MODPATH calls.

```
#mp = fp.modpath.Modpath(modelname=ml.name, modflowmodel=ml, dis_file=dis_file,
#                         head_file=head_file, budget_file=bud_file)
mp = fp.modpath.Modpath(modelname=ml.name, modflowmodel=ml, dis_file=dis_file, head_file=head_file, budget_file=bud_fil
e)

mpnf = '{}.mpnam'.format(ml.name)
mplf = '{}.mplst'.format(ml.name)
mpef = '{}.mpend'.format(ml.name)

mpsim = fp.modpath.ModpathSim(mp, mp_name_file=mpnf,
                              mp_list_file=mplf,
                              option_flags=options,
                              ref_time_per_stp=[1, 1, 1.0],
#                               stop_zone=0,
#                               zone=zone_array,
                              extension='mpsim')

# July 19 added to detach modpath from modflow; kept def of ibound above with modflow pars
hnoflo = -999.99
hdry = -888.88
mpbas = fp.modpath.ModpathBas(mp, hnoflo=hnoflo, hdry=hdry, def_face_ct=0,
                              laytyp=0, ibound=ibound, prsity=0.20, prsityCB=0.20)

#mpbas = fp.modpath.ModpathBas(mp, hnoflo=bas.hnoflo, hdry=upw.hdry, def_face_ct=0,
#                              laytyp=0, ibound=ibound, prsity=0.20, prsityCB=0.20)

mp.write_input()
```

Figure A.5: Example of MODPATH Flopy calls to create transport packages: (ModpathSim) and Basic (Bas).

```
mpsf = '{}.mpsim'.format(ml.name)
# July 14 - changed name and location of .exe for modpath
mp_exe_name = 'E:/WRDAPP/modpath.6_0/bin/mp6x64.exe'
xstr = '{} {}'.format(mp_exe_name, mpsf)

exstat = os.system(xstr)
if exstat != 0:
    print ('MODPATH did not execute properly')
else:
    print ('MODPATH executed properly')
```

```
MODPATH executed properly
```

Figure A.6: Example of call to run modpath with result message shown when it runs properly.

See RunModflowModpathGetAgeObsErr Jupyter Notebook.

# 7 Appendix B Software and Hardware Reference

Table B.1: Software Applications.

| Software Applications | Description | Version | Link for download/executable |
|---|---|---|---|
| SGeMS | Stanford Geostatistical Modeling Software | V2.5b Build Sept 13, 2011 | http://sgems.sourceforge.net/<br><br>sgems-x64.exe |
| MODFLOW | USGS's 3-D Finite Difference Groundwater Model | 1.0.9 Downloaded June 23,2014 | http://water.usgs.gov/ogw/modflow/MODFLOW.html#downloads<br><br>MODFLOW-NWT.exe<br>MODFLOW-NWT_64.exe |
| MODPATH | USGS's particle-tracking post-processing program that uses MODFLOW output files to perform transport calculations. | Modpath.6_0 Downloaded July 14, 2016 | http://water.usgs.gov/ogw/modpath/#downloads<br><br>mp6x64.exe<br><br>E:WRDAPP/Modpath.6_0/bin/mp6x64.exe |
| Pest++ | Watermark Numerical Computing's Parameter Estimation (Calibration Program) – author John Doherty | Interim version with bug fixes from Jeremy White (pest++ support). Downloaded July 28, 2016 | http://wi.water.usgs.gov/models/pestplusplus/<br><br>Pest++.exe |

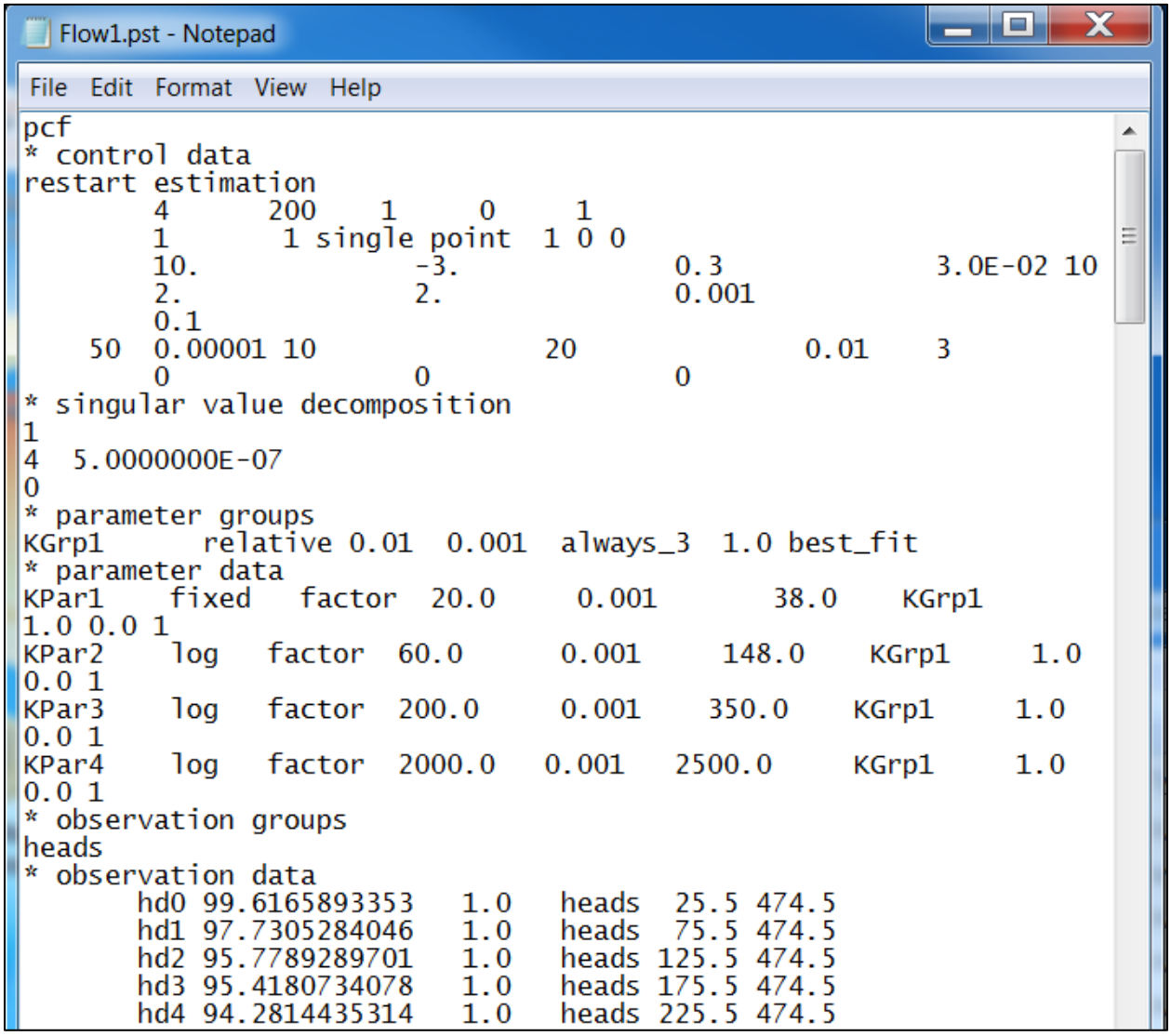
Table B.2: Programming Languages and Libraries.

| Programming Language/ Library | Description | Version | Link for Download/Executable |
|---|---|---|---|
| Python | Anaconda for Jupyter notebooks. Also includes a script editor named Spyder, which would remove the need for the Enthought Software | Version 3.5.1 Downloaded June 15, 2016 | https://www.continuum.io/downloads<br><br>Python.exe<br><br>E:users/vivian/Anaconda3 |
| Python | Enthought Canopy for viewing/editing python scripts | Version 1.5.1 (64-bit)<br><br>Downloaded Jan 21, 2015 | Canopy.exe<br><br>C:\Users\Vivian\AppData\Local\Enthought\Canopy\App\Canopy.exe |
| FloPy | Python library to create, run, and post-process MODFLOW-based models using a programming interface | Version 3.2.5 | http://modflowpy.github.io/flopydoc/ |
| NumPy | Python library for array operations and manipulation | Version 1.11 | https://docs.scipy.org/doc/numpy/reference/ |
| Panda | Python library for data analysis | ~Version 0.19.0 | http://pandas.pydata.org/ |

Table B.3: Hardware/OS Utilized for Research.

| Hardware/OS | Description | Version | |
|---|---|---|---|
| Prostar Laptop High Performance Computer | Gaming-style computer custom made by Prostar; 16GB RAM | PRP150SM-A-R-B P150SM-A, 15.6" FHD/MATT | http://www.pro-star.com/ ordered online; customer service assists in selection; purchased 8/2014 ($1858.00) |
| Intel Core i7-4810MQ CPU | 4 multi-processor CPU | 2.8 GHz | |
| Windows 7.0 Professional | 64-bit | Service Pack 1 | |

# 8 Appendix C Example PEST++ Control and Record Files




…hd4 through hd192…

```
     hd195 66.2764758714   1.0   heads 775.5  24.5
     hd196 65.2833142212   1.0   heads 825.5  24.5
     hd197 63.7619141421   1.0   heads 875.5  24.5
     hd198 59.2436067805   1.0   heads 925.5  24.5
     hd199 52.2080939377   1.0   heads 975.5  24.5

* model command line
E:/users/vivian/anaconda3/python RunParsModflow1v4.py
* model input/output
        Flow1.tplt Flow1_hkpars.txt
        Flow1.inst Flow1_hdobs.txt
++N_ITER_BASE 1
++N_ITER_SUPER 5
++SUPER_EIGTHRES 0.01
```

Figure C.1: Example Pest++ Control File used for Sequential Flow Calibration Run. See Run5anewObsErr3Final/master/Flow1.pst for complete file.

Flow1.rec - Notepad

File Edit Format View Help

```
             PEST++ Version 3.3.0

                by Dave Welter
     Computational Water Resource Engineering


using control file: "Flow1.pst"


PEST++ run mode:-
   estimation

Case dimensions:-
    Number of parameters = 4
    Number of adjustable parameters = 3
    Number of observations = 200
    Number of prior estimates = 0

Model command line(s):-
    E:/users/vivian/anaconda3/python RunParsModflow1v4.py

Model interface files:-
    template files:
      Flow1.tplt
    model input files:
      Flow1_hkpars.txt

    instruction files:
      Flow1.inst
    model output files:
      Flow1_hdobs.txt


PEST Control Information
    relparmax = 2
    facparmax = 2
    facorig = 0.001
    phiredswh = 0.1
    noptmax = 50
    phiredstp = 1e-05
    nphistp = 10
    nphinored = 20
    relparstp = 0.01
    nrelpar = 3
```

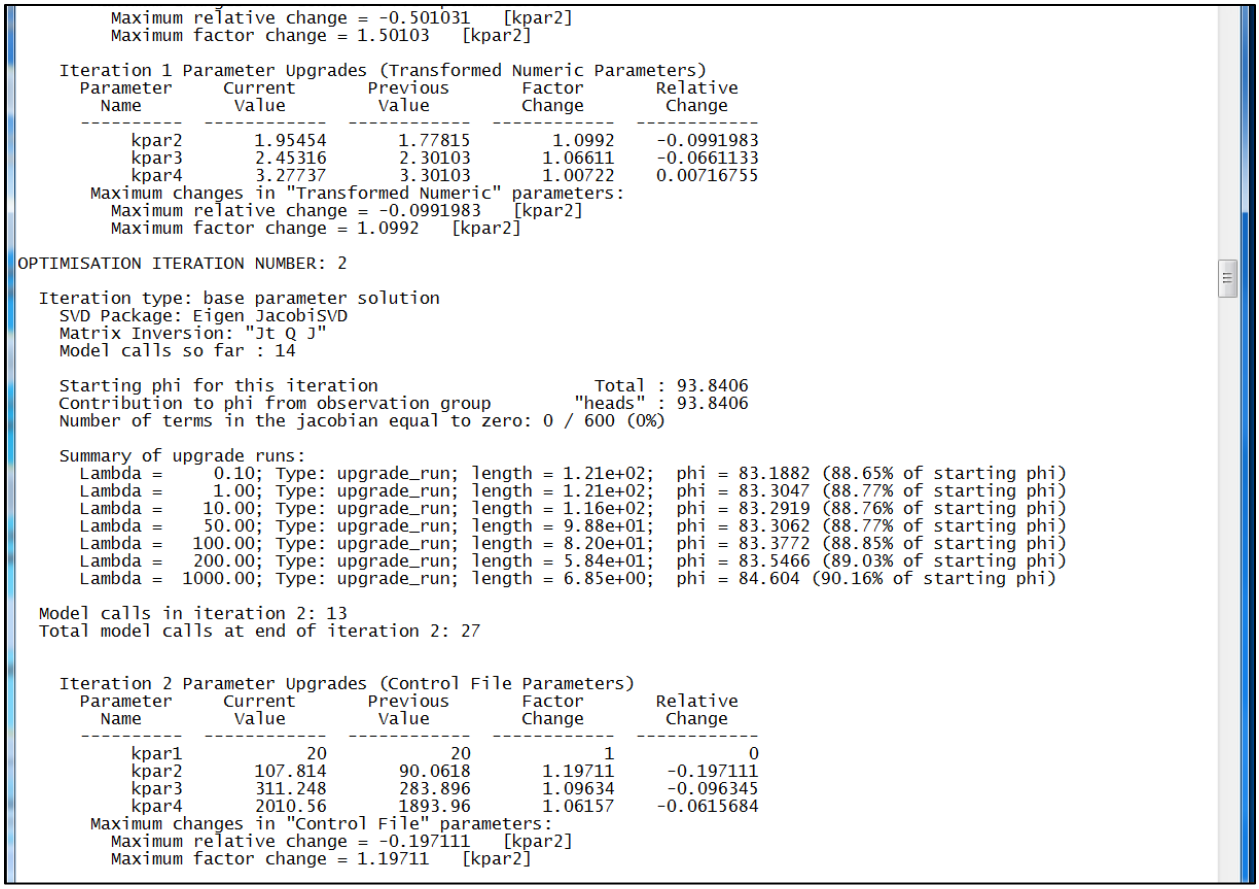
Figure C.2a: Example Pest++ Record File (part a), output from Flow Calibration Run. See Run5anewObsErr3Final/master/Flow1.rec for complete file.

```
PEST++ Options
    n_iter_base = 50
    n_iter_super = 0
    max_n_super = 4
    super eigthres = 5e-07
    svd pack = 0
    auto norm = -999
    super relparmax = 0.1
    max super frz iter = 5
    mat inv = 1
    max run fail = 3
    max reg iter = 20
    lambdas =
            0.1
              1
             10
            100
           1000
    uncertainty flag = 1
    parameter covariance file =
    forecast names =
    derivative run failure forgive = 1
    run overdue reschedule factor = 1.15
    run overdue giveup factor = 100
    base parameter jacobian filename =
    use prior parameter covariance upgrade scaling = 0


Parameter group information
NAME            INCREMENT TYPE     DERIVATIVE INCREMENT    INCREMENT LOWER BOUND  FORCE CENTRAL      INCREMENT
MULTIPLIER
kgrp1                RELATIVE                     0.01                   0.001       ALWAYS_3
  1

Parameter information
NAME           TRANSFORMATION        CHANGE LIMIT  INITIAL VALUE    LOWER BOUND    UPPER BOUND          GROUP
  SCALE         OFFSET  DERIVATIVE COMMAND
kpar1          fixed                   FACTOR             20          0.001             38          kgrp1
  1             0                   0
kpar2          log                     FACTOR             60          0.001            148          kgrp1
  1             0                   0
kpar3          log                     FACTOR            200          0.001            350          kgrp1
  1             0                   0
kpar4          log                     FACTOR           2000          0.001           2500          kgrp1
  1             0                   0

Observation information
NAME                                  VALUE               GROUP              WEIGHT
hd0                                 99.6166               heads                   1
hd1                                 97.7305               heads                   1
hd2                                 95.7789               heads                   1
hd3                                 95.4181               heads                   1
hd4                                 94.2814               heads                   1
```

Figure C.2b: Example Pest++ Record File (part b), output from Flow Calibration Run.

```
hd195                                   66.2765             heads                   1
hd196                                   65.2833             heads                   1
hd197                                   63.7619             heads                   1
hd198                                   59.2436             heads                   1
hd199                                   52.2081             heads                   1

Prior information

   no prior information provided

PEST SVD Information
    maxsing = 4
    eigthresh = 5e-07


   -----    Starting PEST++ Iterations    ----


OPTIMISATION ITERATION NUMBER: 1

  Iteration type: base parameter solution
    SVD Package: Eigen_JacobiSVD
    Matrix Inversion: "Jt Q J"
    Model calls so far : 0

    Starting phi for this iteration                    Total : 207.697
    Contribution to phi from observation group        "heads" : 207.697
    Number of terms in the jacobian equal to zero: 0 / 600 (0%)

    Summary of upgrade runs:
      Lambda =     0.10; Type: upgrade_run; length = 1.23e+02;  phi = 95.8091 (46.13% of starting phi)
      Lambda =     1.00; Type: upgrade_run; length = 1.23e+02;  phi = 95.3697 (45.92% of starting phi)
      Lambda =    10.00; Type: upgrade_run; length = 1.20e+02;  phi = 94.8647 (45.67% of starting phi)
      Lambda =    20.00; Type: upgrade_run; length = 1.18e+02;  phi = 94.5461 (45.52% of starting phi)
      Lambda =    40.00; Type: upgrade_run; length = 1.20e+02;  phi = 94.0329 (45.27% of starting phi)
      Lambda =   100.00; Type: upgrade_run; length = 1.39e+02;  phi = 93.8406 (45.18% of starting phi)
      Lambda =  1000.00; Type: upgrade_run; length = 2.08e+02;  phi = 105.024 (50.57% of starting phi)

  Model calls in iteration 1: 14
  Total model calls at end of iteration 1: 14


    Iteration 1 Parameter Upgrades (Control File Parameters)
      Parameter     Current       Previous       Factor       Relative
        Name         Value         Value         Change        Change
      ----------  ------------  ------------  ------------  ------------
           kpar1            20            20             1             0
           kpar2       90.0618            60       1.50103     -0.501031
           kpar3       283.896           200       1.41948     -0.419478
           kpar4       1893.96          2000       1.05599     0.0530224
       Maximum changes in "Control File" parameters:
```

Figure C.2c: Example Pest++ Record File (part c), output from Flow Calibration Run.



```
        Maximum relative change = -0.501031   [kpar2]
        Maximum factor change = 1.50103   [kpar2]

    Iteration 1 Parameter Upgrades (Transformed Numeric Parameters)
      Parameter     Current       Previous       Factor       Relative
        Name         Value         Value         Change        Change
      ----------  ------------  ------------  ------------  ------------
           kpar2       1.95454       1.77815        1.0992    -0.0991983
           kpar3       2.45316       2.30103       1.06611    -0.0661133
           kpar4       3.27737       3.30103       1.00722    0.00716755
       Maximum changes in "Transformed Numeric" parameters:
         Maximum relative change = -0.0991983   [kpar2]
         Maximum factor change = 1.0992   [kpar2]

OPTIMISATION ITERATION NUMBER: 2

  Iteration type: base parameter solution
    SVD Package: Eigen JacobiSVD
    Matrix Inversion: "Jt Q J"
    Model calls so far : 14

    Starting phi for this iteration                    Total : 93.8406
    Contribution to phi from observation group        "heads" : 93.8406
    Number of terms in the jacobian equal to zero: 0 / 600 (0%)

    Summary of upgrade runs:
      Lambda =     0.10; Type: upgrade_run; length = 1.21e+02;  phi = 83.1882 (88.65% of starting phi)
      Lambda =     1.00; Type: upgrade_run; length = 1.21e+02;  phi = 83.3047 (88.77% of starting phi)
      Lambda =    10.00; Type: upgrade_run; length = 1.16e+02;  phi = 83.2919 (88.76% of starting phi)
      Lambda =    50.00; Type: upgrade_run; length = 9.88e+01;  phi = 83.3062 (88.77% of starting phi)
      Lambda =   100.00; Type: upgrade_run; length = 8.20e+01;  phi = 83.3772 (88.85% of starting phi)
      Lambda =   200.00; Type: upgrade_run; length = 5.84e+01;  phi = 83.5466 (89.03% of starting phi)
      Lambda =  1000.00; Type: upgrade_run; length = 6.85e+00;  phi = 84.604 (90.16% of starting phi)

  Model calls in iteration 2: 13
  Total model calls at end of iteration 2: 27


    Iteration 2 Parameter Upgrades (Control File Parameters)
      Parameter     Current       Previous       Factor       Relative
        Name         Value         Value         Change        Change
      ----------  ------------  ------------  ------------  ------------
           kpar1            20            20             1             0
           kpar2       107.814       90.0618       1.19711     -0.197111
           kpar3       311.248       283.896       1.09634     -0.096345
           kpar4       2010.56       1893.96       1.06157    -0.0615684
       Maximum changes in "Control File" parameters:
         Maximum relative change = -0.197111   [kpar2]
         Maximum factor change = 1.19711   [kpar2]
```

Figure C.2d: Example Pest++ Record File (part d), output from Flow Calibration Run. See Run5anewObsErr3Final/master/Flow1.rec for complete file.