PARAMETRIC UNCERTAINTY ANALYSIS OF URANIUM TRANSPORT SURFACE COMPLEXATION MODELS

By

GEOFFERY L. MILLER

A Thesis submitted to the Department of Scientific Computing in partial fulfillment of the requirements for the degree of Master of Science

Degree Awarded: Spring Semester, 2011
The members of the committee approve the thesis of Geoffrey L. Miller defended on November 18, 2010.

Ming Ye  
Professor Directing Thesis

Robert van Engelen  
Committee Member

Tomasz Plewa  
Committee Member

Approved:

Max Gunzburger, Chair, Department of Scientific Computing

Joseph Travis, Dean, College of Arts and Sciences

The Graduate School has verified and approved the above-named committee members.
# TABLE OF CONTENTS

List of Tables........................................................................ v
List of Figures...................................................................... vi
Abstract.............................................................................. viii

1 Introduction........................................................................
1.1 Laboratory Experiments and Numerical Models of Uranium Transport........ 2
  1.1.1 Column Experiment .................................................. 2
  1.1.2 Surface Complexation Models ................................. 2
  1.1.3 SCM Parameters .................................................... 3
1.2 Objective Function ........................................................ 4
1.3 Parametric Uncertainty .................................................... 5
  1.3.1 Linear and Nonlinear Analysis ................................. 7
  1.3.2 Naive Monte Carlo Application .............................. 8
  1.3.3 Parametric Domain and Prediction Interval Results ....... 9
1.4 Scope and Objectives ....................................................... 13

2 Global Optimization of the Computational Model............................. 15
  2.1 Multi-start BFGS Method ............................................ 15
    2.1.1 DAKOTA Application ........................................... 16
  2.2 Results of Global Optimization .................................... 17
  2.3 Parameter Variance-Covariance Matrix Approximation ............... 19
    2.3.1 Application ....................................................... 19
    2.3.2 Test Case Validation ........................................... 20
    2.3.3 Variance-Covariance Matrix ................................ 22

3 Morris Method Sensitivity Analysis ........................................ 28
  3.1 Morris Method Equations ............................................ 29
  3.2 DAKOTA Application .................................................. 30
    3.2.1 Test Case Validation ........................................... 30
    3.2.2 Analysis for a Flow Study .................................... 31
    3.2.3 Model C4 Analysis .............................................. 32
  3.3 Morris Method Results ............................................... 32

4 Markov Chain Monte Carlo Analysis ....................................... 35
  4.1 Metropolis-Hastings Random-Walk Algorithm ....................... 35
    4.1.1 The Algorithm .................................................. 35
    4.1.2 Likelihood Function .......................................... 37
LIST OF TABLES

2.1 Model C4 Calibrated Parameters .................................................. 18
2.2 Approximate Fisher Test Case Data ................................................. 21
2.3 Fisher Information Matrix Solutions: Analytical and Numerical .......... 22
4.1 MCMC Special Considerations ....................................................... 46
5.1 Test Realization Points, Extreme with High Likelihood .................. 59
5.2 Test Realization Points, Close with Low Likelihood ....................... 61
5.3 Lilliefors Hypothesis Test Results ............................................... 65
5.4 Chain Acceptance Ratios and Scaling Parameters ......................... 66
# LIST OF FIGURES

1.1 Kohler et al. (1996) Surface Complexation Column Model C4 .......................................................... 4
1.2 Linear and nonlinear intervals for model C4 ......................................................................................... 7
1.3 Parallel Monte Carlo Application ........................................................................................................ 10
1.4 Monte Carlo Test 1 Interval Results ...................................................................................................... 10
1.5 Monte Carlo Test 2 Interval Results ...................................................................................................... 11
1.6 log(K1) Response Surface for Experiments 1,8 ................................................................................... 11
2.1 DAKOTA Input File for Global Calibration ......................................................................................... 17
2.2 DAKOTA Application Structure .......................................................................................................... 18
2.3 Local-UCODE Calibrated Model Fit ..................................................................................................... 24
2.4 Global-DAKOTA Calibrated Model Fit ................................................................................................ 25
2.5 Weighted Model Fit .............................................................................................................................. 26
2.6 Model C4 Approximated Variance-Covariance Matrix ..................................................................... 26
2.7 Model C4 Approximated Correlation Matrix ....................................................................................... 27
3.1 MOAT Test Case [Adams et al., 2010] ................................................................................................. 31
3.2 MOAT Results Yucca Flat .................................................................................................................. 32
3.3 DAKOTA Input File for MOAT Analysis ............................................................................................... 33
3.4 MOAT Results Model C4 ................................................................................................................... 34
4.1 Parameter Range Isotherm Test ........................................................................................................... 40
4.2 Steps to solve a multivariate normal PDF [Burkardt, 2009] ............................................................... 41
4.3 Histogram Comparison of Custom Library vs. IMSL ...................................................................... 42
4.4 Probability Plot Comparison of Custom Library vs. IMSL ............................................................... 43
4.5 Model Runner Application .................................................................................................................. 46
ABSTRACT

Parametric uncertainty analysis of surface complexation modeling (SCM) has been studied using linear and nonlinear analysis. A computational SCM model was developed by Kohler et al. (1996) to simulate the breakthrough of Uranium(VI) in a column of quartz. Calibration of parameters which describe the reactions involved during reactive-transport simulation has been found to fit experimental data well. Further uncertainty analysis has been conducted which determines the predictive capability of these models. It was concluded that nonlinear analysis results in a more accurate prediction interval coverage than linear analysis. An assumption made by both linear and nonlinear analysis is that the parameters follow a normal distribution. In a preliminary study, when using Monte Carlo sampling a uniform distribution among a known feasible parameter range, the model exhibits no predictive capability. Due to high parameter sensitivity, few realizations exhibit accuracy to the known data. This results in a high confidence of the calibrated parameters, but poor understanding of the parametric distributions. This study first calibrates these parameters using a global optimization technique, multi-start quasi-newton BFGS method. Second, a Morris method (MOAT) analysis is used to screen parametric sensitivity. It is seen from MOAT that all parameters exhibit nonlinear effects on the simulation. To achieve an approximation of the simulated behavior of SCM parameters without the assumption of a normal distribution, this study employs the use of a Covariance-Adaptive Monte Carlo Markov chain algorithm. It is seen from posterior distributions generated from accepted parameter sets that the parameters do not necessarily follow a normal distribution. Likelihood surfaces confirm the calibration of the models, but shows that responses to parameters are complex. This complex surface is due to a nonlinear model and high correlations between parameters. The posterior parameter distributions are then used to find prediction intervals about an experiment not used to calibrate the model. The predictive capability of Adaptive MCMC is found to be better than that of linear and non-linear analysis, showing a better understanding of parametric uncertainty than previous study.
CHAPTER 1

INTRODUCTION

This study investigates parametric uncertainty of a groundwater reactive transport model. This model is an abstraction of advection, dispersion, and chemical reactions through a groundwater domain. The model was developed by Kohler et al. (1996) to predict the breakthrough of Uranium(VI) in column experiments through a quartz medium. The models are probabilistic, as the response is subject to variability depending on variables input to the model. The input parameters to the model describe the chemical equilibrium constants driving the chemical reaction simulations. The response is described by the concentration of U(VI) seen exiting a column. Calibration of these variables is achieved by minimizing an objective function including the difference between the simulation and observation and a weighting variable to the measurement accuracy during experimental observation. In our study, we use the calibration to find the most likely parameters.

While calibration of the model provides what appears to be an acceptable simulation, the reliability of the predictions using the calibrated parameters is unknown. This parametric uncertainty was first quantified using linear and nonlinear analysis. Second, a naive Monte-Carlo sampling was used in attempt to explain deficiencies in the linear and nonlinear analysis. A global calibration was completed to verify previous model calibrations. The Morris-One-At-a-Time method is used to screen the sensitivity of the parameters with respect to the objective function. Finally numerical approximation of the Fisher information matrix is used to create proposal distributions to drive numerous adaptive Monte-Carlo Markov Chain simulations to understand the posterior parameter distribution.
1.1 Laboratory Experiments and Numerical Models of Uranium Transport

Kohler et al. (1996) describes the column experiments in which water can flow and chemical species, e.g. Uranium U(VI), are transported. U(VI) is adsorbed by the mineral media. The adsorption of U(VI) causes a retardation effect in the rate of transport through the groundwater system. A series of column experiments were conducted and subsequently simulated using surface complexation models.

1.1.1 Column Experiment

Six column experiments were conducted by Kohler et al. (1996) using 99% purified quartz as the mineral media and solutions of Uranium(VI). The experiment is first given a baseline pulse of water to fill the column to determine the pore volume for the media. The pore volume is defined as the volume which occupies the pore space between particles of quartz. The solutions pulsed through the column are maintained at a steady flow rate. This allows the experiment to calibrate pore volumes of solution exiting the column to the elapsed time of the experiment. It is assumed that the low concentration of Uranium(VI) does not affect the flow rate of the solution through the column. The solution of Uranium(VI) is pulsed through a column of quartz using the same constant flow rate.

The goal of the experiment is to observe the breakthrough of U(VI) exiting the column over the course of several pore volumes of water flushing over the first pulse of U(VI). This breakthrough curve of the concentration of U(VI) shows the retardation effect of reactive-transport. For a system with a high affinity for reactions with the metal ion, the initial pore volumes may indicate clean volumes of water. Depending on the equilibrium constants involved in the assumed reactions through the geochemical system, the U(VI) will exit faster or slower.

1.1.2 Surface Complexation Models

Previous studies cited by Kohler et al. (1996) suggest using an isotherm constant to describe the adsorption of U(VI). This approach, inspired by chemical thermodynamics, is limited in its ability to vary the isothermal constants in the system. A constant isotherm requires the assumption of constant chemical conditions: e.g., pH, temperature, aqueous speciation, and loading rates. This
approach does not account for more realistic situations where chemical properties vary in time and space, consequently varying isotherm values throughout the domain.

Surface Complexation Modeling (SCM) captures the dynamic nature of chemical reactions by describing equilibrium and kinetic chemical reactions in a reaction network. The reaction network is solved as a coupled equation with groundwater transport. As chemical species are transported through the domain, the concentrations of the species change, subsequently changing the concentrations of species produced from chemical reactions. This study uses RATEQ a reactive-transport simulator developed by Curtis (2005). RATEQ makes use of MODFLOW-2000 (Harbaugh et al., 2000) and MT3DMS (Zheng and Wang, 1999) for the simulation of groundwater flow and simulate the transport of chemical species, respectively. Finally, RATEQ solves multi-component geochemical reactions.

1.1.3 SCM Parameters

It is unknown what reactions occur within the geochemical system. The reactions correspond to functional groups on the surface reaction sites. Kohler et al. (1996) tested 7 column models, C1-C7. The column model C1 is a 1-site model where only one functional group is considered. Models C2, C3, C4 consider 2 reaction sites. Models C5, C6, C7 consider three reaction sites. These functional group is reacted with U(VI) to create an immobile chemical species. All of the reactions in these models are equilibrium reactions. This requires the knowledge of equilibrium constants, or affinity, to solve the geochemical reaction network. It is also required to know the fraction of the reaction surface which correlates to each functional group. $S_iOH$ in Table 1.1 describes the $i^{th}$ functional groups in model. The site fraction $f$ and equilibrium constants are unknown values which must be calibrated to the observational data.

The equilibrium constants $K_1$, $K_2$, and $K_3$ are described in the log space as $log(K_1)$, $log(K_2)$, and $log(K_3)$. This is similar to pH, a common parameter for groundwater models. pH is described as a base 10 logarithm of hydrogen ion activity. This study refers to the site fraction parameter as Sites or $f$, also represented in the log space.

This study will investigate parametric uncertainty as it relates to the computational Kohler et al. (1996) model C4. C4 was chosen as the primary model to investigate as it is observed through previous calibration and studies by Tebes-Stevens et al. (2001) and Kohler et al. (1996) to
be the best in terms of model fitting and model complexity.

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Functional Group</th>
<th>Parameter Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1OH + UO_2^{2+} + H_2O = S_1OUO_2OH + 2H^+$</td>
<td>$S_1OH$</td>
<td>$\log(K1)$</td>
</tr>
<tr>
<td>$S_1OH + UO_2^{2+} + H_2O = S_2OUO_2OH + 2H^+$</td>
<td>$S_2OH$</td>
<td>$\log(K2)$</td>
</tr>
<tr>
<td>$S_1OH + UO_2^{2+} = S_2OUO_2^+ + H^+$</td>
<td>$S_2OH$</td>
<td>$\log(K3)$</td>
</tr>
<tr>
<td>Site Fraction</td>
<td>$Sites, f$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.1: Kohler et al. (1996) Surface Complexation Column Model C4

## 1.2 Objective Function

An objective function is used to quantify how well any simulation fit the observational data. The objective function serves as a value which describes how well the computational model fits an experiment. The objective function used for the purposes of calibration and parametric uncertainty analysis is the sum of squares weighted residuals (SSWR). SSWR describes the predictive accuracy of the computational model to experimental observation. The weights included in the objective function are related to the error estimates of measurement error. These measurement errors are described as the standard deviation around the observational data point. This error is assumed to be normally distributed with the observed data point as the mean. Numerical error in the simulations is assumed to be small and ignored in this study.

Three independent experiments known as 1, 2, and 8 from Kohler et al. (1996) are used in this metric. Experiments 1, 2, and 8 contain 39, 32, and 49 data points, respectively. The residual of these data points is summed, satisfying the requirement for the metric to depend on the number of experimental measurements. Each of these data points has a respective standard deviation, indicating the observational error. There is no level of satisfaction included in this validation metric, satisfying the final requirement.

Each experiment has its own SSWR objective function value. To combine these into one objective function, each experiment was assigned a group weighting based on how many experimental measurements it contributes to the total 120 measurements. Each experiment is considered as its own group of data, summarizing a total of 3 groups. Therefore, experiments 1, 2, and 8 have the group weights 120.0/39.0, 120.0/32.0, and 120.0/49.0, respectively. The entirety of this objective
function is summarized with the following equations where $\sigma = \text{is the standard deviation of measurement error for a given data point}$, $N = \text{is the total number of data points}$, $n_g = \text{is the number of data points in group } g$, $\theta = \text{is the observed experimental data}$, and $s = \text{is the computationally simulated data}$.

$$w_i = \frac{1}{\sigma_i^2} \quad (1.1)$$

$$SSWR_g = \sum_{i=1}^{n_g} w_i (\theta_i - s_i)^2 \quad (1.2)$$

$$\phi = \sum_{g=1}^{G=3} \frac{N}{n_g} SSWR_g \quad (1.3)$$

This objective function $\phi$ is used throughout the rest of this study as the objective function.

### 1.3 Parametric Uncertainty

Parametric uncertainty refers to the confidence one has in the parameters chosen to drive a computational model. As described previously, there are two sets of distinct parameters in this study. The reaction equilibrium constants for modeled reactions and the site fraction describing the amount of the surface of the reaction sites assigned to a particular ligand. These parameters are calibrated to match the model to the experimental data. Kohler et al. (1996) compared the goodness-of-fit using a consistent objective function over several models, but this is not the only work necessary to understand the usefulness of the model. These parameters are not only unknown, but even for a perfect calibration, there is not necessarily a true value which describes the state of the natural phenomenon. To be strict, since all models are simplifications which have error due to some bias, the parameters used to drive the models are simplifications of reality themselves. From this, it can be deducted that there is no true value for a parameter. This is the most basic motivation for a thorough understanding of the parametric uncertainty involved with a model. It has been recognized for some time that uncertainties in reactive-transport models are significant. Understanding these uncertainties is necessary to determine the effectiveness of the model (Helton, 1993), (Gallagher and Doherty, 2007).
It is discussed by Rojas et al. (2008) that a good calibration does not imply a good model. It is also mentioned by Rojas et al. (2008) that parametric uncertainty does not account for error which occurs as a result of the conceptual model. These concepts are by no means unique to this specific problem. Any conceptual model is a simplification which deviates from the reality of a naturally observed phenomenon. Hydrogeological models have an especially difficult task, as data collection under the ground is especially expensive and sparse over a large domain.

Parametric uncertainty can be quantified using a prediction interval. The prediction interval encompasses error from two sources. First the prediction interval includes the error residual between simulation and experiment propagated from the uncertainty of the parameter estimation. A variance-covariance matrix associated with a set of parameters describes the confidence of a given parameter estimation. In this study, this was estimated by numerically calculating the Fisher information matrix. Secondly the prediction interval includes the random error from the measurement of the predicted value (Hill and Tiedeman, 2007). In order to find a confidence interval of a prediction, the random error from the measurement of the predicted value must be known, which requires more work than will be investigated by the scope of this study.

The lower and upper bounds of a prediction interval describe the range which the model is able to simulate. To consider a model to be correct, it must have a complete predictive cover of experimental data. If an experimental observation is outside of the simulated prediction interval, there is a function in nature which is not captured by the simulation. If a feasible parameter value is used and results in an poor prediction, this also indicates the model is flawed. An ideal prediction interval would have a mean value close to the observed data and a prediction interval which encompasses all of the observed data when a feasible parameter range and distribution is used.

To perform this analysis the distribution of the parameters must be known. Parametric distributions can take on an assumed generalized function such as an analytical Gaussian distribution. When it is clear that a parametric distribution cannot be assumed, Monte Carlo methods can be used to discover more complex distribution functions.
1.3.1 Linear and Nonlinear Analysis

Linear and nonlinear studies are regression analysis techniques that are employed to assess the predictive capability of a model. Both of these techniques assumed parameters follow a normal distribution. In a previous study completed by a collaborator, analysis using both the linear and nonlinear approaches was completed to generate prediction intervals obtained using an optimal parameter set. The linear study takes a calibrated parameter set as the mean and approximates the variance of each parameter around this mean. Assuming a symmetrical normal distribution using this mean and variance, prediction intervals are be generated. The nonlinear study uses equation [1.4] to create a parameter confidence region. Equation [1.4] is given with calibrated objective function $S(b')$, the error variance $s^2$ given a critical value and significance level derived from the statistical distribution chosen, and the accuracy of the prediction $a$ (Hill and Tiedeman 2007, p.178). This parameter confidence region, unlike the linear analysis, is not necessarily symmetric. The nonlinear confidence interval still assumes the parameters are normally distributed to find the boundary of the confidence region.

$$S(b) \leq S(b') + s^2 \cdot \text{critical}_\text{value} + a \quad \text{(1.4)}$$

Figure 1.2 shows the results from the previous linear and nonlinear prediction interval study on model C4. It is seen that both cover part of the predicted values of the breakthrough curve,
but neither provide an exhaustive predictive cover of the experimental data. The tightly bounded linear prediction interval which includes very little of the observational data shows that model is nonlinear and the linear analysis is improper. The large prediction interval in the nonlinear case, which also does not include all of the observational data, implies the possibility that the parameters follow a distribution with a complexity which cannot be generalized as normal.

### 1.3.2 Naive Monte Carlo Application

In a first attempt to understand the model by finding a more appropriate prediction interval, a naive Monte Carlo sampling simulation was proposed. An application was created on the FSU HPC which makes use of the MOAB job scheduler. A uniform distribution of 1000 realizations of the parameter sets was proposed for simulation.

Each realization in a naive Monte Carlo sampling is independent of any other realization, and therefore embarrassingly parallel. Three programs were created using Python 2.4 to create parameter distributions, dispatch sections of these parameter distributions over multiple compute nodes, and execute the computational model. Figure 1.3 shows the application execution path for this parallel model analysis.

The left column displays the first step of the naive MC analysis. First, the parameter sets to drive the computational model need to be generated. Given an input which describes the number of parameters, each parameters’ distribution, and the number of samples to generate, the Parameter Distribution Program (PDP) creates parameter input files with the number of realizations requested. The parameter distribution is described by the type of distribution to generate and specifications required for the distribution. For example, a uniform distribution requires a bounds $[min, max]$ for parameters to be selected. This program makes use of the Python 2.4 random library which provides univariate uniform and normal random number generators.

The MOAB Dispatcher Program (MDP) is a script written for the FSU High Performance Cluster (HPC) to handle the creation of multiple directories for each node to use. The computational model is copied into each node directory. The MDP then calculates the number of sequential iterations that each node is to perform, given the number of parameters generated by the PDP and the number of processors requested. As a graduate student user, one has access to a general access queue and a backfill queue. The general access queue allows for 14 days of computation. The
backfill queue only allows for 4 hours of computation. The advantage to the backfill queue is that it will make use of any idle nodes in the cluster, as it knows you will only use the node for 4 hours. The general access queue can have a longer wait time, as compute nodes may be requested with a complex architecture to use MPI or OpenMP for parallelization. This embarrassingly parallel problem does not require such a complex architecture, so the backfill queue is more desirable as it can be more responsive. The MDP can be configured to use either queue. In this study, the number of sequential iterations per node is configured to have a runtime less than 4 hours so that the backfill queue can be used.

The Model Execution Program (MEP) is then executed on every node as the MOAB workload manager grants them. The MEP uses its node number as a parameter to read the correct section of parameters to run sequentially. Each node runs its unique dataset in its own node directory which was created by the MDP. At the end of execution, all of the results can be analyzed using analysis programs written to collect the data that is distributed over the node directories. At the time of writing this application, each analysis program read in the data in a distributed fashion. Later in the development of the study, a program was created to collect all of the resulting simulation data into one data file, so that the node directories could be archived. Due to numerous directories and input files required to run the model, the node directories could total to gigabytes of disc space. After data collection, the result data file and the node directories are highly compressible into a few megabytes.

In this case, for 1000 realizations, 100 processors were selected to run 10 realizations each. With each realization taking up to 20 minutes, this reduces a worst-case computational time of about 14 days to only about 4 hours. The FSU HPC backfill queue allowed all 100 requested processor to run simultaneously and confirmed a computational time of less than 4 hours.

1.3.3 Parametric Domain and Prediction Interval Results

The parameter distribution is still an unknown quantity. With no proper distribution known to study, a uniform distribution was chosen centered about a known calibrated parameter set. Each parameter distribution is independent of the other. Choosing an appropriate parametric subspace to sample is an open question, which is not completely solved by this study. Two naive MC studies were performed using different parameter ranges based on provided calibrated parameters.
1. 10% varied about a given calibrated parameter.

2. 100% varied about a given calibrated parameter.
Figure 1.4 shows the results for MC test 1, where the parameters are varied 10% around a given calibrated parameter set. The 90% prediction interval is a band around the mean of simulations, but there is very little predictive coverage. The mean breakthrough curve does not represent the experimental data. This preliminary study indicates further that the parameter distribution must be understood to determine an accurate understanding of the model’s predictive capability.

The results of test 2 are presented Figure 1.5 which is surprising in comparison to Figure 1.4. The predictive coverage has improved, but the band neither follows the experimental data nor provides a feasible range of uncertainty around the data. The upper predictive bound does not cover the
peak concentration of U(VI) during experimentation. The upper bound also does not rise or drop off after this peak. The lower bound of prediction is zero, indicating no concentration of U(VI) was observed within the course of the simulation. This implies that a significant number of simulations predict no concentration of U(VI), which is in contradiction with the physical experimental data. The mean of the predictions, just as in Figure 1.4, does not mimic the breakthrough curve at all.

To investigate the behavior of the parameters over the ranges chosen, the first parameter was varied over the 100% range to generate a response surface for experiments 1 and 8. Figure 1.6 shows that the optimal calibration for one experiment is not the same as another. Further, it shows that the optimal calibration for one experiment is a poor value for to make a prediction for the other experiment. This implies that using multiple experiments to calibrate the models will force an average of conflicting results for the optimal parameter values. These curves also show that at extreme values $\log(K1)$, error levels to a constant line. This constant error lines approximately at SSWR values of 2500 is less than the worst error simulated at SSWR values greater than 8000. This is because these values of $\log(K1)$ describe a model that, during the experimental time frame, either heavily retards the movement of U(VI) so that the concentration occurs later or accelerates the concentration so quickly that it all flows through the column before experimental data was collected. Both of these situations generate a flat breakthrough curve over the given experimental time frame. This flat experimental time frame will always have constant error with the observed data points, no matter how much more the parameter is varied in these extreme directions. This presents a possible problem with the proposed objective function, as it does not require mass conservation in the time frame of experimental data. This also presents a problem, as these flat error areas have no direction to indicate a proper direction to vary a parameter.

These results raise a number of questions about the parametric uncertainty to be investigated. First, it was assumed that the calibrated parameters were a global optimum. It may be the case that these parameters are a local optimum. A true global optimal parameter set may result in a feasible prediction intervals. Second, it appears that the first equilibrium constant parameter $\log(K1)$ is very sensitive over a very small range. It is also seen that an optimal $\log(K1)$ for one experiment cannot be applied to another experiment. This presents a problem to our objective function, as multiple experiments are averaged. Finally, the distributions and ranges of parameters are unknown. A feasible parameter range needs to be determined to perform analysis in a compu
tationally efficient manner. The failure of linear and nonlinear analysis may be attributed to the assumption made that the parameters are normally distributed. It is seen that values for $\log(K1)$ in the sampled space for the MC simulation quickly have large objective function values relative to the calibrated value. It is likely the case that the use of uniform distributions for naive Monte Carlo analysis introduced significant bias, the entire parameter space was sampled with an equal likelihood.

1.4 Scope and Objectives

It has been observed that linear and nonlinear analysis was not useful for assessment of this computational model. Linear and non-linear analysis techniques have the advantage of being computationally efficient. This efficiency is gained by making an assumption that parameter distributions are normally distributed around a given mean, taken from a calibrated parameter set. This assumption enables the method to make significantly fewer realizations of the computational model. Monte Carlo Markov Chain methods are considered to be more reliable estimators of parametric uncertainty, which can give more useful information to conduct a predictive uncertainty analysis (Gallagher and Doherty, 2007). This study employs a computational model which requires a significant amount of computational effort. Each model run can last up to 15 minutes. To perform a Monte Carlo analysis, the number of model runs requires is quite large (Gallagher and Doherty, 2007). This study will perform 30,000 realizations to be determined by a MCMC analysis, and then perform predictive analysis using the resulting parameter distributions. Naively and in serial, this computation could take almost a year: up to 312 days. This study will thoroughly discuss the methods used to reduce this computational task to a few weeks or days, depending on the simulation.

If a model continues to fail in predicting observational data after understanding its predictive capability, many hydrogeological studies involve a multi-model analysis which ranks the models in order to perform a statistical weighting and averaging of results (Ye et al., 2008). Developing a single model requires an accurate perspective of reality that is not only expensive, but time consuming. This study does not use multi-model analysis, but will provide an analysis of parametric uncertainty of a single model, which can provide direct insight into how to rank and weight the usefulness of individual models.
First, an independent global calibration will be completed to verify the calibrated values found in previous studies. Second, the Morris One-At-a-Time method will be used to assess the complexity of the effects that the parameters have on the model. Third, an adaptive Monte Carlo Markov Chain simulation will be used to determine the posterior parameter distributions. Finally, these parameter distributions will be used to make a prediction on an experiment not used to calibrate the model. The 95% prediction interval around this prediction will be generated, and compared with results of the linear and nonlinear study.
Global optimization applied to reactive-transport modelling employs the concept of inverse modelling. Inverse modelling uses a given data set to find the model parameters which give the model the best fit to the given data. The objective function [1.3] is used in this case to summarize the data provided by three column experiments.

A calibration was completed for the [Kohler et al. (1996)] model C4 by a collaborator using UCODE. UCODE uses the modified Gauss-Newton algorithm to optimize the model given a parameter range and a start point in the parameter space ([Poeter et al. 2005]). This algorithm is a local optimization method. It is possible that a local optimum, rather than a global optimum, satisfies specified stopping criteria for the algorithm.

Global optimization methods attempt to remove the possibility of missing the best optimal parameter set for a local optimum. The DAKOTA software package ([Adams et al. 2010]) is be used to implement a global optimization method. This is to either confirm the previous calibration, or find a better calibration that was missed in previous study.

### 2.1 Multi-start BFGS Method

The multi-start Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is a global optimization method which uses a Latin hypercube sampling over the given parameter space. This sampling provides multiple starting points for the local BFGS method to perform optimization. BFGS uses an approximate Hessian matrix which described partial second order derivatives for each parameter to determine a direction for optimization ([Adams et al. 2010]).
2.1.1 DAKOTA Application

DAKOTA \cite{Adams:2010} was selected as a software package to perform Morris method analysis on model C4. DAKOTA provides a framework which provides an interface to numerous uncertainty and sensitivity analysis packages. It is compiled externally from the computational model, which was necessary for this study as the source code for the \cite{Kohleretal:1996} models is not available. DAKOTA provides an interface to the model through temporary files sent to the computational model on standard input. A Python driver script was written to accept the temporary files, read the given parameter values, execute the model using the Model Runner Application (MRA) in Figure 4.5, and write out the resulting objective function. The MRA will be described at length as it is presented in Figure 4.5. The model realizations database is implemented in this application, and it will be most useful to discuss this in reference to a Markov Chain Monte Carlo simulation, as it provides significant computational speedup to MCMC, but not to DAKOTA analysis.

Figure 2.1 is the input file required for dakota to specify the optimization method to be used. A total of 100 random starting points are chosen by Latin hypercube sampling. The previous UCODE calibration is also specified as a starting point. The method optpp_q_newton refers to the BFGS method provided by a library DAKOTA makes use of called Opt++ by \cite{Meza:1994}, a nonlinear optimization package. The variables section describes a parameter range of $\pm 100\%$ around the calibrated parameters given from UCODE. The \cite{Kohleretal:1996} model implemented in RATEQ involves numerous input and output files referenced in the current working directory, so the interface is set to only run one model run at a time. The responses section instructs DAKOTA to expect one objective function and to approximate gradients numerically. This section can be configured to accept an analytical gradient or a numerical gradient approximated by the model as input, rather than numerically approximating it on its own.

The analysis driver dakota_driver.py is shown in Figure 2.2 as the steps providing an interface to the input and output for the MRA described in Figure 4.5. The driver accepts parameters from a temporary file provided to dakota_driver.py on standard input provided from DAKOTA’s execution of the driver. The driver is also provided a temporary file to write the objective function value at the end of execution. When the driver terminates, this signals the DAKOTA executable to
perform optimization calculations. The DAKOTA executable, based on provided stopping criteria, will then either provide a new parameter input file, or end the simulation and provide an analysis output file. This decision is made by the optimization method inside of the DAKOTA executable in Figure 2.2.

![DAKOTA Input File for Global Calibration](image)

**2.2 Results of Global Optimization**

It does not appear that UCODE failed to find a true optimal point, as the point found by DAKOTA is similar. The extent to how similar these values are at this point is unknown, as no sensitivity analysis has been completed. A sensitivity analysis can determine the if the magnitude of the change in the optimal value is significant. Table 2.1 shows that the objective function found by Global-DAKOTA is smaller. The optimal point found by DAKOTA was a result of the starting point manually selected from the previous calibrated value. None of the Latin hypercube starting points found a better optimal point. This result will be discussed by the MCMC analysis.
Figure 2.2: DAKOTA Application Structure

<table>
<thead>
<tr>
<th>Optimization Method</th>
<th>Log(K1)</th>
<th>Log(K2)</th>
<th>Log(K3)</th>
<th>Log(Site-fraction)</th>
<th>Obj. Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local-UCODE</td>
<td>-4.919280214</td>
<td>-3.361820996</td>
<td>0.963847388</td>
<td>-1.705162275</td>
<td>97199.2072294</td>
</tr>
<tr>
<td>Global-DAKOTA</td>
<td>-4.864609696</td>
<td>-3.121198247</td>
<td>1.225374656</td>
<td>-1.970447545</td>
<td>91503.360996</td>
</tr>
</tbody>
</table>

Table 2.1: Model C4 Calibrated Parameters

Figure 2.3 displays the breakthrough curves for experiments 1, 2, and 8 for the calibrated simulation and experimental observation. The weighted residual is given by \( \sqrt{w_ir_i} \) where \( w_i \) is the weights discussed in Equation 1.1 and \( r_i \) is the residual of simulation and observation. The weighted simulation is given by \( \sqrt{w_is_i} \) where \( s_i \) is only the simulated value. Ideally, the weighted residual for every simulated value would be zero. The weighted residual plots show an upward trend bias for experiments 1 and 2. Generally speaking, this shows that lower concentrations have overestimation and higher concentrations have underestimation. Experiment 8 has less of a trending bias with respect to concentrations.

Figure 2.4 shows similar breakthrough curve results as Figure 2.3. It is most notable that Experiment 8 does not appear to have as good of a fit for the DAKOTA calibration as the UCODE calibration at the peak concentration of U(VI). This is also seen by an upward trend in weighted residuals for Experiment 8. This occurs despite an overall better objective function. Experiments 1 and 2 appear to have slightly better fits and less weighted residuals.
Figure 2.4 shows similar weighted residual plots for the DAKOTA calibration, and are compared with the UCODE calibration in Figure 2.3. To simplify the comparison of weighted residuals, the plots are combined in Figure 2.5. Figure 2.5 shows that the bias introduced by DAKOTA for some points in Experiment 8 is compensated for by nearly every point in Experiments 1 and 2 having a weighted residual closer to zero. To optimize Experiments 1 and 2, some sacrifices were made in Experiment 8. This is seen in Figure 2.5 for Experiment 8 for weighted simulation values greater than 80. This was found even though Experiment 8 has a larger group weight in the objective function (Equation 1.3), as it has more experimental data points than either Experiment 1 or 2.

The resulting objective function from the Global-DAKOTA results was rechecked by using the globally calibrated parameters in Local-UCODE. This second run of Local-UCODE did not find a better objective function value than that found by the Global-DAKOTA method using this point as a starting point. The stopping criteria for UCODE was set to a parameter change being less than 0.001 or a max iterations of 20. DAKOTA stopping criteria was set to a gradient tolerance of 0.0001 and max iterations of 100. Since the Global-DAKOTA method is using the change in the approximation to the Hessian matrix as it’s stopping criteria, the stopping criteria being used by Local-UCODE cannot be directly compared. The fact that the global method found a better optimal point than the local method is explained by the more rigorous stopping criteria. In the global method, along with the starting points sampled using LHS, the previously known calibrated value from UCODE was also used as a starting point. This parameter set was the starting point which led to the more optimal parameter set found by DAKOTA. None of the starting points sampled by DAKOTA led to an objective function even as good as the previously given parameter set. This leads this study to conclude that the parameter set found is the global optimal parameter set to use for uncertainty analysis.

### 2.3 Parameter Variance-Covariance Matrix Approximation

#### 2.3.1 Application

The variance-covariance matrix measures the parameter estimation uncertainty of individual parameters and the correlations between each parameter. The inverse of the Fisher information matrix \( I \) is the variance-covariance matrix \( C \), shown in Equation 2.1 (Berger, 1985). The resulting
The variance-covariance matrix is positive semi-definite.

\[ C = I^{-1} \quad (2.1) \]

The Fisher Information matrix was computed by numerically approximating the second order partial derivatives for each element of the matrix. In this application, the steps taken to approximate the derivative are 0.01. The algorithm for this is given by adapting a collaborator’s implementation of a numerical second order derivative using the function dfridr from Numerical Recipes given by Press et al. (1992). This solves Equation 2.2 given by Berger (1985) with the observed Fisher information matrix \( I_{ij} \) having elements \( i,j \) where \( N \) is the number of parameters, \( B \) is the vector of parameters, \( \hat{B} \) is the calibrated parameter set, \( x \) is the data, and \( f \) is the likelihood function. In this study, the objective function is used in place of \( \ln(f(x_i|B)) \).

\[
I_{jk}(x) = -\sum_{i=1}^{N} \frac{\partial^2}{\partial B_j \partial B_k} \ln(f(x_i|\hat{B})) \quad (2.2)
\]

### 2.3.2 Test Case Validation

An analytical test case was developed by a collaborator to solve for the expected Fisher information matrix. This numerical approximation calculates an observed Fisher information matrix. The governing equation for the synthetic test case is given by Equation 2.3. First, random standard deviations are generated from \( \sigma_i^2 \sim U(0,1) \). Next, random errors are generated from \( e \sim N(0, \sigma_i^2) \). \( B \) are the coefficient parameters, \( z \) are the synthetic observations, and \( x \) as the input to the model. Synthetic observation data \( z_i \) and variances of the error \( e_i \) were generated and provided in Table 2.2 with \( x = 0, 0.1, ..., 1 \), \( B_0 = 2 \), and \( B_1 = 1 \). Function \( f(x) \) is the simulated data and given observational data \( z \) are used in calculation of the SSWR function as in equation 1.2.

Equation 2.3 can be analytically determined for it’s partial second order derivatives. The first derivatives are given in Equations 2.4 and 2.5. The partial second order derivatives are given in Equations 2.6-2.9. The analytical solution of the Fisher information Equation 2.2 using the analytical equations and the numerical approximation are both given in Table 2.3. There was no error between these solutions given the analytical result provided by my collaborator. This verifies
the functionality of the approximate Fisher information program.

\begin{equation}
  f(x) = B_0 e^{B_1 x} + e
\end{equation}

\begin{equation}
  \frac{\partial f}{\partial B_0} = e^{B_1 x}
\end{equation}

\begin{equation}
  \frac{\partial f}{\partial B_1} = B_0 e^{B_1 x}
\end{equation}

\begin{equation}
  \frac{\partial^2 f}{\partial B_0 \partial B_0} = 0
\end{equation}

\begin{equation}
  \frac{\partial^2 f}{\partial B_0 \partial B_1} = e^{B_1 x}
\end{equation}

\begin{equation}
  \frac{\partial^2 f}{\partial B_1 \partial B_0} = e^{B_1 x}
\end{equation}

\begin{equation}
  \frac{\partial^2 f}{\partial B_1 \partial B_1} = B_0 x^2 e^{B_1 x}
\end{equation}

\begin{table}[h]
\begin{tabular}{|c|c|c|c|}
\hline
\( x_i \) & \( c_i \) & \( \sigma_i^2 \) & \( z_i \) \\
\hline
0.0 & 0.69318 & 0.75367 & 2.6932 \\
0.1 & 0.65011 & 0.79387 & 2.8604 \\
0.2 & 0.98299 & 0.91996 & 3.4258 \\
0.3 & 0.55267 & 0.84472 & 3.2524 \\
0.4 & 0.40007 & 0.36775 & 3.3837 \\
0.5 & 0.19879 & 0.62080 & 3.4962 \\
0.6 & 0.62520 & 0.73128 & 4.2694 \\
0.7 & 0.73336 & 0.19389 & 4.7609 \\
0.8 & 0.37589 & 0.90481 & 4.8270 \\
0.9 & 0.09987 & 0.56921 & 4.9291 \\
1.0 & 0.41986 & 0.63179 & 5.8564 \\
\hline
\end{tabular}
\end{table}

Table 2.2: Approximate Fisher Test Case Data

21
### Table 2.3: Fisher Information Matrix Solutions: Analytical and Numerical

<table>
<thead>
<tr>
<th>Matrix Element</th>
<th>Analytical Result</th>
<th>Numerical Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta f / \delta B_0$</td>
<td>70.327</td>
<td>70.327</td>
</tr>
<tr>
<td>$\delta f / \delta B_1$</td>
<td>87.099</td>
<td>87.099</td>
</tr>
<tr>
<td>$\delta f / \delta B_1^*$</td>
<td>87.099</td>
<td>87.099</td>
</tr>
<tr>
<td>$\delta f / \delta B_1$</td>
<td>137.59</td>
<td>137.59</td>
</tr>
</tbody>
</table>

#### 2.3.3 Variance-Covariance Matrix

The Fisher information approximation application was applied to the Kohler et al. (1996) model C4 by making use of the MRA. This provides significant speedup, as the matrix is symmetric about the diagonal. The Model Runner Application automatically recognizes pre-computed parameter sets and returns the objective function value so that the computational model does not have to be run. Using the calibrated values found through global calibration, the variance-covariance matrix was approximated using the Fisher approximation and Equation 2.1.

It is seen in the resulting variance-covariance matrix given by Figure 2.6 that the parameters do not vary greatly about the mean. This results from the second order derivative Fisher Information matrix having very large values, a result of the objective function given by Equation 1.3. It is also notable in Figure 2.6 that many of the covariances are just as significant as variances of the parameters. This implies significant parameter correlation. Approximate correlations based on the values in the variance-covariance matrix are given in Figure 2.7. Parameter correlations are calculated using Equation 2.10 with $C_{ij}$ as the covariance between parameter $i, j$ and $\sigma_i$ as the variance for parameter $i$. A strong correlation can be physically explained. Reaction equilibrium constants and the fraction describing the area for each reaction site are correlated by definition in a reaction network. The fraction of a site which relates to an equilibrium constant will change how much that equilibrium constant will contribute to the simulation. Equilibrium constants are correlated because the amount of U(VI) adsorbed by one reaction will change the amount available to be adsorbed by another reaction.

The small variance-covariance matrix presents a problem for MCMC analysis. If this variance-covariance matrix is used as-is in an MCMC simulation, all of the parameters proposed would have a very high likelihood, so acceptance is almost a certainty. Also, as this algorithm is seen as a random walk, the steps taken will be very small. This presents a case similar to the naïve
MC method when a very small range was sampled. This resulted in a high confidence around the simulated breakthrough curve, which had very little predictive capability. With such a high confidence that the calibrated parameters are the best parameters for simulation, there is little data coverage by the prediction interval. This problem will be addressed using the Covariance-Adaptive MCMC method proposed in section 4.3.

\[ r_{ij} = \frac{C_{ij}}{\sigma_i \sigma_j} \]  

(2.10)
Figure 2.3: Local-UCODE Calibrated Model Fit
Figure 2.4: Global-DAKOTA Calibrated Model Fit
$C = \begin{pmatrix}
\log(K1) & \log(K2) & \log(K3) & f \\
\log(K1) & 2.2501e-07 & 1.5858e-06 & 1.3784e-06 & -1.4205e-06 \\
\log(K2) & 1.5858e-06 & 1.3467e-05 & 1.0163e-05 & -1.0978e-05 \\
\log(K3) & 1.3784e-06 & 1.0163e-05 & 9.2182e-06 & -9.2390e-06 \\
f & -1.4205e-06 & -1.0978e-05 & -9.2390e-06 & 9.5638e-06
\end{pmatrix}$

Figure 2.5: Weighted Model Fit

Figure 2.6: Model C4 Approximated Variance-Covariance Matrix
\[ \mathbf{r} = \begin{pmatrix}
\text{log(K1)} & \text{log(K2)} & \text{log(K3)} & f \\
\text{log(K1)} & 1.0 & 0.9110 & 0.9571 & -0.9886 \\
\text{log(K2)} & 0.9110 & 1.0 & 0.9121 & -0.9681 \\
\text{log(K3)} & 0.9571 & 0.9121 & 1.0 & -0.9840 \\
f & -0.9886 & -0.9681 & -0.9840 & 1.0 
\end{pmatrix} \]

Figure 2.7: Model C4 Approximated Correlation Matrix
CHAPTER 3

MORRIS METHOD SENSITIVITY ANALYSIS

The Morris One-at-a-Time method (MOAT) is a Monte-Carlo based sensitivity screening technique useful for application to models with a large number of parameters or is computationally expensive. The method calculates the elementary effects of each parameter by varying it over a certain range with a given step size. The calculation of an elementary effect is similar to a first-order derivative with respect to one parameter, but completed with a large step size (Adams et al., 2010). Other global sensitivity techniques require a high resolution grid to be computed over the parameter space of the model. The MOAT method gives an estimate of the relative importance of certain parameters with a low resolution grid, due to a large step size and only varying one parameter at a time (Campolongo et al., 2007). The goal of the elementary effect quantity is to quantify the changes a single parameter has on the model, given all other variables to the model being constant. This is done by varying a parameter one-at-a-time while the rest of the parameters are held at a given point. The elementary effects of the model are then analyzed by computing their mean and standard deviation. The mean of the elementary effects describes the magnitude of the effects. The density of the effects is described by the standard deviation.

A downfall of the original MOAT method as proposed by Morris (1991) is that it does not investigate the interactions of parameters. Since the elementary effects only describe the effect of one parameter on the model at a time, the correlation between two parameters cannot be clearly quantified. Each parameter is described 1-dimensionally. A new morris method was proposed by Campolongo et al. (2007) which performs sensitivity analysis of a parameter relative to a group of parameters. MOAT appears to be a useful and powerful method, but is not yet extensively used due to lack of exposure (Campolongo et al., 2007).
Campolongo et al. (2007) applied the MOAT method to a non-linear multi-phase chemical reaction model with 56 parameters and 5 model results. The model describes the presence of sulfur gas in the atmosphere. It was determined that 15-20 of the parameters in this model were insignificant and could be set to any value within their uncertainty ranges, because the parameters produce a negligible effect on the mean or variance of the model outputs. This allows the modelers to concentrate on understanding the effects of more important parameters. Such important parameters were found to have significant non-linear effects on the model. For example, varying one unknown parameter describing the frequency of rainfall within the bounds of uncertainty resulted in simulations predicting the concentration of sulfur gas ranging from being relatively high and complete removal. From this MOAT result, Campolongo et al. (2007) concluded that the model should be reassessed to include a more accurate assessment of rainfall. It appears that MOAT is effective and efficient in its ability screening parameters used to drive a computational model. This method was applied using DAKOTA for model C4.

The Kohler et al. (1996) model C4 only involves 4 reaction parameters, but at the start of this study it was unknown if all of those parameters are important. It could arise that certain reactions modeled in C4 which lead to correct model predictions are implemented in other models to produce a more accurate model. Also, it could be the case that some model parameters may be insignificant, which would allow the modeler to exclude certain reactions from computation.

### 3.1 Morris Method Equations

The MOAT method requires a given starting parameter vector, \( \mathbf{X} \). The model is varied over a given range with \( p \) partitions over the parameter space. The vector \( \mathbf{X} \) describes the center point in the parameter space which the parameters will be varied about one-at-a-time. An elementary effect of a parameter \( i \) is solved for using the model’s objective function \( y(\mathbf{X}) \) over the number of parameters \( k \). In this study, \( y(\mathbf{X}) \) will be given by the objective function in Equation 1.3. The Equations 3.1 and 3.2 given by Campolongo et al. (2007) describe an of elementary effect subject to distribution \( F_i \). \( d_i(\mathbf{X}) \) \( F_i \) is an elementary effect of one parameter calculated by varying \( X_i \) by \( \Delta \). All other parameters are kept constant to their respective values in \( \mathbf{X} \).

\[
d_i(\mathbf{X}) = \frac{y(X_1, \ldots, X_{i-1}, X_i + \Delta, X_{i+1}, \ldots, X_k) - y(\mathbf{X})}{\Delta}
\] (3.1)
Random samples $N$ are taken over partitions $p$. An even number of partitions $p$ is usually taken to choose a $\Delta = p/(2(p-1))$ (Campolongo et al., 2007). Random samples are chosen uniformly over $[0, 1]$ and transformed into the unit hypercube of the parameter’s distribution. This results in a distribution $F_i$ containing $p^{k-1}(p-\Delta(p-1))$ elementary effects. $F_i$ is then quantified by computing the mean $\mu$ and standard deviation $\sigma$. These quantities are plotted against each other to show the parameter’s effect on the model.

### 3.2 DAKOTA Application

In the same manner as the global optimization study performed with DAKOTA, a MOAT study was completed. DAKOTA provides an interface to the PSUADE MOAT analysis package which can be used by configuring a proper DAKOTA input file. The Problem Solving Environment for Uncertainty Analysis and Design Exploration (PSUADE) package was developed by the Lawrence Livermore National Laboratory in 2005 (Adams et al., 2010).

#### 3.2.1 Test Case Validation

Morris (1991) provides a test model with 20 parameters. These parameters were designed to identify three types of parameters: unimportant, linear, and nonlinear. The model is given in Equation 3.3 where $w_i = 2(x_i - 0.5)$ except for $i = 3, 5, 7$ where $w_i = 2(1.1x_i/(x_i + 0.1) + 0.5)$. $B_i$ are the parameters to the model which are screened by the MOAT method. The terms are given as first-order through fourth-order terms. Figure 3.1 is the resulting plot for both Morris (1991) and the test implementation of this model provided by DAKOTA Adams et al. (2010).

Figure 3.1 shows that the first 10 parameters are important in some respect. The blue points 8, 9, and 10 show that they have a significant influence on the mean of the model, but not the standard deviation. This indicates a linear effect on the model, where these points have a similar mean effect on the model depending on how they are varied. The red points 1 through 7 indicate elementary effects which have significant mean and standard deviation. This suggests nonlinear effects; varying these parameters has varied effects on the model. The rest of the parameters 11 through 20 show relatively unimportant parameters to the model, as their influence is neither large
in mean or in variance. This analysis indicates that parameters 1-10 are the most important to consider when performing uncertainty or sensitivity analysis.

\[ y = B_0 + \sum_{i=1}^{20} B_i w_i + \sum_{i<j}^{20} B_{ij} w_i w_j + \sum_{i<j<l}^{20} B_{ijl} w_i w_j w_l + \sum_{i<j<l<s}^{20} B_{ijls} w_i w_j w_l w_s \]  

Figure 3.1: MOAT Test Case (Adams et al., 2010)

3.2.2 Analysis for a Flow Study

A MOAT analysis was completed using the DAKOTA Application developed for this study by a colleague also studying groundwater flow. The computational model estimates groundwater flow (Qt) into the northern Yucca Flat. A MOAT analysis was performed to identify the most critical parameters out of 44 used to drive the model. Plot (a) in Figure 3.2 shows the mean and standard deviation of the elementary effects for the 44 parameters. Plot (b) in Figure 3.2 shows the same effects plotted for 42 parameters, excluding those in red in plot (a). This result shows a significant difference in the effect of the 7 parameters in green and the 2 parameters in red. This allowed research on this computational model to focus on these 9 parameters for further study, rather than all 44.
3.2.3 Model C4 Analysis

The MOAT method was set up for DAKOTA using the input file in Figure 3.3. It was unknown at the time of this study what parameter range would be appropriate to study. The lower and upper bounds was arbitrarily chosen as ±10% around the calibrated parameters. The variables are defined as continuous and uniform over the specified range. The uniform distribution is specified by default using continuous_design in the variables section. Each dimension of the parameter space is partitioned into 5 subregions which define the $\Delta$ in Equation 3.2. 1000 samples are taken over the parameter space. The dakota_driver.py file accepts the parameter input file, runs the model runner application, and returns the model response file just as with the global calibration procedure described by Figure 2.2. Configuring DAKOTA to run the MOAT method is done by specifying the method as psuade_moat and configuring the number of partitions $p$ and samples $N$. A random seed value can be specified for the purpose of creating repeatable MOAT simulations.

3.3 Morris Method Results

The results for the MOAT analysis of the Kohler et al. (1996) model C4 using the objective function, Equation 1.3 is summarized in Figure 3.4. It is seen that each parameter has a mean and standard deviation effect of greater than $10^6$ on the objective function 1.3. It is seen that the parameter $\log(K1)$ has the greatest effect on the model. This result makes sense with the physical meaning of this parameter, as it is the equilibrium constant of the strongest reaction which occupies.
most of the surface. Sites(f) has the second greatest effect on the model, which has a direct effect as to how greatly log(K1) is considered in simulation. The last two parameters, log(K2) and log(K3), are reaction constants for sites which are weak and occupy a small fraction of the surface.

The MOAT results make intuitive sense for each parameter relative to the other, but all parameters have a surprising ability to change the objective function for the model. If the objective function was changed by a few 10^4 about the calibrated mean parameter, as was seen the difference in calibration techniques, a parameter could be said to possibly have a linear effect. However, seen that every parameter in this model has the ability to generate objective function values which would represent very great amounts of error in simulation. All of these parameters can only be characterized as nonlinear in this respect. The MOAT analysis succeeded in showing the importance of every parameter to the model, but did not suggest any parameters which might be considered insignificant to the modeling procedure.
Figure 3.4: MOAT Results Model C4
CHAPTER 4

MARKOV CHAIN MONTE CARLO ANALYSIS

Markov Chain Monte Carlo (MCMC) is a Monte Carlo method which creates a chain of realizations by using the likelihood of a current realization to determine the next realization to be made. This differs from a naive Monte Carlo analysis because realizations are dependent on each other.

This study will first describe the basic Metropolis-Hastings random walk MCMC algorithm. The steps taken to satisfy requirements of the Metropolis-Hastings algorithm will be described. As a component of the analysis, a custom multivariate normal distribution generator will be described. Implementing the MRA for MCMC analysis will be discussed. The Gelman-Rubin convergence statistic will be used to determine chain stability. A test case of 4 chains completing 1,000 realizations each will show convergence for all parameters. Finally, the method will be applied to an MCMC analysis of 3 chains with 10,000 realizations each. This final analysis will be discussed in the results chapter.

4.1 Metropolis-Hastings Random-Walk Algorithm

4.1.1 The Algorithm

The basic Metropolis-Hastings MCMC algorithm accepts or rejects a proposed parameter set given (1) its likelihood, (2) a previously accepted likelihood, (3) a prior probability distribution for that parameter, and (4) a probability $\alpha$. Proposed points are made using a proposal distribution. There are many ways an MCMC algorithm can satisfy these requirements. The following sections describe the considerations taken in completing these steps for the purposes of this study.

Each realization of the MCMC simulation is dependant on the last. This makes the parallel Monte Carlo simulator developed earlier inapplicable to the MCMC method of analysis. The
parallel MC simulator requires all realizations to be independent of each other. With a naive MC study, the parameter selections can be all proposed simultaneously before the start of the computation. The direction of states in an MCMC simulation through the parameter space depends on the previous state’s model likelihood. However, each MCMC simulation, or chain, is independent of each other, so multiple chains can be computed simultaneously.

The goal of the MCMC simulation is to solve for the posterior probability of parameters $a$ given a computational model and dataset summarized by $b$. Bayes’ Theorem is used to solve for this in equation 4.1 $p(a|b)$ is the posterior probability of parameters $a$ subject to the dataset $b$. $p(b|a)$ is the likelihood of the dataset $b$ given the parameters $a$. $p(a)$ is the prior probabilities for the parameters. $p(a)$ is an unknown quantity to this study. A suitable $p(a)$ will be addressed in Section 4.1.3. For notation, the likelihood $p(b|a)$ will be called $L(a|b)$. Since $p(b)$ is constant, Equation 4.1 is simplified by Equation 4.2 (Gallagher and Doherty, 2007). The prior probability of the parameters $a$ will have a role in this study, discussed in Section 4.1.3. The likelihood function will be solved in Section 4.1.2.

$$p(a|b) = \frac{p(b|a)p(a)}{\int p(b|a)p(a)da} = \frac{p(b|a)p(a)}{p(b)} \quad (4.1)$$

$$p(a|b) \propto L(a|b)p(a) \quad (4.2)$$

The Metropolis-Hastings algorithm first requires a starting point to calculate the posterior probability, $p(a|b)$. Next, another point is chosen to calculate $p(a'|b)$. Hastings introduced the calculations $q(a', a), q(a, a')$ to represent the transitional probability between point $a$ and $a'$. This allows for biased probability distributions to be used in the simulation. This study will only use symmetrical proposal probability distributions, so these terms cancel. Equation 4.3 gives a linear comparison between the posterior probability of the first point $a$ and the new proposed point $a'$. The proposed point $a'$ is accepted into the chain only if it meets a condition set by $\alpha$. A random number $r$ is generated on a uniform distribution over $(0, 1)$. If $r$ is greater than $\alpha$, given by Equation 4.4, the proposal point $a'$ is rejected. Otherwise, the proposal point $a'$ is accepted. When the proposal point $a'$ is rejected, the original point $a$ is accepted into the chain another for the next iteration (Gallagher and Doherty, 2007).
The next iteration of the MCMC simulation uses the resulting accepted point from the previous step and proposes a new \( a' \) based on that point. The stopping criteria for this study will be the number of iterations of the chain. An acceptance ratio can be calculated, given the number of proposed points which were accepted divided by the number of iterations. The acceptance ratio \( \tau \) is considered to be an important statistic in an MCMC simulation. For high dimensions of parameters, the acceptance ratio has been empirically found by Roberts et al. (1997), Roberts and Rosenthal (2001) to be optimal at approximately \( \tau = 0.234 \). Other acceptance rates have been shown to be effective. Gallagher and Doherty (2007) even suggests an acceptance rate between 20%-70% is sufficient. The optimal acceptance rate for an MCMC chain requires a problem to be well understood and the computational time of the MCMC simulation to be short. For this study, \( \tau = 0.25 \) was chosen as the acceptance rate goal as a starting point.

These equations imply a few behaviors of the chain. The chain will always accept a proposal point which has a greater probability than the previous point. This is because \( \beta \) will be greater than 1, so \( \alpha = 1 \). Since no random number on the range of \((0, 1)\) will be greater than 1, the proposal point will always be accepted. Another property of the chain is that it may accept points of lesser probability, depending on the random variable \( r \). For example, if \( P(a'|b) = 0.25 \) and \( P(a|b) = 0.5 \) then \( \beta = 0.5 \) and \( \alpha = 0.5 \). If \( r \) is greater than 0.5, \( a' \) will be rejected. However, even though \( a' \) had a lesser probability than \( a \), if \( r \) is less than or equal to 0.5, \( a' \) will be accepted into the chain. These implied properties of the chain are designed to always optimize with better selections, but attempt to avoid local optima through the \( r \) random variable.

\[
\beta = \frac{P(a'|b)Q(a', a)}{P(a|b)Q(a, a')} \quad (4.3)
\]
\[
\alpha = \min(1, \beta) \quad (4.4)
\]

4.1.2 Likelihood Function

Rojas et al. (2008) describes the analytical Gaussian likelihood function given by Equation 4.5, with a computational model \( M \), a set of parameter vectors \( \theta \), input variable vectors \( Y \), observed data vector \( D \), a simulated data vector \( D' \), and a variance-covariance matrix \( C_D \) for the observed
data, and a number of observations $N$.

$$L(M_k, \theta_l, Y_m, |D) = (2\pi)^{N/2} \cdot |C_D|^{-1/2} \cdot \exp\left(\frac{-1}{2} (D - D^*)^T C_D^{-1} (D - D^*)\right)$$ \hspace{1cm} (4.5)$$

For the purposes of this study, many simplifications can be made to this equation. First, only one model is being considered, so the notation delineating multiple $M$ is not needed. Second, $N$, $Y$, and $C_D$ are constant factors, as the only variation made in this study is in the simulation parameters $\theta$. Finally, for this model, $\phi$ is given as the objective function from Equation 1.3 to replace $(D - D^*)^T C_D^{-1} (D - D^*)$. This results in Equation 4.6 which is constantly linearly related to the true likelihood function. The MCMC simulation will only require a linear comparison between two likelihood functions, not the true value of likelihood.

$$L(\theta|D) \propto \exp\left(\frac{-1}{2} \phi\right)$$ \hspace{1cm} (4.6)$$

A difficulty using this equation is presented by the scale of $\phi$. As seen previously, this objective function ranges from is on the order of 100,000. The calculation of the likelihood function using this objective function results in calculations of values similar to $\exp(0.5*100000)$ which is well beyond the scope of a double precision float. The reason for the objective function’s large value is the high confidence in the experimental measurements. The weighting matrix given by Equation 1.1 depends upon the experimental standard deviation. These values are of the order of $10^{-3}$ for about 100 observations. As an example calculation, a residual may have the magnitude of 0.05, which is a common magnitude even for some points for the calibrated data. If the standard deviation of the observation in this calculation is $10^{-3}$, the weighting becomes $w = 1/(10^{-3})^2 = 1,000,000$. Then the squared weighted residual (SWR) becomes $SWR = wr^2 = 1,000,000 \ast 0.05^2 = 2,500$ For 100 observations of the same error, the sum of these SWRs will be 250,000. To solve this problem, the magnitude of each weight was multiplied by $10^{-6}$. This is not an precise representation of the SSWR, but it preserves the relative weighting for each residual required to determine a good fit. The new SWR given as an example with the new weights equals 0.25 and the SSWR becomes 25. The likelihood of the objective function 25 is $3.7267 \ast 06$. This example calculation has a very small likelihood, but the magnitude of the likelihood is not important. The MCMC simulation only
requires these values to calculate $\beta$ which is a relative comparison of two likelihoods of the same magnitude.

As discussed in Section 1.3.3, flat breakthrough curves will not provide a proper direction for the MCMC chain to evolve. This problem is addressed by proposing a boundary condition when a breakthrough curve is simulated with no concentration seen at any pore volume during the experimental time frame. There are also situations where the simulation will fail to generate a breakthrough curve for certain parameter selections. This is due to the computational model recognizing some failure to converge to a stable solution. This situation has no objective function to calculate. Both of these boundary condition is imposed by returning a likelihood of zero, so that there is no likelihood that the new proposed point will not be accepted into the chain.

### 4.1.3 Prior Distributions

A prior distribution describes the probability of a given value given a previous knowledge of the behavior of the random variable the value represents. For the Kohler et al. (1996) model, this behavior is unknown. Understanding the behavior of the parameters in model C4 is, in fact, the driving motivation behind this study. With this distribution being unknown, two prior distributions were tested.

The first prior distribution consisted of four (4) MCMC simulations assuming a prior distribution with a uniform distribution from $(-\infty, \infty)$. To be specific, the prior distribution was ignored, considering all values to be equally as likely. A preliminary study with this prior distribution resulted in more divergent chains. One chains arrived to a flat objective function area, which had no slope to determine direction for the chain to evolve. This flat area was one with a high amount of error of low likelihood, but by the nature of the MCMC simulation still accepted. When the chain diverged into this area, it did not move back out to an area of higher likelihood within a feasible amount of simulation iterations. Theoretically, the chain would eventually find a path back to higher likelihood areas which would be accepted into the chain. However, given the long simulation time this is not a feasible event to wait through multiple unreasonable simulations. This provided a motivation to determine a feasible prior distribution of the parameters.

The second prior distribution was also uniform, but over the range of $+/−2$ log units centered about the globally calibrated parameter set. This range was determined by taking the column-
derived isotherm value and plotting them against the simulated isotherm values for 100 uniform realizations over this range for experiments 1,2,8. As seen in Figure 4.1, the simulated isotherms cover all of the experimental data for each of these experiments. This parameter range was selected by a collaborator, Gary Curtis at the United States Geological Survey.

![Figure 4.1: Parameter Range Isotherm Test](image)

### 4.1.4 Proposal Distributions

The MCMC simulation hinges upon the selection of the proposal distribution. The proposal distribution drives the new proposed points to be proposed for acceptance into the chain. The proposal distribution will be generated using two functions. First, a variance-covariance matrix approximated using the Fisher Information matrix. Second, a multivariate normal pseudo-random number generator. The parameters proposed into the MCMC chain was be given by a multivariate normal distribution. This distribution was initially centered about the calibrated parameter set and varied by the approximated variance-covariance matrix.

### 4.2 Multivariate Normal Distribution Generation

Proposal points for this simulation follow a multivariate normal distribution about a given mean and variance-covariance matrix. Equation \[ p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu) \right) \] is the equation for this probability distribution.
function (pdf), with μ as the mean and Σ as the variance-covariance matrix. In this study, the
given mean is the current location of the MCMC simulation, starting at the calibrated values for
the parameters. The given variance-covariance matrix starts from the variance-covariance matrix
numerically determined from the Fisher information matrix. The variance-covariance matrix used
to generate new proposal points is scaled throughout the course of the MCMC simulation by the
adaptive method.

A multivariate normal dataset class was written in Python 2.4 to provide the functionality
required to generate proposal points. This class was adapted from code provided by the library
NORMAL_DATASET (Burkardt, 2009). Python 2.4 provides a random library which provides
uniform and univariate normal generators implemented using the Mersenne Twister. The Mersenne
Twister is an efficient pseudo-random number generator with a period of $2^{19937} - 1$ (Matsumoto
and Nishimura, 1998). This period is more than sufficient for the Monte Carlo analysis being
completed in this study with $N = 30000$. Converting the univariate normal generators to provide
a multivariate normal distribution based on a given covariance matrix is given by Bratley et al.
(1983). The probability density function is given by equation 4.7. The algorithm to solve Equation
4.7 is summarized by the steps in Figure 4.2

$$
pdf(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \hspace{1cm} (4.7)
$$

1. Create a matrix Y [k,n] where each row k follows a normal distribution with mean 0 and
   variance 1.
2. Determine the upper-triangular Cholesky factor R of the variance-covariance matrix where
   $\Sigma = R^T R$.
3. Compute $x = \mu + R^T Y$

Figure 4.2: Steps to solve a multivariate normal PDF (Burkardt 2009)

Step 1 is satisfied using the Mersenne Twister pseudo-random number generator. Step 2 is
a decomposition which requires the matrix to be positive-definite. Since the pdf will only exist
for a variance-covariance matrix which is positive-definite, this does not post a problem. It was
also seen in the approximation of the Fisher information, that the variance-covariance matrix is positive-definite. Step 3 is a simple matrix computation.

### 4.2.1 Verification Test Case

A test of this application was proposed to be compared with the International Mathematics and Statistics Library (IMSL) developed by Visual Numerics. 1000 test points were generated for two random variables $x_1, x_2$ with $\mu = (1, 2)$ and $\Sigma = (1, 0; 0, 3)$. The points generated were plotted as a histogram to show the probability density function in Figure 4.3. These points were also used to generate a probability plot in Figure 4.4. The custom multivariate normal random number generator results are denoted as Miller_X1 and Miller_X2. The IMSL results are IMSL_X1 and IMSL_X2. Both the custom and IMSL generators arrived at the same mean and standard deviation requested by the mean and variance-covariance matrix. Figure 4.4 shows that both programs deviate from the 95% confidence interval about the mean. Figure 4.3 shows that IMSL_X1 failed to generate some points near the mean to create a smooth curve. Both of these failures are likely due to the low number of realizations $N = 1000$. Due to the similar results, the custom library was considered to be accurate for use in the MCMC study.

![Histogram of Miller_X1, IMSL_X1, Miller_X2, IMSL_X2](image)

Figure 4.3: Histogram Comparison of Custom Library vs. IMSL
4.3 Covariance-Adaptive Method

4.3.1 Motivation

Without the covariance-adaptive method, the MCMC simulation’s acceptance rate was $\tau = .9997$. For $N = 10000$ realizations, this implies the MCMC simulation only rejected 3 points. This is a result of the small proposal distribution used, which was calculated from the fisher information approximation. The objective function never changed significantly from .0915, or a likelihood of 0.9553. This is because proposal points were extremely close to the previous points, due to a small variance-covariance matrix used in the multivariate normal pseudo-random number generator. This results in every calculation of beta results in a number equal to or close to 1, so the new proposed point is nearly always accepted.

With an acceptance rate this large, it is clear that the parameter space is improperly sampled (Gallagher and Doherty, 2007). If the parameter space is not sampled on a wide enough range, the MCMC simulation gives no new information about the posterior probability distributions. This is the motivation for an adaptive MCMC method which will attempt to create a better picture of parameter space.
Adaptive MCMC methods are simply modifications of the MCMC algorithm to change the way the chain evolves, based on some status of the chain. This study’s problems arise from the variance-covariance matrix used for the proposal distribution and the acceptance rate being too high. These two features of the chain are obvious choices to construct an adaptive method. Atchade and Rosenthal (2005) suggests such an algorithm where the variance-covariance matrix is scaled depending on the current value of the acceptance rate during the MCMC simulation. A factor $\gamma$ is introduced to generate scaling parameters $\sigma_i$ for $i = 0..N$ iterations of the MCMC simulation.

The MCMC algorithm is modified by multiplying the scaling parameter $\sigma_i$ to the variance-covariance matrix $C$ at each step of the simulation, given by Equation 4.8. The value $\sigma_i$ is calculated from the previous step $\sigma_i - 1$ in Equation 4.9. The resulting covariance matrix to use at iteration $i$ of the MCMC simulation is given by $C_i$.

To solve Equation 4.9, the current acceptance rate for the MCMC chain $\tau_i$ is calculated. The acceptance rate is the number of new proposed parameters accepted into the chain. This occurs when $r \leq \alpha$ where $\alpha$ is given by Equation 4.4 and $r \sim U(0,1)$. In five dimensions, Roberts and Rosenthal (2001) found 0.234 was found to be the optimal target acceptance rate. The optimal target acceptance ratio not only depends on the parameters, but also the computational model. Sometimes complicated models benefit from a target acceptance ratio greater than 0.234. For example, Roberts and Rosenthal (2001) found 0.44 to be appropriate for some cases. Since this is the first MCMC study for this model, the target acceptance ratio $\tau_{\text{target}}$ was chosen to be 0.25.

The function $\text{check}(x)$ in Equation 4.10 is used to put a bounds on the scaling parameter. This is completed with the given definition of $\Delta = \{\sigma : \epsilon \leq \sigma \leq A\}$ where $\sigma_{\text{opt}} \in \Delta$. In this study, a feasible scaling parameter is unknown. For this case, the range of $\Delta$ was chosen to be infinite.

\[ C_i = \sigma_i C \]

\[ \sigma_{i+1} = \text{check}(\sigma_i + \gamma(\tau_i - \tau_{\text{target}})) \]

\[ \text{check}(x) = \begin{cases} x & : x \in \Delta \\ \epsilon & : x < \epsilon \\ A & : x > A \end{cases} \]
Atchade and Rosenthal (2005) uses $\gamma = \sigma_0/n$ but this value did not generate scaling factors large enough to have a meaningful effect on the simulation in this study. The factor was modified to be $\gamma = 100(\sigma_0/n)$ to allow the scaling parameters to vary more quickly. For this study, the initial scaling parameter $\sigma_0$ was chosen to be 100,000 to have the variance-covariance matrix start with values on the order of 1.

4.3.2 Implementation

The Model Runner Application (MRA) was developed using Python 2.4. Running the Kohler et al. (1996) computational model involves changing two input data files. The first input file specifies the equilibrium constants. The second file specifies the site fraction between the strong and weak reaction sites. A model run results with a simulated breakthrough curve, which is used to calculate the objective function.

The primary function of the MRA is to create a simple interface to running a Kohler et al. (1996) model. This application includes the specifications required to automatically modify the input files, and know what lines to read from the simulated breakthrough curve through the use of a Model Data class. The MRA is currently capable of running any Kohler et al. (1996) column model or experiment, and could be expanded to be used for any other computational model.

A secondary function of the MRA is to store completed realizations in a database. The model database is used as a surrogate model. When a parameter set is proposed for computation has already been computed by a simulation, the resulting objective function value is simply returned from the database. The model database is an unsorted flat file database which can dynamically store any number of input parameters or requested objective functions.

For model C4, the model database is configured to use four (4) parameters as input and return one (1) objective function. The model database search procedure was implemented using a naive approach, so the search complexity is $O(N \times M)$. $N$ is the number of realizations currently input into the database. $M$ is sum of model parameters and objective functions. The complexity of adding a new realization to the database is $O(1)$, as there is no sorting on the database file implemented. This is an inefficient implementation of a database. However, this database implementation is extremely useful, as a database lookup in this study never exceeds $N = 30000$ and $M = 5$.

On FSU HPC, the timing of the worst-case lookup is approximately 1 second. While this time
could be improved using a more complex insertion procedure, it only adds 1 second to a simulation which normally takes approximately 10 minutes to run. For a model that takes 15 minutes to run, if the model realization is found in the database even with a worst-case time, the MRA runs 900 times faster. The execution time of the MRA is dominated by the complexity of the unknown method used to solve a linear system of equations $Ax = B$, which is certainly greater than $O(N)$. The covariance-adaptive MCMC method proposed was implemented by a custom Python 2.4 code. Figure 4.6 shows the MCMC control flow, as described previously. Two parameter sets P1 and P2 are generated and run by the MRA. A decision to accept or reject parameter P2 is made by Equations 4.3 and 4.4. When $r > \alpha$, the parameter P2 is accepted.

![Figure 4.5: Model Runner Application](image)

This MCMC study requires some special considerations which need to be made to obtain a posterior probability distribution. These considerations which have been previously discussed are summarized in Table 4.1

<table>
<thead>
<tr>
<th>Problem</th>
<th>Caused By</th>
<th>Solution</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnitude of SSWR</td>
<td>Experimental weights</td>
<td>Scale weights by $10^{-6}$</td>
<td>Section 4.1.2</td>
</tr>
<tr>
<td>Small covariance matrix</td>
<td>Experimental weights</td>
<td>Covariance-Adaptive Method</td>
<td>Section 4.3</td>
</tr>
<tr>
<td>Unknown prior distribution</td>
<td>No previous analysis</td>
<td>Isotherm cover check</td>
<td>Figure 4.1</td>
</tr>
<tr>
<td>Runtime of simulation</td>
<td>Computational complexity</td>
<td>MRA</td>
<td>Figure 4.5</td>
</tr>
</tbody>
</table>

Table 4.1: MCMC Special Considerations
Figure 4.6: Monte Carlo Markov Chain Application
4.4 Chain Convergence Test

The Gelman-Rubin convergence statistic [Gelman et al. (2004)] was chosen for this study to determine the proper convergence of the MCMC simulations. A common term with MCMC simulations is known as the burn-in period. The burn-in period refers to the realizations at the start of an MCMC simulations before the chain has converged to an area of feasible likelihood. This is essentially a period of calibration to determine a good starting distribution for the MCMC simulations. According to [Meyn and Tweedie (1993)], if a MCMC simulation is to converge for any starting distribution, then any initial distribution will hold the same property. The goal is to then evolve a stationary, or converged, set of chains. However, when a simulation starts from a distribution centered about the calibrated parameter, the burn-in period does not need to represent a region of realizations to be excluded from the posterior distribution.

The setup for this study starts the simulation about the calibrated parameter with the approximated variance-covariance matrix. This is in attempt to remove the need for a burn-in period of model calibration to find a good starting distribution. This was done to reduce the necessary computational cost. The convergence will be observed using the Gelman-Rubin convergence statistic that involves between chain convergence testing. Probability distribution functions will be compared for both cases where a burn-in period is considered and where it is ignored.

The Gelman-Rubin convergence statistic is calculated for each variable assessed by the MCMC simulation individually for $J$ chains of $N$ realizations. An MCMC study with multiple chains will consist of a total of $JN$ realizations of the computational model. Equation 4.11 calculates the average value for one chain. Equation 4.12 calculates the average over multiple chains. Equation 4.13 calculates the variance of one chain. The average and variance of the scalar variable $\phi$ is calculated between $B$ and within $W$ each chain, respectively. These equations are given by 4.14 and 4.15 [Gelman et al. (2004)].

\[
\psi_j = \frac{1}{N} \sum_{i=1}^{N} \psi_{ij} \quad (4.11)
\]

\[
\psi_\cdot = \frac{1}{J} \sum_{i=1}^{J} \psi_{.j} \quad (4.12)
\]
\[ s_j^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\psi_{ij} - \psi_j) \]  
\[ B = \frac{N}{J-1} \sum_{j=1}^{J} (\psi_{j.} - \psi_{..})^2 \]  
\[ W = \frac{1}{J} \sum_{j=1}^{J} s_j^2 \]

\[ W \text{ and } B \text{ are used to calculate the variance of } \psi \text{ given the model } y \text{ in equation 4.16} \]

This variance is then used to calculate the convergence statistic \( \sqrt{R} \) in equation 4.17.

\[ \text{var}(\psi|y) = \frac{N-1}{N} W + \frac{1}{N} B \]  
\[ \sqrt{R} = \sqrt{\frac{\text{var}(\psi|y)}{W}} \]

When the Gelman-Rubin convergence statistic \( \sqrt{R} \) approaches 1, the MCMC chains have converged. Some values close to 1 are acceptable, 1.2 is normally considered close enough for most problems (Gelman et al., 2004). The analysis of the exact target value for this statistic is left for future study.

### 4.4.1 MCMC Analysis Test Case

A test of the MCMC application developed in this study was created for \( N = 1000 \) realizations and 4 chains. Two test cases were run, the first where no prior distribution was considered and the second where the prior distribution discussed in Section 4.1.3 was applied. Each chain resulted with the target acceptance ratio of 0.25, verifying the functionality of the adaptive method. Figure 4.7 shows the first test without the prior distribution on the left, and the second test with the prior distribution on the right. The first test resulted in some parameters failing to converge to 1 or close to 1. For example, it appears that K2 has converged slightly below 2. With the prior distribution, all parameters quickly converged close to 1. This shows the necessity for including a prior distribution.
Figure 4.7: Test Cases 1 and 2, Convergence
CHAPTER 5
MCMC RESULTS

The primary goal of this study is to understand parametric uncertainty. This is done by estimating a posterior probability distribution function (PDF) using MCMC analysis. The MCMC analysis evaluates Equation 4.2 for the posterior PDF. The parametric uncertainty is validated by generating prediction intervals for an experiment based on a calibration of other independent experiments. To determine how reasonable the estimation of the posterior PDF is, multiple chain convergence is a criteria to satisfy. The likelihood response surfaces show the effectiveness of the calibrations completed. Parameter correlation response surfaces show the trends between parameters. The correlations surfaces are colored with respect to likelihood to show the complexity of the parameter space. The posterior PDF will also be analyzed using a the cumulative distribution function and the Liliefors test to determine the normality of the distributions. Normal distributions were assumed in both the linear and nonlinear analysis previously completed for Kohler et al. (1996) model C4. If this is not a valid assumption, it is trivial to conclude that the linear and nonlinear analysis is invalid.

5.1 Chain Evolution and Posterior Probability Distributions

Figures 5.1, 5.2 show the chain evolution and resulting posterior PDF for 30,000 and 27,000 realizations, respectively. Each chain of 10,000 realizations is placed one after the other to determine an overall posterior PDF for all chains. This is given in two figures to show the effect of removing a 10% burn-in period as discussed in Section 4.4. Between these figures, it is seen that removing the first 10% merely removes some realizations around the calibrated points for log(K1) = −4.8, log(K2) = −3.1, log(K3) = 1.2, and Sites(f) = −1.9, as shown by the global calibration in Table
2.1 The burn-in period refers to a length of the MCMC simulations where the parameters have not stabilized around a calibrated point. In this study, a distribution around the calibrated point was used as the starting point, removing the need for a burn-in period to be considered. There is no justifiable reason to discount these points of high likelihood from the chains.

The trace plots in Figure 5.1 do not appear to converge to a stationary distribution for any parameter. This is especially the case for parameters \( \log(K_1) \) and \( \log(K_3) \). With the burn-in period removed in Figure 5.2, the trace plots have less parameter selections around the calibrated parameters. The convergence of these trace plots will be discussed in Section 5.4.

The posterior PDF plots is shown as histograms with 20 bins over the range of \( \pm 2 \) log units. At a glance, the resulting posterior PDF does not appear to be normal for any parameter. \( \log(K_1) \) has a significant peak around the calibrated parameter in comparison to the other parameters. \( K_1 \) also appears to have possibly 2 more peaks at \(-5.5\) and \(-4.4\) other than the peak near the calibration at \(-4.7\).

The prior distribution for \( \log(K_3) \) was not implemented correctly for the range of \( \pm 2 \) log units around the global calibrated parameter. A coding error allowed values to \(-3\) to be the lower bounds of the prior distribution. However, rather than this being an unrecoverable error, it shows that many parameter selections far from the calibration are realistic realizations could still be accepted for these values of \( \log(K_3) \). In contrast, despite the parameter range chosen for \( \log(K_1) \) being \( \pm 2 \) log units around \(-4.8\), no points greater than \(-4.3\) were entered into the chain.

The rest of the parameters found parameters for the \( \pm 2 \) log unit range. Parameter \( \log(K_1) \) transformed from a uniform distribution over the \( \pm 2 \) log unit range to what appears to be a log normal distribution. This explains the sensitivity seen by MOAT analysis for \( \log(K_1) \). The less sensitive parameters from MOAT analysis, \( \log(K_2) \) and \( \log(K_3) \) show distributions closer to a uniform distribution. This implies less sensitivity to the those parameters. The PDF for \( Sites(f) \) shows a greater variance in the points selected than \( \log(K_1) \). This agrees with the MOAT analysis which determined that \( Sites(f) \) is the second most sensitive parameter.
Figure 5.1: Trace plot and Probability Distribution, 30000 Realizations
Figure 5.2: Trace plot and Probability Distribution with 10% burn-in removed, 27000 Realizations
5.2 Likelihood Surfaces

The likelihood surfaces for the MCMC data show the likelihood of parameter selections over the parameter range given by the prior distribution. Likelihood response surfaces will show the parameter selections against likelihood for all 30,000 realizations. Parameter correlation surfaces will show trends between parameters and areas of high likelihood. The parameter correlation surfaces will be compared with an estimation of the parameter correlations derived from the covariance matrix.

5.2.1 Likelihood Surfaces

The calibration study is confirmed by Figure 5.3. The maximum values seen in likelihood points are identical to the global calibration. Similar to the results presented in the MOAT study by Figure 3.4, more sensitivity in $\log(K1)$ and $Sites(f)$ is seen. $\log(K2)$ and $\log(K3)$ show a much flatter response surface than $\log(K1)$ and $Sites(f)$. While it might appear that one would expect a linear or negligible behavior from a flat likelihood surface, it is important to note the quantity of realizations nearing zero likelihood throughout the range of all of the parameters. It is important to note that this is not a regular grid for the range of parameters, the other parameters vary for each point in the likelihood surface. This indicates significant correlation between the parameters as discussed previously in Figure 2.6.

Figure 5.4 shows the correlations between parameter selections and are colored according to the likelihood of that selection. There are observable trends in all parameter pairings except for parameters $\log(K2)$ and $\log(K3)$. This surface implies a very jagged parameter space, where some high likelihood values are next to very low parameter values. The global calibration parameter agrees with the point of highest likelihood in all of the plots in Figure 5.4. It is notable that surfaces for $(\log(K1), \log(K2))$ and $(\log(K1), Sites)$ have regions of relatively high likelihood far away from the calibration region. These regions are separated by a band of low likelihood. It is clear from Figure 5.4 that $\log(K1)$ is the most sensitive parameter. For each plot involving $\log(K1)$, there is a band of many realizations surrounding the calibration value for $K1$. However, there are numerous points found by MCMC around $-6.0$ which is far away from the calibration. Some of the high likelihood points far away from the calibration will be verified for correctness in Figure 5.5.
The jagged surface depicted in Figure 5.4 explains why no Latin Hypercube starting point
selection in the global calibration found a better calibration point than when starting from the pre-
viously known local calibration point from UCODE. There are many local optimal points over this
surface surrounded by low likelihood which would easily give the appearance of a global optimum.

Referring back to Figure 2.7 parameter correlations \((\log(K1), Sites(f))\); \((\log(K2), Sites(f))\);
\((\log(K3), Sites(f))\) are negative. A negative trend is seen in all of the parameter correlation
surfaces in Figure 5.4 including the \(Sites(f)\) parameter. In Figure 2.7 parameter correlations
\((\log(K1), \log(K2))\); \((\log(K1), \log(K3))\); \((\log(K2), \log(K3))\) are positive. Positive trends are
also seen in \((\log(K1), \log(K2))\) and \((\log(K1), \log(K3))\), but it is not clear there is any trend in
\((\log(K2), \log(K3))\). The failure for the correlation approximation and the parameter correlation
surface to agree for parameters \((\log(K2), \log(K3))\) can be due to either a poor sampling for these
two parameters in MCMC or the approximation of the variance-covariance matrix is too coarse.
Figure 5.3: Likelihood Response Surface
Figure 5.4: Parameter Correlation Surfaces
5.2.2 Verification of Extrema

Four points listed in Table 5.1 were chosen to see if the some high likelihood realizations found far away from the calibration points or any trends around those calibrations are still reasonable breakthrough curves. Breakthrough curves with low likelihood, seen in numerous realizations in the likelihood surfaces, are commonly near zero concentration for all data points. A point near the calibrated point has the highest likelihood of any realization in the MCMC simulations. Not many realizations show a likelihood of 0.9 or greater, but there are many with a likelihood of 0.8 or greater. Point T1 is a point which is close to the calibrated point, but varies significantly for $\log(K2)$ and $Sites(f)$. Point T2’s value for $Sites(f)$ is almost an 1 log unit different than the calibration. Given the high sensitivity of $\log(K1)$ discussed throughout this study, points T3 and T4 have significantly different $\log(K1)$ values. Point T3 also varies greatly in $\log(K3)$, while all parameters for T4 almost differ a full log unit. Figure 5.5 shows the breakthrough curves resulting from points T1, T2, T3, and T4. The remarkable similarity to the calibrated breakthrough curve shows they are not accepted due to error in the MCMC simulation or likelihood calculation, but rather because they are reasonable. There are many breakthrough curves in the simulation with near zero concentrations seen over all pore volumes, and others with much more significant retardation or early breakthrough of U(VI).

<table>
<thead>
<tr>
<th>Test Point</th>
<th>Likelihood</th>
<th>$\log(K1)$</th>
<th>$\log(K2)$</th>
<th>$\log(K3)$</th>
<th>$Sites(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration</td>
<td>0.955</td>
<td>-4.864609696</td>
<td>-3.121198247</td>
<td>1.22374656</td>
<td>-1.970447545</td>
</tr>
<tr>
<td>T1</td>
<td>0.933</td>
<td>-4.821052106</td>
<td>-2.825493246</td>
<td>1.548920337</td>
<td>-2.281653042</td>
</tr>
<tr>
<td>T2</td>
<td>0.904</td>
<td>-4.801995712</td>
<td>-1.861911515</td>
<td>1.383442327</td>
<td>-2.626996530</td>
</tr>
<tr>
<td>T3</td>
<td>0.925</td>
<td>-5.043395034</td>
<td>-3.518511246</td>
<td>0.593423264</td>
<td>-1.372703895</td>
</tr>
<tr>
<td>T4</td>
<td>0.831</td>
<td>-6.015234412</td>
<td>-4.066155811</td>
<td>0.054575613</td>
<td>-0.718095211</td>
</tr>
</tbody>
</table>

Table 5.1: Test Realization Points, Extreme with High Likelihood

Four more points listed in Table 5.2 and run for the prediction of Experiment 5. Figure 5.6 shows the breakthrough curves for these points. The point P4 is a point which is not significantly different from either the calibration point, P1 or the other points P2 and P3. P4 is closer to the calibrated point P1 for the most sensitive parameter $\log(K1)$ than P3. However, due to the correlation of $\log(K1)$ to the other parameters, such as $Sites(f)$, the realization P4 has a much lower likelihood. The correlation between the parameters makes a very complex model likelihood surface for the MCMC simulations.
A PDF of the 30,000 realizations is plotted in Figure 5.7. This shows only realizations for the observed peak pore volume. Many realizations have a concentration near zero. The reason for this can be seen in Figure 5.6. Point P4 has a peak concentration near zero, but still had a likelihood which could be accepted into the MCMC chain. A breakthrough curve is still simulated for the data points observed, but appears much later than points P2 and P3.
<table>
<thead>
<tr>
<th>Test Point</th>
<th>Likelihood</th>
<th>$\log(K_1)$</th>
<th>$\log(K_2)$</th>
<th>$\log(K_3)$</th>
<th>$\text{Sites}(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration P1</td>
<td>0.9553</td>
<td>-4.8646</td>
<td>-3.1212</td>
<td>1.2254</td>
<td>-1.9704</td>
</tr>
<tr>
<td>P2</td>
<td>0.8669</td>
<td>-4.8144</td>
<td>-2.7842</td>
<td>1.8533</td>
<td>-2.3938</td>
</tr>
<tr>
<td>P3</td>
<td>0.7232</td>
<td>-4.7233</td>
<td>-3.0080</td>
<td>2.3048</td>
<td>-1.8208</td>
</tr>
<tr>
<td>P4</td>
<td>0.1044</td>
<td>-4.9435</td>
<td>-3.4925</td>
<td>1.4966</td>
<td>-2.6320</td>
</tr>
</tbody>
</table>

Table 5.2: Test Realization Points, Close with Low Likelihood

![Sample Realizations P1-4](image1)

Figure 5.6: Test Breakthrough Curves, Experiment 5

![Histogram PV(9), 30000 Realizations, 20 Bins](image2)

Figure 5.7: Peak Observed Pore Volume, Plotted for 30000 Realizations
5.2.3 Regular Grid Test Case

It is common with MCMC analysis that the ability to sample extreme values in the parameter space is limited by the constraint that all dimensions of the space are varied at the same time (Gallagher and Doherty, 2007). In attempt to see a regular response of one parameter at a time, parameter \( \log(K_1) \) was varied on a regular grid of \( d(\log(K_1)) = 0.1 \) over \([-6.0, -4.0]\). The other 3 parameters are held constant to the calibrated values. Error is calculated by squared residuals. It is seen by Figure 5.8 that low error realizations of \( \log(K_1) \) cannot be seen with this resolution. Instead of finding reasonable breakthrough curves, all of these realizations of \( \log(K_1) \) are low likelihood points, even if they are near high likelihood points, such as the point found by global calibration.

Figure 5.8: Error Pore Volumes vs. \( \log(K_1) \)
5.3 Assessment of the Normal Assumption

Figure 5.9 shows cumulative distribution functions for each parameter. The CDF is compared to a normal curve generated from the mean and standard deviation of the posterior PDF. The CDF of $\log(K1)$ is very different from what would be a normal CDF. $\log(K3)$ has significant bias from the mean value of the CDF to the global optimum. This is also seen by Figure 5.3 for $\log(K3)$ where most realizations were around $-2.5$ instead of the calibrated parameter $-1.9$.

$K2$ and $Sites(f)$ appear to similarly mimic a normal CDF, with some bias and deformation. It is not clear that a qualitative judgement can be made from Figure 5.9 to discount any of these CDFs as normal. To accurately determine if the MCMC posterior distributions are normal, the Lilliefors test is used.

![Figure 5.9: Cumulative Distribution Functions, 30000 Realizations](image-url)
5.3.1 Lilliefors Test

The Lilliefors test is a variation of a Kolmogorov-Smirnov test. The Kolmogorov-Smirnov test compares two cumulative distribution functions, a hypothetical distribution and a test distribution. The test proposes the null hypothesis that the test model comes from a distribution which similarly describes the hypothetical distribution. The Kolmogorov-Smirnov test requires the knowledge of such a hypothetical model to calculate the statistic $D$ by equation (5.1) where the CDF is the cumulative distribution function from the hypothetical model and the EDF is the empirical distribution function found from the test distribution (Davis, 2002).

\[
D = \max |CDF - EDF|
\]  

The statistic $D$ is compared to a given table of critical values for a desired significance level (Davis, 2002). If $D$ exceeds the critical value, the null hypothesis is rejected, stating that the test sample is not drawn from the given hypothetical model for the given significance level.

A requirement of the Kolmogorov-Smirnov which this study cannot provide is a known hypothetical model to describe a cumulative distribution function. A question this study seeks to answer is if the parameters follow normal distributions. The Lilliefors test modifies the Kolmogorov-Smirnov test to compare an empirical data set to a normal distribution with unknown mean and variance. The test statistic equation (5.1) remains the same. The null hypothesis in this case is such that the test model comes from a normal distribution (Davis, 2002).

A hypothesis being accepted or rejected depends upon the significance level of the test. The significance level is related to Type I and Type II errors. A Type I error occurs when the hypothesis is rejected by the test statistic, but the hypothesis is true in reality. Conversely, a Type II error occurs when the hypothesis is accepted, but the hypothesis is actually false. This is summarized in figure 5.10. The probability of a Type I error is defined by the significance level. If the significance level is too high, it is more likely that a Type I error can occur as the hypothesis is more likely to be rejected. In general, the probability of a Type II error increases when the significance level decreases (Haan, 1977).

MATLAB provides a function lillietest which tests a vector of empirical data for a significance level between $[0.001, 0.50]$. The lillietest function returns 1 if the null hypothesis is rejected, and
0 if it cannot be rejected for the given significance level. For a baseline comparison, a single normal distribution $x_{test}$ was generated using the random normal multivariate generation program discussed earlier in section 4.2 with a mean of 0 and standard deviation of 1. Each parameter’s distribution arrived from the MCMC simulation and the baseline dataset was tested for significance levels 0.001, 0.05, 0.50. The results of this test are given in figure 5.3.

<table>
<thead>
<tr>
<th>Significance Level</th>
<th>Log(K1)</th>
<th>Log(K2)</th>
<th>Log(K3)</th>
<th>Log(Sites)</th>
<th>$x_{test}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Accepted</td>
</tr>
<tr>
<td>0.05</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Accepted</td>
</tr>
<tr>
<td>0.50</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Rejected</td>
<td>Accepted</td>
</tr>
</tbody>
</table>

Table 5.3: Lilliefors Hypothesis Test Results

The Lilliefors test results show that the null hypothesis is rejected for the entire range of significance levels provided by MATLAB. The null hypothesis that the baseline $x_{test}$ random variable is subject to a normal distribution was accepted. This does not mean that the baseline test truly is normal, but it is possible, as the hypothesis is not rejected. For the parameters of interest, the null hypothesis was rejected. This means that none of the parameters follow a normal distribution.

## 5.4 Chain Convergence

MCMC chain convergence is an open field of research. A problem with the concept of convergence for MCMC simulations is that it is not possible to be certain that a finite chain is representative of an underlying stationary distribution (Cowles and Carlin, 1996). Given that a computational machine is finite, no convergence analysis can guarantee that the true posterior PDF has been determined.

The scaling parameter introduced by the covariance-adaptive method did not converge to any stationary distribution. The final values for the acceptance ratio and the scaling parameters are presented in Table 5.4. Ideally, with enough realizations an optimal scaling parameter would be
found when using the covariance-adaptive method (Roberts and Rosenthal, 2001). The target acceptance ratio of 0.25 was roughly achieved, despite the scaling parameter discrepancies.

<table>
<thead>
<tr>
<th>Chain Number</th>
<th>Acceptance Ratio</th>
<th>Scaling Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2523</td>
<td>209880.020729</td>
</tr>
<tr>
<td>2</td>
<td>0.2398</td>
<td>135068.787824</td>
</tr>
<tr>
<td>3</td>
<td>0.2571</td>
<td>54601.9486985</td>
</tr>
</tbody>
</table>

Table 5.4: Chain Acceptance Ratios and Scaling Parameters

Figure 5.11 shows the mixing of the 3 MCMC chains for all 10000 realizations. Chain 3 has a clear divergence in parameter \( \log(K3) \) from realizations 4000 to 8000 to values of \(-2.0\). However, both chains 1 and 2 traverse these values of \( \log(K3) \) from realizations 2500 to 5000. A similar behavior is seen in parameter \( \log(K1) \). Chains 2 and 3 move to values for \( \log(K1) \) around \(-6.0\) from realizations 5000 to 7000. Chain 1 also moved to this value of \( \log(K1) \) briefly around realization 1000. Figure 5.11 shows with test point T4 that \(-6.0\) for \( \log(K1) \) is a reasonable value to generate a breakthrough curve. The \( \log(K3) \) response surface in Figure 5.3 shows realizations at value \(-2.0\) in the likelihood range of \((0.8, 0.9)\). These multiple modes is in contrast to the ideal model for an MCMC simulation, where there is one mode to converge to for a parameter. It appears that the mixing behavior in Figure 5.11 for parameters \( \log(K2) \) and Sites(\( f \)) is good.

To quantify the convergence of all parameters, the Gelman-Rubin convergence test is used. The Gelman-Rubin convergence test is passed if the test statistic scalar is less than 1.2 for most models (Gelman et al., 2004). For these parameters, Figure 5.12 shows that the parameters \( \log(K2) \) and Sites(\( f \)) satisfy this requirement. \( \log(K1) \) and \( \log(K3) \), where it was previously observed may exhibit multi-modal behavior, do generally remain under 1.2. However, it appears they may be telescoping away from convergence sinusoidally. \( \log(K1) \) first shows a spike of divergence from realizations 500 to 1000. Another, more severe, spike of divergence for \( \log(K1) \) is seen from realizations 2000 to 7000. This spike is due to the values of \( \log(K1) \) explored around \(-6.0\) from realizations 5000 to 7000 seen in Figure 5.11. All of the parameters converge nearly to 1 from realizations 7000 to 10000, but there is no guarantee that a parameter will not diverge again in the future if the MCMC simulation were to continue.
Figure 5.11: Chain mixing for 3 chains, 10000 realizations each
Figure 5.12: Gelman-Rubin Chain Convergence, 3 Chains of 10000
5.5 Prediction Intervals

A prediction was made using the distributions generated from the three (3) MCMC chains simulated in this study. The likelihood of the parameters given to the MCMC chains were subject to model C4 and experiments 1, 2, and 8 given by Equation 4.5. The prediction is completed using the same model C4, but for experiment 5 of the Kohler et al. (1996) column experiments. The prediction interval is generating by sorting each value predicted by $C/C_0$, grouped by each point in time given by pore volumes. The median line plotted in this interval shows the middle of the simulated values for a particular pore volume.

Five prediction intervals are shown in Figure 5.13 with intervals of 50%, 70%, 90%, 95%, and 99%. None of these prediction intervals include a full coverage of the experimental data. For the 99% interval, one data point is left out. Even though this one point is missed, this is a better result than both the linear and nonlinear analysis given by Figure 1.2. The linear interval missed all but 2 data points. The nonlinear analysis missed 8 data points.

The lower interval is only visible in Figure 5.13 for low prediction intervals 50% and 70%. This indicates many breakthrough curves with very little concentration of U(VI) seen in simulation for these pore volumes. The upper interval fails to capture the peak concentration, but covers all but a few data points for all intervals.

The 95% interval in Figure 5.13 includes the linear and nonlinear analysis from Figure 1.2. MCMC results in more predictive coverage than either linear or nonlinear analysis. The nonlinear analysis is better than the linear analysis due to the nonlinearity of the model. However, the nonlinear analysis doesn’t make physical sense. The nonlinear analysis shows concentrations much greater than 1, which is physically impossible. The intervals generated by MCMC do not violate physical laws, which makes it a more reliable uncertainty analysis of the model.
Figure 5.13: Various prediction intervals for experiment 5, given realizations determined by 30000 accepted MCMC points
CHAPTER 6

CONCLUSIONS

This study completed global calibration, MOAT, MCMC, and prediction interval analyses for the reactive-transport column model provided by Kohler et al. (1996) C4. The Kohler et al. (1996) model C4 was not the best fit to the experimental data. It was chosen because it was the best in terms of fit and model complexity. Model C5 has the best fit to experimental data, but includes more parameters. It may be the case that a more complicated model may correct the errors seen in model C4. The methods employed and code written are able to study other Kohler et al. (1996) models.

The global optimization performed was more effective than the previous local calibration. Due to the complexity of the model response surface, there are many local optima. The second-order optimization BFGS method from DAKOTA was able to find a better optimal point than UCODE.

The MOAT method was able to show the relative sensitivity of parameters. This result was confirmed by MCMC analysis. The high sensitivity of all parameters is compounded by a very high correlation between every parameter.

The three MCMC simulations which ran were able to achieve acceptance rates close to the target 25% by the use of the covariance-adaptive method. The use of the covariance-adaptive method was necessary and successfully searched the parameter space. Gelman et al. (2004) suggests that a value of the Gelman-Rubin convergence statistic less than 1.2 is satisfactory, which is achieved by this study for all model parameters in less than 1000 realizations.

The scaling parameter used for the proposal distribution did not converge between chains in 30,000 realizations. This may be due to a target acceptance rate that is too low. It may also be corrected by completing more realizations in each chain.
The prediction intervals generated using the realizations from MCMC analysis effectively generated posterior parameter distribution functions. None of the parameter distributions are normal. This invalidates any methodology used to assess parametric uncertainty which assumes a normal distribution of parameters.

This study found that model C4 is unable to predict some data points of the peak concentration of U(VI). The peak concentration is an important aspect to have a high confidence in prediction. Without a high confidence in understanding the peak concentration of U(VI), a decision cannot be properly made using the model for either determining whether a safe amount of U(VI) will be seen.

Another important aspect of simulation of the breakthrough of U(VI) is the long tail of the curve after the peak. This is critical in assessing the effectiveness of a remedial clean-up. The observational data for the tail of the breakthrough curve is completely covered by the prediction intervals generated from the MCMC realizations. This may show that the model is effective in describing the continued breakthrough of U(VI) over time.

The Model Runner Application was able to store previous realizations of the model. MCMC simulations were run in parallel. This decreased the computational time of a large MCMC simulation significantly from a worst-case scenario of nearly a year to two weeks. The naive parallel Monte Carlo application was also successful in decreasing the computational time required to make predictions from the posterior PDFs generated from the MCMC study.

6.1 Method Effectiveness for Parametric Uncertainty

Both MOAT and MCMC analyses was useful for assessment of parametric uncertainty. MOAT analysis would be more useful for a model with a large number of parameters. MOAT still resulted in a relative sensitivity of the parameters. MCMC was computationally expensive, but individual chains were easily parallelizable. The MCMC simulations benefit greatly from remembering the result of previous realizations.

6.1.1 Effectiveness of MOAT Analysis

The MOAT analysis determined the mean and standard deviation of the elementary effects of each parameter to the objective function used to assess the model, Equation 1.3. Each parameter was found to have a mean and standard deviation effects on the order of $10^6$. The MOAT analysis
was able to determine that all of the parameters involved are important to the predictive capability of the model. The relative importance of each parameter cannot be inferred with certainty from the result of this MOAT analysis. However, the relative importance of these parameters was confirmed by the MCMC analysis.  

Campolongo et al. (2007) concluded using a similar MOAT study that when drastic changes in a model output are observed using parameter values within the bounds of uncertainty that the model needs reassessment. These drastic changes were not as simple as nonlinear effects. The variation of the parameter over a reasonable range for that parameter resulted in predictions with very little certainty in predictive capability. Model C4 could benefit from a more sophisticated understanding of all of the parameters, more attention to the objective function used, and possibly an assessment of the conceptual model itself. This conclusion led to the decision to complete an MCMC study to determine a posterior parameter distribution. The understanding of the parameter distributions and correlations may lead to a more useful prediction interval than the linear and non-linear analyses already completed.

6.1.2 Effectiveness of MCMC

Results from the Gelman-Rubin convergence statistic in the model parameters is concerning. It appears that chain 2 was stuck in a parameter subspace for parameters $\log(K_1)$ and $\log(K_3)$ for a significant portion of the simulation. Before and after the divergent during that chain’s simulation, it mixed correctly with the other chains. Fortunately, this divergent range was not outside of the range the other chains searched. Without considering the order of the realizations made during the simulation, all of the parameters appear to be mixed within the same range. The divergent behavior was exhibited primarily in the first equilibrium constant, which has been seen to be the most sensitive. Despite the divergent period from realizations 2000-6000 for chain 2, the likelihood still converged very quickly in the first 1% of the simulation. However, it could be that the parameters $\log(K_1)$ and $\log(K_3)$ have not converged to a stationary distribution. This may imply a point in the future of the chain that becomes more divergent.

A concerning aspect of the MCMC simulations is the scaling parameter results. The scaling parameters used did not converge to an optimal scaling value for the MCMC simulations over the three chains completed. The scaling parameters differed on several orders of magnitude. This
resulted from the different directions the MCMC simulations took, which cause different amounts of parameters to be accepted or rejected. When a chain diverges to an area of high or low acceptance, the scaling parameter is adjusted each time accordingly. The difference in the scalings occurs from overcorrection the scaling parameter attempts. The MCMC simulation could not be run long enough to find the optimal scaling parameter for this study.

The MCMC analysis also revealed points in the parameter space of high likelihood that are vastly different than the calibrated parameters previously accepted. None of these points had a higher likelihood than the calibration, but still fit experimental data well. This indicates that the calibration of the parameters almost certainly does not represent a true value of reaction equilibrium constants seen in nature. Since the calibration can be mimicked by random points far away in the parameter space, it is a leap in logic to assert that the calibration values are more correct to nature simply because it matches a limited amount of data collected slightly better. It is already seen that even the calibration does not accurately match the breakthrough curves, especially for peak concentrations.

6.2 Critical Analysis of Results

There are some problems with the results from this study for the model chosen. A critical analysis of the prediction intervals, objective function, parameter space, PDFs, and convergence may lead to a better continued study of parametric uncertainty for reactive transport models.

6.2.1 On the Prediction Intervals

A significant number of realizations within the bounds of parameter uncertainty result in breakthrough curves representing almost a complete absence of U(VI) exiting the column over the course of the experiment. This is shown by the lower bounds of predictions intervals being close to zero concentration. The simulation period is not subject to a mass balance conservation to be achieved over the course of the experimental time frame. This is not a concern for predictive uncertainty. However, this leads to the question whether the uranium will eventually be seen exiting the column at any time. Since the simulation does not find the complete concentration of U(VI) at the end of
the experiment, it must be assumed that the U(VI) is still in the column. It still may be the case that U(VI) will eventually exit the column, but this event was not predicted by many simulations.

6.2.2 On the Objective Function

The most critical element to this research is the objective function used [1.3]. This function is used to both calibrate the model and also compute model likelihood during MCMC analysis. Making a prediction using the model not only pivots on the effectiveness of the model, but also the usefulness of the objective function in describing the model. It was a clear choice to use every data point in the breakthrough curve with its respective measurement error to construct the SSWR objective function. Very low measurement error, quantified by variance, creates objective function values which are highly sensitive to simulated deviations from the observations. The high sensitivity seen in MOAT analysis primarily due to the large variation in the objective function seen (90,000-2,000,000) when the parameter space only varied ±10% centered around the calibrated values. This was not an artifact caused by changing the objective function values for a zero breakthrough curve, failed simulation, or the prior distribution. The likelihood was only set to zero for the MCMC simulations for those cases, and was not used in the MOAT analysis.

It is also important to note that the numerical variance-covariance matrix computed in this study not only has very small values, \(10^{-7}, 10^{-5}\), for the variance of the parameters, but also covariance. It might seem obvious to simply conclude that this means we have a high confidence in finding the proper calibrated values, as the parameters do not vary around those values in any direction. However, what this leads to is a very poor predictive capability, similar to linear analysis.

6.2.3 On the Parameter Space

The parameter space for the reactive transport model used in this study is unknown not only in range, but also in dimensions. The purpose of this study was to assess the parametric uncertainty and predictive capability of To determine the best number of functional groups used in the model, the Kohler et al. [1996] experiment essentially performed a Monte Carlo analysis in the reaction parameter space with extremely low number of realizations. Not only is the reaction affinity and site fraction unknown quantities for a given model, but the number of reactions and functional groups is unknown in itself. Kohler et al. [1996] created 7 column models, which can be seen as
7 realizations along the dimensions of reaction types and number of functional groups. This study concentrated on model C4 which assumes there are three (3) reactions and two (2) functional groups. There is no verification for the validity of such an assumption except for achieving the best calibrated fit to experimental data.

Another contribution to the extreme sensitivity of the objective function to the model parameters is the use of a log scale. A huge parameter space of $10000^4$ is compressed into the log space of $4^4$. Considering the log scale, which is commonly used in chemical studies for chemical equilibrium, $N = 30000$ realizations may not cover the parameter space as well as it appears when the parameter range is only 4 in each dimension. Certainly, $N = 1000$ realizations completed during the naive Monte Carlo analysis is not enough to cover the parameter space.

### 6.2.4 On the Posterior Distribution Functions

The posterior parameter distributions determined in this study resulted in a more useful prediction interval than linear and non-linear analyses. These parameter distributions, however, tell us nothing about the true nature of reactive-transport phenomena. The reactions chosen to be simulated in the Kohler et al. (1996) model C4 could both be gross approximate subset of the actual reactions which exist in the column. The reactions assumed to exist in the column for simulation may not truly occur. The posterior parameter distributions for the equilibrium constants only describe the effectiveness of the parameters chosen to drive the model to represent a single property of reactive-transport, the concentration of U(VI) exiting the column over time. The manner in which U(VI) arrived at the exit of the column experimentally was not assessed by this study.

### 6.2.5 On the Convergence of MCMC

The Gelman-Rubin convergence test may not be a proper method to determine the convergence of these simulations. This statistic has been criticized by Cowles and Carlin (1996) for three major reasons. First, it diagnoses convergence based on normal approximations. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. In this study it has been clearly seen that the parameters do not follow a normal distribution. It appears that parameters $\log(K1)$ and $\log(K3)$ may exhibit a multi-modal PDF. In the case of multi-modality it may not be feasible to expect that an MCMC chain would converge to only one parameter estimation. Second,
the Gelman-Rubin statistic is univariate. It has been seen in this study that the parameters have a similar order of covariance as variance. This observation may call for a convergence statistic to consider a multivariate simulation. Finally, Cowles and Carlin (1996) criticizes the Gelman-Rubin convergence statistic for requiring multiple chains to calculate between chain convergence. This criticism does not necessarily apply to this study, as individual chains are computed in parallel, so there is not a significant cost for this requirement. However, it is uncertain how many chains would be sufficient to present reliable data to the Gelman-Rubin convergence test.

6.3 Future Study

The model runner class was expanded to make use of a model data class. The model data class provides the necessary instructions for modification of input files to the reactive-transport computational simulations for all of the column experiments and models constructed by Kohler et al. (1996). The model data class only requires the model number as input, making the study of the other models simple. However, due to the computational time required to run these simulations, in continued study there was only time to complete a global calibration of models C5 and C6. With the exception of computational time, it would be trivial for the user of the applications created in this study to continue research on models C5, C6, and the other models provided by Kohler et al. (1996) in a manner outlined by this study.

Model C4 may not have the best predictive capability. The other models may result in a more reliable predictive coverage. If the reactions being modeled more closely match reality, the parametric sensitivity may not be as high. It is my hypothesis from this study that the extreme sensitivity seen in the parameters to model C4 is not only due to the high confidence in the experimental data. I believe it is also due to over-simplifications made by the model. Rojas et al. (2008) concludes that parametric uncertainty cannot compensate for conceptual model uncertainty. The conceptual model constructed by Kohler et al. (1996) includes 2-site and 3-site surface complexation models, but the number of reactive sites on the surface of the particles involved is unknown. Furthermore, the concept of surface reaction sites is a simplification of reality in itself.

The adaptive scaling parameters did not converge to an optimal scaling parameter value. This in itself indicates a divergence of the MCMC simulations. It is discussed by Atchade and Rosenthal.
that one-dimensional simulations may benefit from a larger acceptance rate than \( \tau = 0.25 \) used in this study, such as \( \tau = 0.44 \). This study involved a 4-d parameter space, so further understanding of the consequences of the acceptance rate should be explored. Also concerning the adaptive method, Atchade and Rosenthal (2005) completed a simulation example with \( w = 1,10,100 \) with \( w \) being the number of MCMC iterations before updating the scaling parameter. Atchade and Rosenthal (2005) did not see a benefit to waiting a certain number of MCMC iterations before making a scaling parameter update for \( N = 250,000 \) realizations. This study may benefit from employing a \( w \) value, rather than performing a scaling update every step. This study involved a factor of 10 less realizations than the study completed by Atchade and Rosenthal (2005) which may have an effect on results. However, the computational cost of such a number of realizations would take months to perform for the Kohler et al. (1996) models.

For parameters \( \log(K1) \) and \( \log(K3) \), the distributions may even be multi-modal. A convergence criteria for MCMC which can assess convergence for non-normal, multi-modal, and highly correlated parameters is required for further research on this topic.

Tebes-Stevens et al. (2001) did not conclude the same extreme sensitivity as seen in this study. This is due to using a only two synthetic time locations to perform sensitivity analysis of the total Uranium concentration seen. While this objective function is not applicable to model calibration, it does suggest that a better objective function could be chosen. In this study, the objective function used was extremely sensitive to any change from the experimental data, due to a high confidence in the experimental data points. The objective function values were so large, in fact, that the order of magnitude in the experimental error variance had to be ignored when applied to the Gaussian likelihood function. The proposed objective function may not be particularly useful for parametric studies. Other objective functions could be devised to validate the computational model. Other likelihood functions, such as those proposed by Rojas et al. (2008) may be useful tools as well.
BIBLIOGRAPHY


