2014-12-01

Design and Implementation of an Inverse Modeling Framework Using the Method of Anchored Distributions

Carlos Andres Osorio Murillo
Brigham Young University - Provo

Design and Implementation of an Inverse Modeling Framework Using the Method of Anchored Distributions

Carlos Andres Osorio Murillo

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Daniel P. Ames, Chair
Everett J. Nelson
Gustavious P. Williams
Norman L. Jones
Michael A. Scott

Department of Civil and Environmental Engineering

Brigham Young University

December 2014

ABSTRACT

Design and Implementation of an Inverse Modeling Framework Using the Method of Anchored Distributions

Carlos Andres Osorio Murillo
Department of Civil and Environmental Engineering, BYU
Doctor of Philosophy

Estimation of spatial random fields (SRFs) such as transmissivity or porosity is required for predicting groundwater flow and subsurface contaminant movement. Similarly, distributed parameter fields such as terrain roughness and evapotranspiration coefficients are required by other areas of environmental and earth sciences modeling. This dissertation presents an inverse modeling framework for characterizing SRFs called MAD#, which is an end-user software implementation of the Bayesian inverse modeling technique Method of Anchored Distributions (MAD). The MAD# framework allows modelers to "wrap" existing simulation modeling tools using an extensible driver architecture that exposes model parameters to the inversion engine. A compelling aspect of this model wrapping approach is that it does not require end-users to modify model configuration files; rather the model driver manages dynamic changes to model input and configuration files at run time. The MAD# framework is implemented in an open source software package with the goal of significantly lowering the barrier to using inverse modeling in education, research, and resource management. Toward this end, we introduce and test an intentionally simple user interface for simulation configuration, model driver integration, spatial domain and model output visualization, and evaluation of model convergence.

Keywords: Inverse modeling, Method of Anchored Distribution, Groundwater

ACKNOWLEDGEMENTS

I would like to particularly thank my supervisor, Professor Daniel Ames, who accepted me as a Ph.D. student to develop this research and has supported my work during these years. I express my sincere gratitude for his useful suggestions, invaluable advice, and patience throughout my Ph.D. My deep gratitude for everything professor Ames has done during this time.

Special thanks to Professor Yoram Rubin at University of California, Berkeley, for his time, comments, and contributions to enhance this research. I would like to address special thanks to the National Science Foundation for the funding of my research which has made my work possible. Along with Professor Ames, Professor Rubin and his Research group at Berkeley provided me with all the support needed to develop my research and get my degree.

I would like to express my special gratitude to my committee members: for their time, interest and collaboration to fulfill the requirements for my Ph.D. degree. In addition to them, I must acknowledge to Brigham Young University (BYU) my gradtitude for having been accepted, supported, and allowed to meet incredible people.

Finally, I express my deepest gratitude and love towards my dear wife. She always has been there for me, encouraging me over the years. With all her faith and love, I am able to fulfill my goals and pursue further goals.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Problem Statement

Environmental simulation modeling generally begins with the definition of a mathematical equation and input parameters that describes a particular physical phenomenon (Yeh W. W., 1986). When this model is applied in another scenario, it typically must be calibrated by changing its parameters. Inverse modeling is the process of using observations to estimate parameters of the model. This technique uses the available observations and the model itself to determine the characteristics of the parameters.

In environmental simulation models, key parameters often take the form of gridded or "field" data. Because these parameters are not easily characterized at all locations in a model domain, such spatially varying fields are sometimes represented as spatial "random" fields (SRFs) which, in reality are not truly "random" but can be used to describe heterogeneous variables distributed spatially (e.g., air temperature, soil permeability, etc.). The characterization of an SRF requires estimating the geostatistical or other spatial model parameters that define its behavior (e.g., nugget, sill, range, etc.). Inverse modeling techniques have been used to estimate these parameters. These techniques have also been applied in different scientific areas to calibrate models. In general, inverse modeling methods use optimization techniques that provide the so-called "best" solution of the model parameters. But, these optimization techniques can be affected by the partial evaluation of the parameter space and "the oversampling of high goodness of fit regions of the parameter space" (Mugunthan and Shoemaker, 2006). Alternatively, stochastic inverse modeling techniques can explore the parameter space reducing the bias in the parameter estimation. These techniques can be applied for estimating the parameter of SRFs. The purpose of this dissertation is to implement a stochastic inverse modeling for SRF characterization.

Inverse modeling usually requires configuring multiple software applications and demands programming skills to manage data generated during the inversion. This complex process reduces the adoption of the approach. In hydrology, Carrera and others (2005) identified some challenges in the adoption of inverse modeling. These challenges are:

1) Incorporate geological data;

2) Improve flexibility of the code and procedures to handle any and all relevant data types;

3) Complete quantification of uncertainty;

4) Reduce difficulty of code operation; and

5) Coupling inverse modeling techniques with a Geographic Information System (GIS) platform.

The Method of Anchored Distributions (MAD) presented by Rubin and others (2010) is a stochastic inverse modeling technique that addresses the first three challenges posed by Carrera (2005). The incorporation of geology data allows the representation of geological features through random fields modeled using structural parameters; handles multiple relevant data types through use of direct measurements and measurements that are indirectly related to them; and accommodates uncertainty by explicitly incorporating observation uncertainties and quantifying uncertainty of geostatistical structural parameters and specific locations in the domain. In spite of the contributions of the MAD method, it has not been widely adopted, largely because it still is extremely complicated to implement. Also it, in itself, does not address the last two challenges presented by Carrera. My expectation is that by building a generic software to fully implement MAD, I can help break down all of the Carrera-noted barriers, thereby helping the scientific community to incorporate inverse modeling as a common tool.

The challenge of a generic inverse modeling tool is to control the complexity of forward simulation model elements like configuration, platform, domain, format, units, etc. This dissertation will explore mechanisms to support different forward models, specifically hydrological models, for the implementation of a generic inverse modeling tool focused on determination of SRFs using MAD.

The approach presented here can be considered in terms of a numerical methods research project where our governing equation (forward model), at least in the groundwater case studies presented, is generally the simple dispersion model proposed by Henry Darcy in 1856.

$$\boldsymbol{Q} = \frac{-\boldsymbol{kA}}{\boldsymbol{\mu}}\frac{(\boldsymbol{p_b} - \boldsymbol{p_a})}{\boldsymbol{L}} \qquad (1\text{-}1)$$

where Q is total discharge, *k* is permeability of the soil, *A* is the cross sectional area of the flow unit, *L* is the water travel distance, and p_b-p_a represents the total pressure drop and μ is fluid viscosity. Unfortunately, when dealing with groundwater problems, it is impossible to perfectly characterize the coefficients or model parameters, μ and *k*, hence we use a Monte Carlo and Bayesian approach to invert the model and solve for the coefficients. In practical terms, this equation and related equations defining the movement of water in the subsurface are coded into open and closed source software packages. Hence through the remainder of this dissertation, I will use the term “forward model” to refer to both the underlying governing equations of a particular process of interest, and to refer to the software implementations of such equations.

1.2 **Related Work**

Data assimilation is a general term referring to methods that use a model together with observations to improve model performance. Data assimilation includes methods for parameter estimation, sensitivity analysis, and observation system design (Anderson et al., 2009). Inverse modeling techniques are a subset of data assimilation technologies. This research is focused on inverse modeling.

Inverse modeling techniques are often applied in hydrology for quantifying the uncertainty of parameters using observed data (Yeh et al., 2002), in hydrogeology for estimation of flow parameters and geological structures (Farmani et al., 2008), and in oceanography and atmospheric science for estimating large spatially distributed parameter fields (Bennet, 2002). Generic inverse modeling algorithms are available in software many commercial software applications including Mathematica (http://www.wolfram.com/mathematica/) and MatLab (http://www.mathworks.com/). Custom inverse modeling tools have also been developed to support parameter estimation of specific simulation models. For example, Parameter Estimation (PEST) (Doherty 1994), can be linked with MODFLOW to estimate groundwater parameters. PEST can also be used by other models such as iTOUGH2 (Finsterle & Zhag, 2011), PMWIN (Chiang & Kinzelbach 2001) and eWater (McCloskey et al, 2011) for modeling of flow through porous media, generic groundwater modeling, and catchment modeling respectively.

While model calibration using optimization algorithms in general is a widely understood and commonly executed task by modelers and scientists, stochastic inverse modeling procedures are not particularly widely used likely due to difficulty of implementation. The complicated nature of most hydrologic simulation models also contributes to the low adoption of an inverse modeling algorithms. Scientific scripting applications such as R (http://www.r-project.org/), use internal packages to execute inverse modeling procedures and hence are more suited to models completely written within the R environment – not externally compiled and executed models (e.g., written in FORTRAN). These types of model executables need external applications for parameter estimation. I believe that users and developers of specialized forward simulation models can take advantage of inverse modeling algorithms and reduce redundant development time (i.e., time that would be spent building a custom inverse modeling tool) by using an external inverse modeling software framework.

1.3 Research Objective and Scope

This research is focused on the characterization of SRFs using stochastic inverse modeling. The main objective is the implementation and testing of the inverse modeling technique, called MAD, using an extensible, user-friendly software framework approach which addresses Carrera's challenges. The framework contributes in the adoption of stochastic inverse modeling that reduces the complexity in the setup of inverse problems in hydrology. A secondary goal is to demonstrate the utility of such a framework within a HPC environment. The final goal is to demonstrate the utility of the framework for solving complex parameter optimization problems – in this case the Levenberg-Marquardt solution space for nonlinear least-squares assessment.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 describes the core contribution of the research, development of an inverse modeling software framework (MAD#) for implementing the MAD inverse modeling technique with any external, pre-compiled, model executable. This chapter has been submitted to the journal, Environmental Modeling & Software and was returned for revisions. The revised paper has since been re-submitted and is in review.

Chapter 3 is a brief peer-reviewed conference paper that was presented at the 7th International Congress on Environmental Modeling & Software (iEMSs 2014) and describes the integration of MAD# with a distributed computing platform (HTCondor). Chapter 4 presents a MAD# case study focused on the evaluation of the solution space of an optimization algorithm. This chapter has been prepared as a standalone technical paper that will be submitted to the journal, Computers & Geosciences for publication. A complete user manual for the developed MAD# software is provided in Appendix A. All references from all chapters are included in a single references section.

.

2 A NOVEL INVERSE MODELING AND UNCERTAINTY CHARACTERIZATION SOFTWARE FRAMEWORK AND GROUNDWATER ASSESSMENT CASE STUDY

Co-authors: Matthew Over, Heather Savoy, Daniel P. Ames, Yoram Rubin

Submitted for publication to Environmental Modelling & Software (09/2014)

2.1 Overview

Spatial phenomena variability is typically evaluated through analytical and numerical models that describe the general properties of spatial random fields (SRFs). These models employ parameters and observations to define spatial variability. The characteristics – and hence variability – of an SRF can be discerned by the relationship between model parameters, direct, and indirect information. A number of hydrogeological studies have been conducted using SRF analysis (Delhomme, 1979; Carrera & Neuman, 1986; Dagan, 1987; Bates & Townley, 1988; Bellin & Rubin, 1996; Yeh et al., 2002; Kanso et al., 2003; Gallagher & Doherty, 2007; Farmani et al., 2008). This chapter introduces an open source inverse modeling framework, called MAD# (pronounced "mad sharp"), focused on the characterization of SRFs using the Method of Anchored Distributions (MAD), a Bayesian inverse modeling technique (Rubin et al., 2010).

The process of estimating model parameters from the inversion of governing equation(s) and observations is called inverse modeling. For over fifteen years, researchers have advocated for the development of flexible and easy-to-use inverse modeling tools, with the understanding that the shortage of such tools hinders the development of comprehensive and credible uncertainty quantification tools (Poeter & Hill, 1997; Poeter & Hill, 1999; Rubin, 2004; Dagan, 2011). Carrera et al. (2005) identified five features that are needed for broad adoption of inverse modeling tools in hydrogeology: 1) incorporating geological data, 2) improving the flexibility of the code and procedures to handle any and all relevant data types, 3) a complete quantification of uncertainty, 4) reducing the difficulty of code operation, and 5) coupling inverse modeling techniques with a geographic information system (GIS) platform.

A number of existing simulation model software tools include model parameter estimation and uncertainty characterization as embedded functions within the program. For example, WEAP (Yates et al., 2005) and PMWIN (Chiang & Kinzelbach 2001) both are applications that use forward models (FMs) and model parameter estimation software applications like PEST (Doherty, 1994). These and related software tools have aided adoption of uncertainty characterization and inverse modeling to some degree. However, additional tools are needed that provide a more general set of capabilities and that address the issues raised by Carrera et al. (2005).

MAD has been shown by Rubin et al. (2010), Murakami et al. (2011), and Chen et al. (2012) to be a flexible stochastic inverse modeling technique that addresses the first three challenges posed by Carrera et al. (2005). Specifically, MAD can account for geology (Challenge #1) via the representation of geological features through SRFs modeled using structural parameters; handles multiple relevant data types (Challenge #2) through use of direct measurements and measurements that are indirectly related to the variable modeled; and accommodates uncertainty (Challenge #3) by explicitly incorporating observation uncertainties and quantifying uncertainty of geostatistical structural parameters and a new concept called "anchors".

2.2 Chapter Goals

The purpose of this research is to address Carrera's Challenges #4 and #5 by implementing and testing MAD in an extensible, user-friendly software framework. Specific goals for the developed framework include:

1) It should be capable of generically accommodating FMs that relate target variables with observations.

2) It also should be flexible in supporting the use of other user-specified software packages for random field generators (RFGs).

3) It should be able to characterize the uncertainty associated with SRFs.

4) It should be well-documented and transparent with independently verifiable results.

The remainder of this chapter presents the approach to meeting the research goals noted above in the form of an open source inverse modeling software framework called MAD#. This new inverse modeling application builds upon a prototype architecture (Osorio et al., 2012), in which MAD was implemented as a HydroDesktop (Ames et al., 2012) plugin using an embedded steady-state head solver written in R statistical software. MAD# is a standalone desktop application and includes an architecture for adding custom random field generator drivers (RFGDs) and forward model drivers (FMDs) for incorporating new models. We present an architectural overview of MAD# and descriptions of drivers currently implemented. We also present a demonstration of MAD# in two synthetic pumping experiments using a MODFLOW (Harbaugh, 1996) project created in the PMWIN interface (Chiang & Kinzelbach, 2001).

2.3 MAD Theoretical Background

Although a complete description of MAD is outside the scope of this paper, a brief introduction to the method is presented here. MAD is a Bayesian inverse modeling technique focused on characterizing SRFs by using Bayes' theorem and the following concepts intended to address the challenges stated in the previous section:

- Geostatistical models are used to capture large-scale trends and reproduce patterns of spatial variability in terms of SRFs.
- Data classification – MAD classifies data (measurements) in a general format that is not limited (or specific to) any particular discipline or application. MAD categorizes data as:
  - Type A data, z_a=$y(x_i)$ + ε_a, i=$1,..,N$, which could include direct measurements (including measurement error ε) of the target variables (e.g., hydraulic conductivity) at location x_i, i=$1,..,N$, or other types of measurements (e.g., transmissivity) at xi that could be directly related to the target variable at x_i,
  - Type B data, z_b=$M(x_i)$ + ε_b, j=$1,..,M$, which include all measurements (including measurement error ε) that do not conform with Type-A, but are related to the target variable via a forward model, M (e.g., pressure head)
- Localization through anchored distributions (or "anchors"). An anchor is a statistical distribution of a target variable at a given location. Anchors can be employed for multiple target variables and/or locations. Anchors intend to capture local effects in the field of the target variables by conditioning realizations on fields.

MAD defines a target variable as an SRF, which is represented by a vector of geostatistical structural parameters (θ) capturing the global tendency, and anchors (ϑ) for quantifying local variations of the parameter field. MAD relies on the following proportionality (Rubin et al., 2010)

$$p(\theta,\vartheta|z_a,z_b)\ \propto p(\theta,\vartheta|z_a)\,p(z_b|\,\theta,\vartheta,z_a) \qquad (2\text{-}1)$$

where p indicates a probability density function (pdf) and $p(\theta,\vartheta|z_a)$ is the joint prior distribution of the structural parameters and anchors conditional on Type-A data vector z_a, and $p(z_b|\,\theta,\vartheta,z_a)$ is the likelihood of observing the Type-B data vector z_b given the structural parameters, anchors and Type-A data. Finally, $p(\theta,\vartheta|z_a,z_b)$ is the joint posterior distribution of the structural parameters and anchors conditional on both Type-A and Type-B data.

2.4 **MAD Methodological Approach**

MAD is applied in three stages: 1) Strategy, 2) Implementation, and 3) Assessment. These three stages are described in Figure 2-1 and are discussed in depth in the following three subsections.

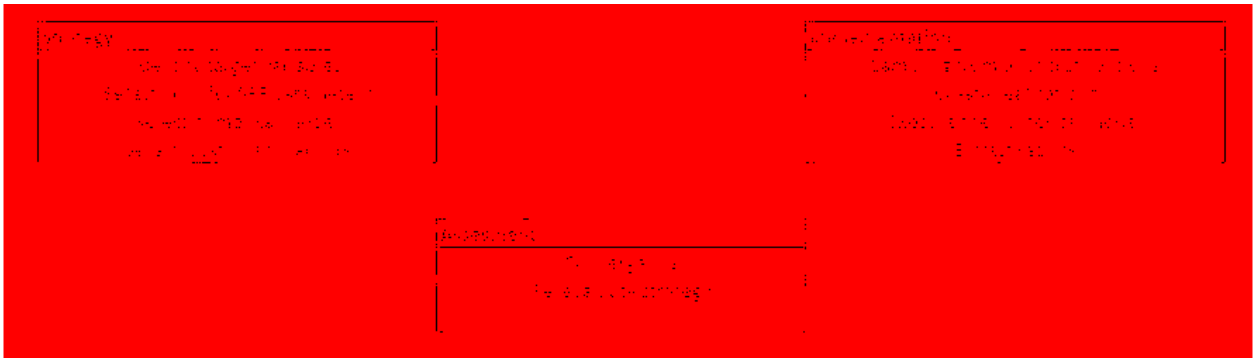


Figure 2-1: Structure of the MAD process

2.4.1 Strategy

The first stage of applying MAD to a case study is the formulation of a strategy. This strategy entails the following six elements: (1) identifying target variables, (2) selecting appropriate priors for the SRF parameters, (3) identifying types of data available, (4) selecting numerical modeling strategy, (5) selecting locations for anchors, and (6) planning post-calibration model testing. A target variable is typically a heterogeneous variable that needs to be characterized. Measurements of target variables or directly related variables are classified as Type-A data. An SRF model type to describe a target variable's spatial variability in the field needs to be chosen, and this choice can be made from analyzing the measurements and previous literature. The second consideration is choosing appropriate priors for the parameters of the SRF model types chosen for the target variables. After these first two steps, the target variable is set for MAD.

The third step focuses on identifying the data that could be used for the inversion process. Since the inversion requires indirect data, measurements need to be taken of a variable that the target variable influences via a mathematical model. These measurements are classified as Type-B data. This inversion data type generally describes larger-scale phenomena than the target variable and thus the combination of the two better informs the inversion process.

The fourth step is creating a numerical model. The numerical model needs to take the target variable, or a variable directly related, as an input and produce the inversion data type as an output. Any relevant environmental influences from the site (e.g., wells or streams) need to be numerically represented. Other considerations that are necessary for building a well-posed model are also required, including dimensionality, boundary conditions, and time dependence. The

Type-A and Type-B data need to be collected from within the domain selected and adequately distant from the boundaries to prevent interference.

The fifth step is placing the anchors in locations that are influential in the environmental process being modeled. Optimal selection of anchor locations was discussed by Yang et al. (2012). Each anchor also requires a prior distribution defined by prior knowledge.

Finally, one must choose an evaluation method for assessing the success of the inversion approach and testing calibration. Multiple inversions can be performed and cross-validation can be used to determine which approach is most successful. At this point, the strategy is developed and MAD can be implemented.

2.4.2 Implementation

The four steps of the implementation stage are (1) sampling from prior distributions, (2) creating realizations, (3) executing numerical model simulations, and (4) extracting results.

The first two steps cover the sampling strategy. In the first step, each SRF model parameter and anchor will need to have its prior distribution sampled, creating θ_i and ϑ_i, i in 1,..., N where N is the number of samples. The number of samples needs to be high enough to cover the parameter space, but choosing the number is not an exact science. Evaluating if there are not enough samples is covered in the assessment stage. In the second step, the samples from the previous step are used to create realizations of the target variable field. Each θ_i will define an SRF model to create realizations conditional on ϑ_i and the Type-A data.

The third step in the implementation stage is to run the simulations. For each realization created, the numerical FM is applied and a simulation is created. In the fourth step, the relevant Type-B simulated values are extracted and used to calculate likelihoods and posterior distributions. The extracted Type-B values are the $z_b$vector for which likelihoods are calculated.

The method for calculating likelihood is not specific to MAD such that any applicable to the $z_b$vector and Type-B measurements may be used. The likelihood method is applied to each sample, yielding a likelihood distribution across the sample space. This is multiplied by the prior distributions resulting in the posterior distributions and concluding the implementation stage.

2.4.3 Assessment

The assessment stage of the MAD approach focuses on assessing: (1) convergence and (2) general strategy. Convergence is assessed in two ways: if there were enough realizations and if there were enough samples. Generally, enough realizations are needed to estimate accurate likelihood values and enough samples are needed to resolve accurate posterior distributions. Graphical examples are given in section 3.0 on how to assess convergence.

The second kind of assessment, assessing general strategy, is more open-ended. The success evaluating technique chosen in the strategy stage is applied here. If the chosen success criteria are not met, then the strategy can be modified in several ways, including: increasing the total number of measurements if possible, changing the SRF model type or which parameters are random, or modifying the FM. If the success criteria are indeed met, it is still advisable to compare different parameters of the SRF model (e.g., covariance functions), or FMs to address model uncertainty. With this evaluation and acceptance of success, MAD has been thoroughly applied.

2.5 **MAD# Software Framework**

This section describes the design and development of an open source software framework that implements the MAD methodological approach described above. The software, MAD#, is designed as an extensible architecture that uses generic functions for sharing information, executing processes and extracting data of FMs and RFGs, which can be stand-alone software applications, packages of a statistical frameworks, or libraries. To support these diverse applications, MAD# framework uses a driver approach to connect the FMs and RFGs through forward model drivers (FMDs) and random field generator drivers (RFGDs) respectively with the framework. These drivers are libraries that implement a set of interfaces of the *MadInterfaces* library shown schematically in Figure 2-2.

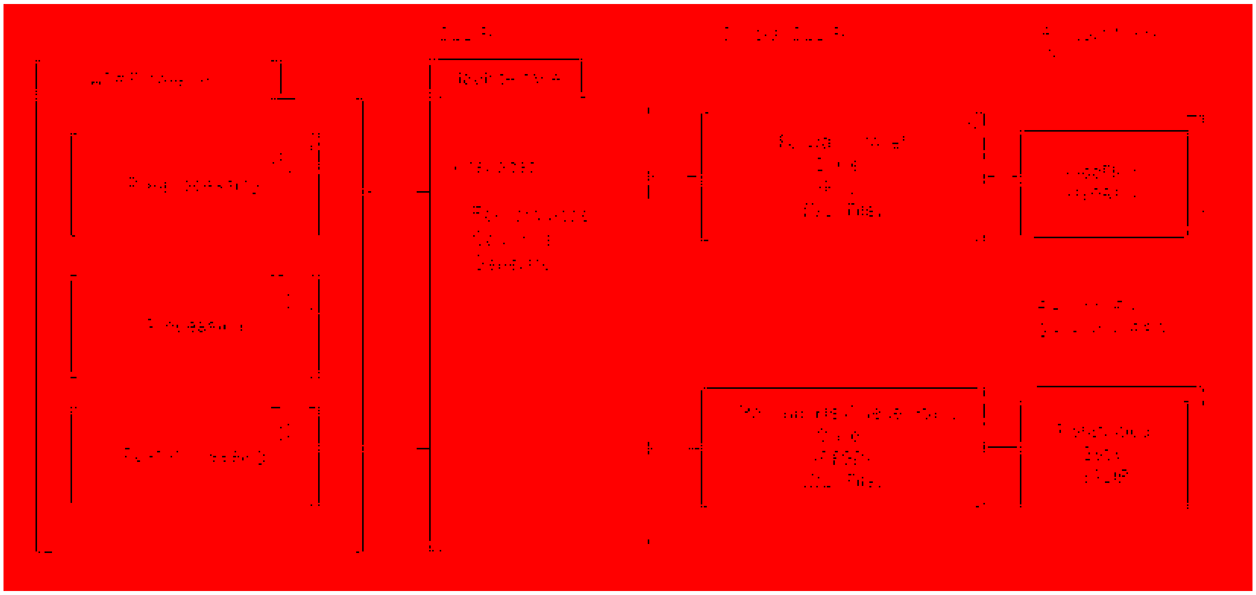


Figure 2-2: General MAD# framework architecture

This driver-based approach is expected to facilitate adoption and extension of the system by 3rd parties who can create new drivers for supporting new FMs or RFGs. MAD# is programmed using the .NET framework and the open source DotSpatial GIS programming library (http://www.dotspatial.org/) following a similar approach as used in Ames et al. (2012). An open source approach was chosen to support transparency and adaptability of the software

(Alexandrov et al., 2011). The DotSpatial library provides the geographic and display functionalities of MAD#, including data management, control, projection, symbology, and extension management. Statistical libraries including Math.NET (Math.NET, 2014) and ALGLIB (Bochkanov, 2014) are also used in the framework. A customized version of MapWindow 6 (Dunsford & Ames, 2011) serves as the final user interface of MAD#

2.5.1 Data Structure

The MAD# data structure is based on three entity-relationship models (Figure 2-3) implemented using open source database SQLite. Three SQLite file databases "XMAD", "XRESULT" and "XDATA" are created in different stages of the MAD# process.

The XMAD file stores the information generated in pre-processing module using tables. The variable table contains the list of variables provided by the FMD. These variables are used as key words during MAD# process. The Domain table stores the geographic information of the FM domain. The Measure table manages the Type-A and Type-B data and anchors. The prior information of the structural parameters and anchor are stored in the PriorData table. The z_b vector is stored in the SelectionValues table. This table is used to link the information generated by the simulation of the FM.

The processing module creates an XDATA database file for each sample. This approach works well with a large number of simulation files. These files store the simulated Type-B per realization in the ResultSelection table, which is related to the SelectionValues table matching each output with the z_b vector. The processing module creates a XRESULT file database that contains a copy of all tables of the XMAD file and the parameter used by the user to execute the simulation in the ConfigurationResult table. The XRESULT file is used in the post-processing module to store the likelihood calculation in the LikelihoodGroupValue table.

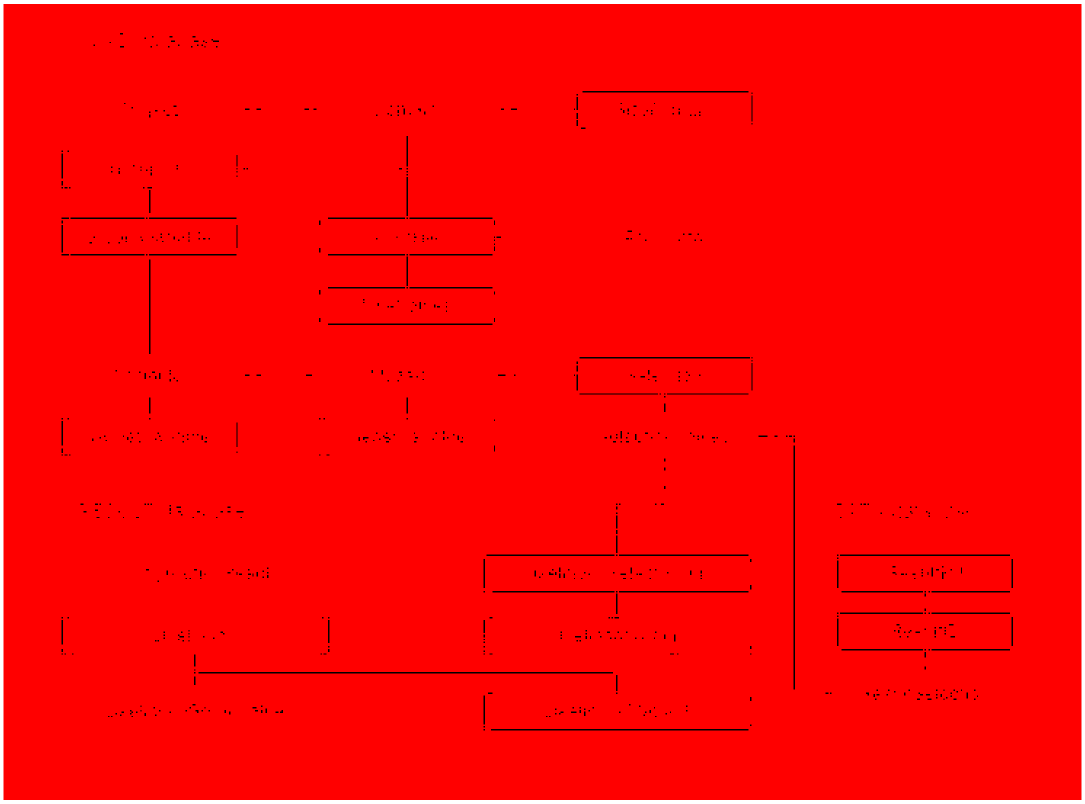


Figure 2-3: MAD# database structure

2.5.2 Drivers

The FMDs are simple model wrappers that can be developed by 3rd parties to enable use of specific models with MAD#. The FMDs also expose the list of inversion target variables, domain and temporal types. MAD# supports, by default, two forward models: PMWIN – MODFLOW 96 (Chiang & Kinzelbach, 2001), and HYDRUS-1D (Šimůnek et al., 1998).

A second type of MAD# driver is intended to support external programs capable of generating random spatial data fields (e.g., using geostatistics). RFGDs are used to define structural parameters (e.g., for a geostatistical model); establish random and deterministic structural parameters; and generate conditional fields. The RFGDs implemented in MAD# are: GSTAT – Based on GSTAT R package and stand-alone (Pebesma & Wesseling, 1998), R_Base_Package – Based on Mvtnorm (Genz & Bretz, 2009), Msm (Jackson, 2011), and Tmvtnorm (Wilhelmm & Manjunath, 2013) R packages.

Although the likelihood calculation is not a driver, in future releases, the calculation of the likelihood will managed as a driver. This approach will support coupling of different likelihood calculation methods with the framework. In the current version, the likelihood is calculated using a nonparametric kernel method through the R statistical package *NP* (Hayfield & Racine, 2008).

2.5.3 Implementation of MAD Stages

The strategy stage is addressed in the pre-processing module (Figure 2-4). This is an input module where the user selects a FMD and RFGD. The selected FMD obtains information about the geographical domain of the FM and the temporal nature of project (steady-state or transient). Geographic domain information is used to reproduce the same domain of the FM. A list of available variables classified in Type-A and Type-B are through FMD. These variables allow users to identify the target variable and type of data available. The target variable is considered a SRF, and defined by SRF parameters through RFGD. The selected SRF parameters are managed as inversion parameters. The module contains tools for introducing the location of observations, and anchors that are also inversion parameters. These parameters should be associated to prior distributions, which can be generated by the MAD# framework or imported by the user.

The MAD implementation stage is handled by the processing module (Figure 2-5) which is the core of the MAD# framework. The user defines the number of samples and realizations per samples to be executed. Using each sample, MAD# requests from the RFGD a number of realizations (defined by the user). The RFGD returns the realizations of the sample. MAD# processes the realizations of each sample in the FM using the FMD. The FMD extracts the simulated data at the location of each Type-B data measurement. Finally, the processing module generates a result file, which is a database file with all parameters used in the process, and output files with the information extracted from multiple executions of the FM.

The post-processing module completes the assessment stage (Figure 2-6). This module begins by assembling the simulated Type-B of the processing module by sample. The MAD# user can define different subsets of the Type-B data vector to evaluate the likelihood. The simulated Type-B and the Type-B subset are compared to calculate the likelihood per sample.

The convergence of the likelihood of each sample is evaluated using a graphical tool, which uses the likelihood of a sample with different amount of realizations. The evaluation of convergence of number of samples is done comparing the likelihood of all samples with different number of realizations. When the number of realizations or the number of samples is insufficient, it is necessary to add more realizations or samples. This process is executed again until the likelihood converges adequately. Using the likelihood of all samples is used to generate a posterior of the structural parameters and anchors. The appendices B.1, B.2, and B.3 show the pseudo code of each module.

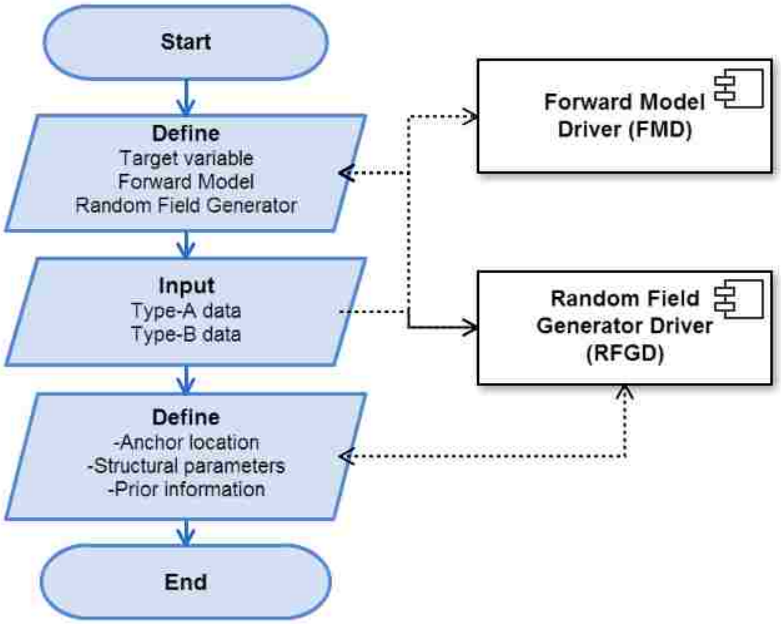


Figure 2-4: MAD# pre-processing module flow chart

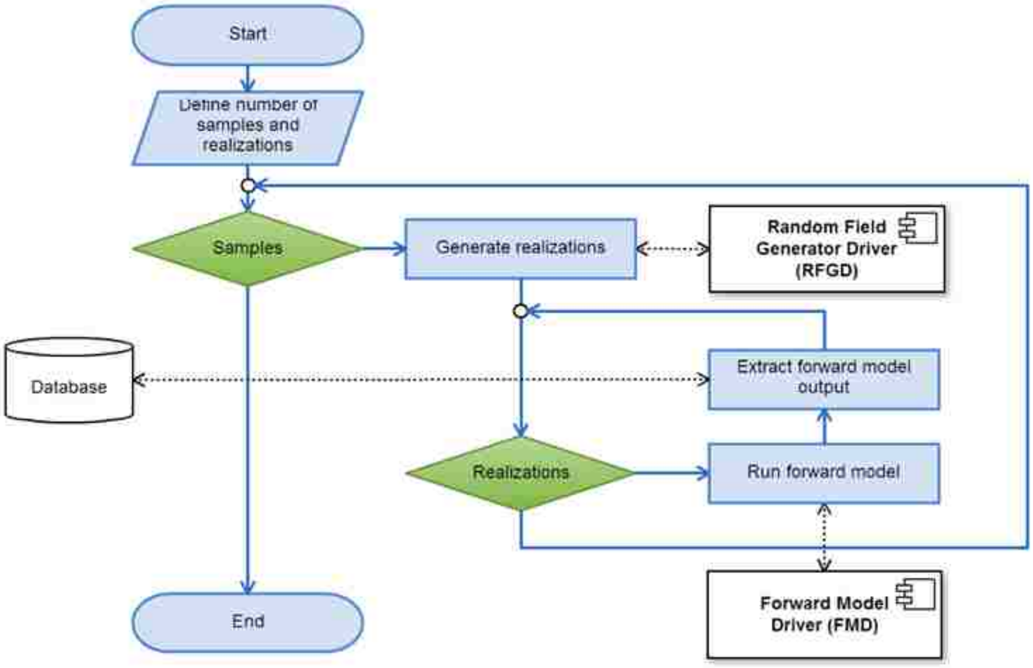

Start
Define number of samples and realizations
Samples
Generate realizations
Random Field Generator Driver (RFGD)
Database
Extract forward model output
Realizations
Run forward model
Forward Model Driver (FMD)
End

Figure 2-5: MAD# processing module

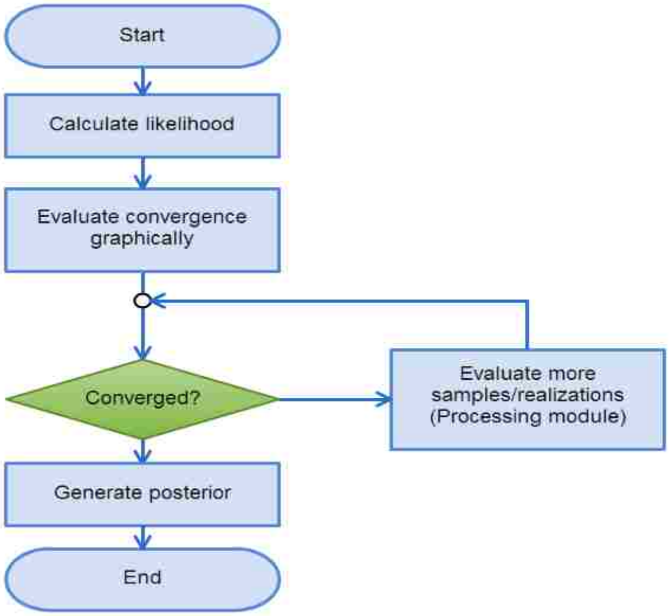

Start
Calculate likelihood
Evaluate convergence graphically
Converged?
Evaluate more samples/realizations (Processing module)
Generate posterior
End

Figure 2-6: MAD# processing flow chart

2.6 Example Test Cases

A base synthetic case with lateral confined groundwater flow through a heterogeneous aquifer was used for creating four test cases. These heterogeneous aquifers are of interest because in reality aquifers are heterogeneous and this heterogeneity can affect the travel time of contaminants to as sensitive areas like drinking water wells. A synthetic Ln(Transmissivity) field was created using GSTAT (Pebesma & Wesseling, 1998) with isotropic exponential covariance and no trend, which is common on synthetic projects (Li et al., 2005). The SRF uses the following structural parameters: mean = -2, variance = 0.15, range (length scale) = 28 meters, and nugget = zero meters. The field is 400x400 meters discretized into a 40x40 uniform rectangular grid.

From this baseline field, the Type-A data were collected and the anchor values are known. Also, the specific storage was constant of 0.001. The aquifer was evaluated as steady and transient state. Three periods of 3, 10, and 15 days were defined in the transient state. A well was placed at center of the domain, pumping only in the second period. The boundary conditions in all test cases were: constant heads 105m and 100m at south and north respectively, and no lateral flow, generating a hydraulic gradient of 1.25%.

2.6.1 Case Studies: Strategy

The general strategy for this example is as outlined in Section 2.4.1 and starts with choosing the target variables in each test case (Table 2-1). The structural parameters are not assumed and will be random variables in the inversion process. The prior distributions of all structural parameters were chosen to be uniform distributions in order to be conservative. The Type-A measurements in all test cases were taken from the three locations that form a triangle placed at the center of the domain (Figure 2-7).

Table 2-1: Description of case studies

Test case	Type	Pumping	Target variable	Structural parameters	Priors bounds	Anchors	Type-B measurements
I	Steady	-	Transmissivity	Mean Partial Sill Range	[-5, -1] [0.1, 0.7] [10, 120]	8	4
II	Transient	5 m^3/d	Transmissivity	Mean Partial Sill Range	[-5, -1] [0.1, 0.7] [10, 120]	8	12
III	Steady	-	Transmissivity	Mean Partial Sill Range	[-5, -1] [0.1, 0.7] [10, 80]	0	11
IV	Transient	5 m^3/d	Transmissivity	Mean	[-5,-1]	0	11
			Specific storage	Mean	[0.0005, 0.002]		

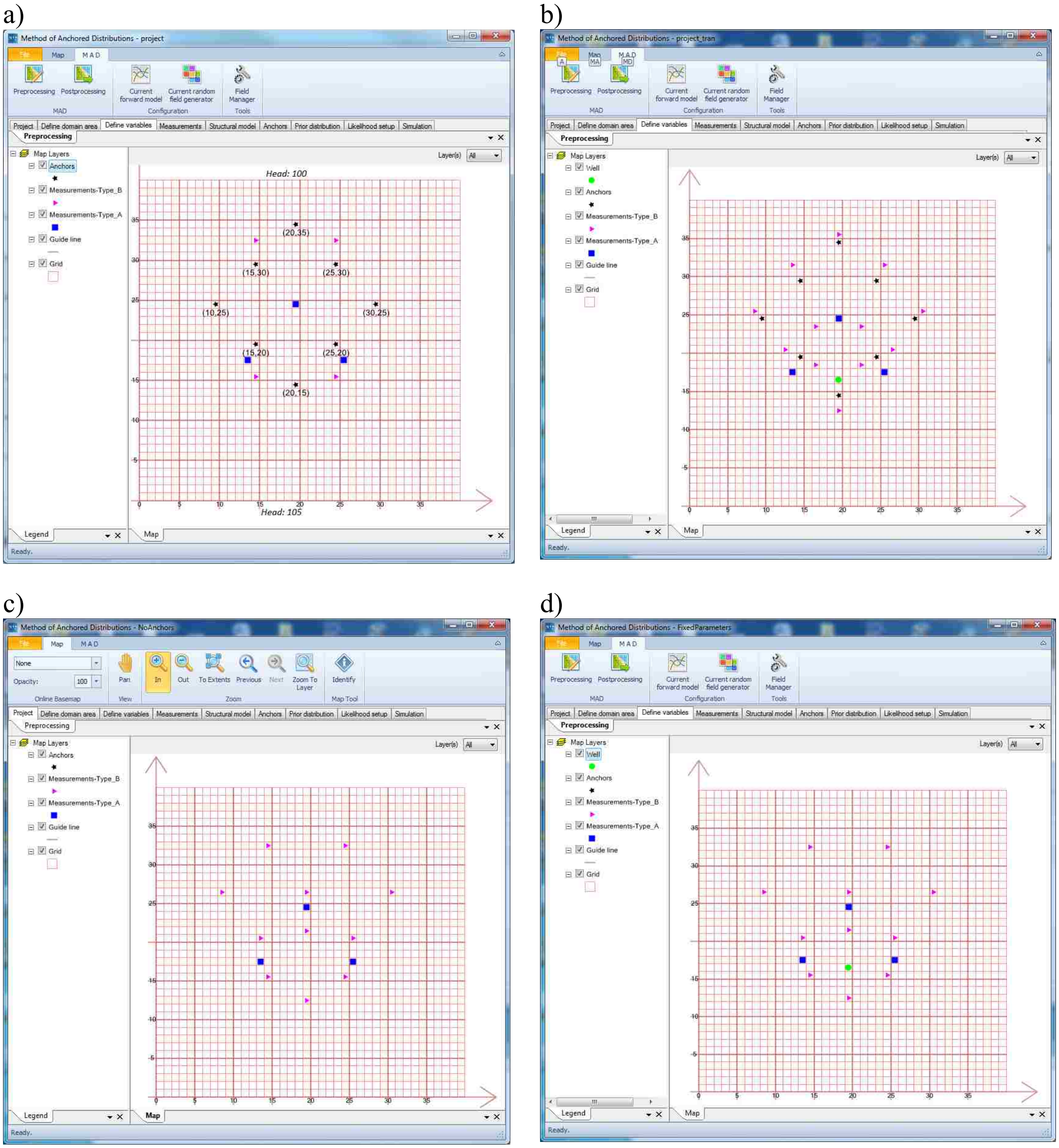

a)
Method of Anchored Distributions - project
Project | Define domain area | Define variables | Measurements | Structural model | Anchors | Prior distribution | Likelihood setup | Simulation
Preprocessing
Map Layers
Anchors
Measurements-Type_B
Measurements-Type_A
Guide line
Grid
Head: 100
Head: 105
(20,35)
(15,30)
(25,30)
(10,25)
(30,25)
(15,20)
(25,20)
(20,15)
Legend
Map
Ready.
b)
Method of Anchored Distributions - project_tran
Well
c)
Method of Anchored Distributions - NoAnchors
Pan
In
Out
To Extents
Previous
Next
Zoom To Layer
Identify
d)
Method of Anchored Distributions - FixedParameters

Figure 2-7: Test Cases. a) Test case I. b) Test case II. c) Test case III. d) Test case IV.

The inversion data type for these test cases is hydraulic head. Head provides ideal Type-B data since head gradients are a function of the hydraulic conductivity field, being this field numerically equals that transmissivity field managed in the test cases, where the thickness is one.

The Type-B measurement locations are along the predominant flow path (south to north) for test case I. The test case II, the Type-B measurements are along of the flow direction generated by the pumping. The test cases III and IV use the same Type-B measurement configuration.

The FM used was MODFLOW-96 (Harbaugh, 1996) since that FM software is well-established in the groundwater community, is easily available which appeals to the MAD# community resource objective, and already has a FMD written for it. The MODFLOW-96 project used for the FM simulations in MAD# is based off of that used for creating the synthetic Type-B data in all test cases. This eliminates the possibility of model uncertainty associated with the FM, a condition which does not reflect reality, but this is an elementary example to show the basic application of MAD#. The one disparate aspect of the FM used in MAD# compared to the baseline FM is that the synthetic transmissivity field is not utilized in the former. The same domain extent and discretization along with boundary and initial conditions were used.

The placement of anchors is in a diamond configuration settled between the Type-B data locations. There are eight anchors, and the distances between them and a Type-B data location varies. In the assessment stage, the influence of the distance between the anchors and Type-B data will be analyzed. Strategic placement of anchors is a subject of current research, and is addressed in Yang, Over & Rubin 2012.

The goal of this example is to compare the difference between the posterior distributions to the respective prior distributions, for both over the parameter space and in relation to the true values of the synthetic baseline case. Three structural parameters and anchors will be assessed this way. The test case I and II will be compared in order to evaluate the effect of more Type-B measurements and transient state in the inversion. The test cases III and IV are linked with the objective to determine the posterior distribution of the structural parameter mean of transmissivity and specific storage. The maximum likelihood value of the structural parameters range and variance of the test case III were used in the test case IV.

All of the information provided in this subsection is entered into the pre-processing module of MAD#, as discussed in Section 2.4.

2.6.2 Case Studies: Implementation

The implementation of the strategy is carried out in the MAD# processing module; the first step of sampling prior distributions was conducted in each test case using MAD#. To obtain samples for the structural parameters, the prior distributions were independently randomly sampled using Math.NET (Math.NET, 2014) library included in MAD#. The number of unique samples for each structural parameter was 100. To create samples for the anchor location, first a unique sample from the structural parameter set was used to generate conditioned fields of the Type-A data. The conditioned fields at the anchor locations were then used to obtain a vector of values that describe normal distribution. In total, with the 100 unique structural parameter samples each having 10 anchor samples, there were 1000 prior samples.

Each of the 1000 prior samples had 300 realizations generated with the GSTAT as the RFG. Each of these realizations were passed to MODFLOW-96 (Harbaugh, 1996) and the FM was executed a total of 300,000. Since the forward simulations are independent, MAD fits into the ‘embarrassingly parallel’ category of parallel algorithms. The high-throughput computing resource HTCondor was utilized to distribute the forward simulations in a computer lab. The total processing time for the simulations is presented in the Table 2-2. HTCondor controls the number of instances in each simulation process. The computer lab has computers with different CPUs from Intel I7 - 8 cores, 8 GBytes of memory to Intel I5 – 4 GBytes of memory.

After the simulations were run and MAD# extracted the simulated Type-B results from the Type-B measurement locations, the likelihood distributions were calculated. The likelihood calculation method utilized was non-parametric kernel density estimation (Hayfield & Racine 2008). The advantage of the non-parametric method is that there is no assumption on the shape of the distribution, but the disadvantage is that more realizations are needed to resolve the shape compared to parametric methods.

Table 2-2: Computational cost of case studies

Test case	Physical hours	CPU hours	Number of computers
I	3.83	230.1	51
II	6.37	282.9	46
III	1.67	88.4	22
IV	1.40	124.8	30

2.6.3 Case Studies: Assessment

Within the MAD assessment stage, if a sample has consistent likelihood over increasing numbers of realizations, and has the same relative likelihood when compared to other samples, then its likelihood is converged. Figure 2-8 shows one convergence plot for each of the four test cases. Seven random samples are chosen, and their likelihoods over a range of realizations are calculated. After 260 realizations, all but one sample out of all test cases holds a consistent likelihood value. This suggests that 300 realizations are enough to proceed to calculating posterior distributions.

By multiplying the converged likelihood distributions by the prior distributions, the posterior distributions are calculated. Figures 2-9 to Figure 2-12 compare posterior distributions to their respective prior distributions and the true values. There are two ways to determine if a posterior is an improvement over a prior. Primarily, the posterior should cover a narrower range of values in the parameter space, which concentrates the probability density. Second, the posterior can increase the probability of the true value, however this is only applicable in synthetic case studies.

Comparing the four test cases, some insight can be gained on the influence of Type-B data on posteriors. Test case II – which has more Type-B measurements in both space and time compared to test case I – had posteriors with narrower ranges and higher probabilities for the true values for all but one anchor (Figure 2-9). The same success can be seen for the partial sill (Figure 2-10).

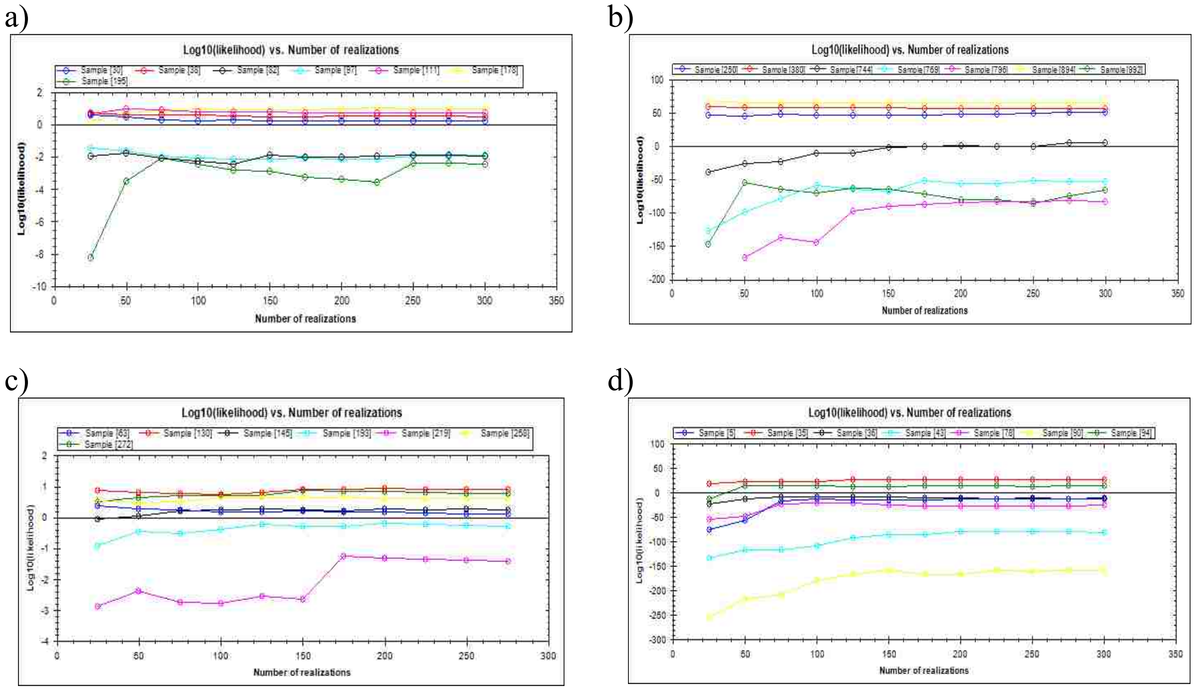


Figure 2-8: Likelihood convergence. a) Case I. b) Case II. c) Case III. d) Case IV.

The mean's posterior is narrower in test case II, but the true value's probability is approximately the same as in the prior, while test case I had a high probability for the true value. For the range structural parameter, the posterior had a narrower range, but the true value does not fall within it. However, in test case III, which has more Type-B variables in space than test case I, the range structural parameter has a posterior with the probability density concentrated closer to the true value. The estimation of the range structural parameter shows a large uncertainty with respect the true value similar result were found by Firmani, Fiori & Bellin 2006.

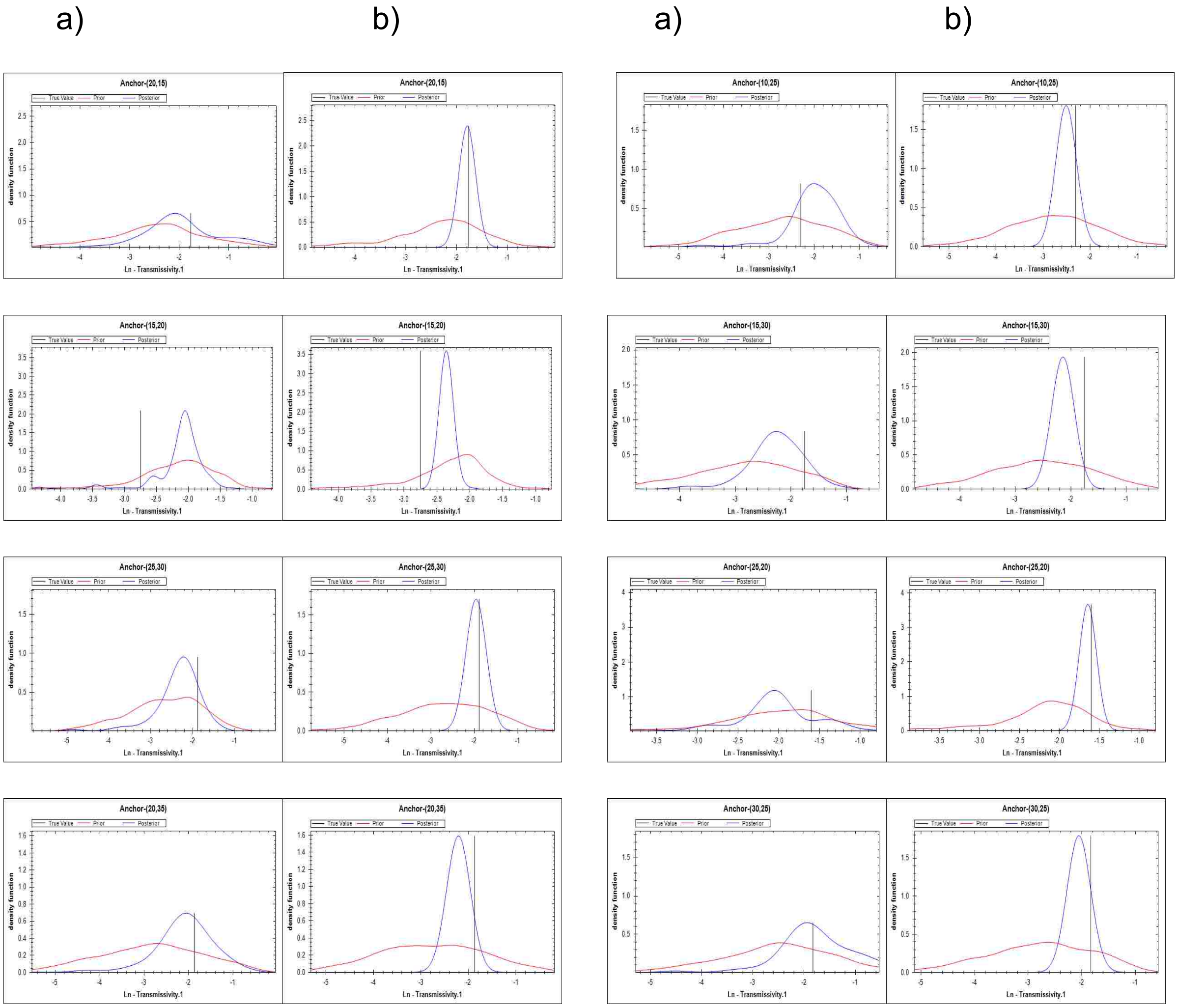


Figure 2-9: Comparison of priors (red) and posteriors (blue). a) Case I. b) Case II.

The maximum likelihood (ML) value of the structural parameters range and partial sill of the test case III (Figure 2-11) were used in the test case IV. The ML value of the range 14.33 m and partial sill 0.17 were considered constant values in the test case IV. The posterior pdf in test case IV (Figure 2-12), where there are transient Type-B measurements but one less than test case II, the mean has the best posterior out of all the test cases and the storage coefficient’s mean also had a successful posterior.

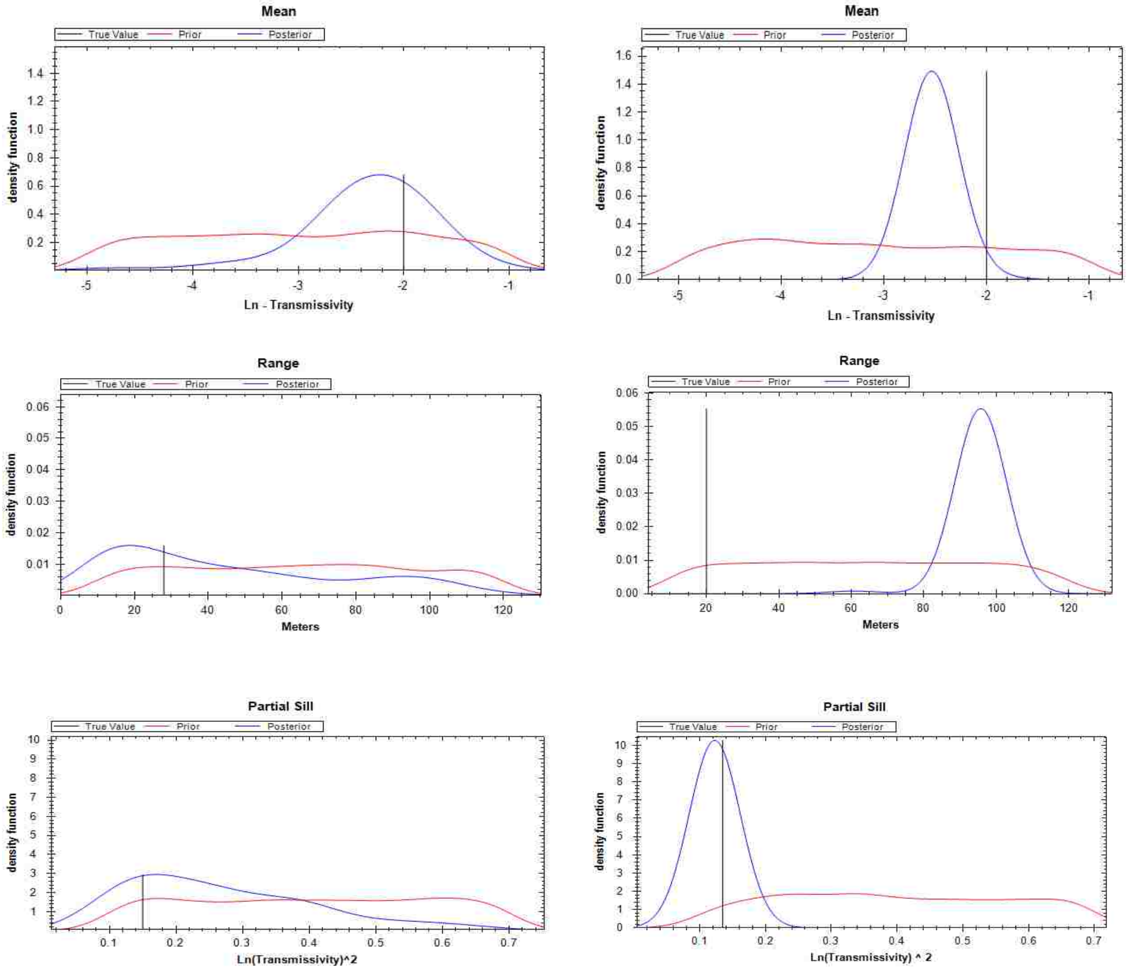


Figure 2-10: Prior and posterior comparison of structural parameters. a) Case I b) Case II.

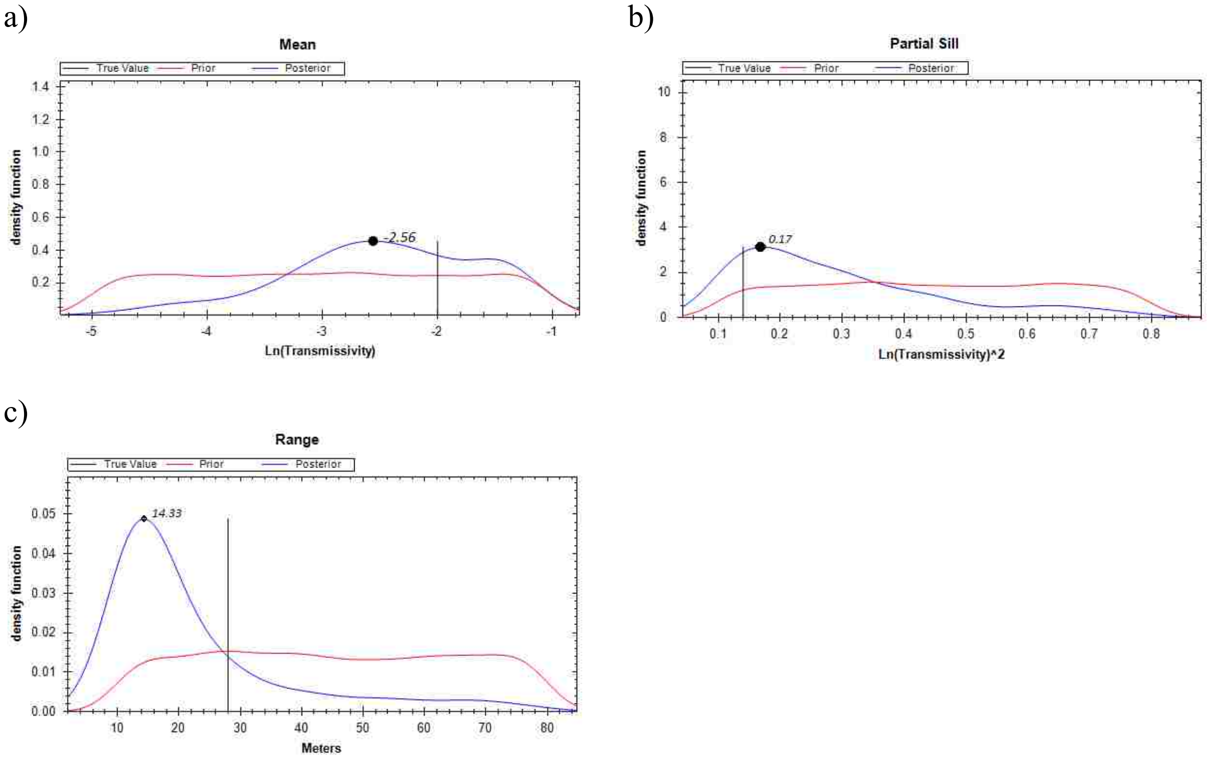


Figure 2-11: Prior and posterior structural parameters case III. a) Mean. b) Sill c) Range.

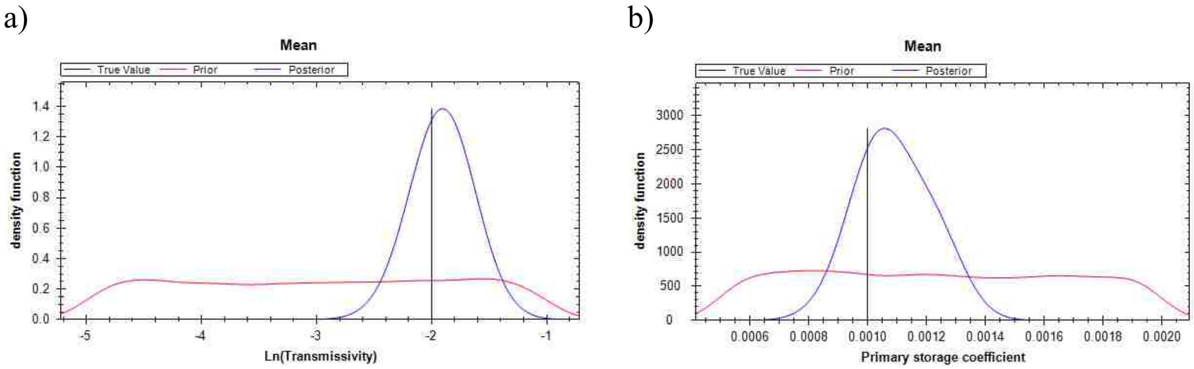


Figure 2-12: Prior and posterior case IV. a) Mean transmissivity. b) Mean storage coefficient.

2.7 Discussion and Conclusions

The MAD approach and the MAD# software differ significantly from other common inverse modeling methods and tools. PEST (Doherty, 1994), UCODE (Poeter & Hill, 1999) and

ITOUGH2 (Finsterle & Zhag, 2011) have as a primary purpose, deterministic parameter estimation through minimizing linear or nonlinear objective functions and determining a single value per parameter. Conversely, MAD# uses a Bayesian method for transferring information from observations to anchors and structural parameters to obtain posterior distributions for each parameter.

Inverse modeling applications generally require configuring and changing input and output files, executing a forward model, and evaluating results. While methods such as Joint Universal Parameter Identification and Evaluation of Reliability (JUPITER) (Banta et al., 2008) have been developed to reduce the complexity of parameter estimation, users are required to alter input files of the FM to fit within the specific parameter estimation framework. MAD# implements a new approach, in which users are not required to create template files. Using a plug and play approach, the user just specifies the FMD for executing the inverse process. The expectation is that this approach will increase the ease of using new FMs and simplify the application of inverse modeling techniques.

The generic configuration of the MAD# framework allows accommodation of different FMs via drivers. This chapter demonstrated how a FM can be linked with the MAD# framework. The generation of realizations in MAD# is also managed via a driver. The MAD# user can select the appropriate generator for each project. The generated information in the inversion process is stored in simple SQLite databases, which can be accessed by generic SQLite manager applications and integrated with the GIS environment. The inversion process required a long processing time, which suggests that MAD# team should work in mechanism to execute in High Performance Computing frameworks.

In the case studies presented in this chapter, the SRF Ln(Transmissivity) is characterized using the indirect measurements of pressure head via solving head equation implemented in MODFLOW. The anchors defined in the SRF domain show the posterior distribution of Ln(Transmissivity) and the uncertainty at the anchor locations. The posterior pdfs of structural parameters and anchors characterized the global and Ln(Transmissivity) field, respectively. The posterior pdfs obtained with more Type-B measurements produced values closer to true values. MAD# provides a user interface that allows comparison of multiple scenarios.

In summary, the MAD# software framework was designed, developed, and described here to aid scientists, modelers, and students in the application of inverse modeling and SRF characterization. MAD# specifically addresses the five criteria proposed by (Carrera et al., 2005) and represents a potentially valuable step forward for inverse modeling in general and the MAD method specifically.

3 CHARACTERIZING SPATIAL RANDOM FIELDS USING MAD# AND THE HIGH THROUGHPUT COMPUTING SOFTWARE – HTCONDOR

Co-authors: Daniel P. Ames, Heather Frystacky, Yoram Rubin

Conference paper published in the proceedings of the 7th Intl. Congress on Environmental Modelling and Software, San Diego, CA, USA.

3.1 **Overview**

The high cost of collecting measurements for characterizing SRFs is an incentive to evaluate more accurate techniques that integrate additional resources such as models, secondary information, and related variables. Existing information can contribute to the reduction of the uncertainty in the SRFs. Using inverse modeling techniques; it is possible to extract information from simulation or other "forward" models to improve parameter estimation. Although both deterministic and stochastic inverse modeling techniques have been introduced, stochastic methods can be more appropriate for SRFs since they do not require following strict Gaussian assumptions usually required in deterministic models. The Method of Anchored Distributions (MAD) is a stochastic inverse modeling method focused on characterizing the uncertainty of SRFs (Rubin et al., 2010).

MAD# uses forward models that relate a variable of interest to measurable variables as a means for characterizing the uncertainty at specific locations called "anchors" – statistical devices located in the SRF domain. The inversion process requires evaluating conditional SRFs in a forward model multiple times. The number of evaluations is proportional to the number of observations included in the process. The strategy to localize anchors in the domain depends on characteristics of phenomena and measurements (Yang, Over, & Rubin, 2012). The most time-consuming step in the execution of MAD# is the evaluation of each conditional SRF in the forward model. Although MAD# can run the process in multiple cores, large projects could require multiple computers. MAD# addresses this issue submitting the simulation to a high throughput computing (HTC) environment such as the HTCondor system, which is a software application that shares the unused computing resources of a network. This chapter presents the use of MAD# together with HTCondor to improve simulation time depending on the number of nodes in the HTCondor environment. We also explore how the system works when the network is totally available (weekend) and the network is used in a normal day (weekday).

3.2 **Inverse modeling in High Performance Computing**

High Performance Computing (HPC) environments have been used extensively in recent years to solve inverse problems in different scientific areas (Goncharsky & Romanov, 2013) (Bertshinger, 2001). The management of large meshes in simulation requires large memory and CPU capacity. The grid dimension in MAD# projects depends on the size and complexity of the forward model and its required inputs. The number of simulations also depends on the number of observations used in the inversion process. MAD# does not improve the performance of the forward simulation model code itself. Rather, MAD# runs multiple instances of the forward model in parallel using a Monte Carlo simulation approach. Monte Carlo simulations are easily parallelized in stochastic models (Barry, 1990). In groundwater modeling, SRFs represent the structure and distribution of geological features, which are used for representing variability at different resolutions. The high-resolution representation of SRFs requires more computational resources (Tompson et al., 1998).

Stochastic inverse modeling applications use large computational resources; hence HPC is a logical choice for running these applications. However, the adoption of inverse modeling techniques depends on the accessibility of HPC resources. HPC access is more readily available through technologies such as HTCondor, which allows users access to a large amount of unused computational cycles in a computing infrastructure. Monte Carlo applications have been successfully tested in this platform (Zhou & Mascagni, 2000), which indicates that it is an appropriate technology for testing inverse modeling methods.

3.3 MAD# and HTCondor

The University of Wisconsin-Madison developed a software system called HTCondor that administers and accesses the unused computational resources from a computing infrastructure. HTCondor provides several mechanisms to identify idle workstations candidates to execute jobs. A flocking system allows deployment of jobs in other groups of computers (pools). When a workstation is working on a job, this job can be interrupted by user activity. Then, HTCondor transfers this job to another node in the network. The scheduling process is executed according to the characteristics requested by users like available CPU cycles, memory and operating system. HTCondor works in a Master-Worker schema, where the master node controls the worker nodes and the worker nodes execute the jobs. Each node can also be configured to be able submit jobs in the network.

The execution environment of HTCondor allows users to submit jobs in several so-called “universes.” The universes provide mechanisms to control the communication between the worker nodes. The distributed approach used by MAD# is the universe "vanilla”; although this universe does not provide information on how the jobs are been executed. The output of each job can be transferred using a mechanism for transferring files enabled in this universe. The vanilla environment allows one to submit software packages that can be executed in each node. The MAD# core executable is transferred using this method.

MAD# stores the forward model output per sample; this is a convenient for the parallelization of the Monte Carlo process. Figure 3-1 shows the process that will be executed in each node of HTCondor. It is clear that is necessary to transfers the random field generator application and the forward model to each node to complete the simulation per sample.

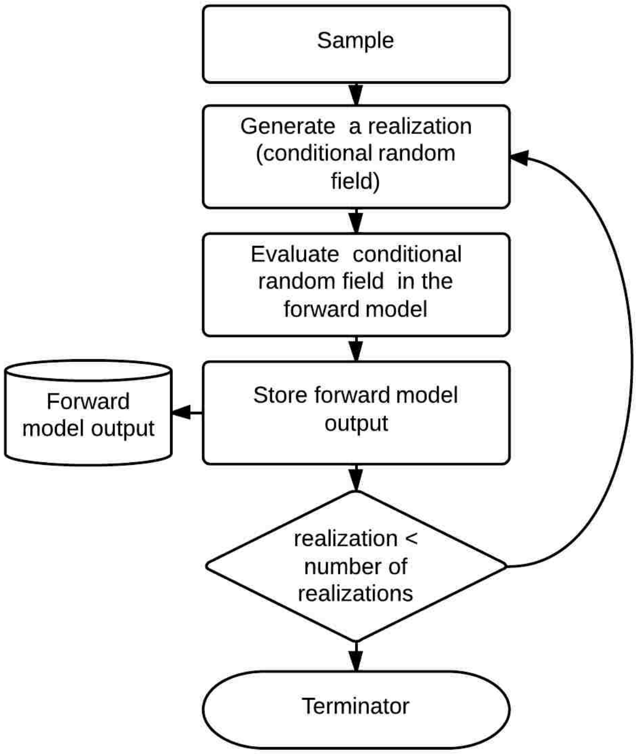
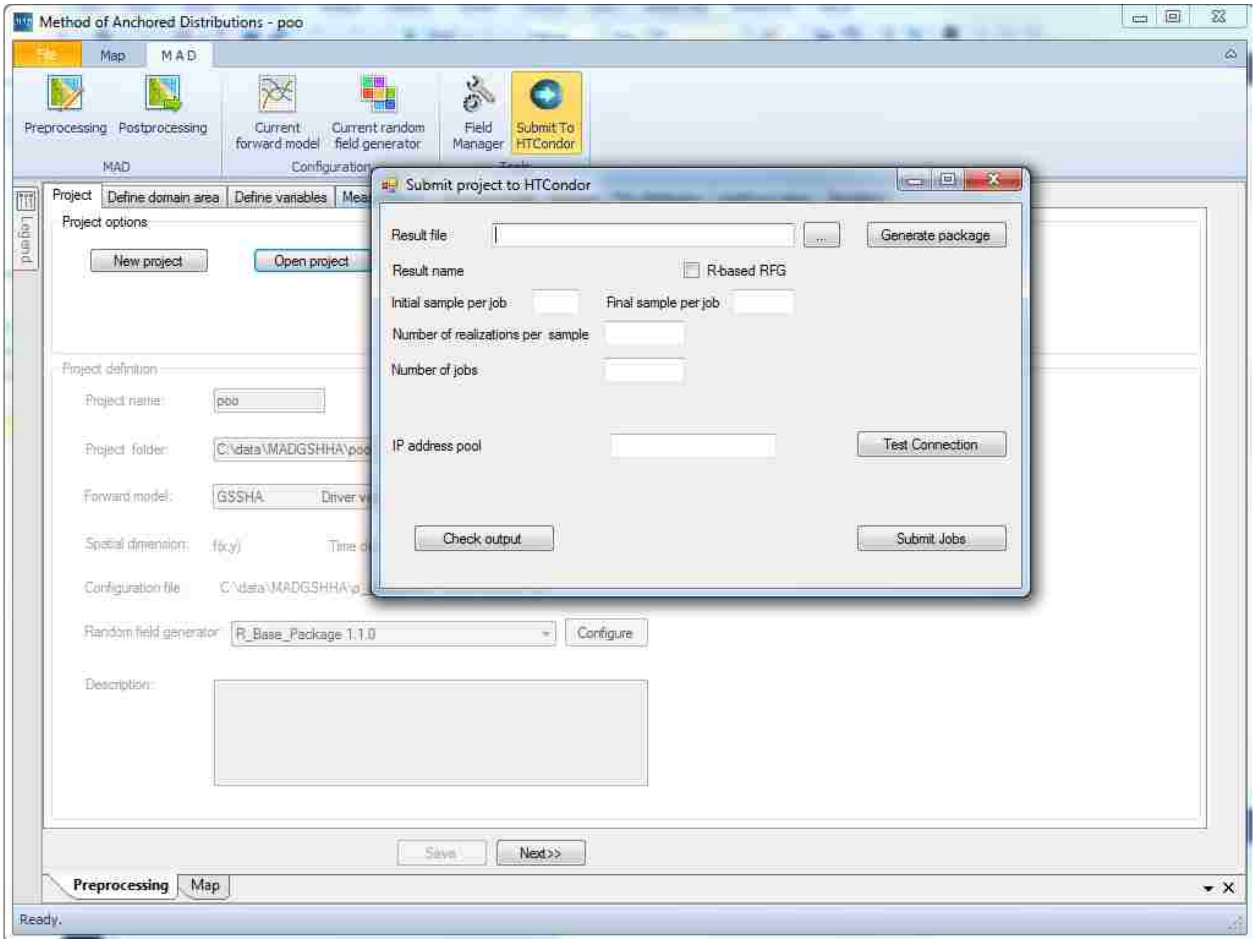

Sample
Generate a realization (conditional random field)
Evaluate conditional random field in the forward model
Forward model output
Store forward model output
realization < number of realizations
Terminator

Figure 3-1: Internal loops in MAD#

MAD# generates a compressed package with MAD# core, the forward model and project data. The MAD# allows users to send the packages directly to HTCondor through a simple user interface.



Method of Anchored Distributions - poo

Submit project to HTCondor

Project name: poo

Project folder: C:\data\MADGSHHA\poo

Forward model: GSSHA

Spatial dimension: f(x,y)

Configuration file: C:\data\MADGSHHA\p

Random field generator: R_Base_Package 1.1.0

Figure 3-2: User interface of MAD# to submit job in HTCondor

3.4 Performance Evaluation HTCondor Deploying MAD# Jobs

The performance of HTCondor working with MAD# was evaluated using three case studies with different grid densities (Figure 3-3a, 3-3b and 3-3b). The common objective of each of these MAD# case studies is to characterize the Log-Transmissivity at the anchor locations.

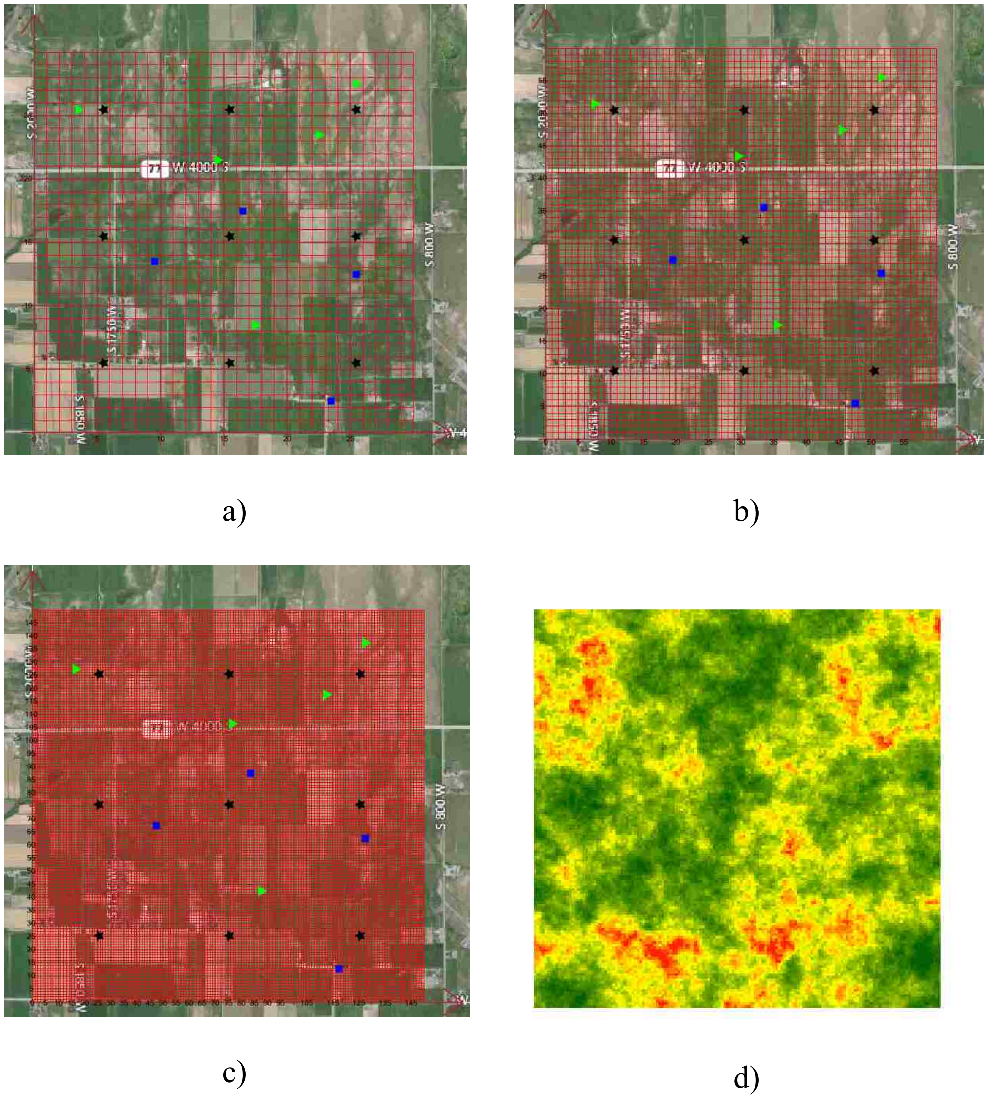
For each case study, the Type-A target variable is Log-Transmissivity and Type-B data are head pressure. The forward model MODFLOW-96 (Harbaugh, 1996) is used to relate the Type-A measurements to the Type-B variable. Each MODFLOW project was configured as steady-state with two boundary conditions at 100m and 90m north and south respectively. GSTAT (Pebesma & Wesseling, 1998) was used to generate conditional random fields. The true field in each synthetic dataset is the same as is shown in Figure 3-3d. The true field was resampled to fit the density grid of each project. The number of anchors and measurements are the same in each case. Additionally, the test cases are evaluated in two scenarios where the status of the HTCondor network is different:

- Weekend Scenario: Computers are fully available.
- Weekday Scenario: Computer availability can change at any moment.

The global variability of the SRF is defined by an exponential model with the following structural parameters: range, partial sill and mean with values of 300m, 0.15, and -4 respectively. The MAD theory allows one to determine the posterior distribution of an anchor when the structural parameters are known. Equation 3-1 shows the anchor posterior distribution $\mathrm{p}(\vartheta|\mathrm{z_a},\mathrm{z_b})$ conditioned to Type-A and Type-B data which is a simplification of the Equation 2-1. The prior distribution of anchors is just conditioned to Type-A data $\mathrm{p}(\vartheta|\mathrm{z_a})$.

$$\boldsymbol{p(\vartheta|z_a, z_b)\ \propto p(\vartheta|z_a)\, p(z_b|\,\vartheta, z_a)} \tag{3-1}$$

The number of samples evaluated is 300 and the number of realizations per sample is 30, which means that MODFLOW will run 9000 times. It is not necessary to check the convergence or specific quality of the solution because our objective is determining the time-consumption of the simulation process using the HTCondor infrastructure. Our HTCondor network was divided between two pools of Windows operating system computers with 72 and 243 nodes, respectively.



a)

b)

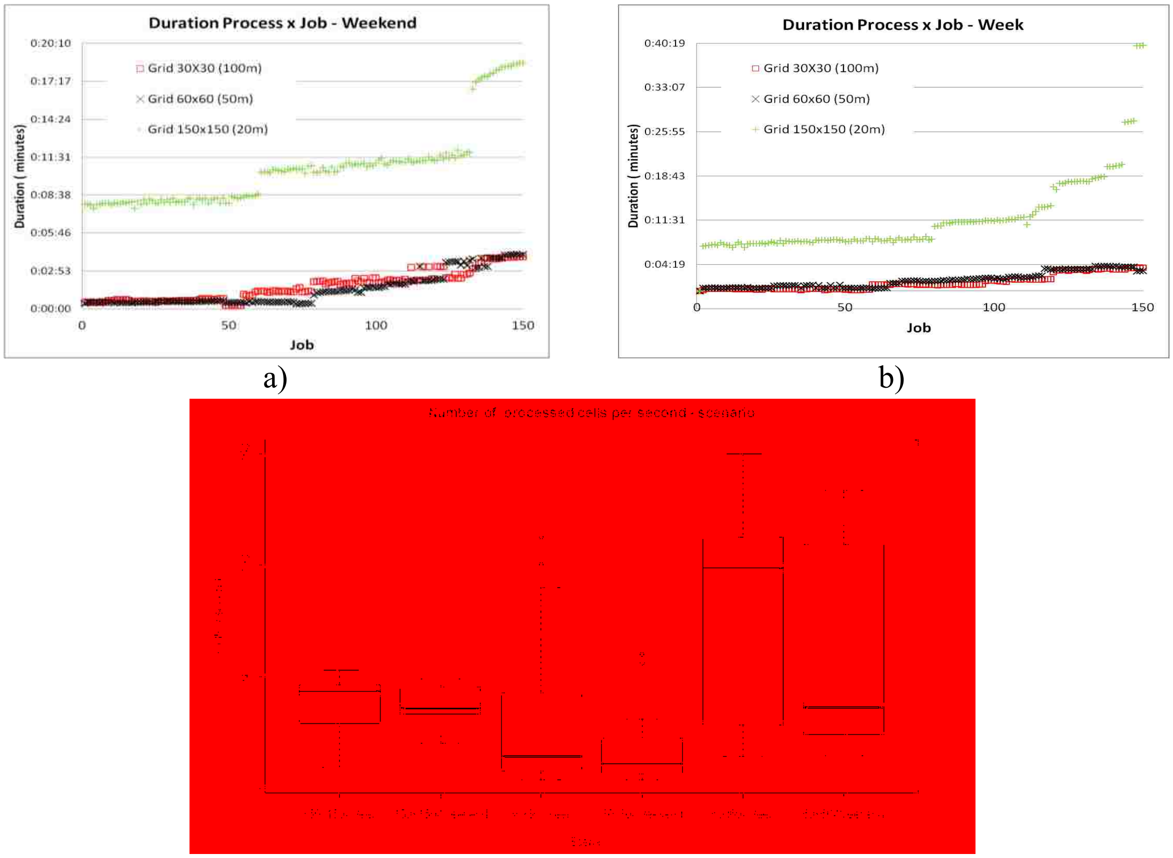
c)

d)

Figure 3-3: Project grids: a) 100m, b) 50m, c) 20m. Type-B (green), Type-A (blue), Anchors (black). d) SRF Transmissivity (True field)

In each test case, 150 jobs are submitted with two samples per node. The HTCondor scheduling process first assigns the jobs in the pool with 72 nodes. When there are not available nodes, the second pool is used through the HTCondor flocking process. Figure 3-4a and Figure

3-4b shows the duration time of all jobs in the three test cases per scenario. Using the indicator number of processed cells per second (PCs/S), the performance of the execution of HTCondor among test cases is evaluated (Figure 3-4c). The variability of the indicator PCs/S in the density grid 60x60 is higher than other cases. The 150x150 grid shows the lowest variability in the indicator PCs/S. This can be explained by the longer duration of this test case in comparison with the other test cases. Although the amount of information transferred for each node is the same in all test cases, the latency can contribute in the variability of the test cases with small sizes (Cai et al., 2014).



a)

b)

c)

Figure 3-4: Duration jobs in HTCondor. a) Weekend b) Weekday c) Indicator processed cells/second.

The total duration of the simulations for the 30x30 grid and 60x60 grid in both scenarios were similar (Table 3-1). The 150x150 grid test case in the weekday scenario had the maximum duration, where some jobs were three times higher than other jobs--noting that the last eight jobs were re-assigned to other nodes (Figure 3-4b). This was produced by a suspended mechanism of HTCondor, which suspends jobs when a user takes control of a computer.

Table 3-1: Duration simulation

Evaluation	Total duration one computer (hours)	Total duration (min)	Ratio HTCondor/one computer	Mean(min)	Variance	p-value<0.05
30x30 weekend	0.9	3.97	13.61	1.70	1.18	
30x30 week	0.9	3.83	14.09	1.42	1.52	0.04*
60x60 weekend	1	4.15	14.46	1.46	1.48	
60x60 week	1	4.10	14.63	1.80	1.52	0.02*
150x150 weekend	12.9	18.68	41.43	10.71	9.22	
150x150 week	12.9	40.05	19.33	11.10	14.78	0.02*

The duration mean of the jobs in each test case is compared in both scenarios through a significance test, where the null hypothesis considers the duration mean is equal for jobs executed in both scenarios. Table 3-1 shows the duration mean is different rejecting the null hypothesis in all cases with a standard significance level of 0.05, which indicates that the execution of the MAD# projects can be affected by the state of the HTCondor network. We observed that the PCs/S indicators were lower in the test cases of the weekend scenario but with lower variability.

We compared the total duration of the simulation in the HTCondor network with the estimated total simulation time of a single computer with 8 cores. Table 1 shows the duration statistics and the ratio between total time in HTCondor and approximate duration in one computer. When the grid density is larger the ratio increases. The 150x150 grid computation completed 41 times faster than a single computer in the weekend scenario. Conversely, the total duration in a weekday was 19 times faster. The performance reduction was produced because some computers were claimed by users; the reassignment of processes increases the time twice in the last eight nodes as is shown in Figure 3-4b.

3.5 **Discussion and Conclusions**

HTCondor showed a considerable improvement in the total duration of the Monte Carlo process in MAD#. The high variability in the duration time in the small cases studies (30x30 grid and 60x60 grid) is produced by the network latency. The effect of the latency can be reduced by increasing the simulation time per node, which can be achieved by processing more MAD# samples per node. The HTCondor "vanilla" environment was used to submit MAD# jobs, allowing MAD# core to run in each node and to transfer the simulation output files to the master node. Although, the transfer file system works adequately, it is necessary to evaluate the Hadoop File System (HDFS) to reduce the high variability in the response of small MAD# projects.

The availability of idle computers in the HTCondor network changes the duration mean and total duration of the execution of MAD#. Although, the main characteristic of HTCondor is the management of idle computers, the reassignment of jobs decreases the performance of the simulation. We recommend that large projects use a time frame with greater availability.

4 EVALUATION OF THE LEVENBERG-MARQUARDT ALGORITHM SOLUTION SPACE USING A METHOD OF ANCHORED DISTRIBUTIONS IN A GROUNDWATER PROBLEM

Co-authors: Daniel P. Ames, Yoram Rubin

Planned submission for publication to Computers & Geosciences (12/2014)

4.1 **Introduction**

The characterization of the parameters of Spatial Random Fields (SRFs) is usually done through Geostatistical techniques and inverse modeling procedures. Using observations of an SRF and indirect measurements related through a model, the inverse modeling methodology can be applied to estimate the parameters. In Geostatistics, SRFs are usually defined by covariance functions with unknown parameters. SRFs represent variables in mathematical models such as the hydraulic conductivity field in groundwater models. The hydraulic conductivity field has a very heterogeneous distribution. This variable can be characterized by SRFs. Inverse modeling techniques can be used to characterize these parameters through observations, for example, pressure head in ground water models, and the relationship with the target variable through a forward model (e.g., MODFLOW). The number of parameters in SRF depends on its discretization. Usually, the number of parameters is larger than the number of observations, which produce an ill-posed problem with non-unique solution.

The relationship between the target variable, hydraulic conductivity, and the pressure head is established by a partial differential equation that uses the law of mass balance and Darcy's

Law. This equation generates a non-linear relationship between both variables that can be solved by non-linear algorithms. Stochastic and deterministic inverse modeling methodologies have been used to estimate the SRF parameters (Doherty, 2003; Nowak & Cirpka, 2004). Stochastic inverse modeling techniques quantify parameter uncertainty through posterior distributions. These methods are based on the evaluation of SRF realizations conditioned to available observations and prior information of the parameters. The computational cost of this method is a disadvantage in comparison with deterministic methods. Deterministic methods such as least-square methods for fitting the parameters generally use the Gauss-Newton algorithm which works efficiently with linear problems. Variations of the Gauss-Newton algorithm, such as, the Levenberg-Marquardt (LM) algorithm provide a solution for non-linear problems. The LM algorithm is an iterative method that uses a damping term to minimize the objective function. Hybrid methods combine Bayesian and LM algorithms (Nowak & Cirpka, 2004).

The adaptation of the LM algorithm with different regularization methodologies has been studied by several authors (Doherty, 2003; Alcolea, Carrera, & Medina, 2006; Fienen, Muffels, & Hunt, 2009). This regularization approach reduces the parameter space to covert the ill-posed problem in well-posed problem. The sensitivity matrix generated during the optimization process defines the parameters with low impact in the solution. The group of low sensitivity parameters can be associated with large sensitivity parameters through a mathematical function. The new group parameters reduce the parameter space. This mathematical approach could include bias in the estimation process. The basic regularization schema is an ill-posed problem with a non-unique solution.

We implemented a modified LM algorithm for SRF parameter estimation that customizes the step size derivative of each parameter. The algorithm also fits the semivariogram of the spatial field. As a deterministic method the algorithm just uses interpolated fields, which are conditioned by observations of the target variable and auxiliary points distributed in the spatial domain. This approach is similar to the Pilot Point method but without regularizing the problem, maintaining the ill-posed characteristics of the problem and generating multiple local minima. In general, the characterization of an SRF is a non-linear problem that can produce multiple minima (Nowak & Cirpka, 2004). A multiple starting point method is applied in our modified LM algorithm using random samples for each parameter. The solution space is used to generate a set of prior distributions that are evaluated in the Bayesian inverse modeling framework MAD for characterizing SRF parameters. Although MAD is not an optimization algorithm itself, it estimates the posterior distributions of the parameters reducing the associated uncertainty.

Our modified LM algorithm is implemented in the open source inverse modeling framework MAD# (http://mad.codeplex.com/). The multiple starting point approach requires executing the optimization process *N* times (*N* number of samples) generating a new prior set which is evaluated in a Bayesian method. To support the computational challenge of both methods, a distributed computing system is coupled with the MAD# framework. The high throughput computing platform (HTCondor) is used to distribute the simulations over a computing network. The modified LM algorithm and the MAD technique are executed by HTCondor. Section 4.1.3 explains the link between MAD# and HTCondor.

A synthetic groundwater project is prepared having as a target variable the SRF Ln(Transmissivity), The project is steady state with constant head as a boundary condition. The SRF is modeled using a geostatistical approach where we have global and local parameters. We generate five test cases to evaluate the solution space of parameters generated by LM algorithms and use MAD technique to assess the parameter through posterior distributions.

4.1.1 Spatial Random Fields in Optimization Algorithms

An SRF can represent a spatial variable randomly distributed, such as hydraulic conductivity in hydrology. These variables can be described by Gaussian and non-Gaussian processes (Meerschaert et at., 2013). In our case, it was assumed that the SRFs are a stationary Gaussian process, modeled using a typical geostatistical model:

$$\boldsymbol{Y}(\boldsymbol{s}) = \boldsymbol{\mu}(\boldsymbol{s}) + \boldsymbol{\varepsilon}(\boldsymbol{s}) \tag{4-1}$$

where $Y(s)$ represents the variable, $\mu(s)$ is the large-scale variation and $\varepsilon(s)$ is the small-scale variation associated with a covariance function. This covariance function was modeled through theoretical models with the following parameters, variance σ^2 and range ϕ. When the variability is attributable to measurement errors, a nugget parameter ε_n is used.

The variable Y can be represented by $\bar{Y}(s)$ field, which can be characterized by global and local parameters (Rubin et at., 2010). Geostatistical models represent the global variability in SRFs. Local variations of SRFs were estimated using artificial points in the domain. Methods such as Pilot Points (Doherty, 2003) and MAD (Rubin et at., 2010) use these points to determine local variations. Using a MAD nomenclature, conditional realizations $\tilde{y}$ and interpolated fields $\bar{y}$ can represent an approximation of the $\bar{Y}(s)$, where both $\tilde{y}$ and $\hat{y}$ fields are conditioned to the available information z_a and the artificial points x_a, and represented by a function $f(\mu, \sigma^2, \phi, z_a, x_a)$.

The realizations $\tilde{y}$ and interpolated fields $\hat{y}$ describe the variability of the field, but the interpolated field depends on the size of z_a to obtain a better approximation of the variability of the field. As shown in Figure 4-1 the fields are conditioned to the same amount of points and both are using the same structural parameters. The interpolated fields cannot capture the heterogeneity of the base field with few points.

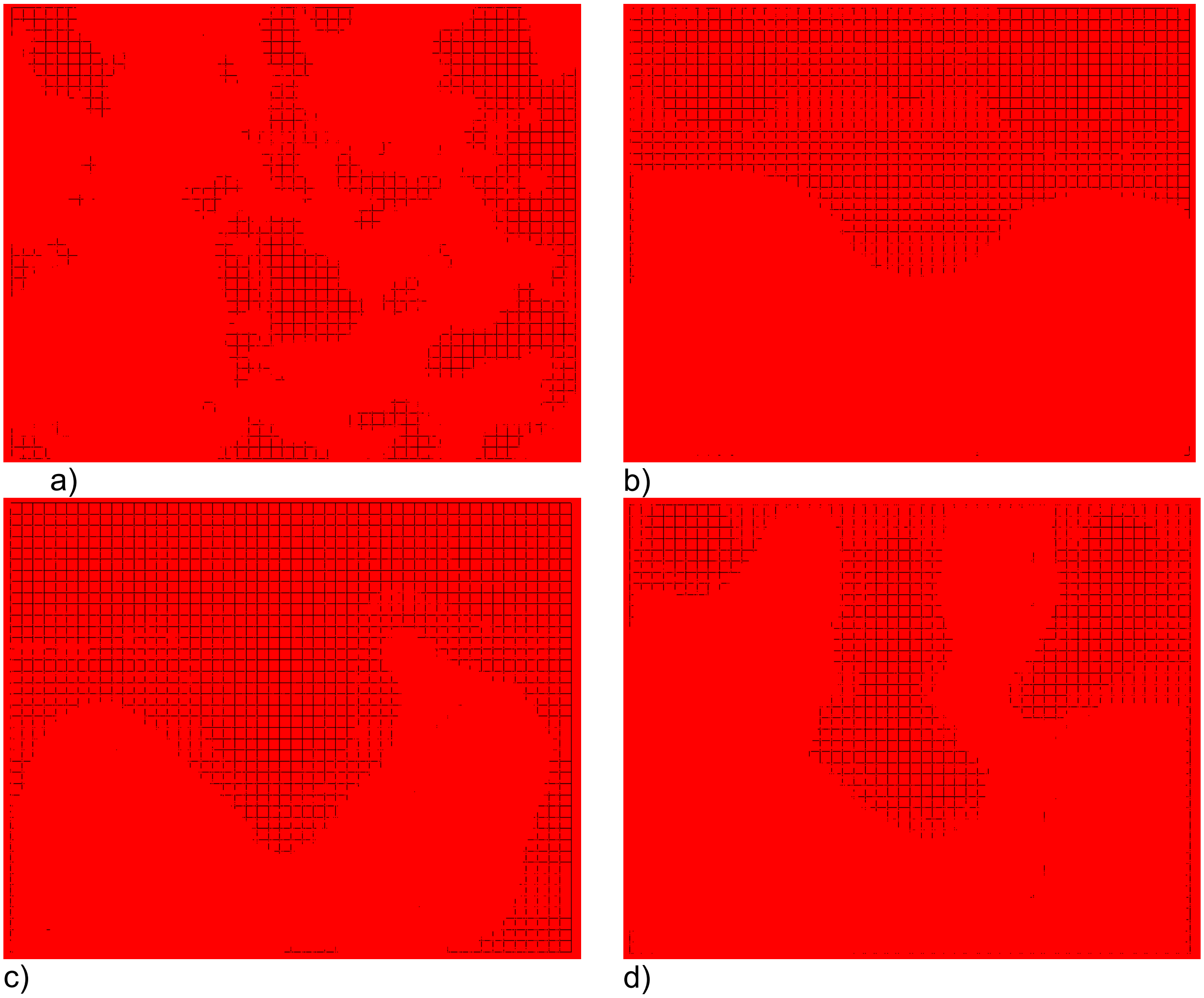


Figure 4-1: Effect of the number of points to characterize an SRF using interpolated fields. a) Realization conditioned to 9 points, b) interpolated field using 9 points, c) interpolated field using 15 points, and d) Interpolated field using 39 points

4.1.2 Inversion Data

The inverse modeling technique determines the structural parameters of an SRF through inversion data. These data are observations z_b related to target variable through a forward model. The forward model, $M,$ generates modeled observations, $\dot{z}_b,$ used in the inversion. Stochastic and deterministic inverse methods estimate parameters by finding the differences between $\dot{z}_b$ and $z_b.$

In stochastic methods, using conditional realizations, $\tilde{y}$, the SRF heterogeneity is represented. These conditional realizations are evaluated in forward model, *M,* to obtain $\dot{z}_b$, where the output estimation error is ε_b. The following equation represents the relationship:

$$\dot{z}_{\boldsymbol{b}} = \boldsymbol{M}(\tilde{y}) + \boldsymbol{\varepsilon_b} \tag{4-2}$$

The modeled observations, $\dot{z}_b$, are generated by an interpolated field $\hat{y}$ using a deterministic method, as is shown in the following equation:

$$\dot{z}_{\boldsymbol{b}} = \boldsymbol{M}(\hat{y}) + \boldsymbol{\varepsilon_b} \tag{4-3}$$

Although $\hat{y}$ is not a random field it can describe a smooth version of the SRF, which is used in the inverse process. The $\dot{z}_b$ vector can contribute to the parameter estimation depending on its approximation to true observations z_b.

4.1.3 Levenberg-Marquardt Algorithm

The LM algorithm is an optimization method that minimizes the objective function in non-linear problems (Lourakis & Argyros, 2005), and it is used as an inverse modeling technique for the model calibration (Hanke, 1997). In groundwater problems, modified versions of the LM algorithm have been evaluated using a regularization approach (Finsterle & Kowalsky, 2011; Hanke, 1997; Nowak & Cirpka, 2004). The LM algorithm uses a starting point to find an optimal solution, but in some cases multiple starting points can result in multiple solutions. In those cases a regularization term is introduced in the algorithm to well-pose the problem (Doherty, 2003). The LM algorithm combines steepest descent and least square method (*Gauss-Newton*) to identify the minimum of a function.

In terms of the SRF parameter estimation, the LM algorithm defines an objective function based on true observations, z_b and model output. The objective function is defined in the following equation:

$$\boldsymbol{S}(\boldsymbol{\beta}) \approx \sum [\boldsymbol{z_b} - \boldsymbol{M}(\hat{y}(\boldsymbol{\beta}))]^2 \qquad (4\text{-}4)$$

The parameter vector β represents the global and local parameters of an SRF, e.g., $\beta = f(\mu, \sigma^2, \phi, \varepsilon_n, x_{ai})$, where $S(\beta)$ is the sum of the squares of the deviations. To solve this system is necessary to initialize the parameter vector β with a set of guessed values. The new β vector is calculated using an iterative procedure adding a δ vector to β. A linear approximation of the function $M(\hat{y}(\beta))$, with respect of the δ vector, is used to estimate the variation of the function as is shown in the following equation:

$$\boldsymbol{M}(\hat{y}(\boldsymbol{\beta} + \boldsymbol{\delta})) \approx \boldsymbol{M}(\hat{y}(\boldsymbol{\beta})) + \boldsymbol{J\delta} \qquad (4\text{-}5)$$

The approximation, above, requires obtaining the Jacobian $J = \partial M(\hat{y}(\beta))/\partial \beta$. To calculate the δ vector, the $S(\beta)$ deviations are minimized with respect to δ using the following approximation:

$$\boldsymbol{S}(\boldsymbol{\beta} + \boldsymbol{\delta}) \approx \sum [\boldsymbol{z_b} - \boldsymbol{M}(\hat{y}(\boldsymbol{\beta})) - \boldsymbol{J\delta}]^2 \approx \|\bar{\boldsymbol{\varepsilon}} - \boldsymbol{J\delta}\| \,;\, \bar{\boldsymbol{\varepsilon}} = \boldsymbol{z_b} - \boldsymbol{M}(\hat{y}(\boldsymbol{\beta})) \qquad (4\text{-}6)$$

The deviations with respect to δ are approximated to zero obtaining the *Gauss-Newton* step of the δ vector, which is shown in the following equation:

$$\boldsymbol{J^T J\delta} = \boldsymbol{J^T}\bar{\boldsymbol{\varepsilon}} \qquad (4\text{-}7)$$

The LM algorithm solves the problem adding a damping term λ in the equation (4-7), this damping term constrains the solution in the iteration process. This term is reduced or increased to achieve a reduction in the error. An identity matrix I is used to introduce the damping term in the Gauss-Newton equation, LM final equation is defined by:

$$(\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J} + \boldsymbol{\lambda I})\boldsymbol{\delta} = \boldsymbol{J}^{\boldsymbol{T}}\overline{\boldsymbol{\varepsilon}} \tag{4-8}$$

The damping term is used as a regularization term for solving ill-posed problems (Lourakis & Argyros, 2005). It is common to find problems where the solution is not global. The LM algorithm depends on estimation of the Jacobian and the δ vector, which is defined as a gradient of step size. Large step sizes can produce unstable solutions and small step sizes increase the processing time. A gain ratio, ρ, controls the approximation to the optimal solution. Large values of ρ indicate that the solution is close (Madsen, Nielsen & Tingleff, 2004). The iteration of the LM algorithm is controlled by ρ, and the Jacobian is updated only when $\rho > 0$. The parameter continuity is an assumption in the LM algorithms that could differ with requirements of the model. In the case of SRFs, the structural parameter range, ϕ, must be a positive value. The step size in the LM algorithm can jump out of bounds. Another critical point in the LM algorithm is the stop signal, which can be managed by adjusting the threshold value that controls the step size, or damping term value. The pseudo code of this algorithm is shown in the appendix B.4.

4.1.4 Implementing an Optimization Algorithm in MAD#

The main inverse modeling technique implemented in MAD# is MAD, but the extensibility architecture of MAD# facilitates to add other inverse modeling techniques. In our case, it has been implemented the LM algorithm in the MAD# core which allow users to execute both inverse modeling techniques. The LM algorithm also use the FMD and RFGD to optimize the SRF parameters (Figure 4-2)

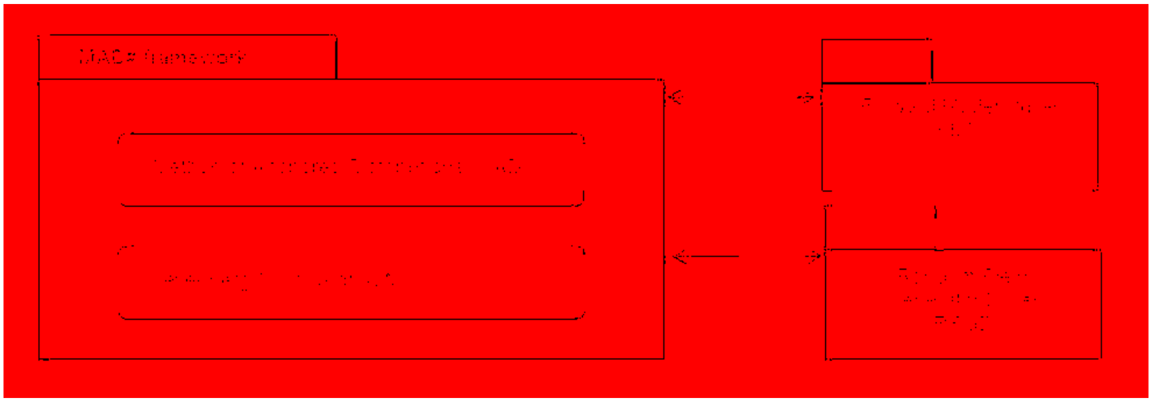


Figure 4-2: MAD# architecture

4.2 **Methods**

4.2.1 Extending the Objective Function in a Modified Levenberg-Marquardt Algorithm

The LM algorithm optimizes the parameters minimizing the objective function equation (4-4), assuming that each parameter is independent, which allows the LM algorithm to freely determine the δ vector. This assumption affects the local parameters, x_a, which depend on the geostatistical model and the global parameters (e.g., σ^2, ϕ and ε_n). To solve this issue we fitted the semivarigram with the geostatistical model and included it in the objective function as is shown in the following equation:

$$\tilde{\boldsymbol{f}}(\boldsymbol{\beta}) = \begin{cases} \boldsymbol{z_b} - \boldsymbol{M}\big(\hat{\mathrm{y}}(\boldsymbol{\beta})\big) \\ \boldsymbol{\gamma_a}(\boldsymbol{h}) - \ \boldsymbol{\gamma}(\boldsymbol{h}, \boldsymbol{\beta}) \end{cases} \tag{4-9}$$

The empirical semivariance, γ_a, is defined by $\gamma(h)_a = (\frac{1}{2n}) \sum_{i=1}^{n}(z(x_i + h) - z(x_i\ ))^2$, where h is the lag distance and n is the number of available $z_a$and $x_a$values. The number of lag values is determined by iterating between the available data points until a smooth semivariogram is found. The theoretical semivariogram, $\gamma(h,\beta)$, represents the geostatistical model. For example, with the exponential model:

$$\boldsymbol{\gamma(h) = \sigma^2 \left(1 - exp\left(-\frac{h}{\phi}\right)\right)} \tag{4-10}$$

The new objective function is defined by the following equation:

$$\boldsymbol{S(\beta) \approx \sum \tilde{f}(\beta)^2} \tag{4-11}$$

The global and local parameters are conditioned to the semivariogram function, which can reduce the solution space. The modified LM algorithm also applies a barrier method in each parameter to avoid invalid values in the forward model (Fletcher, 2013). This constraint can benefit the problem reducing parameter space must be explored (Kehl et al., 2005). The derivative matrix in the modified LM algorithm is calculated using a one-sided finite difference which evaluates *N* times the forward model to generate the Jacobian, where *N* is the number of parameters. The $\Delta\beta,$ perturbation size, used to calculate the derivatives also affects the accuracy of the Jacobian (Burg, 2000). The modified LM algorithm customizes the $\Delta\beta$ according to the scale of each parameter using a threshold based on the prior information of each parameter.

4.2.2 Uniqueness Property in Levenberg-Marquardt Algorithm

The LM algorithm optimizes the parameters of a non-linear model using the observations that are compared within the objective function. The number of parameters should be lower than the number of observations to have a well-posed problem. In our case, the number of parameters in the SRFs is very high in comparison with the number observations of related variables. In groundwater for example, the hydraulic conductivity is discretized and managed as an SRF to be able to model the process, and each element of the hydraulic conductivity field is an unknown parameter. Pressure head observations can be used as inverse data to characterize the hydraulic conductivity, but the number of pressure observations is probably lower than the number of parameters.

This ill-posed problem can be solved using the Tikhonov regularization methodology, which converts an ill-posed problem to well-posed through a regularization parameter using a Γ matrix which enforces the solution (Tikhonov & Arsenin, 1978). Although, the optimization LM algorithms also use a regularized term, λ, the problem is not well-posed because the number of parameter unknowns are not reduced, such as occurs in the Pilot Point method using super-parameters (Doherty, 2003). We study the solution space generated by the LM algorithm using multiple start points. These points are used as "informative" prior distribution in the stochastic method.

4.2.3 Constraining the Solution Space Using Levenberg-Marquardt Algorithm

The parameter estimation of SRF using the modified LM algorithm includes the semivariogram model in the objective function to constrain the solution space. The semivariogram model increases the systematic error in the solution due to the error produced by the fitting process. However, the solution conserves the semivariogram structure. Without this constrain, the local parameters can take values that do not follow the global structure of the SRF.

The guessed starting points in the modified LM algorithm can initialize with values that do not have any geostatistical structure. The modified LM algorithm can use multiple starting points to locate solutions close to a global minimum, but sometimes these solutions do not converge to a minimum. Global optimizations algorithms use the solution space to detect global minimum (e.g., basin of attraction algorithms). The Bayesian MAD technique constrains the characterization of the SRFs using prior information. The prior information is based on knowledge about the target variable. In the remainder of this chapter, we explore the generated solution space in MAD to further characterize the SRF parameters.

4.2.4 Applying Solution Space as Prior Distributions in MAD

The prior information $p(\theta, \vartheta|z_a)$ equation (2-1) in MAD formulation shows that structural parameters and anchors are conditioned to observation z_a. Prior information of structural parameter $p(\theta)$ can be non-informative when using uniform distributions. Anchor distribution ϑ is defined in terms of the target variable Y, demonstrated by the following equation,

$$\boldsymbol{\vartheta_a} = \boldsymbol{y}(\boldsymbol{x_{\vartheta a}}) + \boldsymbol{\varepsilon_a} \tag{4-12}$$

The anchor values are samples from a field y at $x_{\vartheta a}$ position. The y field is conditioned to structural parameters and observations of the target variable. The error ε_a represents the uncertainty associated with measurements error (Murakami, 2010). A practical procedure used to generate anchor distributions is shown in Algorithm 1. The prior information of structural parameters $p(\theta)$ is defined by a non-informative distribution such as uniform distribution. The parameter space should be explored by strategies such as Latin hypercube or hybrid sampling schemas. These strategies determine the number of samples required.

Algorithm 1 (Generate anchor distributions)

(1) Define the probabilistic distribution for each structural parameter (e.g., $p(\mu), p(\sigma^2), p(\phi)$) independently.

(2) Draw n samples for each distribution to explore the parameter space, e.g., Latin hypercube.

(3) Define the number of realizations per sample m

(4) For each set of samples $s_i = \{ \mu_i, \sigma_i^2, \phi_i \}$

- Calculate the covariance, C_i , using a geostatistical model conditioning to z_a.
- Generate m samples for each anchor using Multivariate Gaussian function $\vartheta_{ai} = G(s_i, C_i, x_{\vartheta a}, m)$.
- Add ϑ_{ai} to ϑ_a

The MAD technique uses samples of the prior distributions to explore the parameter space. The sample evaluation process is a very time consuming task. To reduce this process we explore "informative" prior distributions that reduce the parameter space constraining the samples to a small parameter space region. The LM and modified LM algorithm are evaluated to reduce the parameter space (Section 4-3).

The solution space of the ill-posed problem using LM algorithms can provide multiple solutions using multiple starting points. The MAD technique explores the solution space as informative prior information thus reducing parameter space. This evaluation is shown in Algorithm 2. The solution space is generated using the guessed starting points that cover the parameter spaces. Global and local parameters are defined by independent uniform distributions. We can constrain the parameter space using anchor distributions where the local parameters are conditioned to global parameters (Algorithm 1).

Algorithm 2 (Evaluate solution space in MAD)

(1) Generate n starting points.

- It could be based on distributions (e.g., Algorithm).

(2) For each set of samples

- Obtain the optimal solution using LM or a modified LM algorithm
- Add the solution to solution space

(3) Draw *m* samples from the multivariate solution space

(4) Apply the solution space samples to MAD as prior distributions.

4.2.5 Computational Cost

The LM algorithm requires us to evaluate the forward model n_p+1 times per iteration, where n_p is the number of parameters. Although the LM algorithm converges very fast to optimal parameters, the number of iterations depends on how the guess parameters are near to the solution. The number of starting points, *k,* increases the computational cost of this methodology. The total time of the LM algorithm execution is $T_{LM\ time} = t_{fm}*(n_p+1)*k*i_n,$ where t_{fm} is the forward model execution time and i_n is the average number of iterations. The computational cost of MAD depends on the number of samples, n_s, which is equal to *k,* and the number of realizations per samples, n_r , where the number of realizations per samples is defined by a preliminary converge analysis using few samples. The total time of MAD is $T_{MAD\ time} = t_{fm} * k * n_r$. The computational cost of using the solution space is defined by,

$$\boldsymbol{T_{time}} = \boldsymbol{kt_{fm}}((\boldsymbol{n_p} + \mathbf{1})\boldsymbol{i_n} + \boldsymbol{n_r}) \qquad (4\text{-}13)$$

4.2.6 Evaluating Solution Space in a HTCondor

We implemented the LM and modified algorithms within the framework by using the flexibility of MAD# to execute FMDs and RFGDs. To generate the solution space, the LM algorithm requires calculating multiple starting points. This is compared with the process executed by MAD# using the MAD technique, in which it requires us to evaluate multiple samples to estimate the parameters. MAD# can evaluate multiple prior samples in an HTCondor network deploying compressed packages. This package contain the MAD# core, the RFGD, the FMD and the project data that returns $\dot{z}_b$ data sets per sample generated by the forward model, shown in Figure 4-3a. Including the LM algorithm in the MAD# core, we can use the current development of MAD# to deploy compressed packages with the LM algorithm to receive the optimized parameters Figure 4-3b.

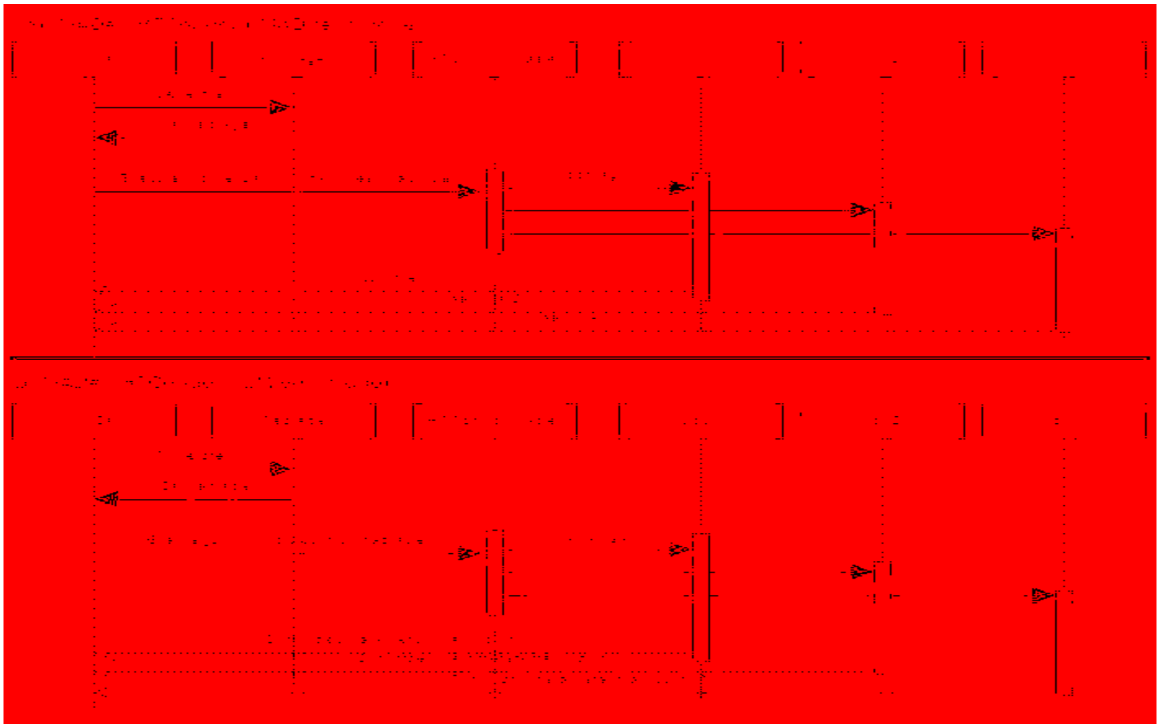


Figure 4-3: MAD # and HTCondor events diagram

4.2.7 Drawback of the Modified Levenberg-Marquardt Algorithm

The modified LM algorithm uses interpolated fields to optimize the global and local parameters of SRFs. To reproduce the variability of an SRF, the interpolated fields require a large amount of local parameters, which include more uncertainty in the parameter estimation. The Jacobian matrix used in the LM algorithm is based on perturbation of the parameters. In the case of global parameter, *partial sill*, a predicted field is not affected by the perturbation, which does not produce any change in the forward model response. The modified LM algorithm introduces the semivariogram model which affects the global parameter *partial sill*.

The modified LM algorithm constrains the parameter space using a barrier method that alters the minimization of the objective function, which can introduce false information in the optimization. To reduce this issue, the perturbation should be controlled by a threshold value that avoids large step sizes, which jump out of the parameter bounds.

The parameter estimation in non-linear problems cannot reproduce exactly the observations (Nowak & Cirpka, 2004). In the case of a modified LM algorithm for SRF estimation, the estimated parameters do not reproduce the exact observations, although this does not mean lack of accuracy. The optimized local parameters, $\dot{z}_a = \bar{y}(x_a)$, always generate residuals $r = y - \bar{y}(x_a)$.

4.3 **Results**

In this section we evaluate the solution space generated by LM algorithms using MAD.

4.3.1 Test Project Base

We used a base synthetic case with lateral confined groundwater flow through a heterogeneous aquifer is used as a test project. These heterogeneous aquifers are of interest because in reality aquifers are heterogeneous and this heterogeneity can affect the travel time of contaminants to as sensitive areas like drinking water wells. A synthetic Ln(Transmissivity) field Y was created using Gstat (Pebesma & Wesseling, 1998) with isotropic exponential covariance and no trend. This is common on synthetic projects (Li et. al, 2005). The SRF is generated by an unconditional realization with the following structural parameters:

- mean μ = -2,

- variance (*partial sill)* $\sigma^2 = 0.1$,
- range ϕ (*length scale*) = 500 meters,
- nugget ε_n = 0 meter.

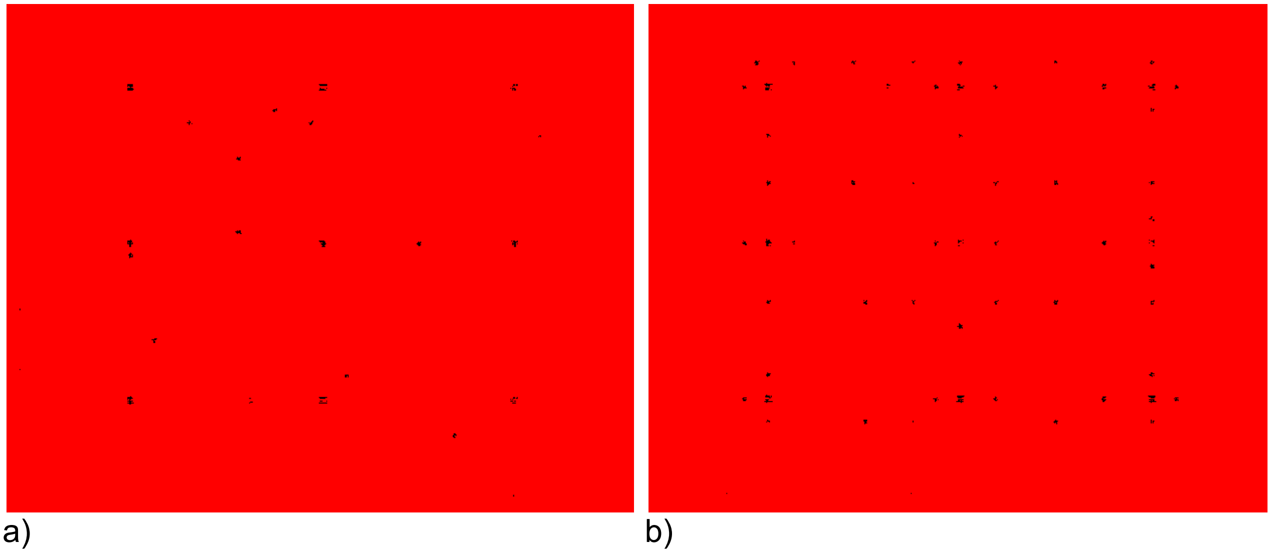


Figure 4-4: Observations and artificial points used in the test cases, a) Test cases 1,2,and 3 setup. b) Test cases 4 and 5 setup. Artificial points (black points), observations Ln(Transmissivity) (blue square), and pressure head observations (magenta triangle)

The field is 5000x4000 meters discretized into a 50x40 uniform rectangular grid. From this baseline field, the target data z_a were collected and the artificial points are known. The specific storage was a constant of 0.001. The boundary conditions in our base project were:

- Constant pressure head East= 100m
- Constant pressure head West=500m
- Hydraulic gradient= 1%.

To evaluate the solution space generated by optimization LM algorithms are used 9 observations, z_a. The transformation, $K = exp(Y)$, generates the transmissivity field K, which is related with pressure head through the forward model, MODFLOW-96 (Harbaugh, 1996). A set of 16 pressure head observations distributed in the domain is used as inversion data, z_b, shown in Figure 4-4.

4.3.2 Test Cases

We used five test cases to compare the solution space generated by the LM algorithms and MAD technique. The LM algorithms use the interpolation method Ordinary Kriging, which generates fields that represent a smooth version of the SRF depending on the number local parameters. The test cases evaluate 12 and 48 artificial points that are consider local parameters. The test cases 1, 2, and 3 include 12 artificial points located randomly, while test cases 4 and 5 use 48 artificial points located close to inversion data $z_{b.}$ For illustration, the interpolated fields generated with different amount of number of local parameters are displayed in Figure 4-5.

The global structural parameters, μ, $\sigma^2$and, ϕ, are estimated in all test cases. In test cases 3 and 5, the global and local parameters are characterized by MAD using prior distributions of Algorithm 1. Cases 1, 2, and 4 use guessed starting point samples uniformly distributed for each global and local parameter. In all test cases, MAD estimates the parameters as a posterior distribution. The setup of each test case is shown in Table 4-1. To illustrate the space solution generated by the LM algorithm, some local parameters are compared in Figure 4-6. The algorithms are evaluated using 500 starting points. The LM algorithm in case 1 produces low accurate and spread solutions, as shown in Figure 4-6(a)-(c). In the case 2, the modified LM algorithm uses the same multiple starting points as case 1, but generates more accurate solutions with lower variance Figure 4-6(d)-(f). Test case 2 shows solutions around the true values, indicating that there is a global minimum.

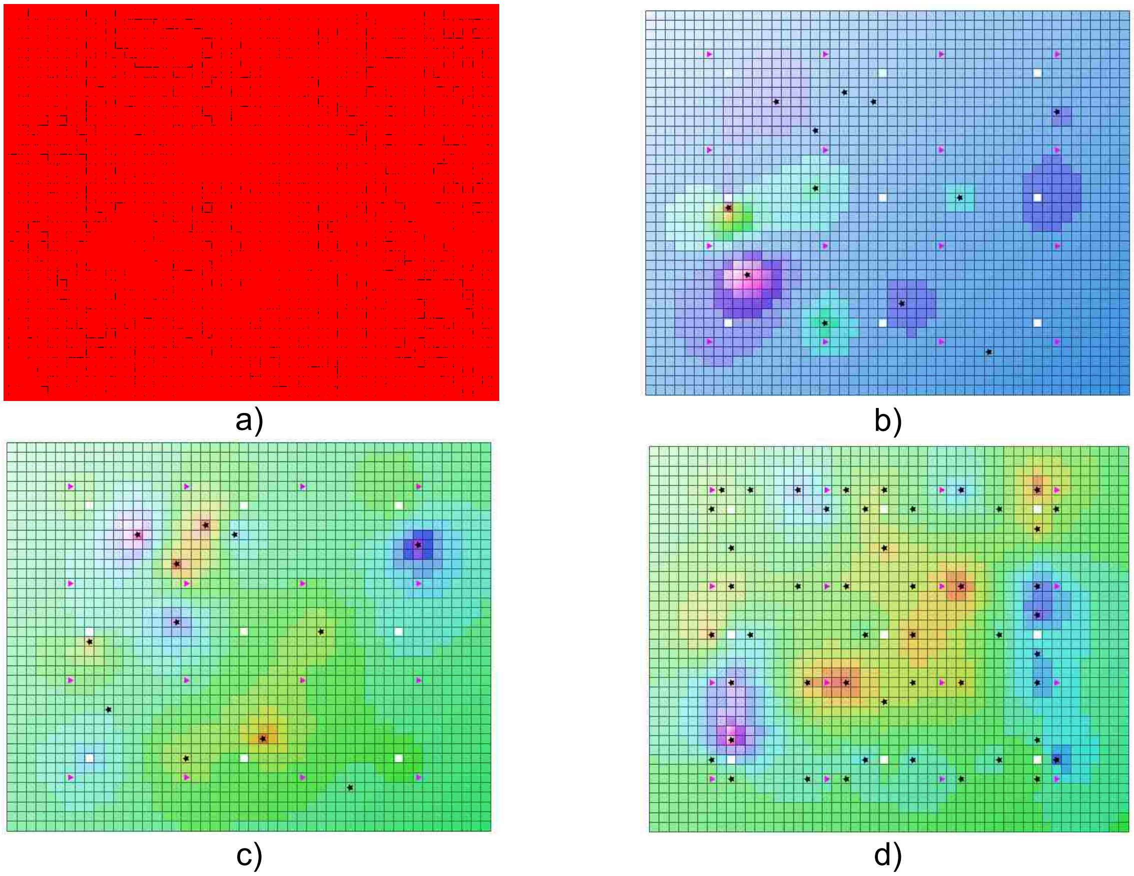


Figure 4-5: Interpolated fields using LM and modified algorithm. a) Base field of Ln(Transmissivity), b) Interpolated field by LM algorithm, c) Interpolated field by modified LM algorithm and d) Interpolated field by modified LM algorithm with more artificial points.

Table 4-1: Test cases setup

	Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5
No. artificial points	12	12	12	48	48
Starting point	Random	Random	-	Random	-
Prior Generated	LM	Modified LM	Algorithm 1	Modified LM	Algorithm 1

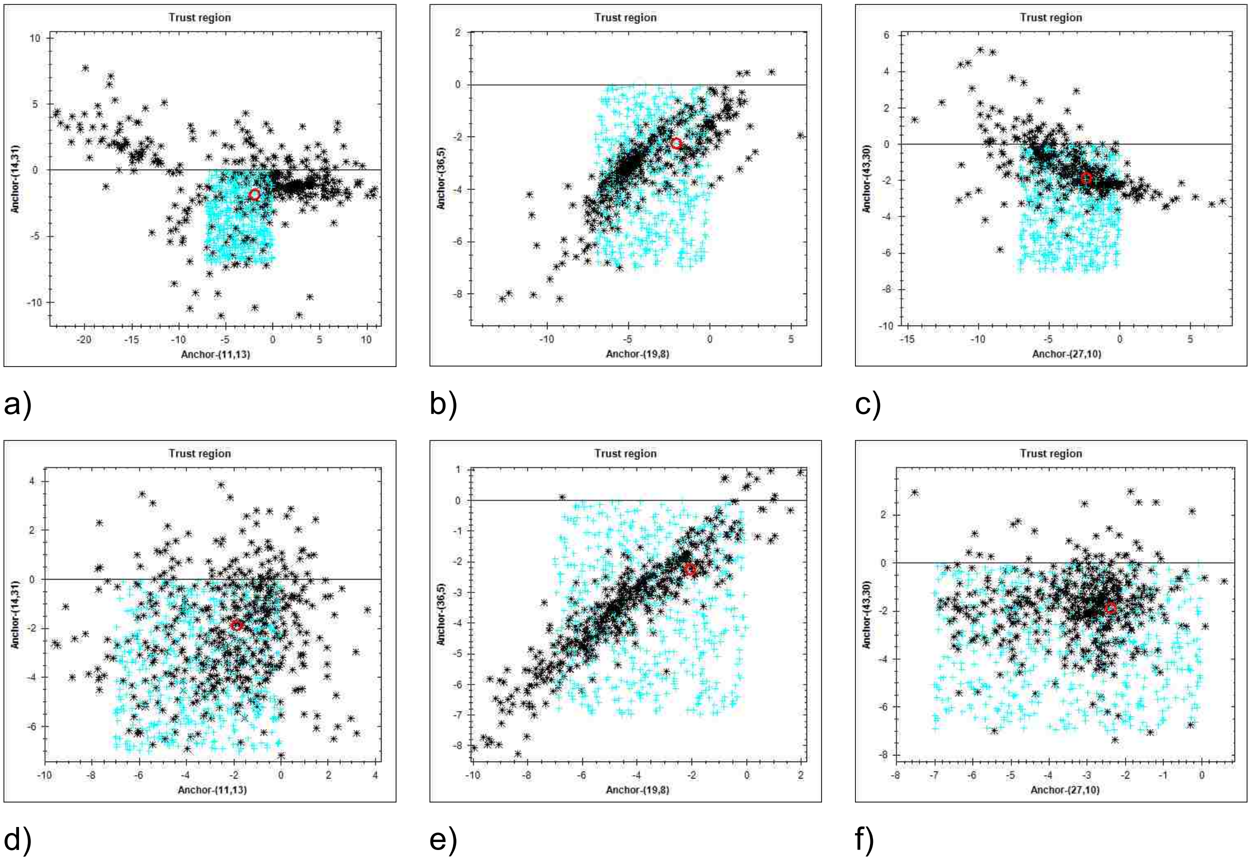


Figure 4-6: Comparison of starting points and space solution using LM and modified LM algorithm. a-c) LM algorithm, d-f) Modified LM algorithm. . a) and d) Location 11,13 vs. location 14,31., b) and e) Location 19,8 vs. location 36,5., c) and f) Location 27,10 vs. location 43,30. Starting point (cyan star), solutions (black star) and true value (red circle).

The results of MAD exploration of test cases 1, 2 and 3 are shown in the Figure 4-7. The number of samples evaluated in MAD is 500 per test case. A convergence analysis showed that the non-parametric likelihood converges in 300 realizations. In case 1, the posterior distributions do not reproduce the true values. This is caused by the large variability in the solution space. In case 2, the accuracy and variability of the posterior distributions improve in comparison with case 1. Using prior distributions generated with Algorithm 1 and test case 3 fully characterizes the parameters appropriately. The global parameter variance, σ^2 (*partial sill*), has a large uncertainty in cases 1 and 2, which indicates that LM algorithms have a large oscillation in the estimation of σ^2.

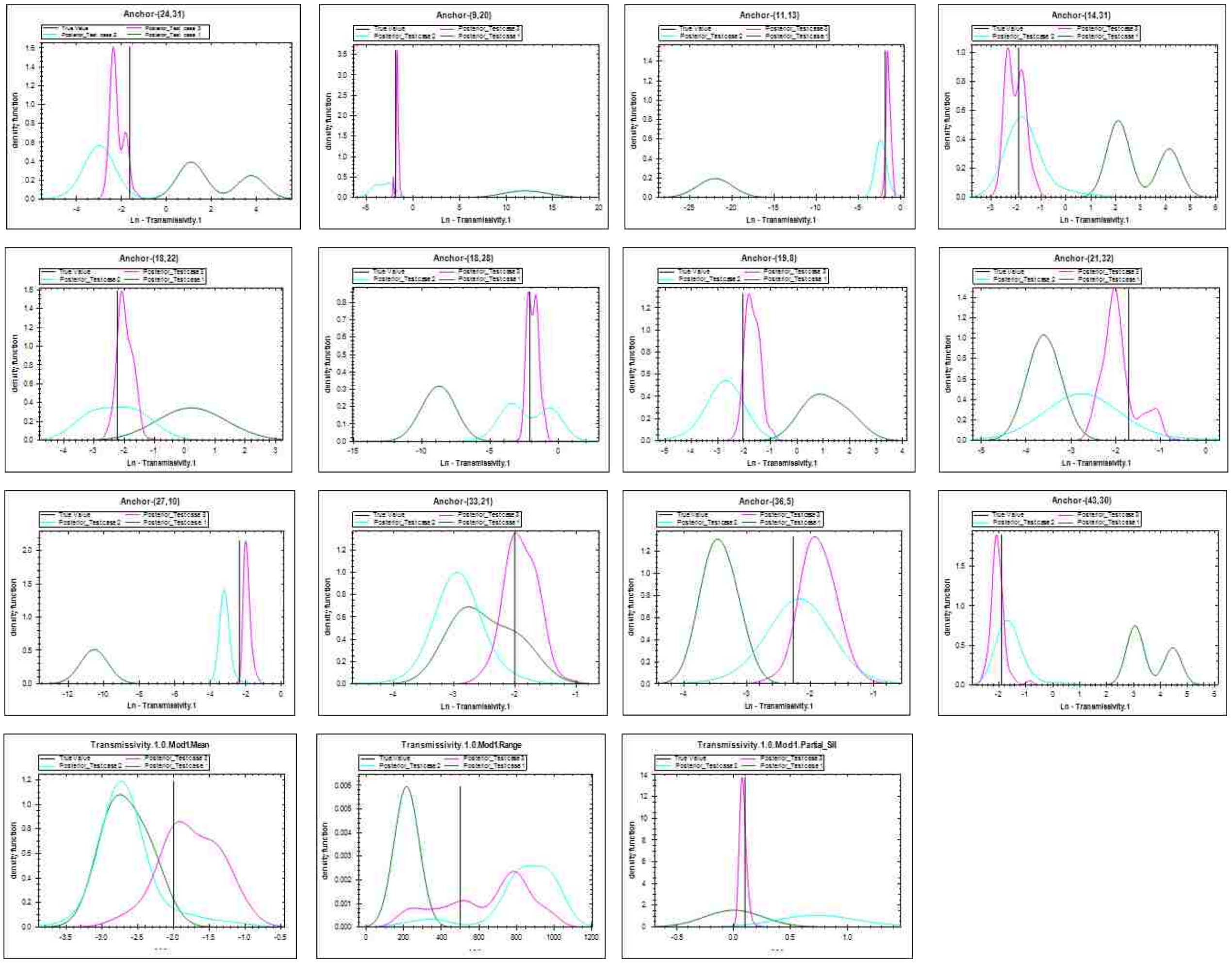


Figure 4-7: Posterior distribution of parameters comparison. Test case 1 (green), test case 2 (cyan), test case 3 (magenta). Vertical line (true value).

The number of local parameters in the project improves the characterization of the global parameters (Figure 4-8). However, the covariance variability is higher in the modified algorithm. Using the maximum likelihood values of all global and local parameters of the case 4, the realizations show larger variability than in case 5 (Figure 4-9).

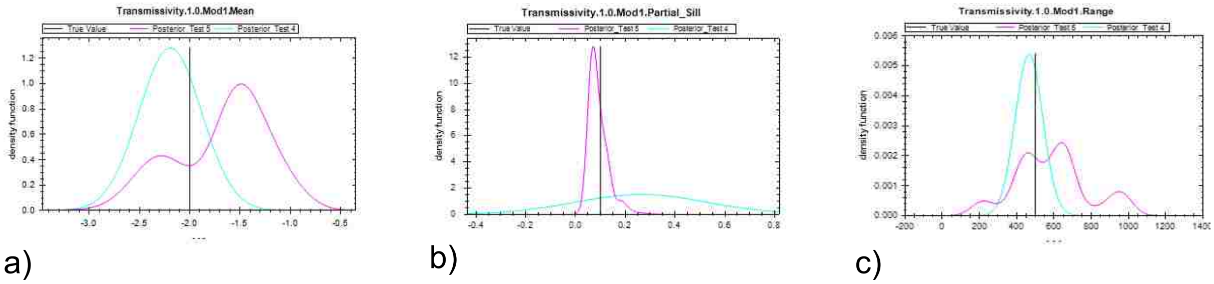


Figure 4-8: Posterior distribution of global parameters in the test cases 4 and 5. a) mean b) variance (partial sill) and c) Range. Test case 4 (cyan), Test case 5 (magenta).

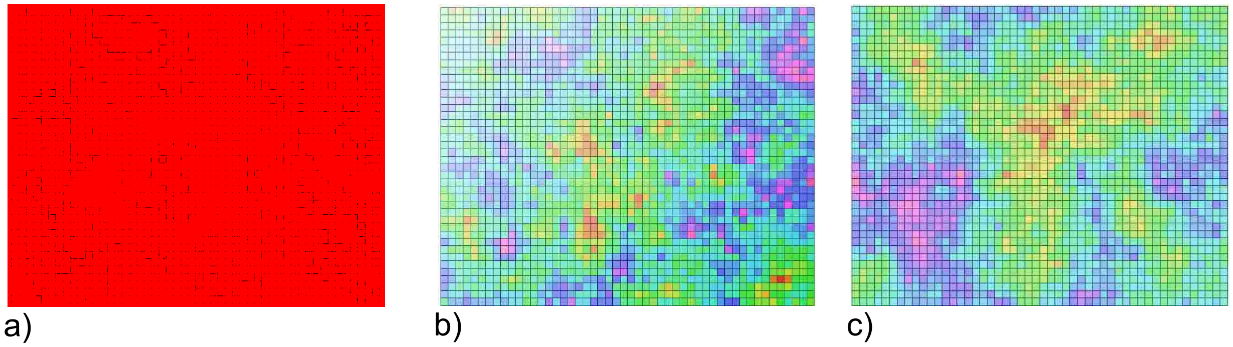


Figure 4-9: Realizations of Ln(Transmissivity) created using the maximum likelihood values of test cases 4 and 5. a) Base field, b) Test case 4, and c) Test case 5.

The number of parameters increases the computational cost of using Algorithm 2, shown in Table 4-2. We observed that the average number of iterations decreased using the modified LM algorithm. The MAD technique is affected by the number of parameters, which is explained by the computing cost to generate realizations with more constrained parameters by RFGD. The average number of iterations indicates that the threshold used, 1e-3, is very low.

Table 4-2: Performance test cases in HTCondor

Test Case	Average number of iterations	CPU time Algorithm 2 (hour)	CPU time MAD (hour)
Test 1	17.9	7	32
Test 2	7.1	9.5	33.5
Test 3	-	-	40.5
Test 4	6.7	21.5	41.5
Test 5	-	-	40

4.4 **Conclusions**

We successfully demonstrated the flexibility of the MAD# framework to support other inverse modeling techniques, such as the LM algorithms. The conventional LM algorithm generates large variability in the solution space, which generates posterior distributions far from true values (Figure 4-7). However, adding the semivariogram in the objective function, the solution space improves posterior distributions. Using the LM algorithms, the global parameter variance, σ^2, generates large uncertainty.

The solution space generated by the LM algorithms showed that interpolated fields can characterize SRFs, although they required large amounts of artificial points. We did not evaluate the effect of the artificial point location in this project. The MAD technique implemented in MAD# helps us to detect the global minimum of the parameters using posterior distributions. The computational cost of this new implementation could be a drawback in the adoption. The prior distribution shows a large impact on parameter estimation. The new method can help generate priors when the distribution of parameters is unknown. The execution sequence of the optimization algorithm and the Bayesian method increases the computational time, but is a useful alternative in the generation of prior distributions conditioned to observations. The computational cost of generating a solution space can be managed successfully through the HTCondor platform.

5 CONCLUSIONS AND FUTURE WORK

The new inverse modeling framework presented in this dissertation is an alternative to setup Bayesian inverse modeling procedures. The platform MAD# can be integrated with other forward models through its "plug-and-play" modular drivers-based architecture. New drivers can increase the adoption of this framework and will help more people to apply inverse modeling as a common practice. Although Bayesian procedures, such as the MAD procedure, can be computational intensive, the framework addresses this issue through the integration with a high throughput computing platform. This integration will allow users to submit the parameter inversion problem in a computer network that can be configured by the user.

In the case studies presented in Chapter 2, the SRF Ln(Transmissivity) is characterized using the indirect measurements of pressure head via solving head equation implemented in MODFLOW. The anchors defined in the SRF domain show the posterior distribution of Ln(Transmissivity) and the uncertainty at the anchor locations. The posterior pdfs of structural parameters and anchors characterized the global and Ln(Transmissivity) field, respectively. The posterior pdfs obtained with more Type-B measurements produced values closer to true values. MAD# provides a user interface that allows comparison of multiple scenarios. Although MAD# was tested using hydrological models, the framework was designed to support generic forward models, which allows developers to create new drivers to solve inverse problems of other scientific areas.

In Chapter 3, HTCondor showed a considerable improvement in the total duration of the Monte Carlo process in MAD#. The high variability in the duration time in the small cases studies (30x30 grid and 60x60 grid) is produced by the network latency. The effect of the latency can be reduced by increasing the simulation time per node, which can be achieved by processing more MAD# samples per node. The HTCondor "vanilla" environment was used to submit MAD# jobs, allowing MAD# core to run in each node and to transfer the simulation output files to the master node. Although, the transfer file system works adequately, it is necessary to evaluate the Hadoop File System (HDFS) to reduce the high variability in the response of small MAD# projects.

In Chapter 4, we successfully demonstrated the flexibility of the MAD# framework to support other inverse modeling techniques, such as the LM algorithms. The conventional LM algorithm generates large variability in the solution space, which generates posterior distributions far from true values (Figure 4-7). However, adding the semivariogram in the objective function, the solution space improves posterior distributions. Using the LM algorithms, the global parameter variance, σ^2, generates large uncertainty.

In addition to the overarching goals of designing, developing and testing a new Bayesian inverse modeling framework, this research also aimed to address two specific inverse modeling challenges proposed by Carrera (2005) including integration with geographic information systems and simplifying adoption of inverse modeling techniques. MAD# addresses the first of these challenges through the adoption of an open source geographic information system software library called DotSpatial (http://www.dotspatial.org), which directly supports the integration of common GIS data formats and online basemaps to facilitate geolocation of the study area.

MAD# is intended to address the challenge of simplifying inverse modeling through its relatively user-friendly Windows-based graphical user interface (GUI). This GUI allows users to setup the inversion procedure and link to existing forward models without coding. The driver-based approach provides the target variables and inversion data that users may use in the inversion. In this way, MAD# reduces the complexity in the configuration of inverse modeling problems. Carerra's challenge regarding the incorporation of geological data is addressed by the integration of geostatistical models that represent geological variables. The geostatistical parameter characterization can be managed by the random field generators supported by MAD#.

Different types of inversion methods such as MAD and Levenberg-Marquardt algorithms provide more alternatives for parameter characterization. Carrera also stated that new inverse modeling frameworks should quantify the model uncertainty, which is addressed by MAD# through the estimation of the fully characterized probability distribution function of the parameters.

MAD# is now a C# implementation; we wish to convert the framework into a multiplatform application through other programming languages such as C and C++. The integration of other likelihood software applications is important for the parameter estimation. New likelihood drivers should be implemented in the future in order to improve the parameter estimation. Indeed, efforts are already underway to port the code to C++ for execution in Linux environments.

REFERENCES

Alcolea, A., Carrera, J., & Medina, A. (2006). Pilot points method incorporating pior information for solving the groundwater flow inverse problem. *Advance in Water Resources, 29*(11), 1678-1689.

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., & Avellano, A. (2009). The data assimilation research testbed: A community facility. Bulletin of the American Meteorological Society, 90(9), 1283-1296.

Alexandrov, G., Ames, D., Bellocchi, G., Bruen, M., Crout., N, Erechtchoukova, M., Hildebrandt, A., Hoffman, F., Jackisch, C. & Khaiter, P. (2011), Technical assessment and evaluation of environmental models and software: Letter to the Editor, *Environmental Modelling & Software*, *26*, 328-336.

Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T. & Valentine, D. (2012), HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis, *Environmental Modelling & Software*, *37*, 146-156.

Banta, E.R., Hill, M.C., Poeter, E., Doherty, J.E. & Banbendreier, J. (2008), 'Building model analysis applications with the Joint Universal Parameter IdenTification and Evaluation of Reliability (JUPITER) API', *Computers & Geosciences*, *34*, 310-319.

Barry, D. A. (1990). Supercomputers and their use in modeling subsurface solute transport. *Reviews of Geophysics, 28*(3), 277-295.

Bates, B.C. & Townley, L.R. (1988), Nonlinear, discrete flood event models, 1. Bayesian estimation of parameters, *J. Hydrol. 99*, 61-76, doi: 10.1016/0022-1694(88)90078-9.

Bellin, A. & Yoram R. (1996). HYDRO_GEN: A spatially distributed random field generator for correlated properties. *Stochastic Hydrology and Hydraulics*, *10*(4), 253-278.

Bennet, A. F. (2002). *Inverse Modeling of the Ocean and atmospher.* Cambridge: Cambridge University Press.

Bertshinger, E. (2001). Multiscale Gaussian Random Fields and Their Application to Cosmological Simulations. *The Astrophysical Journal Supplements*.

Bochkanov, S. (2014),'ALGLIB' [Computer program], Available at www.alglib.net (Accessed 03 September 2014)

Burg, C. (2000). Use of discrete sensitivity analysis to transform explicit simulation codes into design optimization codes. *Fourth Mississippi State Conference on Differential Equations and Computational Simulations, Electronic Journal of Differential Equations*, 13-27.

Cai, Y., Judd, K., Thain, G., & Wright, S. (2014). Solving Dynamic Programming Problems on a Computational Grid. *Comput Econ*.

Carrera, J., Alcolea, A., Medina, A., Hidalgo, J., & Slooten, L. J. (2005). Inverse Problem in hydrogeology. *Springer-Verlag, 13*, 206-222.

Carrera, J. & Neuman, S.P. (1986), Estimation of aquifer parameters under transient and steady state conditions: 1. Maximum likelihood method incorporating prior information, *Water Resour. Res.*, *22*(2),199-210, doi: 10.1029/WR022i002p00199.

Chen, X.M.H., Hahn, M.S., Hammond, G.E., Rockhold, M.L., Zachara, J.M. & Rubin, Y. (2012). Three-dimensional Bayesian geostatistical aquifer characterization at the Hanford 300 Area using tracer test data, *Water Resour. Res.*,*48*, W06501, doi: 10.1029/2011WR010675.

Chiang, W.-H., & Kinzelbach, W. (2001). *3D-Groundwater Modeling With Pmwin: A Simulation System for Modeling Groundwater Flow and Pollution.* Springer Verlag.

Dagan, G. (1987), Theory of solute transport by groundwater, *Ann. Rev. Fluid Mech.*, *19*, 183-215.

Dagan, G. (2011), An overview of stochastic modeling of groundwater flow and transport: From theory to applications, *Eos, Transactions American Geophysical Union*, *83*(53),621-625.

Delhomme, J.P. (1979). Spatial variability and uncertainty in groundwater flow parameters: A geostatistical approach. *Water Resour. Res.*, *15*, 269-280.

Doherty, J. (1994), PEST: a unique computer program for model-independent parameter optimisation., *Water Down Under 94: Groundwater/Surface Hydrology Common Interest Papers*, 551.

Doherty, J. (2003). Ground water model calibration using pilot points and regularization. *Groundwater, 41*(2), 170-177.

Dunsford, H. & Ames, D.P. (2011), MapWindow 6.0: an extensible architecture for cartographic symbology, *OSGeo Journal*, *8*, 31-36.

Farmani, M. B., Kitterød, N.-O., & Keers, H. (2008). Inverse modeling of unsaturated flow parameters using dynamic geological structure conditioned by GPR tomography. *Water Resources Research, 44*(W08401).

Fienen, M., Muffels, C., & Hunt, R. (2009). On Constraing pilot point calibration with regularization in PEST. *Groundwater, 47*, 835-844. doi:10.1111/j.1745-6584.2009.00579.x

Finsterle, S., & Kowalsky, M. (2011). A truncated Levenberg-Marquardt algorithm for the calibration of higjly parameterized nonlinear models. *Computer & Geosciencies, 37*(6), 731-738.

Firmani, G., Fiori., A & Bellin A.(2006). Three-dimensional numerical analysis of steady state pumping test in heterogeneous confined aquifers, *Water Resour. Res., 42*, W03422, doi:10.1029/2005WR004382.

Fletcher, R. (2013). *Practical methods of optimization.* Jonh Wiley & Sons.

Gallagher, M. & Doherty, J. (2007), Parameter estimation and uncertainty analysis for a watershed model, *Environmental Modelling & Software*, *22*(7), 1000-1020.

Genz, A. & Bretz, F. (2009). *Computation of Multivariate Normal and t Probabilities*, Springer-Verlag, Heidelberg.

Goncharsky, A., & Romanov, S. (2013). Supercomputer technologies in inverse problems of ultrasound tomography. *Inverse Problems*.

Hanke, M. (1997). A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems, 13*(1), 76.

Harbaugh, A. (1996). *Programmer's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model: U.S. Geological Survey Open-File.* USGS.

Hayfield, T. & Racine, J.S. (2008). Nonparametric Econometrics: The np Package, *Journal of Statistical Software*, *27*(5).

Jackson, C. (2011), Multi-State Models for Panel Data: The msm Package for R, *Journal of Statistical Software*, *38*(8), 1-29.

Kanso, A., Gromaire, M.C., Gaume, E., Tassin, B. & Chebbo, G, (2003), Bayesian approach for the calibration of models: application to an urban stormwater pollution model, *Water Science and Technology*, *47*(4), 77-84.

Kehl, R., Bray, M., & Van Gool, L. (2005). ull body tracking from multiple views using stochastic sampling. *IEEE Computer Society Conference on. 2*, pp. 129-136. Computer Vision and Pattern Recognition.

Li, W., Nowak, W., & Cirpka, O. A. (2005). Geostatistical inverse modeling of transient pumping tests using temporal moments of drawdown. *Water resources research*, *41*(8).

Lourakis, M., & Argyros, A. (2005). Is Levenber-Maquardt the most efficient optimization algorithm for implementing bundle adjustment? *Tenth IEEE International Conference on Computer Vision*, (pp. 1526-1531).

Madsen, K., Nielsen, H., & Tingleff, O. (2004). *Methods for non-linear least squares problems.* Informatics and Mathematical Modelling, Technical University of Denmark.

Math.NET team (2014), *Math.NET Numerics (version 2.2.0)* [Computer program], Available at https://mathnetnumerics.codeplex.com/releases/view/126119 (Accessed 03 September 2014)

McCloskey, G., Ellis, R., Waters, D., & j., S. (2011). PEST hydrology calibration process for source catchments – applied to the Great Barrier Reef, Queensland. *19th International Congress on Modelling and Simulation.* Perth.

Meerschaert, M. M., M. Dogan, R. L. Van Dam, D. W. Hyndman, and D. A. Benson (2013), Hydraulic conductivity fields: Gaussian or not?, *Water Resour. Res.*, *49*, doi:10.1002/wrcr.20376.

Murakami, H. (2010). Development of A Bayesian Geostatistical Data Assimilation Method and Application to the Hanford 300 Area.(Doctoral dissertation). *University of California, Berkeley*.

Nowak, W., & Cirpka, O. (2004). A modified Levenberg-Marquardt algorithm for quasi-linear geostatistical inversing. *Advance in Water Resources*, 737-750.

Osorio, C., Over, M., Ames, D.P. & Rubin, Y. (2012). An Extensible Inverse Modeling Software Architecture for Parameter Distribution Estimation, *2012 International Congress on Environmental Modelling and Software*, <http://www.iemss.org/sites/iemss2012/proceedings/D6_1005_Osorio_et_al.pdf>.

Osorio-Murillo, C., Over, M., Ames, D., & Rubin, Y. (2014). MAD#: A Framework for Inverse Modeling and Uncertainty Characterization. *Environmental Modeling and Software(in second revision).*

Pebesma, E., & Wesseling, C. (1998). Gstat: a program for geostatistical modeling, prediction and simulation. *Computer & Geosciences, 24*(1), 17-31.

Poeter, E.P. & Hill, M.C. (1997). Inverse Methods: A necessary next step in groundwater modeling, *Ground Water*, *35,* 250-260.

Poeter, E.P. & Hill, M.C. (1999). UCODE, a computer code for universal inverse modeling, *Computers & Geosciences*, *25*, 457-462.

Rubin, Y. & Dagan, G. (1988), Stochastic analysis of the effects of boundaries on spatial variability in groundwater flow: 1. Constant head boundary. *Water Resour. Res.* 24(10), 1689-1697, doi: 10.1029/WR024i010p01689.

Rubin, Y. & Dagan, G. (1989), Stochastic analysis of the effects of boundaries on spatial variability in groundwater flow: 2. Impervious boundary. *Water Resour. Res.* 25(4), 707-712, doi: 10.1029/WR025i004p00707.

Rubin, Y. (2004). Stochastic hydrogeology – challenges and misconceptions, *Stochastic Environmental Research and Risk Assessment*, *18*, 280-281.

Rubin, Y., Chen, X., Murakami, H., & Hahn, M. (2010). A Bayesian approach for inverse modeling, data assimilation, and conditional simulation of spatial random fields. *Water resourses reseach, 46*(W10523).

Šimůnek, J., K., H., Šejna, M., & van Genuchten, M. T. (1998). The Hydrus-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably-saturated media Version 1.0. *International Ground Water Modeling Center*, 186.

Tikhonov, A., & Arsenin, V. (1978). Solutions of Ill-Posed Problems. *Mathematics of Computation*, 1320-1322. doi:10.2307/2006360

Tompson, A. F., Falgout, R. D., Smith, S. G., Bosl, W. J., & Ashby, S. F. (1998). Analysis of subsurface contaminant migration and remediation using high performance computing. *Advances in Water Resources, 22*(3), 203-221.

Torn, R., & A., A. (2009). The Data Assimilation Research Testbed: A Community Facility. *American Meteorological Society.*

Wilhelmm, S. & Manjunath, B.G. (2013). Tmvtnorm [computer program]. Available at http://cran.r-project.org/web/packages/tmvtnorm/index.html (Accessed 03 September 2014).

Yang, Y., Over, M., & Rubin, Y. (2012). Strategic placement of lacalization devices (such as pilot points and anchors) in inverse modeling schemes. *Water Resources Researcg*, *48*. doi:10.1029/2012WR011864

Yates, D., Sieber, J., Purkey, D. & Huber-Lee, A. (2005), WEAP21—A Demand-, Priority-, and Preference-Driven Water Planning Model, *Water International*, *30*(4), 487-500.

Yeh, T. C. J., L. W. Gelhar, and A. L. Gutjahr (1985), Stochastic analysis of unsaturated flow in heterogeneous soils: 1. Statistically isotropic media, *Water Resour.Res.,* 21(4), 447-456, doi: 10.1029/WR021i004p00447.

Yeh, T.-C. J., S., L., J., G. R., Baker, K., Brainard, J., D., Alumbaugh., D & LaBrecque, D. (2002). A geostatistically based inverse model for electrical resistivity surveys and its applications to vadose zone hydrology. *Water Resources Research, 38*(12), 14-1.

Yeh, W. W. (1986, February). Review of Parameter Identification Procedures in Groundwater Hydrology: The Inverse Problem. *Water Resources Research, 22*(2), 95-108.

Zhou, M., & Mascagni, M. (2000). The cycle server: a Web platform for running parallel Monte Carlo applications on a heterogeneous Condor pool of workstations. *2000 International Workshops on*, (pp. 111-118). Toronto, Ont.

APPENDIX A. USER MANUAL OF MAD#

A.1 What is MAD#?

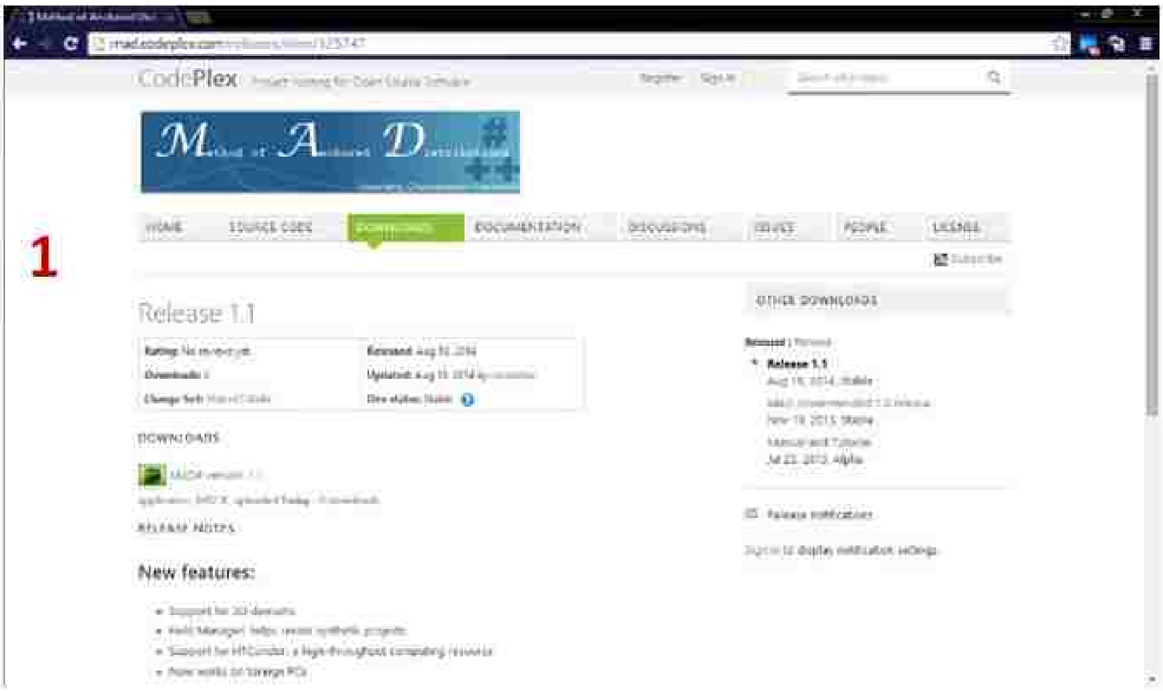
The inverse problem exists for spatially variable fields such as hydraulic conductivity in groundwater aquifers or rainfall intensity in hydrology. Common to all these problems is the existence of a complex pattern of spatial variability of the target variables and observations, the multiple sources of data available for characterizing the fields, the complex relations between the observed and target variables and the multiple scales and frequencies of the observations. The method of anchored distributions (MAD) that we implement here is a general Bayesian method of inverse modeling of spatial random fields that addresses this complexity. The central elements of MAD are a modular classification of all relevant data and a new concept called "anchors." Data types are classified by the way they relate to the target variable. Anchors are devices for localization of data: they are used to convert nonlocal, indirect data into local distributions of the target variables. The target of the inversion is the derivation of the joint distribution of the anchors and structural parameters, conditional to all measurements, regardless of scale or frequency of measurement. The structural parameters describe large-scale trends of the target variable fields, whereas the anchors capture local in-homogeneities.

This project strives to develop a community-based, open-source computational platform for Bayesian inverse modeling and conditional simulations in hydrology. It is driven by the need to provide easily- accessible tools for applications and a platform for broad-based, long-term development, built to meet the challenges brought upon by the ever increasing complexity of modern computational tools, the diversity of data types and the breadth and depth of the subject matters needed for applications. The project is inspired by the success experienced by other communities with a long tradition of community-based development efforts. We have constructed a kernel that includes a modular computational platform and a user-interface that allows users and developers to link the kernel with their models. The kernel is built using open-source architecture. CUAHSI will act as the custodian of the MAD# computational platform. This implies securing the integrity of the MAD# kernel, updating it and making it available to users and developers. This effort is in line with CUAHSI's overall strategy. MAD# will facilitate the science discovery process by making advanced inverse modeling tools available to users and an inviting development platform for developers. It will enhance the quality of hydrologic science by encouraging collaboration between hydrologists and earth scientists, and provide a platform for statisticians and computer scientists. MAD# will provide educators and students with access to modern, well-documented computational tools, and with the motivation to experiment with them, because it will be understood as a general and extensible tool that is useful far beyond the classroom. MAD# will make a contribution to the growing culture of community-based, open-source analytical tools that make science more accessible. Long-standing, well-tested, transparent and hence better-trusted analytical tools form the base for a constructive public discourse on matters relating to environmental regulations

A.2 How do I Install MAD#?

MAD# is open-source software and can be found on Codeplex (http://mad.codeplex.com), a website that hosts open-source projects. MAD# has been successfully tested on Windows 7 and 8. In addition to MAD#, other software needs to be installed on your computer. The supplemental software required are a forward modeling software (MODFLOW96, MODFLOW 2005, HYDRUS 4.15) and the statistical language R (2.15.0-3) with specific packages. Links for downloading these supplements are given below.

1) Go to the MAD#'s Codeplex Downloads page. Click on the recommended 1.1 release name to download the installer.



Release 1.1

Rating: No reviews yet
Downloads: 1
Released: Aug 19, 2014
Updated: Aug 19, 2014
Dev status: Stable

DOWNLOADS

MAD# version 1.1

RELEASE NOTES

New features:

- Support for 3D domains
- Field Manager: helps create synthetic projects
- Support for HTCondor, a high-throughput computing resource
- Now works on foreign PCs

OTHER DOWNLOADS

Released | Planned

Release 1.1
Aug 19, 2014, Stable

MAD# (recommended 1.0 release)
Nov 19, 2013, Stable

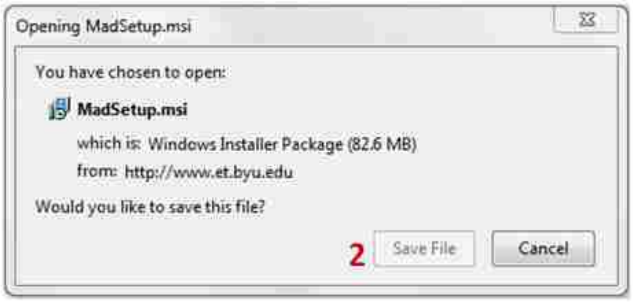
Manual and Tutorial
Jul 22, 2013, Alpha

Release notifications

Sign in to display notification settings

Figure A- 1: Web page MAD#

2) Depending on your web browser, a pop-up may appear. Click on **Save File**. This will start the download.


Opening MadSetup.msi

You have chosen to open:

MadSetup.msi

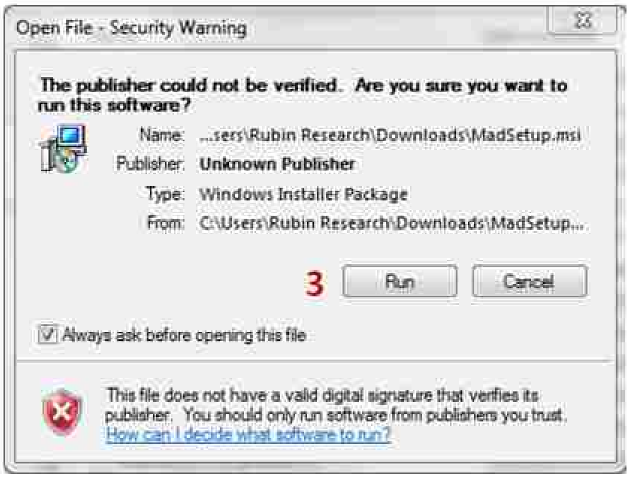
which is: Windows Installer Package (82.6 MB)

from: http://www.et.byu.edu

Would you like to save this file?

Figure A- 2: Installing MAD#

3) Once the download is complete, open the file. A warning may appear (it may look different from the one below depending on your version of Windows OS). If it does appear, click on **Run**.


Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: ...sers\Rubin Research\Downloads\MadSetup.msi

Publisher: **Unknown Publisher**

Type: Windows Installer Package

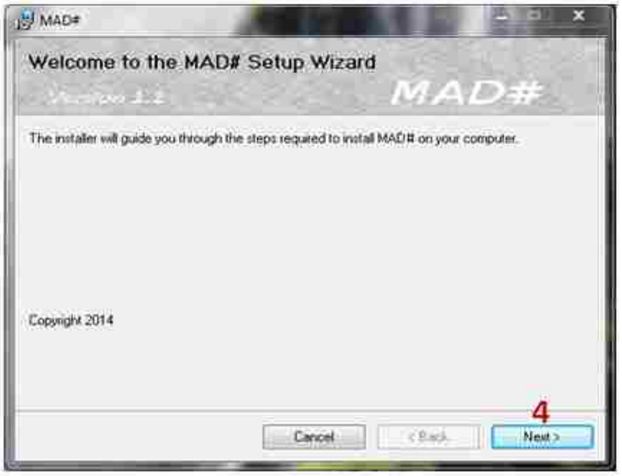
From: C:\Users\Rubin Research\Downloads\MadSetup...

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

Figure A- 3: Warning in the installation

4) The MAD# installer will appear. Click **Next>**.



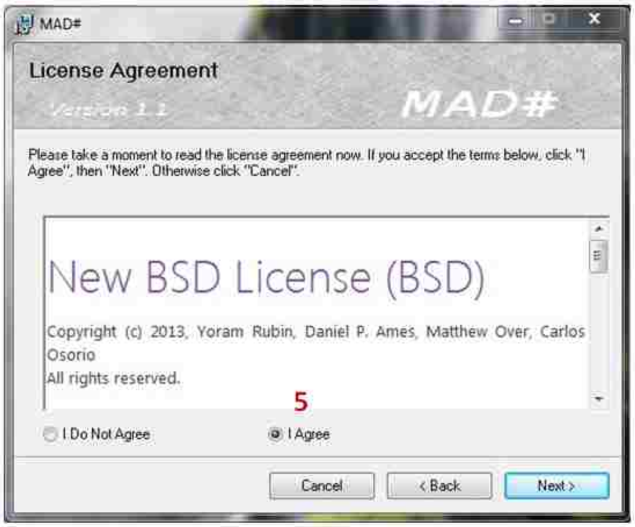
Welcome to the MAD# Setup Wizard

The installer will guide you through the steps required to install MAD# on your computer.

Copyright 2014

Figure A- 4: Welcome installation MAD#

5) The license agreement will appear next. Read through the agreement, click the **I agree** radio button, and then click **Next>**.



License Agreement

Please take a moment to read the license agreement now. If you accept the terms below, click "I Agree", then "Next". Otherwise click "Cancel".

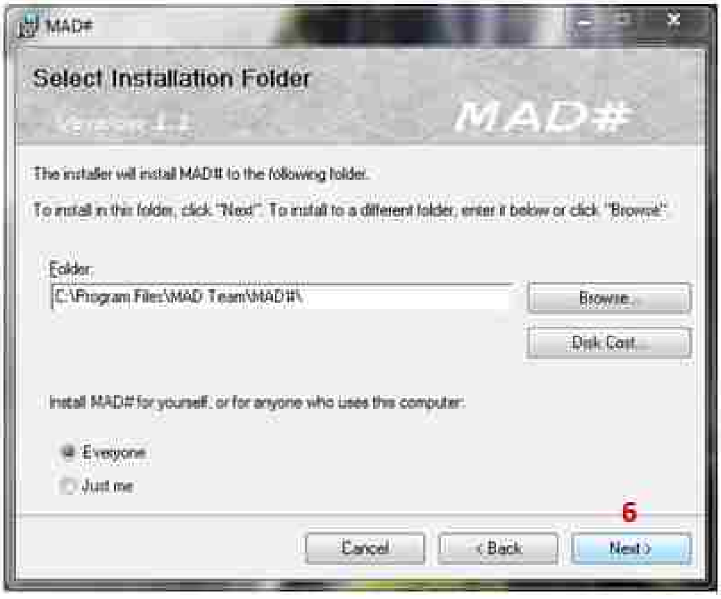
New BSD License (BSD)

Copyright (c) 2013, Yoram Rubin, Daniel P. Ames, Matthew Over, Carlos Osorio
All rights reserved.

○ I Do Not Agree ● I Agree

Figure A- 5: MAD# license

6) Either leave the installation folder as the default, or choose the folder in which you wish to install MAD#. Also choose if you would like every user on your machine to have MAD# installed. Click **Next>**.



Select Installation Folder

Version 1.1

The installer will install MAD# to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

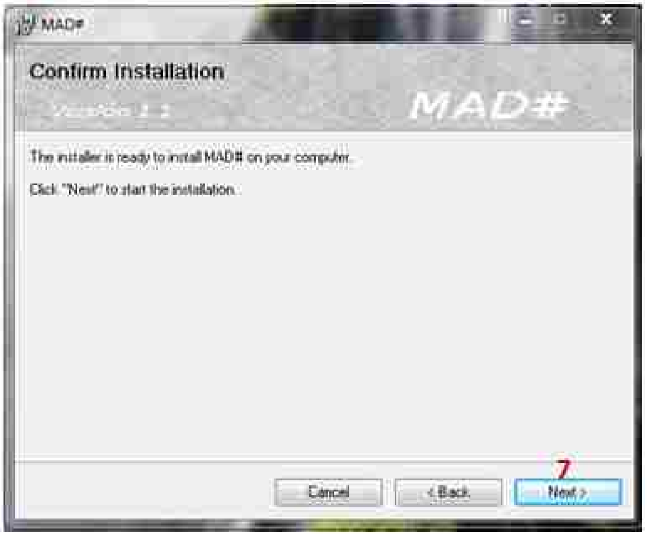
Folder:
C:\Program Files\MAD Team\MAD#\

Install MAD# for yourself, or for anyone who uses this computer:

● Everyone
○ Just me

Figure A- 6: Select installation folder

7) When you wish to start installation, click on **Next>**.



Confirm Installation

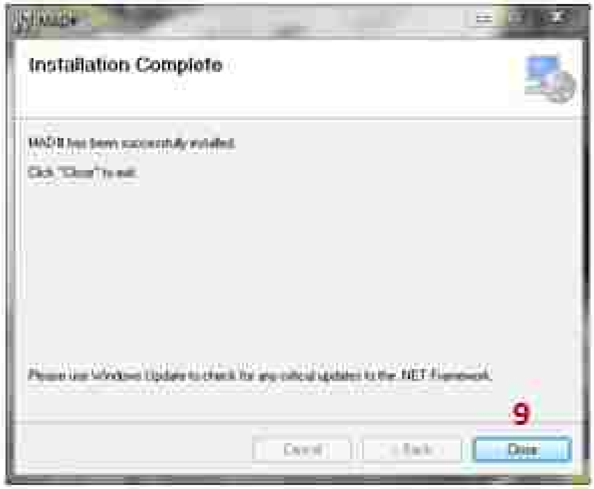
Version 1.1

The installer is ready to install MAD# on your computer.

Click "Next" to start the installation.

Figure A- 7: Confirm installation

8) A loading bar will appear while MAD# installs. Wait for it to fill and you will be automatically progressed to the next step.



Installation Complete

MAD# has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Figure A- 8: Complete installation

9) Now MAD# will have a desktop icon and can be found in the folder in which you installed it.

10) In addition to MAD#, you will also need to install:

At least one of the following forward model software packages:

MODFLOW-96

U.S. Geological Survey, Hydrologic Analysis Software Support Program, Reston, VA. Recommended: **PMWIN 5.3.**

Chiang, H.W., Processing MODFLOW for Windows (PMWIN), Version 5.3, Simcore Software, 2005.

MODFLOW-2005

Harbaugh, A.W., 2005, MODFLOW-2005, The U.S. Geological Survey modular ground-water model—the Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods 6-A16 Recommended: **MODELMUSE 3.2**

Winston, R. ModelMuse, version 3.2, United States Geological Survey, 2014.

HYDRUS 4.15

Šimůnek, J., M. Šejna, H. Saito, M. Sakai, and M. Th. van Genuchten, The HYDRUS-1D Software Package for Simulating the Movement of Water, Heat, and Multiple Solutes in Variably Saturated Media, Version 4.15, HYDRUS Software Series 3, Department of Environmental Sciences, University of California Riverside, Riverside, California, USA, 2012.

Plus the R language:

R Statistics 2.15.1

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Or

R Statistics 3.1

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

The versions of R that have been tested successfully with MAD# are 2.15.0-2.15.3 and 3.1.1. Only the 32-bit option of R is compatible. In addition to the base package, MAD# potentially needs these packages and will handle the installation for you:

- mvtnorm

  Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, Torsten Hothorn (2012). mvtnorm: Multivariate Normal and t Distributions. R package version 0.9-9994. URL http://CRAN.R-project.org/package=mvtnorm

- tmvtnorm

Stefan Wilhelm, Manjunath B G (2013). tmvtnorm: Truncated Multivariate Normal and Student t Distribution. R package version 1.4-8.

- msm

  Christopher H. Jackson (2011). Multi-State Models for Panel Data: The msm Package for R. Journal of Statistical Software, 38(8), 1-29. URL http://www.jstatsoft.org/v38/i08/.

- gstat

  Pebesma, E.J., 2004. Multivariable geostatistics in S: the gstat package. Computers & Geosciences, 30: 683-691.

- intervals

  Richard Bourgon (2010). intervals: Tools for working with points and intervals. R package version 0.13.3. http://CRAN.R- project.org/package=intervals

- np

  Tristen Hayfield and Jeffrey S. Racine (2008). Nonparametric Econometrics: The np Package. Journal of Statistical Software 27(5). URL http://www.jstatsoft.org/v27/i05/.

- RSQLite

  David A. James and Seth Falcon (2012). RSQLite: SQLite interface for R. R package version 0.11.2. http://CRAN.R- project.org/package=RSQLite

- corpcor

  Juliane Schafer, Rainer Opgen-Rhein, Verena Zuber, Miika Ahdesmaki, A. Pedro Duarte Silva and Korbinian Strimmer. (2012). corpcor: Efficient Estimation of Covariance and (Partial) Correlation. R package version 1.6.4. http://CRAN.R-project.org/package=corpcor

- Matrix

  Douglas Bates and Martin Maechler (2012). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.0-6. http://Matrix.R-forge.R-project.org/

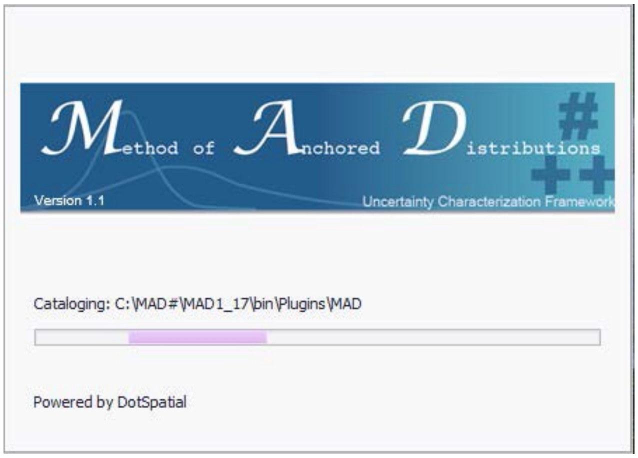
A.3 How do I Start a Project?

MAD# stores project data in databases so that information can be efficiently retrieved, so specifying a location for the databases is one of the first steps in starting a project. MAD# also needs to communicate with external forward modeling software because it is has been built modularly and is not specific to any research question. You will identify which supported software you are using for simulations and specific folders and files that are required to execute your configuration of the forward model. MAD# needs to know if the supplied forward model is transient or if it is steady-state. The last external communication issue to resolve is with the random field generator. Again, MAD# is designed to give the user options in how their project is run, so you are given the choice of a conditional random field or an unconditional one (if the generated random field honors the measurements or not). You will choose which generator is correct for your analysis and will confirm that the appropriate files are on your computer for that generator. All of this information will tell MAD# the necessary technical data so it can coordinate with external software, which allows MAD# to automate the simulation process so you do not have to manage it.

1) Before working with MAD#, the forward model needs to be built with MODFLOW96, MODFLOW2005, or HYDRUS-1D. In one of these programs, create a functioning forward model. You need to specify all the conditions that will be deterministic in the Monte Carlo simulations (boundary conditions, initial conditions, etc.). For the parameters of the forward

model that will be treated as MAD# random variables, you can enter an arbitrary (but physically possible) number like 1 for those fields because MAD# will replace these values with realizations from the random field generator. Run the forward model yourself to verify there are no errors before proceeding to MAD#. Save the forward model and record its location.

2) After building your forward model, open MAD# by finding its shortcut on your desktop or from your Start menu under the MAD folder. The loading screen looks like this:


Method of Anchored Distributions
#
++
Version 1.1
Uncertainty Characterization Framework

Cataloging: C:\MAD#\MAD1_17\bin\Plugins\MAD

Powered by DotSpatial

Figure A- 9: Loading screen MAD#

3) When MAD# first starts, the **Project** tab is open. Choose if you want to create a new project or open a previously configured project (*.xmad* file) by clicking the **New project** or **Open project** buttons, respectively. If you opened a project, skip to step 11.

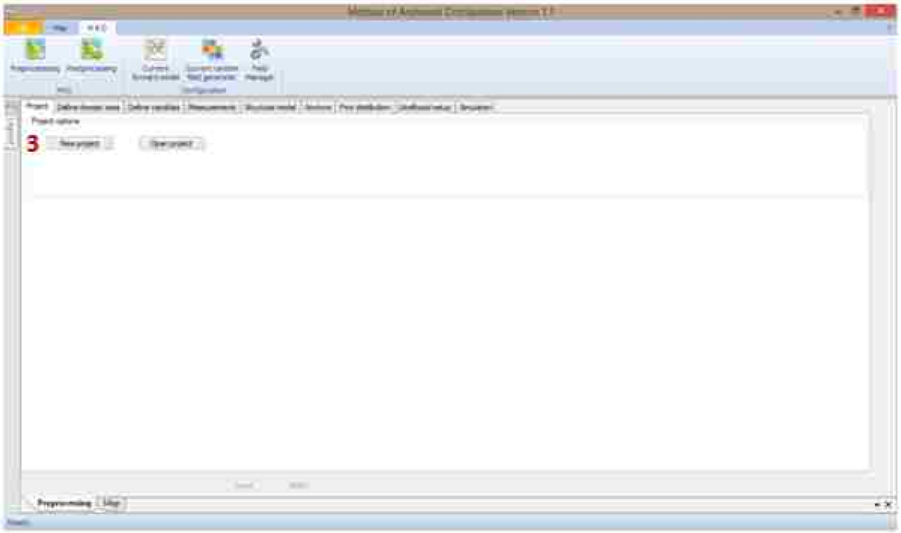


Figure A- 10: Main form of MAD#

4) When creating a new project, you need to first give your project a name and a folder in which to store the appropriate databases. You can either type in the folder path or use the Search button to locate the folder you wish to use

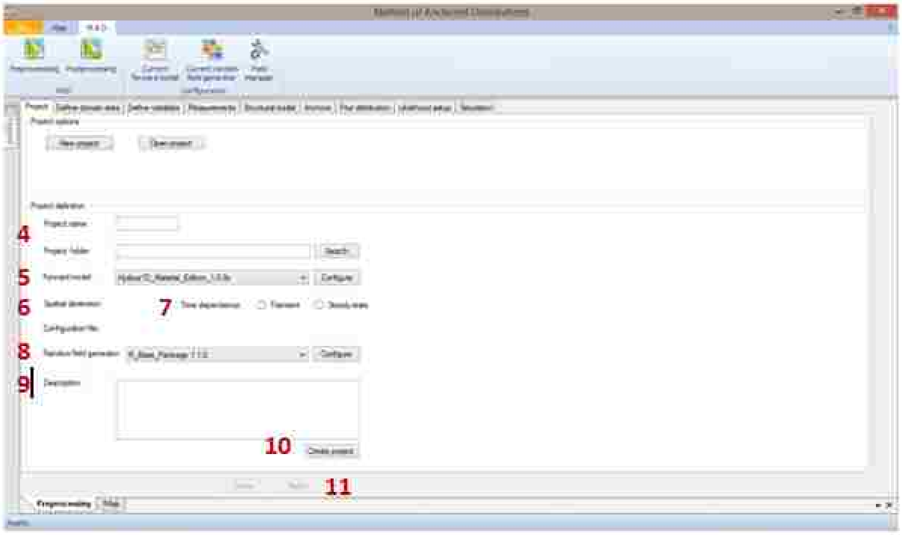


Figure A- 11: Create a new project

5) Now you can configure this project to work with the forward model mentioned in step 1. From the dropdown list, choose the software that you used to build your forward model. Click the **Configure** button next to that list. A separate window will appear, which looks slightly different for each of the forward modeling software. Here are general steps with examples for each of the supported software

   a) Specify the executable for the software. If you installed the software with the default path, then you can find the executable at the locations below (this path will be in the field already by default). If you did not install with the default path, then navigate to where you installed the program and find the .exe from below. These paths will be slightly different depending on the version downloaded.

      - HYDRUS: *C:\Program Files (x86)\PC-Progress\Hydrus-1D 4.xx\H1D_CALC.EXE*
      - MODFLOW-96: *C:\Simcore\PMWIN 5.3\modflow\lkmt1\mflowpm4v.exe*
      - MODFLOW-2005: C:\WRDAPP\MF2005.1_11\bin\mf2005.exe

   b) Then navigate to your saved forward model file from step 1. The project folder will automatically appear with the folder path of the file you chose. Here are examples of the filenames for which to look:

   - HYDRUS: *selector.in*
   - MODFLOW-96 and MODFLOW-2005: *myproject.nam*

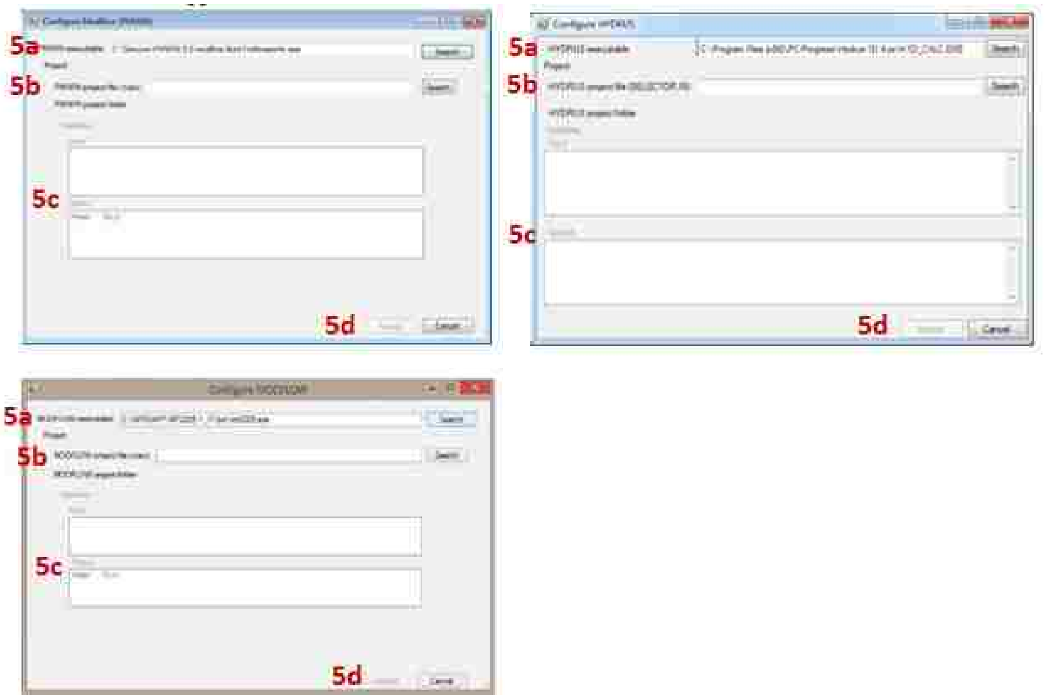


Figure A- 12: Forward model configuration forms

c) The two variable boxes will also automatically populate with the inputs and outputs of the model that you supplied.

i) If both boxes have information and it is not consistent with what you expected, then check to see if you chose the correct project file.

ii) If the input box is blank, it means the configuration was unsuccessful. Check that your forward model is functioning in the respective software (without using the MAD# interface), the executable path for the executable, and the path to the forward model folder.

d) Click **Accept**.

6) Next you will confirm the spatial dimension of the imported forward model. If the text displayed is not accurate, review the model project and/or the provided project file to see if

there any errors. For MODFLOW96 and MODFLOW2005 projects, MAD# will interpret the three spatial dimensions as x, y, and z. MAD# will interpret HYDRUS as z.

7) Indicate if the forward model is a function of time or not by clicking the radio button next to Transient or Steady-state, respectively.

8) From the dropdown list for random field generators, choose if you want to use R with gstat (conditional random fields), the R base package (unconditional random fields), or GSTAT without R. (Note: The version numbers listed are for the drivers, not the packages. Also, the R base driver is only compatible with HYDRUS. See the gstat manual entries on *vgm* and *krige* to learn more about the functions MAD# uses. When using gstat, MAD# will use a value of 10E-6 for the zero parameter to prevent a documented bug). Then click **Configure**.

If you choose the Gstat driver, MAD# will automatically search for the executable. If it is found, a window will appear.

Figure A- 13: Gstat executable

If you choose either of the R drivers (gstat or base package) a separate window will appear, and it will look the same for both drivers.

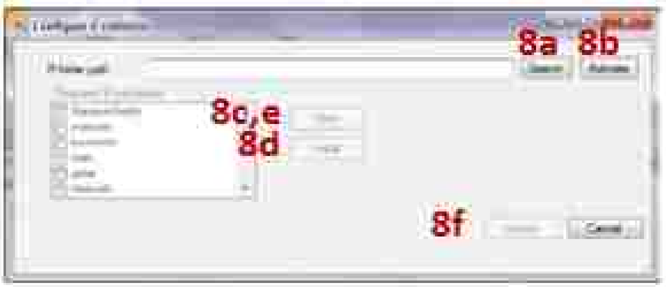


Figure A- 14: Configuration of R-Gstat driver

a) Either type in the path for R, or click on the **Search** button and navigate to where R is installed on your computer. If you installed R in the default path, the folder should be *C:\Program Files\R\R- 3.1.0*. The version number may differ depending on the version you have installed on your computer.

b) Click the **Activate** button. If the pop-up that appears confirms that the path was successfully activated, you can click **OK** and proceed to step 8c. If the pop-up says “Not found,” you should confirm that the path exists and retry.

c) Next check if MAD# can access the listed R packages by clicking the **Verify** button.

d) If all of the packages are installed, then a confirmation pop-up will appear and you can click **OK**. Proceed to step 8f. If some of the packages were not verified, a pop-up will appear indicating the number of packages missing and the **Install** button will become active. Click on the **Install** button. You will see loading bars appear and disappear as MAD# handles the installations.

e) Return to the Verify button to confirm that the installation was successful and activate the **Accept** button.

f) Click **Accept**.

9) [Optional] You can write a description for your project in the text box provided. This is useful for when you load previously made projects and the description is there to remind you what you created.

10) Make sure everything entered in the tab is correct, because proceeding with the incorrect forward model file, wrong generator, or wrong time dependence is irreversible. Once you are confident that everything is correctly entered, click the **Create project** button. This saves the information entered in an *.xmad* file with the name that you provided in the folder that you specified.

    Click the **Next>>** button to proceed to the next tab, **Define Domain Area**, where the spatial domain will be defined.

A.4 How do I Set the Spatial Domain?

Since MAD# will be automating the simulation process, it will need to translate between your forward model's spatial domain and its own. MAD# needs to be able to match the grid from the forward model; therefore the numbers of rows, columns, and layers are automatically matched between the two programs. However, MAD# cannot always interpret the physical scale of your forward model (MODFLOW96 uses generic units of length). MAD# provides a map over which you can display your spatial domain on so you can compare your grid with your physical field site. To accurately compare spatially dependent data in the Bayesian framework, the numerical grid from your forward model must match the MAD spatial domain. Currently, MAD# only supports rectangular grids with uniform density in 1D, 2D, or 3D.

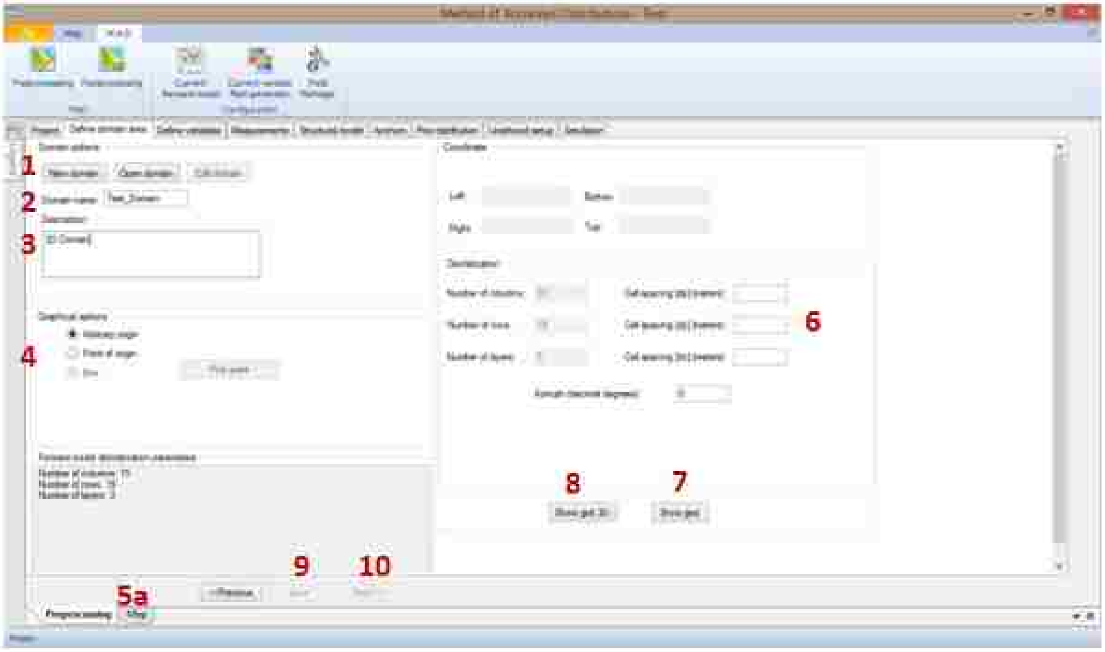


Figure A- 15: Define a spatial domain

1) If this is a new project or you want to create a new domain, click on the **New domain** button which will activate the rest of the screen and proceed to step 2.

If you loaded a previously made project in the previous tab and have a saved domain for that project, you can click the **Open domain** button to load that domain then skip to step 9. (Note: You cannot load a domain from a different project because MAD# will only search this current project's database. Also, you may edit the opened domain, but the rest of the information previously entered in subsequent tabs will need to be re-entered for the project.)

2) Enter a name for your spatial domain so that if you want to run your project again in the future, you can just load the domain.

3) [Optional] You can enter a description for this domain if you want to remind yourself when you load the domain in the future of what its purpose was.

4) If your project is synthetic, HYDRUS is your forward model, or you do not wish to provide the geographic location of your test site, select the **Arbitrary origin** radio button under **Graphical options** and then proceed to step 6. A geographic location can be specified for the spatial domain by first selecting the Point of origin button.

5) To provide a geographic location:

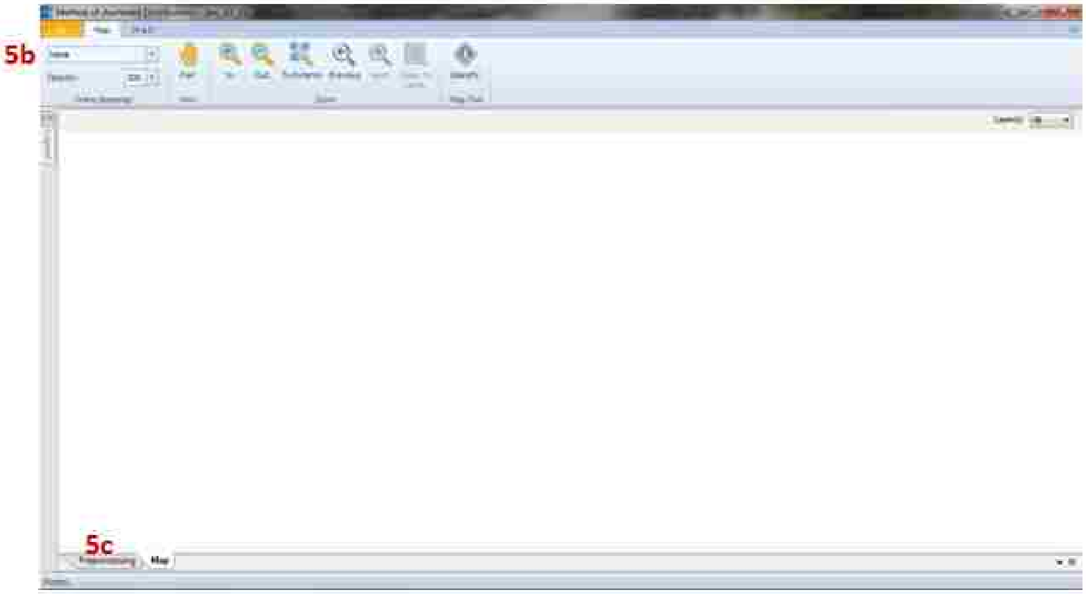


Figure A- 16: Graphical option in domain

a. First, you need to open the **Map** tab. (Note: The Map tab is movable and dockable. You can click and drag the tab to be a separate window have it consume a predefined portion of the current window by dragging your cursor to the docking icons that appear while dragging).

b. Then load a map by clicking on the dropdown menu for **Online Basemap** in the ribbon and selecting the type of map you want. (Note: You need internet access for the map to load). Use the zoom and pan buttons also in the toolbar to navigate to your field site.

c. Click on the **Preprocessing** bottom tab. You can choose the origin of your domain by clicking the **Point of origin** radio button.

Point of origin method: Click the **Pick point** button, click on the **Map** bottom tab, move your cursor to the map, and click on the location of the origin (bottom left) of your grid. You may re-click to adjust your selection. The fields for the coordinates will be populated with the appropriate coordinate information. Click on the **Preprocessing** bottom tab.

6) In the **Discretization** section the number of fields that appear is dependent on how many spatial dimensions the forward model contains. For one-dimensional forward models, only two fields will activate, for two-dimensional forward models two additional fields will activate for describing the columns, and for three-dimensional forward models six fields will activate. In the bottom left hand corner of the window, there is a **Forward model discretization parameters** box that lists the grid size that your forward model has. Currently in this release, MAD# requires the same grid size as that in your forward model and these values are populated for you.

Arbitrary origin OR Point of origin method: Because only the origin is designated, you will have to specify the cell size in your grid. For each of your forward model's spatial dimensions, enter the size of each cell in meters.

If the axes of your grid are not parallel with latitude and longitude, then you can use the **Azimuth (decimal degrees)** field to rotate your grid. Positive numbers will yield a clockwise rotation, and negatives will yield counterclockwise.

7) Click on the **Show Grid** button.

- MODFLOW96 and MODFLOW2005: In the map tab, your grid will appear and will fill the extent of the map tab. In the ribbon above, there are buttons for panning and for zooming in and out (once you select in or out, click on the map itself to zoom). The **To Extents** button will return the map to the full extent of your grid once you click the button. If you click the **Identify** button and then select any grid block (or other features that will appear after completely subsequent tabs), a tab will pop-up giving metadata about the feature. The legend can be accessed by hovering over the word **legend** printed vertically on the upper portion of the left hand side of the window. You can change the layer shown in the map using the **Layer(s)** dropdown menu, which located near the top right of window.
- HYDRUS: A third bottom tab called **Vertical** will appear and will automatically open. Zoom and pan buttons, along with a legend, will appear on the right.

You may return to steps 5 and 6 to adjust the grid until you are satisfied that it accurately represents your forward model and field site. You will not be able to adjust this domain's grid dimensions after proceeding from this step.

From this point forward, when asked to provide the row, column, or layer id-number you will provide the MAD# id-number as shown in the grid, which may differ from your forward model's id-numbers. The drivers for the different forward models are designed to translate between MAD# and the forward models so you do not have to worry about it.

Click on the **Preprocessing** bottom tab

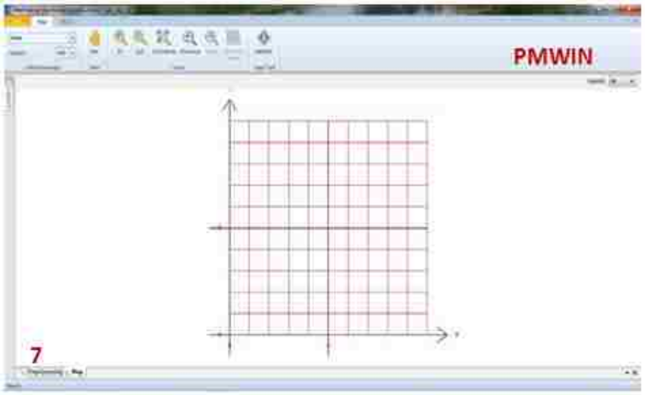


Figure A- 17: Domain generated in a PMWIN project

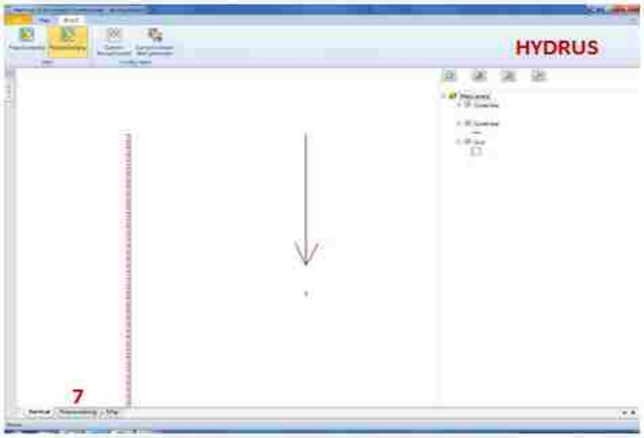


Figure A- 18: Domain generated in a HYDRUS project

8) For 3D domains, click the **Show grid 3D** button to bring up a 3D view of the domain. In order to view the domain in 3D, you must first have clicked the **Show grid** button (step 7). By default, dragging on any view of the domain while holding the left mouse button will translate the domain. To rotate the domain, first right-click somewhere in the window and select **Rotate** from the context menu. Dragging on any view of the domain will then rotate that view. To return to translation movie, select t**ranslate** from the context menu. In the context menu, click "ResetTargetPos" to return the domain origin to the center of the view.

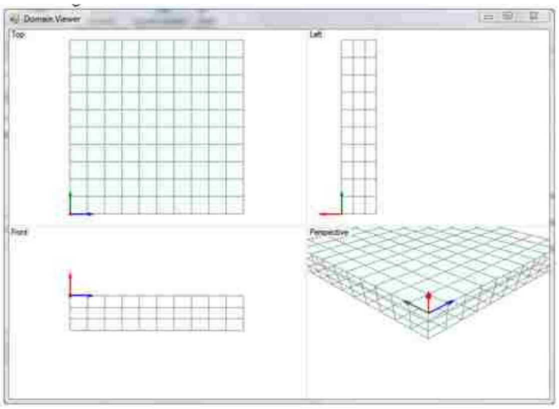


Figure A- 19: 3D domain

9) Click the **Save** button to save this domain in your project database.
10) Click the **Next>>** button to proceed to the next tab, **Define Variables**, where you can choose your variables.

A.5 How do I Define Variables?

There are two distinct variable categories in MAD#: variables that are inputs into the forward model and that will vary between realizations (called inversion target variables or Type-A variables) and variables that are extracted from those simulations and that will used for computing the likelihoods (called inversion data types or Type-B variables). This tab will provide two lists that categorize the variables from your forward model. The objective of this tab is to select the variables of your inversion. The details of when and where these variables have values will be entered in subsequent tabs

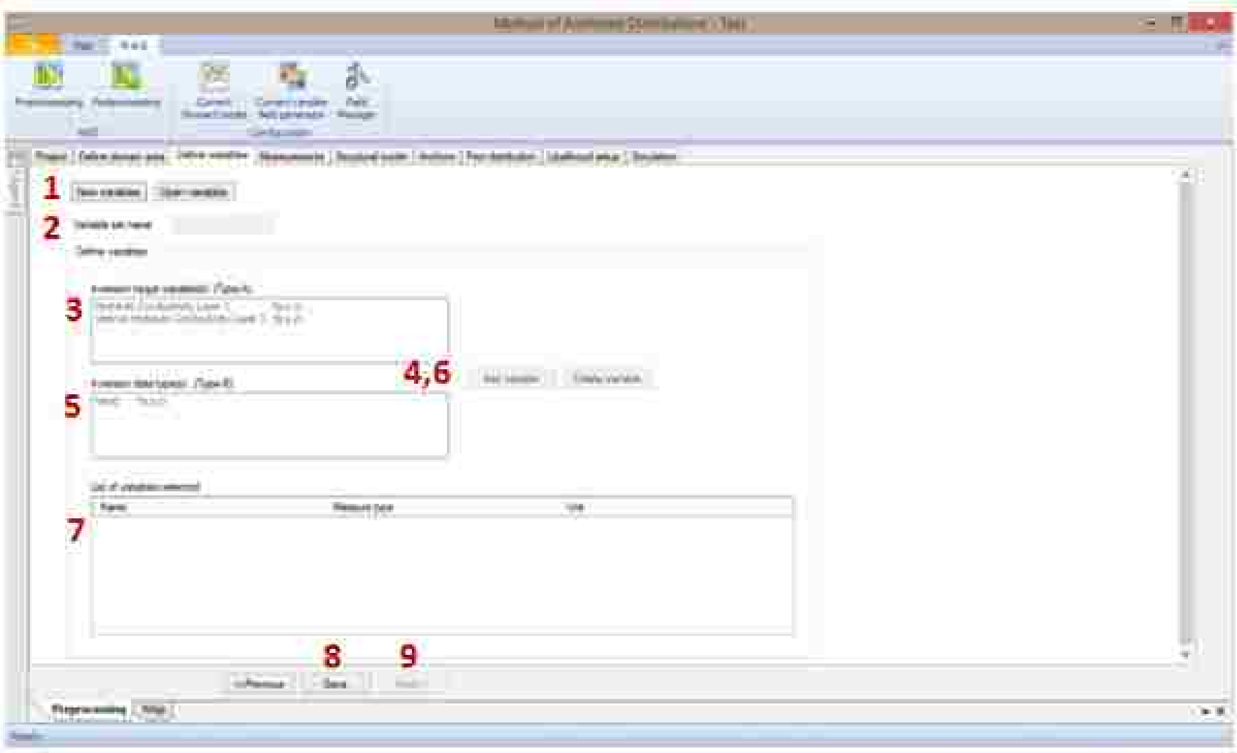


Figure A- 20: Define variables in MAD#

1) If this is a new project or you want to set new MAD# variables, click **New variables** and proceed to step 2.

If you would like to open a previously created set of variables, click the **Open variables** button, choose which set of variables to load, and click **OK**. (Note: Only the sets of variables previously defined for this particular project will be available to load because only this project's database is searched.) Skip to step 9.

2) Provide a name under which you want to save this set of variables in this project's database.

3) From the **Inversion target variable(s)** list, click on a variable that you wish to vary in the realizations.

4) Click the **Add Variable** button. Now the variable is added to the **List variables selected** table.

Return to step 3 if there are any additional inversion target variables you wish to use. (Note: At least one variable must be selected from the **Inversion target variable(s)** list.)

5) From the **Inversion data type(s)** list, click on a variable that you wish to simulate for the purpose of calculating likelihood.

6) Click the **Add Variable** button. Now the variable is added to the **List of variables selected** table.

Return to step 5 if there are any additional inversion data types you wish to use. (Note: At least one variable must be selected from the **Inversion data types(s)** list.)

7) In the **List of variables selected** table, confirm that the information listed is correct (Note: this table is automatically populated with information from the forward model, so if there are errors check to see if your forward model is configured correctly.) In the first column, there should be the variable names you selected in steps 3 and 5. In the second column, the measure type (Type A or Type B, see *Rubin et al., 2010 Section 2.1: Data classification*) is given (Note: The type is dependent on if the variable is an inversion target or inversion data type.) In the third column, the unit for the variable is provided. If the unit is "undefined" then that means that the forward model does not provide the unit (MODFLOW96 does not provide length units). If that is the case, be careful that the values you provide for this variable in subsequent tabs are consistent with the forward model.

8) Click on the **Save** button to save this variable set to the name you provided in step 2. A pop-up stating "Save successful" will appear and you can click **OK**.

9) Click on the **Next>>** button to proceed to the next tab, **Measurements**, where you will place measurements.

A.6 How do I Input Measurements?

Measurements are input in the **Measurements** tab. There are two types of data, Type-A and Type-B data (see *Rubin et al., 2010 Section A.5: Data classification* for a thorough explanation of these data types). Type-A data are measurements of your inversion target variable at locations in the spatial domain and are only supported if the gstat random field generator is used. The Type-B data are measurements for your inversion data type at locations in the spatial domain. For transient Type-B data, you have the option of importing a text file containing both the time and value of the measurement. The format of this file is discussed below, and currently a file can only hold one location's measurements. If you wish to segment or exclude any of your measurement data, you can do this later in the **Likelihood setup** tab; you do not have to alter your text files or create duplicates for multiple time series at the same location.

If your project uses the gstat random field generator, this tab will open with the **Type-A** subtab and you will complete steps 1-7 for both Type-A and Type-B measurements on their respective subtabs. To switch to the **Type-B** subtab, click on the subtab below the main tabs. If you are using the base random field generator, you will automatically start in the **Type-B** subtab and you will only do steps 1-7 on this subtab.

Entering measurements manually:

1) To add a new measurement, click on the **Insert Measurement** button.

2) Click on the blue cell below **Measurement Location Name**. Type in a unique name to identify the measurement location. For example, "borehole 1". (Note: Each location in space

should have a unique name, but you can have multiple measurements at the same location. If you have already entered one measurement for a specific location, you can enter the same name and the coordinates will automatically populate for subsequent measurements. You must either hit tab, or enter plus clicking another cell, for the coordinates to populate. Then skip to step 4.)

3) For each spatial dimension column (row, column, and/or layer), click on the cell below the header.

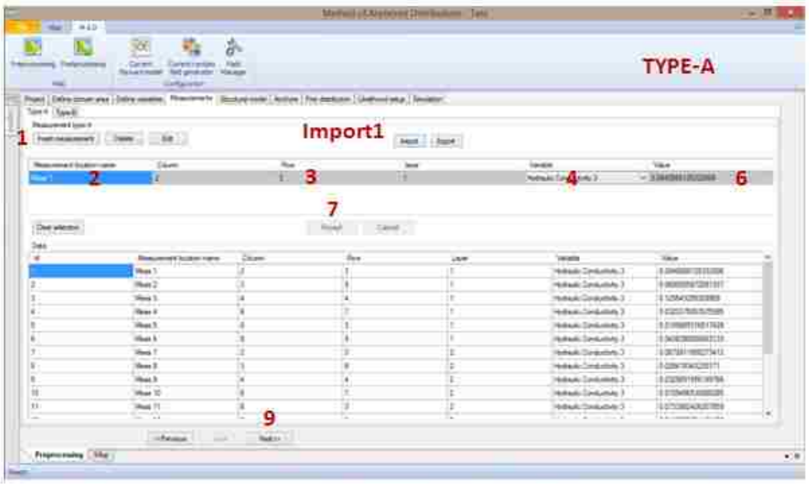


Figure A- 21: Introducing Type-A measurements

Type in the row/column/layer number (from the MAD# grid) where the measurement was taken.

4) Click on the dropdown list below **Variable** and then click on the variable that you wish to enter a value for (Note: the variables listed are the variables you designated in the Define Variables tab.)

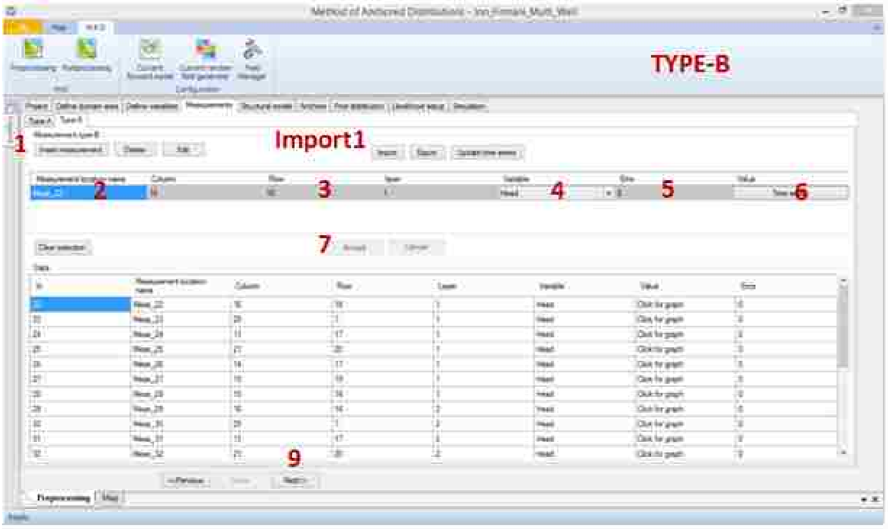


Figure A- 22: Introducing Type-B measurements

5) [For Type-B only] In the **Error** column, type in the error standard deviation associated with the device used to collect the measurement. If you do not want to include device error, type in 0. If this is a transient project, the error is assumed to be constant along the time series.

The device error is assumed to be Gaussian with a zero mean. See Appendices J and K for derivations explaining how these error standard deviations are utilized in the likelihood calculations.

6) How you enter the value of the measurement is dependent on the type of variable and time-dependence of your forward model. If your forward model is steady-state and/or this is Type-A data, click on the cell below **Value** and enter the numerical value for your measurement (Note for Type-A data: this value should <u>not</u> be transformed even if you intend on using a covariance model based on a transformation). Proceed to step 6.

If your forward model is transient and this measurement is Type-B, click on the **Time series** button under **Value**. A new **Time Series** window will appear. Follow either step 6a or 6b (not both).

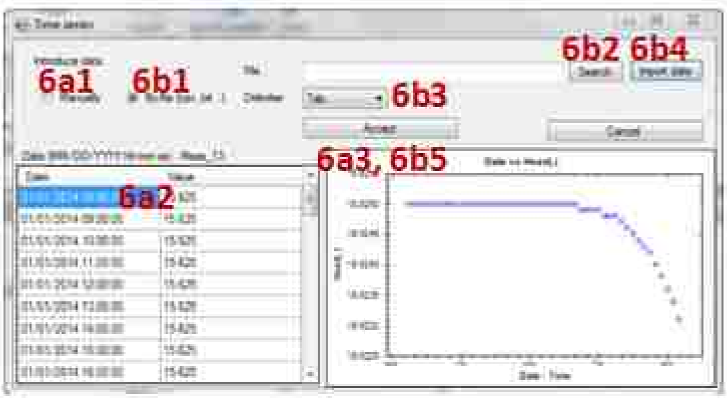


Figure A- 23: Importing Type-B data

a. Manual method:

(1) Click on the **Manually** radio button Click on the cell below **Date**, type in the date in the format MM/DD/YYYY hh:mm:ss. (Note: You may omit the hh:mm:ss and the values will be assumed as 00:00:00. Also, hours are given as 0-23 and there is no AM or PM).

(2) Click on the cell below **Value** and type in the measured value of your Type-B data for the given time. Hit Enter on your keyboard (Note: it is required to do this, don't just click the next row). Repeat entering dates and values until you have provided all data.

(3) A plot of the data will appear on the right side of the window. You can click and drag your cursor to draw a box on the graph and that box will be zoomed in for inspection. If the data looks accurate, click the **Accept** button.

b. By file method: First make sure your file is in the right format. Your file should be tab-, semicolon-, or comma- delimited and should look like this:

Date	Value
MM/DD/YYYY hh:mm:ss	0.00
MM/DD/YYYY hh:mm:ss	1.11

(Note: You may omit the hh:mm:ss and the values will be assumed as 00:00:00. Hours are given as 0-23 and there is no AM or PM. Also, the column headers are case-sensitive and the columns are tab-delimited.)

(1) Click on the **By file (csv,txt,…)** radio button.

(2) To import the data, click on the **Search** button, select the file you wish to import (make sure that it is not locked by being opened in another program, like Excel), click **OK**.

(3) Use the **Delimiter** dropdown menu to specify the delimiter used in the timeseries file.

(4) Click **Import data**.

(5) Your data will appear in the table on the left side of the window. A plot of the data will appear on the right side of the window. You can click and drag your cursor to draw a box on the graph and that box will be zoomed in upon for inspection. If the data looks accurate, click the **Accept** button.

7) Click **Accept** to add the measurement to the project.

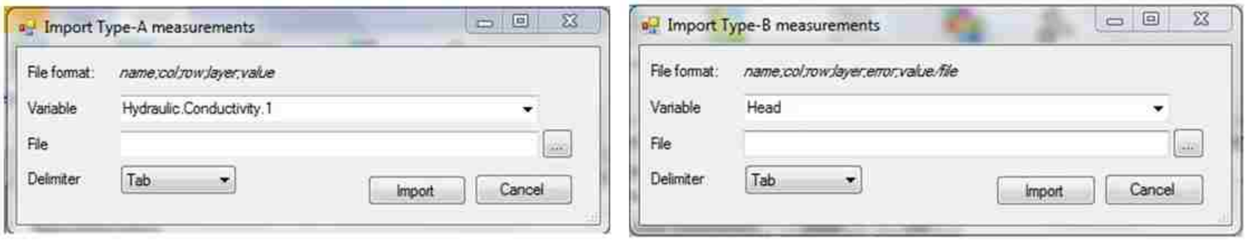
8) Repeat steps 1-7 for each measurement you would like to add. After a measurement is accepted, you can see its location in the Map tab. If you hover over the icons on the grid, or right-click them, information about the item will appear. The legend for the items on the grid is on the left edge of the Map tab and you can hover on the left edge icon for the legend to show up.

9) Click on the **Next>>** button to proceed to the next tab, **Structural model**, where you will define the structural model for the inversion target variable.

Importing measurements from a formatted .txt file

For Type A data and steady-state Type-B data, measurements can be imported directly from a single .txt file containing information about the measurement names, locations, and values. For transient models, the values of Type B data can be imported from a set of .txt files. A file containing the measurement names, locations, errors, and names of the .txt files containing the timeseries' can be imported in order to create the measurements. If necessary, each measurement time series can be updated individually.

1) Click on the **Import** button to open the **Import Measures** window. This window will look slightly different for Type A and Type B Data.



Import Type-A measurements

File format: *name;col;row;layer;value*

Variable: Hydraulic.Conductivity.1

File:

Delimiter: Tab

Import Type-B measurements

File format: *name;col;row;layer;error;value/file*

Variable: Head

File:

Delimiter: Tab

Figure A- 24: Import measurements

2) The required headers for data in the .txt file are displayed near the top of the window next to "File format:". The column headers of the text file should exactly match the headers listed here. Note that the Type B file must include a column for error (described above in step 5 of the *Entering measurements manually* section).

3) From the **Variable** drop down menu, select the variable for which the file provides values (note: the variables listed are the variables you designated in the Define Variables tab).

4) In the **File** field, you can either enter the path of the .txt file you wish to import, or use the search button to locate the file using Explorer.

5) In the **Delimiter** drop-down menu, select the delimiter used to separate the columns in the text file. Available options are Tab, Comma, or Semicolon.

6) Click **Import** to import the measurements.

7) For Type-A data and *steady-state* Type-B data, steps 1-6 are sufficient to import measurement values.

8) For *transient* Type-B data, all measurements and timeseries' can be imported by formatting the .txt files as follows:

- For each measurement location, create a delimited .txt file containing the timeseries information with the format (in this example tab-delimited but can be semicolon- or comma- delimited):

- For each measurement location, create a delimited .txt file containing the timeseries information with the format (in this example tab-delimited but can be semicolon- or comma- delimited):

```
Date          value
  01/02/2014 00:00:00        1.1
  01/03/2014 00:00:00        1.2
```

  The name of each timeseries file should indicate which measurement it corresponds to. In this example, let's call our timeseries file Meas1.txt.

- Create a delimited .txt file containing the measurement name, location (using MAD# indices) and the name of the file containing the timeseries for each measurement, with the following format (in this example tab-delimited but can be semicolon- or comma-delimited). *The delimiter used for this file must be the same as the delimiter for the timeseries files*!:

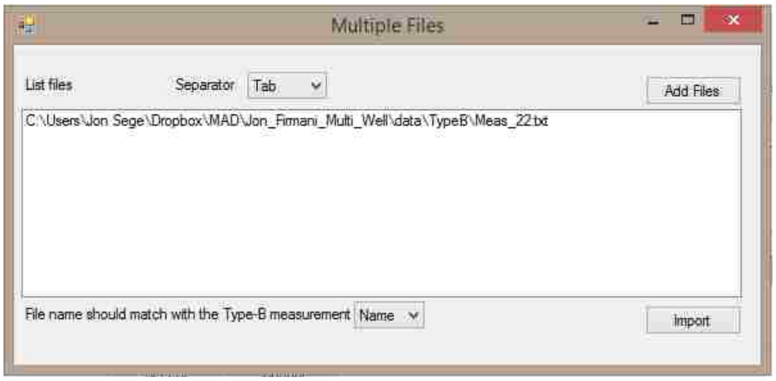
Name	Col	Row	Layer	Error	Value/File
Meas1	1	2	3	0.01	Meas1.txt

Save this file in the same folder containing all of the timeseries text files.

- Using the Type-B data import feature described in steps 1-6, import the second file (containing measurement names, locations, and timeseries files).

If any of the Type-B timeseries' need to be updated or changed, use the following procedure:

a. In the Measurements tab, click **Update time series**. A new window will appear.



Separator Tab

C:\Users\Jon Sege\Dropbox\MAD\Jon_Firmani_Multi_Well\data\TypeB\Meas_22.txt

File name should match with the Type-B measurement Name

Figure A- 25: Insert multiple time series files

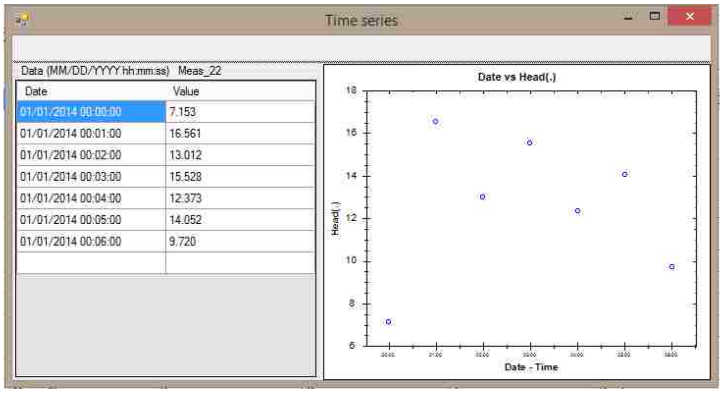
b. There are several options for formatting the time series. However, all time-series' must have the columns "Date" and "Value" (see 6b in the *Importing measurements manually* section). Time series text files may be delimited by tabs, commas, or semicolons. All imported files must have the same separator. Select your files' separator in the **Separator** dropdown menu.

c. The name of the text file (not including the extension) must exactly match either the ID or the Measurement location name of the corresponding measurement. Use the dropdown menu next to **File name should match with the Type-B measurement** to indicate whether the file names match the measurement name or ID.

d. Click **Add Files**, then navigate to the folder containing the time series files. Select the file to add, then click Open. Repeat this step for every file to be imported.

e. Make sure that all measurement time series files appear in the window under "List files." Click **Import.**

f. In the Type B Measurement tab, the value column should display the text "Click for graph." Clicking this field should bring up a window displaying a table and graph of the transient time series. Make sure that each measurement contains the appropriate time series.



Data (MM/DD/YYYY hh:mm:ss) Meas_22

Date	Value
01/01/2014 00:00:00	7.153
01/01/2014 00:01:00	16.561
01/01/2014 00:02:00	13.012
01/01/2014 00:03:00	15.528
01/01/2014 00:04:00	12.373
01/01/2014 00:05:00	14.052
01/01/2014 00:06:00	9.720

Figure A- 26: View time series

9) After measurements have been imported, they can be edited by clicking on the row in the

"Data" table and clicking the **Edit** button. After ensuring that all imported measurements are correct, click on the **Next>>** button to proceed to the next tab, **Structural model**, where you will define the structural model for the inversion target variable.

10) After measurements have been imported, they can be edited by clicking on the row in the "Data" table and clicking the **Edit** button. After ensuring that all imported measurements are correct, click on the **Next>>** button to proceed to the next tab, **Structural model**, where you will define the structural model for the inversion target variable.

A.7 How do I Define the Structural Model?

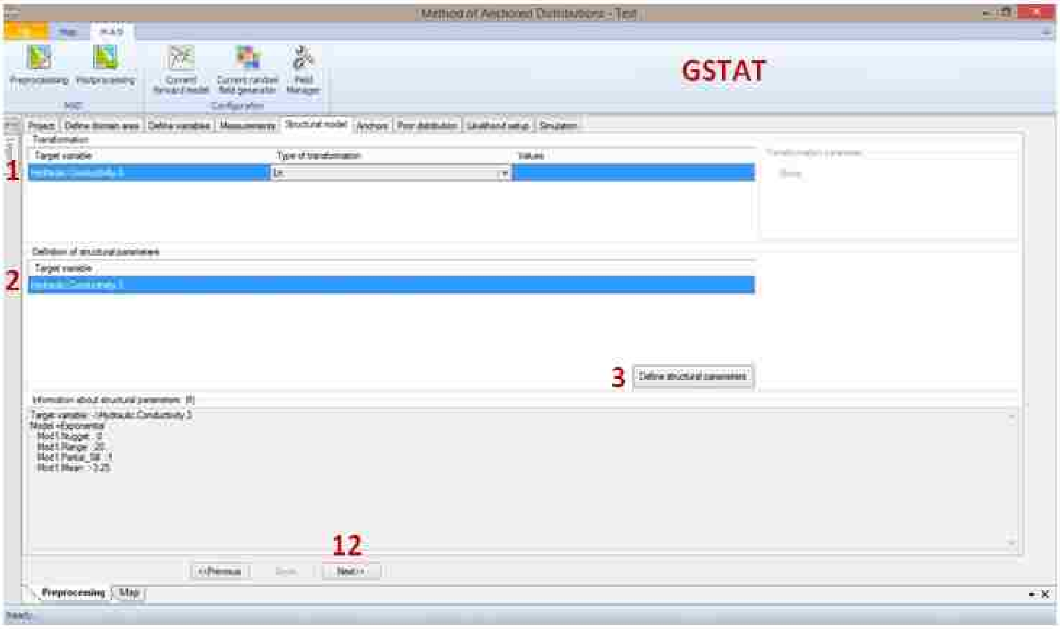
Before providing the samples of structural parameter sets, you must first define the kind of random function that describes the inversion target variable's probability distribution. Once the random function is specified, the structural parameters can be designated as either a random variable or deterministic. The model parameters in MAD# use the same definitions as Gstat (see Gstat manual for more details). Random variables will have their values altered in every batch of realizations (i.e. every batch of realizations has a different structural model) depending on the samples from the random variable's prior probability distribution. Deterministic parameters will have a value that remains constant for every batch of realizations. You have the option of designating a transform for the target variable in this tab, so that you can define a structural model for the transformed target variable. The transform will automatically be applied to the measurements provided in the previous tab.

Note: MAD distinguishes between R-Gstat and Gstat. The latter is the original package, and is in C. The former is an expansion of Gstat and has been built as a package in R. R-Gstat is currently supported by its developers and therefore has more functionality than Gstat which is not. In this section, both R-Gstat and Gstat follow the same procedures.

1) If you would like to work with a transform of your target variable(s), click on the arrow of the dropdown menu under **Type of transformation** and choose which kind of transform you want for this target variable. To the right will appear fields for each of the transform parameters (if any). Fill in these fields for your transform and click **Accept**. Repeat for each target variable you wish to transform.

2) Depending on the random field generator you used, the structural model may describe multiple target variables.

If you are using gstat or are using the base generator with only one target variable, then click on a target variable under **Target variable** in the **Definition of structural parameters** list and proceed to step 3.



Target variable : Hydraulic Conductivity 3
Model =Exponential
Mod1.Nugget : 0
Mod1.Range : 20
Mod1.Partial_Sill : 1
Mod1.Mean : -3.25

Figure A- 27: Structural parameter with R-Gstat and Gstat drivers

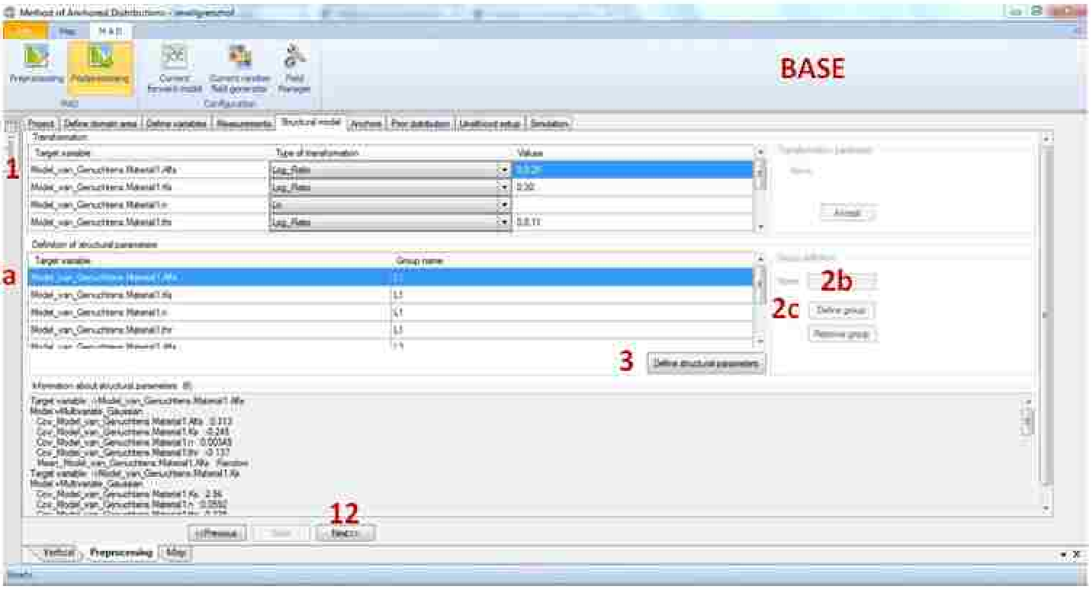


Figure A- 28: Structural parameter with Base driver

If you are using the base generator and have multiple target variables, you can group the variables together:

a. Ctrl-click the variables you wish to group.

b. Type in a name for the group under **Group definition**. (Note: Do not use spaces or special characters in the group name.)

c. Click **Define group**.

d. Repeat for each group.

2) Click the **Define structural parameters** button (Note: all variables in a group need to be selected for this button to work). A new window will appear.

3) Depending on the random field generator you chose, the structural parameters are defined differently.

In the new **Structural Parameter** window:

BASE (univariate): Click on the **Distribution type** dropdown menu arrow and click on the function type you would like. Proceed to step 5.

BASE (multivariate): Click on the **Distribution type** dropdown menu arrow and click on the function type you would like.

a. Under **Covariance matrix**, click on any variable combination for which you would like to set a covariance value. The legend for the matrix is to the right.

b. Select the **Deterministic** radio button and type in the value. Click **Accept**.

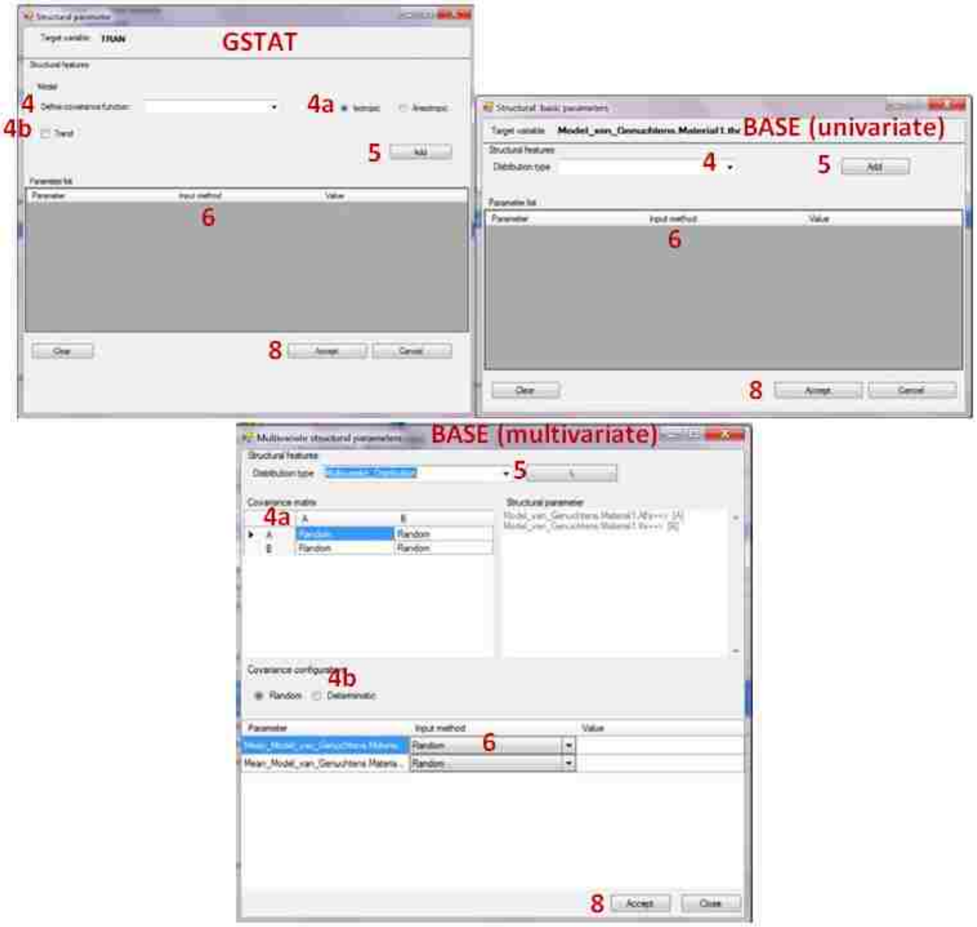


Figure A- 29: Structural parameters forms

GSTAT: Click on the arrow for the **Define covariance function** dropdown list and then click on the type of covariance function that represents the spatial heterogeneity in your target variable.

a. Choose if your target variable is isotropic or not by clicking on either the **Isotropic** or **Anisotropic** radio buttons. For anisotropic situations, the principle axis is defined in degrees clockwise from the local Y axis of the field. The anisotropy ratio is the ratio of the minor integral scale over the principal integral scale (always between 0 and 1).

b. Select if there is a trend for your target variable or not by having the **Trend** checkbox checked or not. If there is a trend, provide the **Number of coefficients**. No trend in this case means that the mean surface is spatially constant.

4) Click the **Add** button (the / button for multivariate base generator). This will populate the **Parameter List** table with the parameters affiliated with the structural model you chose.

5) Click on the first entry below **Input Method** and select if the parameter is **Random** or **Deterministic**. If you choose random for a parameter, you will have to provide a prior distribution for this parameter in the **Prior** tab. If you choose deterministic, then also click on the cell under **Value** and to the right of the current cell and type in the value of the parameter.

6) Repeat step 6 for each of the parameters in the **Parameter List**.

7) Click **Accept**.

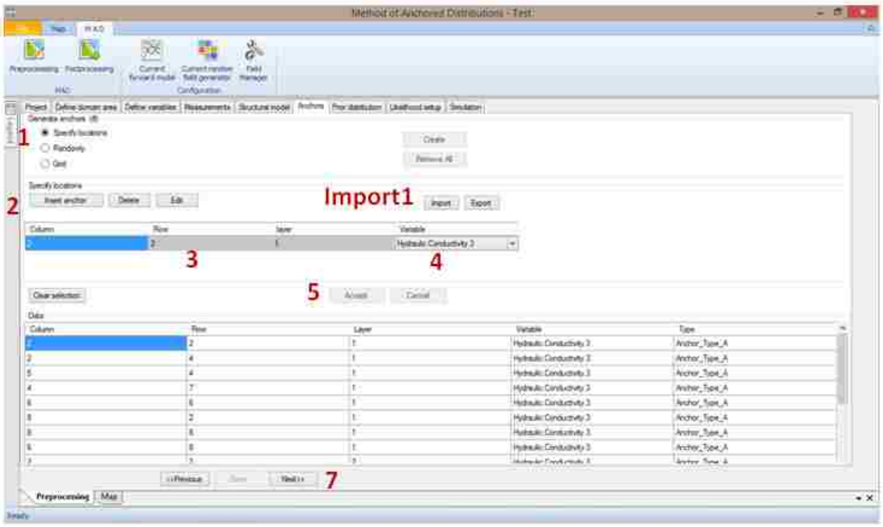
8) If necessary, repeat steps 2-8 for the remaining target variables.

9) Click **Next>>** to proceed to the next tab, **Anchors**, where you can place anchors.

A.8 How do I Add Anchors?

The structure of this tab should look familiar since it has the same basic format as the Measurements tab. Here you can place the anchors, the locations in your domain where a target variable (or some transform thereof) can be statistically characterized. (Note: Anchors are only permitted if the Gstat random field generator was chosen in the **Project** tab). Here you will provide the location of the anchor in the grid and in the **Prior distribution** tab you will provide the samples from the prior PDF for which likelihoods will be computed. See *Y. Yang et al., 2012* for a discussion on strategic placement of anchors. (Note: Multiple anchors of different variable types can be collocated).

Anchors can be added manually or imported from a formatted .txt file. First, the method for adding anchors manually will be outlined, then the steps to import them will be described.



Column	Row	layer	Variable
2	2	1	Hydraulic Conductivity 3

Column	Row	Layer	Variable	Type
2	2	1	Hydraulic Conductivity 3	Anchor_Type_A
2	4	1	Hydraulic Conductivity 3	Anchor_Type_A
5	4	1	Hydraulic Conductivity 3	Anchor_Type_A
4	7	1	Hydraulic Conductivity 3	Anchor_Type_A
6	6	1	Hydraulic Conductivity 3	Anchor_Type_A
8	2	1	Hydraulic Conductivity 3	Anchor_Type_A
8	8	1	Hydraulic Conductivity 3	Anchor_Type_A
6	8	1	Hydraulic Conductivity 3	Anchor_Type_A

Figure A- 30: Introducing anchors

Entering anchors manually:

1) If you would like to add anchors individually, select the **Specify locations** radio button and proceed to step 2.

   If you would like to add anchors randomly or uniformly, click the **Randomly** or **Grid** radio buttons, respectively. Type in the number of anchors you would like placed in the **Number of anchors** field. Click **Create** and proceed to step 7.

   (Note: If you are using PMWIN or Modflow 2005, be aware that MAD# does not check for inactive cells. Anchors should not be placed on these cells. If your project has inactive cells, be sure that anchors are not placed at those locations. See the PMWIN manual (version 5.0, page 17) for an explanation of an inactive cell.)

2) To add a new anchor, click on the **Insert Anchor** button.

3) For each spatial dimension column (row, column, and/or layer), click on the cell below the header.

   Type in the row/column/layer number (from the MAD# grid) where you would like to place the anchor.

4) Click on the dropdown list below **Variable** and then click on the variable that you want to place an anchor for (Note: the variables listed are the variables you designated as inversion target variables in the Define Variables tab).

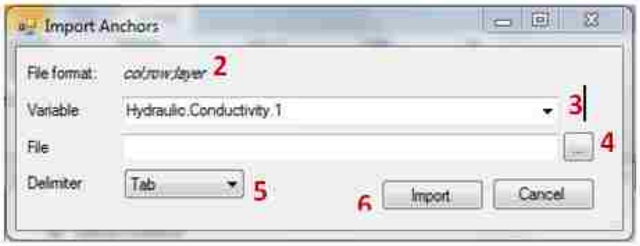
5) Click **Accept** to add the anchor to the project.

6) Repeat steps 2-5 for each anchor you would like to add.

7) Click on the **Next>>** button to proceed to the next tab, **Prior distribution**, where you will define the prior distribution for the random structural parameters and anchors.

Importing anchors from a formatted .txt file

1) Click the **Import** button. A window will appear.



Import Anchors

File format: col;row;layer

Variable: Hydraulic.Conductivity.1

File:

Delimiter: Tab

Figure A- 31: Import anchors

2) The required format of the data in the .txt file is displayed near the top of the window. The column headers of the text file should exactly match this description.

3) From the **Variable** drop down menu, select the target variable for the anchor.

4) In the **File** field, you can either enter the path of the .txt file you wish to import, or use the search button to locate the file using Explorer.

5) Use the **Delimiter** dropdown to select the delimiter used in the text file.

6) Click **Import** to import the anchors.

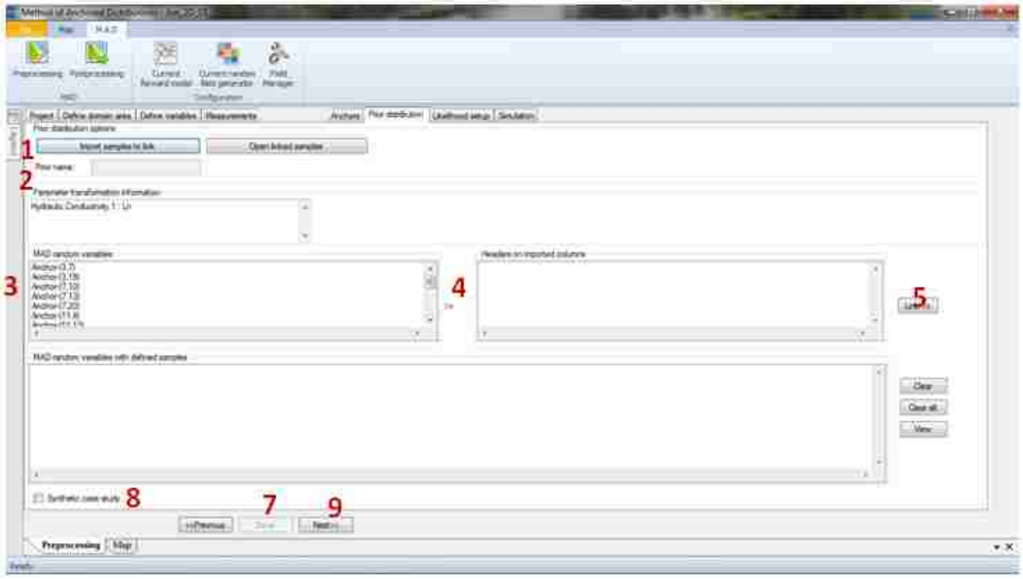
7) After anchors have been imported, they can be edited if necessary by clicking one of the anchor fields in the "Data" table and clicking the **Edit** button.

A.9 How do I Provide Prior Distributions?

In this tab, you will provide samples for each of the Type-A data structural model parameters that were designated as random in the Structural model tab and the anchors if you created any. (Note: You need at least 1 structural parameter as random or at least 1 anchor). The samples should be drawn from the joint prior distribution of the structural parameters and anchors. The samples will be used to create an ensemble of randomly generated realizations of your target variable fields that will be input to your forward model.

The objective of this tab is to define relationships between samples generated by a user from a joint prior PDF and the random variables; this ensures that MAD# can use the samples properly in the Monte Carlo process. For guidance on the number of samples to include in the file, see Over et al., 2013.

Samples can be generated by the user, then imported to MAD#, or samples can be generated using the built-in Field Manager tool. First, the steps to add user-generated a prior samples will be outlined, then the use of the Field Manager to generate samples will be described



Hydraulic Conductivity 1 : Ln

Anchor (3,7)
Anchor (3,13)
Anchor (7,10)
Anchor (7,13)
Anchor (7,20)
Anchor (11,6)

Figure A- 32: Introducing prior distributions

Adding user-generated a prior samples

1) To load previously set prior distributions, click on the **Open linked samples** button and select from the pop-up window the name of the prior distribution configuration you wish to

use. Click **OK**. Proceed to step 9.

To import your samples, click on the **Import samples to link** button and find the file containing your samples. All samples have to be in one *.txt* file and all samples must be of the same length. The samples should be in the transformed space if a transform was chosen in the **Structural model** tab and can be seen in the **Parameter transformation information** box.. The format of the file is that the first row should contain the text headers for the MAD# random variables without quotes and delimited by semi-colons. The remaining rows only have values delimited by semi-colons. An example format is given below, with sample R code that can be used to generate it. Note that this code is an example for parameters only. Anchor priors must be conditional to Type A data (if applicable) and the covariance function.

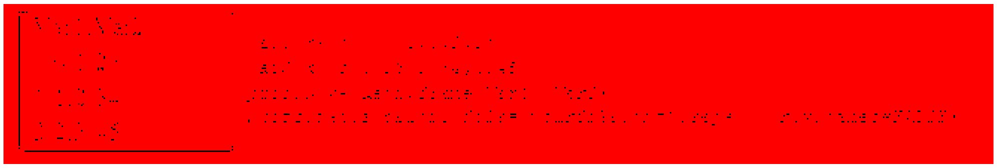


Figure A- 33: R code to export prior distributions

Below the option buttons, there are two lists. On the left, the **MAD Random Variables** lists the random variables that you have thus far defined, which need prior distributions. On the right, there is the **Headers in Imported Columns** list where the column headers from the file you imported are listed if the load was successful. Steps 4 – 7 links the two lists so MAD# can read in the prior distributions for the random variables.

2) Type in a name for this prior set in the **Prior name** textbox.

3) Click on a MAD random variable to highlight it.

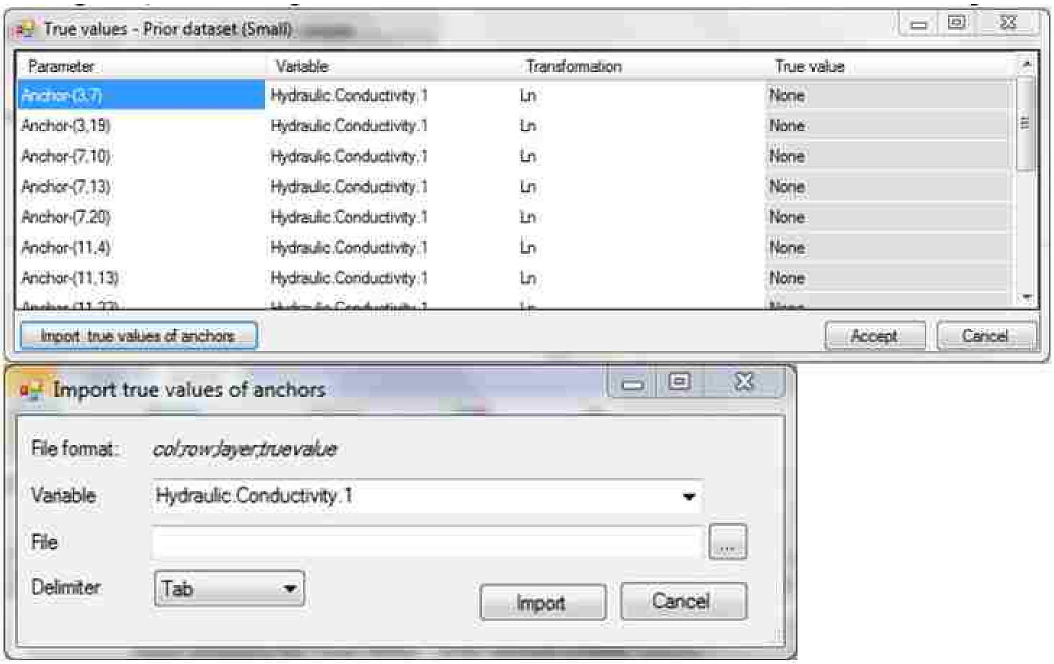
4) Click on the header in the other list that corresponds to the samples of the MAD random variable chosen in step 3

5) Click on the **Link>>** button. See that the two highlighted items in the two lists disappear and that they now show up in the **MAD random variables with defined samples** list together.

6) Repeat steps 3-5 until a relationship has been defined for every MAD# random variables in the list.

7) Click on the **Save** button. A pop-up will appear and you can click **OK**.

8) [Synthetic scenarios only, optional] Click on the **Synthetic case study** check box. A new window will appear. In the new window, type in the true values (transformed, if a transform is being used) for each respective MAD random variable under **True value**. Click **Accept**.


True values - Prior dataset (Small)
Import true values of anchors
Accept
Cancel
Import true values of anchors
Import
Cancel

Parameter	Variable	Transformation	True value
Anchor-(3,7)	Hydraulic.Conductivity.1	Ln	None
Anchor-(3,19)	Hydraulic.Conductivity.1	Ln	None
Anchor-(7,10)	Hydraulic.Conductivity.1	Ln	None
Anchor-(7,13)	Hydraulic.Conductivity.1	Ln	None
Anchor-(7,20)	Hydraulic.Conductivity.1	Ln	None
Anchor-(11,4)	Hydraulic.Conductivity.1	Ln	None
Anchor-(11,13)	Hydraulic.Conductivity.1	Ln	None

File format: *col;row;layer;truevalue*

Variable: Hydraulic.Conductivity.1

File:

Delimiter: Tab

Figure A- 34: True values synthetic case

True values can be imported by clicking **Import true values of anchors**.

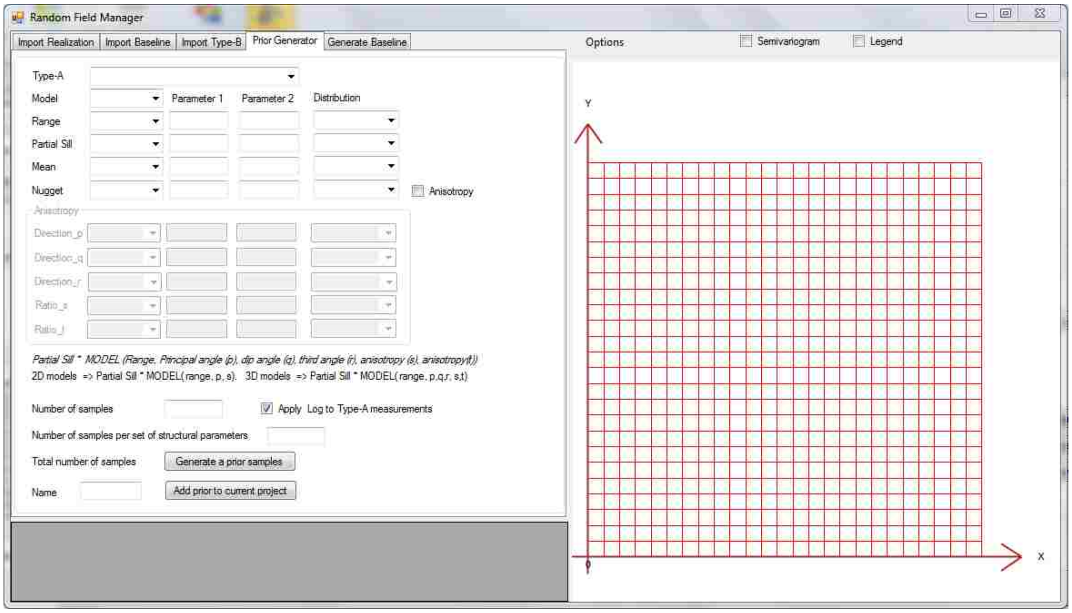
i.) Select the **Delimiter** and the location of the text file containing the true values (this text file must contain these headers: col, row, layer, and truevalue).

ii.) Click **Import**.

9) Click on **Next>>** to proceed to the next tab, **Likelihood setup**, where the measured data to use in calculating the likelihood function will be selected.

Generating and adding samples using the Field Manager tool

1) Click the **Field Manager** button Field Manager found in the toolbar ribbon near the top of the screen. A new window will appear. Click on the "Prior Generator" tab.



*Partial Sill * MODEL (Range, Principal angle (p), dip angle (q), third angle (r), anisotropy (s), anisotropy(t))*
2D models => Partial Sill * MODEL(range, p, s). 3D models => Partial Sill * MODEL(range, p,q,r, s,t)

☑ Apply Log to Type-A measurements

Figure A- 35: Field manager

2) In the Type-A dropdown, choose the Type-A variable to which prior distributions will be conditioned.

3) In the Model dropdown, choose the type of distribution that characterizes the heterogeneity of the Type-A variable.

4) For each model parameter, select whether the parameter is deterministic or random using the dropdown menu. If a random parameter, specify whether the distribution is normal or uniform and enter the two parameter values that define the distribution. In the case of a normal distribution, Parameter 1 is the mean and Parameter 2 is the variance. For a uniform distribution, Parameter 1 and Parameter 2 are the minimum and maximum values of the support, respectively. If a deterministic parameter, only the **Parameter 1** field will remain active. Enter the value of the parameter there.

5) Specify whether the field is anisotropic by checking the **Anisotropy** checkbox. If checked, the fields in the **Anisotropy** box will become active. These parameters should be entered similarly to step 4. For 2D anisotropy, the rotation angle (p) corresponds to an azimuth angle measured in degrees clockwise from the positive Y direction. The anisotropy factor (s) defines the ratio of the minor axis range to the major range. In 3D models, the first rotation angle (p) rotates the principal direction (original Y axis) clockwise in the horizontal plane. The second rotation angle (q) rotates the principal direction clockwise from the horizontal. The third angle (r) rotates the two directions orthogonal to the principal direction clockwise relative to the principal direction when looking toward the origin. The anisotropy factors (s and t) are the ranges in the minor directions divided by the major range. See the GSLIB user's manual for a discussion and illustration of anisotropy parameters.

6) The **Number of samples** field defines the number of parameter sets drawn from the model parameter distributions. If all model parameters are deterministic, this field will become inactive, since the parameter set will always be the same.

7) The **Number of samples per set of structural parameters** field specifies the number of times that random variable values should be drawn given each parameter set. If, for example, you enter 10 here, the Field Manager will draw anchor values from 10 realizations of a field described by each set of model parameters.

8) Enter a name for the a prior sample database to be generated by the Field Manager. This will allow you to open these samples and link them to the anchors in your project.

9) Click **Generate a prior samples**. The space below the **Add prior to current project** button should be populated with a table containing samples for each random parameter and/or anchor location. If an error message appears, it is possible that the model parameters entered in step 4 are not correct.

10) Click **Add** prior to current project.

11) Return to the MAD# preprocessing window and click **Open linked samples**. A new window will appear, which should list the name of the sample database generated the Field Manager (among any other previously-saved prior samples).

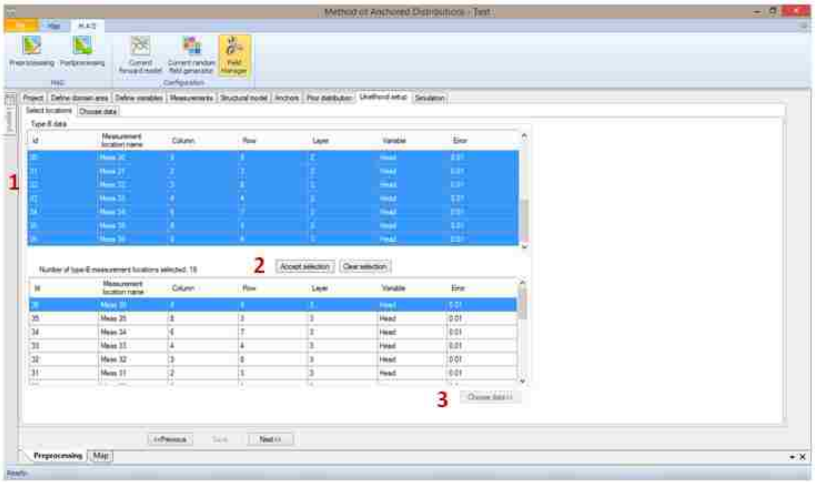
12) Click on the name of the prior dataset. A list should appear indicating the number of samples for each random variable and/or anchor location. Click **OK**. The random variables should be automatically linked to their respective headers in the prior database.

13) If this is a synthetic project, true values can be entered as described above in step 3 of *Adding user- generated a prior samples*.

14) Click on **Next>>** to proceed to the next tab, **Likelihood setup**, where the measured data to use in calculating the likelihood function will be selected.

A.10 How do I Choose the Data Used for Calculating Likelihood?

The purpose of this tab, **Likelihood setup**, is to pick the measurement data that the simulations will be compared with in order to compute the likelihood function. You can choose to either use the actual data values you supplied in the Measurements tab or you can use aggregate representations of the transient data by using regression See *Over et al., 2013* for discussions on the relationship between the cost of MAD# and how much data you choose for the **z**b vector. You do not have to use the entire time-series. Warning: Select all the information you would ever consider using to develop the likelihood function, because you can truncate your selection later on but not add to it. There must be at least one selection on this tab to proceed since there needs to be at least a datum to define the likelihood function.



Id	Measurement location name	Column	Row	Layer	Variable	Error
30	Meas 30	[illegible]	[illegible]	2	Head	0.01
31	Meas 31	2	[illegible]	[illegible]	Head	0.01
32	Meas 32	3	8	[illegible]	Head	0.01
33	Meas 33	4	4	[illegible]	Head	0.01
34	Meas 34	6	7	[illegible]	Head	0.01
35	Meas 35	8	[illegible]	[illegible]	Head	0.01
36	Meas 36	[illegible]	[illegible]	[illegible]	Head	0.01

Number of type-B measurement locations selected: 18

Id	Measurement location name	Column	Row	Layer	Variable	Error
36	Meas 36	[illegible]	[illegible]	3	Head	0.01
35	Meas 35	8	3	3	Head	0.01
34	Meas 34	6	7	3	Head	0.01
33	Meas 33	4	4	3	Head	0.01
32	Meas 32	3	8	3	Head	0.01
31	Meas 31	2	5	3	Head	0.01

Figure A- 36: Select Zb vector

1) Select all of the measurement locations from the **Type-B data** table you wish to choose data from.

2) Click on the **Accept selection** button. Below there is a table listing the locations you have selected. If your project is steady-state, skip to step 13

3) [Transient projects only] Click on **Choose data>>** to proceed to the second part of the **Likelihood setup** tab.

4) If you do not wish to use an aggregate representation of your transient data, skip to step 10.

   If you wish to aggregate, examine the plot of your temporal Type-B data. Distinguish intervals over which you would like to aggregate data. You can have one or more intervals, they can overlap, and the entire time series does not have to be covered. You can click and drag your cursor to draw a box on the graph and that box will be zoomed in upon so that you can determine which index are the data points the different intervals begin and end. Also, you can select data from the **Temporal** data list and the data point will be highlighted in the plot.

5) Click the arrow for the **Number of intervals** dropdown list and select the number of intervals you determined in step 4.

6) Click on the cell below **Start at** and type in the index of the first point in the interval. (Note: The minimum index is reported above).

7) Click on the cell below **End at** and type in the index of the last point in the interval. (Note: The maximum index is reported above).

8) Click on the cell below **Order** and type in the order of the polynomial regression to perform in the interval.

9) Repeat steps 6-8 for each of the intervals.

10) Ctrl-click the items in the **Temporal data** and **Aggregate data** lists that you would like to

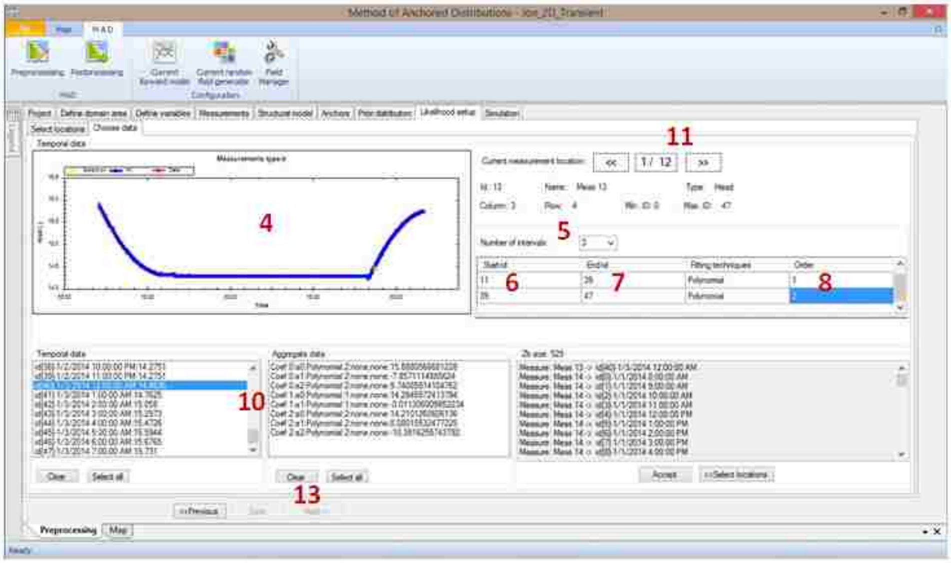
include for likelihood calculations (items from both lists can be used). You can also use the **Select all** buttons for the two lists to add all the items from the list. As items are selected, they will appear in the **z**$_b$ vector list in the bottom right hand corner of the window.

11) Click on the >> button to proceed to the next measurement or << to return to the previous measurement.

12) Repeat steps 4 – 11 until you are finished with every measurement location.

13) Click **Accept** if all locations and measurements are correct. If you need to change locations, click on <<**Select** locations to return to the measurement selection screen.

14) Click on **Next>>** to proceed to the next tab, **Simulation**, where you will start the execution of the simulations

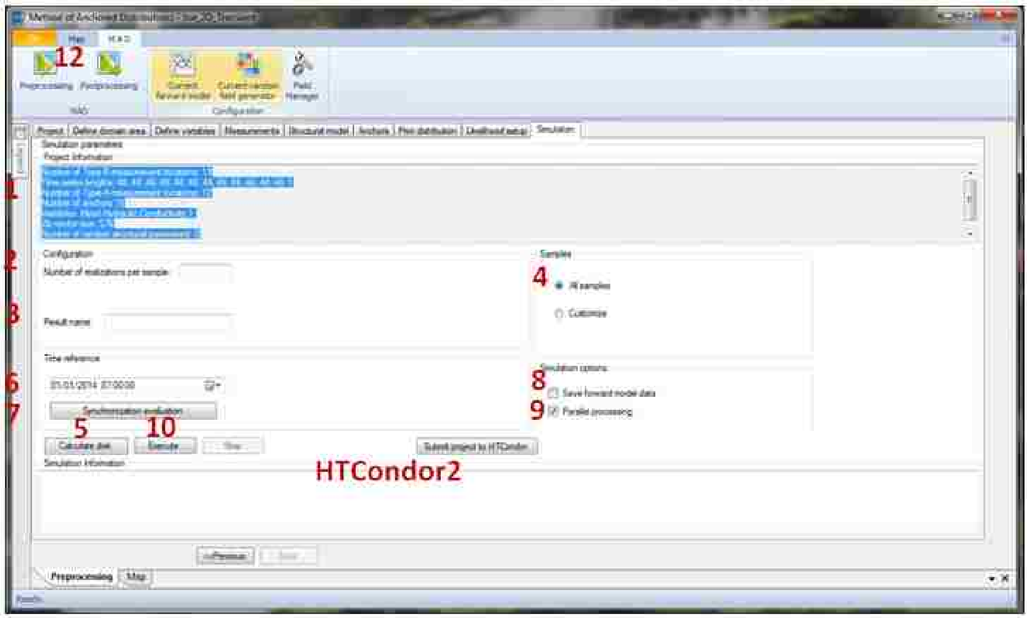


Start id	End id	Fitting techniques	Order
11	26	Polynomial	1
26	47	Polynomial	2

Figure A- 37: Polynomial approximation

A.11 How do I Run the Simulations?

Here you will start the simulation process. You will specify which samples you wish to run. It is possible to break up the samples across multiple computers to expedite the simulation process without affecting the results since the samples have already been provided. MAD# has a built-in feature which allows the user to execute simulations using an HTCondor high-throughput computing cluster. How long it takes for the simulations depends on how many samples you have, how many realizations you want, and the computation resources you have. See *Over et al., 2013* for a discussion on choosing the number of samples and realizations.



Time reference: 01/01/2014 07:00:00

● All samples

○ Customize

☐ Save forward model data

☑ Parallel processing

Figure A- 38: Run simulations

1) Review the **Simulation parameters** list for accuracy. If any of the information is unexpected, return to previous tabs to correct any errors (Note: If you make changes in a previous tab, be careful to change any subsequent tabs that may be affected).

2) Click on the field labeled **Number of realizations per sample** and type in the number of random fields of your Type-A data to create for each sample of structural model parameter combinations.

3) Click on the field labeled **Result name** and type in a name to identify this run. The resulting output file will be your project name followed by an underscore followed by this result name with the extension .xResult (for example: *exampleProject_exampleResults.xResult*). More results can be appended to these results if you use this name again.

4) Under **Samples**, select if you want to use **All Samples** or if you want to **Customize** the number of samples to use by clicking on the respective radio buttons. If you click customize, two fields will appear to the right. Enter in which samples you would like to **Start** and **End** with.

5) [Optional] Click on the **Calculate disk** button to get an estimate of the hard disk space required to store the full project

6) [Transient projects only] Provide the reference time (one time step before your measurement time series begins) in the dropdown menu by either typing in the data or using the dropdown arrow. You can scroll through the months and years to quickly find your reference time.

7) [Transient projects only; Optional] Click on the **Synchronization evaluation** button. A window containing a plot will appear like the figure below. The plot will have two timelines, a blue one representing what times your forward model will cover given the reference time you just provided in step 6 and a red one that covers the times which you provided measurements for. If the blue timeline overlaps the entire red timeline, then click **Accept**. If they don't overlap properly, check that the measurement times are correct and possibly extend the time period in the model

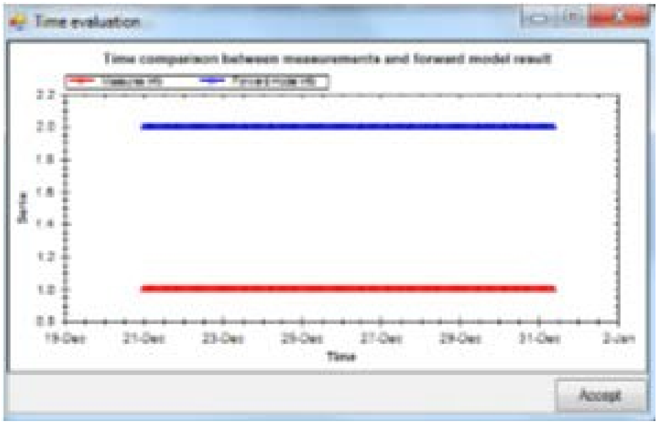


Figure A- 39: Synchronization evaluation

8) If you would like to save the full Type-B time-series at each measurement location produced in each realization of your forward model, then check the **Save Forward Model data** checkbox. This data can be seen in the post-processing part of MAD# and is stored in the individual *.xdata* files for each sample.

9) If you would like to use all cores on your machine for the simulations, then have a check in the **Parallel processing** checkbox. If unchecked, only one core will be used. (Note: it is all cores or one core, no option in between).

10) When you are certain that all information has been entered correctly, click **Execute**. A pop-up message will appear asking if you wish to delete a folder called MADTemp. If you know that you have changed anything in your forward model, or are unsure, click **Yes**. If you are certain that no changes have been made to your forward model, you may click **No**. Once you answer the question, the simulations will begin.

If you wish to stop the simulations, you can click the **Stop** button. This button will stop the program when the current realization is complete (Note: this means that the current sample will not be completed).

11) Currently, to monitor the progress of the simulations: look into the project folder and see if *.xdata* or

    .txt files are being created. Each individual *.xdata* should be growing from 4kb. The files will have numbers appended to their names and once the files for the number of samples you requested have been created the simulations are complete.

12) To start the post-processing, click on the Post-processing button in the toolbar. A new window will appear with the post-processing tabs.

Running simulations on an HTCondor pool

HTCondor is a full-featured batch system for managing compute-intensive jobs. It allows batch processes to be run over multiple machines simultaneously, while managing queueing, scheduling, prioritization, and resource monitoring. In order to run MAD# simulations using HTCondor, your machine must be configured as part of an HTCondor pool, or with the capability to flock to one. Once your machine is configured on HTCondor, you may run simulations using MAD#'s built-in HTCondor submission form.

1) Before submitting to a pool, one batch job must be run using the regular MAD# simulation screen.

    Follow steps 1-10 from the previous section, setting the **Number of realizations per sample** field to 1, checking the **Customize** radial button under **Samples**, and entering 1 in both **Start** and **End** fields. Choose a name for the simulation. This will be the name of the result files throughout the HTCondor process, as well as the final .xresult file that is created after the HTCondor processes is completed.

2) After the initial simulation is completed, click on **Submit project to HTCondor**, a new window will appear.

3) In the **Result File** field, specify the path of the .xresult file created in step 1. Either enter the file path directly in the field or use the search button to locate the file using Windows explorer.

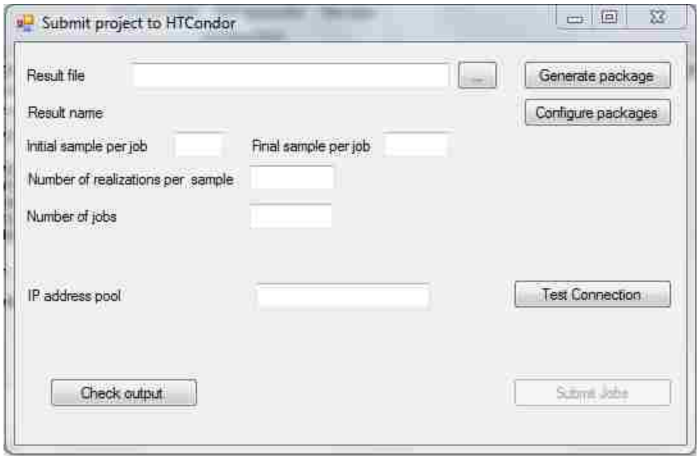


Figure A- 40: Submit project to HTCondor

4) *For R-Gstat RFG only:* If you specified R-Gstat as your random field generator in preprocessing, there will be a button **Configure packages**. Click this button to open a new window.

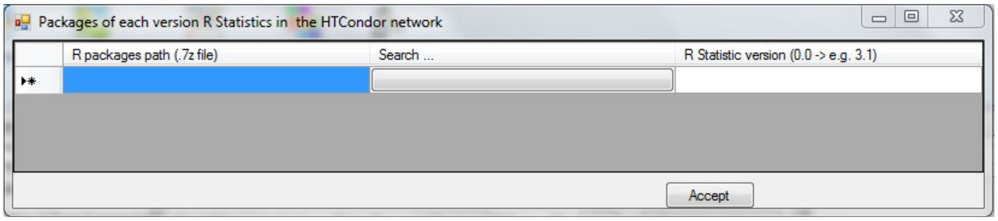

Packages of each version R Statistics in the HTCondor network
R packages path (.7z file)
Search ...
R Statistic version (0.0 -> e.g. 3.1)
Accept

Figure A- 41: Configuring packages for R based drivers

In order to run simulations on HTCondor when R-Gstat or R-Base is the random field generator, the necessary R packages must be provided in a .7z (7-zip) archive. A .7z archive containing all packages for the R-Gstat driver using R-3.1 is available for download on Codeplex. Enter the path of the .7z archive containing the necessary packages in the **R packages path (.7z file)** field, or locate the archive using the **Search** button.

- In the **R Statistic version** field, enter the first two numbers of the version of R you are using.
  For example if you are using 2.1.5, enter 2.1.

Click **Accept**

5) Before running a simulation in HTCondor, a folder must be created which contains several files required for the batch processes to be executed on the pool. Click on **Generate package** to have MAD# automatically generate the required folder. A window may appear warning that a folder will be deleted, and asking if you want to continue. Click **Yes**. This process may take some time. When it completes, a window will appear saying "HTCondor package generated successfully." Click **OK**.

6) HTCondor breaks your total simulation processing workload into a specified number of jobs. Each job consists of a specified number of realizations for a specified number of samples. In the **Initial sample per job** field, identify the number of the first sample you

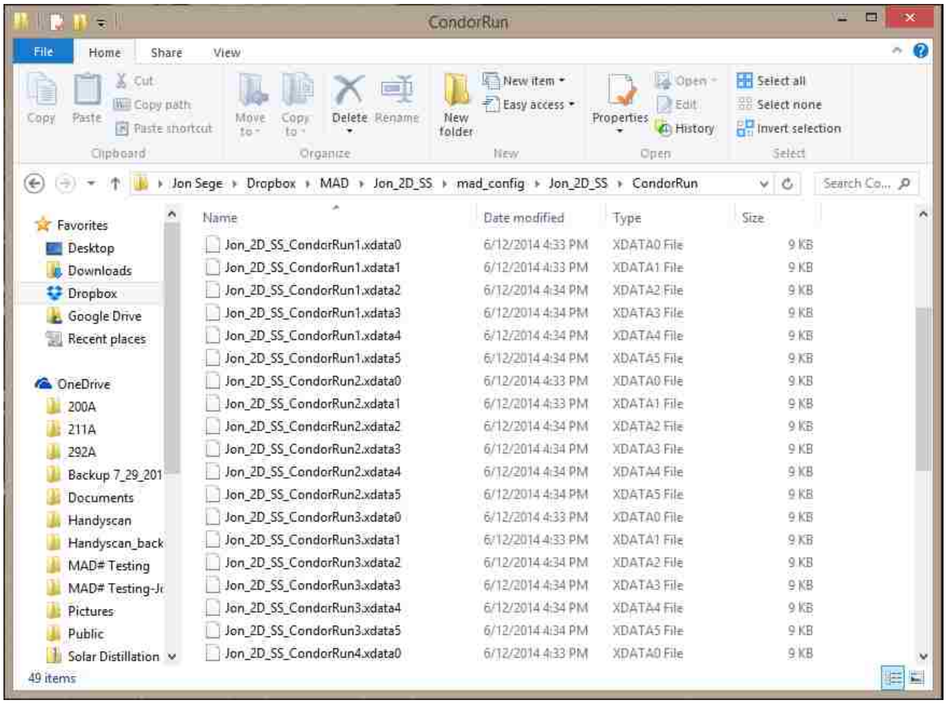
want to run for each job. In the **Final sample per job** field, identify the number of the final sample you want to run for each job. For example, if you want to run 5 samples in each job, starting with sample 1, you could enter 1 in the **Initial sample per job** field, and 5 in the **Final sample per job** field. If you wanted to run 5 samples starting with sample 5, you could enter 5 in the **Initial sample per job field**, and 10 in the **Final sample per job field**.

7) In the **Number of realizations per sample** field, enter the number of realizations you would like to run for each sample.

8) In the **Number of jobs** field, enter the total number of batch processes you want to run, which should be equal to the total number of samples you want to run divided by the number of samples per job. For example, if you want to run 1000 samples, and you specified 5 samples per job, then the number of jobs should be 200.

9) As the HTCondor process progresses, .xdata files will be added to the result folder (the results folder is in the project folder and has the same name as the **Result name** specified in the **Simulation** tab). You can also check the progress of the simulations using the *condor_q* and *condor_status* commands in the Windows command line.

10) After the simulations are complete (*condor_q* shows no more of the submitted jobs running, pending, idle, etc.), click on **Check output**. This will bring up a window showing the results of a *condor_q* command, and will also rename the .xdata files in the result folder. After clicking **Check output**, the files in the result folder should appear similar to the following.

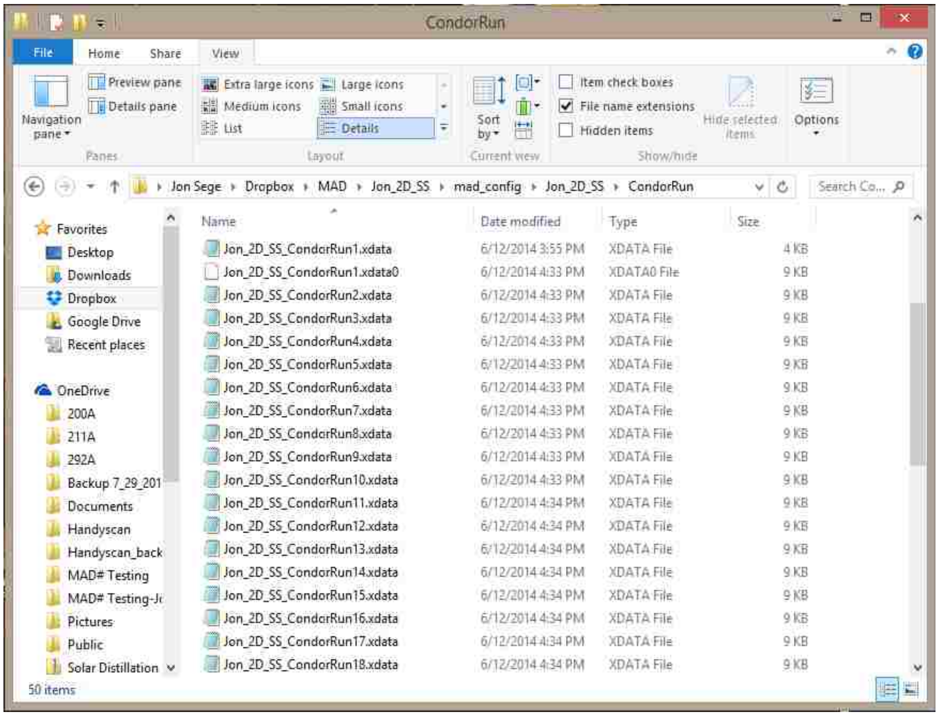


Jon Sege ▸ Dropbox ▸ MAD ▸ Jon_2D_SS ▸ mad_config ▸ Jon_2D_SS ▸ CondorRun

Name	Date modified	Type	Size
Jon_2D_SS_CondorRun1.xdata0	6/12/2014 4:33 PM	XDATA0 File	9 KB
Jon_2D_SS_CondorRun1.xdata1	6/12/2014 4:33 PM	XDATA1 File	9 KB
Jon_2D_SS_CondorRun1.xdata2	6/12/2014 4:34 PM	XDATA2 File	9 KB
Jon_2D_SS_CondorRun1.xdata3	6/12/2014 4:34 PM	XDATA3 File	9 KB
Jon_2D_SS_CondorRun1.xdata4	6/12/2014 4:34 PM	XDATA4 File	9 KB
Jon_2D_SS_CondorRun1.xdata5	6/12/2014 4:34 PM	XDATA5 File	9 KB
Jon_2D_SS_CondorRun2.xdata0	6/12/2014 4:33 PM	XDATA0 File	9 KB
Jon_2D_SS_CondorRun2.xdata1	6/12/2014 4:33 PM	XDATA1 File	9 KB
Jon_2D_SS_CondorRun2.xdata2	6/12/2014 4:34 PM	XDATA2 File	9 KB
Jon_2D_SS_CondorRun2.xdata3	6/12/2014 4:34 PM	XDATA3 File	9 KB
Jon_2D_SS_CondorRun2.xdata4	6/12/2014 4:34 PM	XDATA4 File	9 KB
Jon_2D_SS_CondorRun2.xdata5	6/12/2014 4:34 PM	XDATA5 File	9 KB
Jon_2D_SS_CondorRun3.xdata0	6/12/2014 4:33 PM	XDATA0 File	9 KB
Jon_2D_SS_CondorRun3.xdata1	6/12/2014 4:33 PM	XDATA1 File	9 KB
Jon_2D_SS_CondorRun3.xdata2	6/12/2014 4:34 PM	XDATA2 File	9 KB
Jon_2D_SS_CondorRun3.xdata3	6/12/2014 4:34 PM	XDATA3 File	9 KB
Jon_2D_SS_CondorRun3.xdata4	6/12/2014 4:34 PM	XDATA4 File	9 KB
Jon_2D_SS_CondorRun3.xdata5	6/12/2014 4:34 PM	XDATA5 File	9 KB
Jon_2D_SS_CondorRun4.xdata0	6/12/2014 4:33 PM	XDATA0 File	9 KB

49 items

Figure A- 42: HTCondor Output



□ Item check boxes
☑ File name extensions
□ Hidden items

Jon Sege ▸ Dropbox ▸ MAD ▸ Jon_2D_SS ▸ mad_config ▸ Jon_2D_SS ▸ CondorRun

Name	Date modified	Type	Size
Jon_2D_SS_CondorRun1.xdata	6/12/2014 3:55 PM	XDATA File	4 KB
Jon_2D_SS_CondorRun1.xdata0	6/12/2014 4:33 PM	XDATA0 File	9 KB
Jon_2D_SS_CondorRun2.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun3.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun4.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun5.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun6.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun7.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun8.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun9.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun10.xdata	6/12/2014 4:33 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun11.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun12.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun13.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun14.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun15.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun16.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun17.xdata	6/12/2014 4:34 PM	XDATA File	9 KB
Jon_2D_SS_CondorRun18.xdata	6/12/2014 4:34 PM	XDATA File	9 KB

50 items

Figure A- 43: Rename output files generated by HTCondor

11) The results are now ready for post-processing, which will be described next. To start post-processing, click on the **Post-processing** button in the toolbar. A new window will appear with the post-processing tabs.

A.12 How do I Start Post-processing?

This is the first step after the simulations have all be calculated, and the only purpose of this tab is to open the file those simulations are saved in. You may choose to open results from one simulation, or you may choose to examine results from multiple simulations in order to compare differences between runs.

When opening the results from one simulation, the appearance of the tab is the same as the **Project** tab from the beginning of your project. That familiar form will populate with the data you entered earlier once you open your results.

If you want to compare results from different runs, you may draw priors and posteriors from multiple .xresult files on the same plot using the **Compare multiple runs** feature. In order to do this, likelihoods must first be calculated for the different simulations following the steps outlined in the following sections.

Opening one result file

1) Click the **Open result** button. From the pop-up window, find the results file. This file will be your project name followed by an underscore followed by the result name you provided in the **Simulation** tab with the extension *.xresult* (for example: *exampleProject_exampleResults.xresult*). Click **OK**.

2) If the preprocessing steps were carried out on the current machine, then the random field generator should be automatically configured. However, if you are opening the result file on

a different machine, then you will need to reconfigure the random field generator.

a. If the Gstat random field generator was selected, a popup may appear saying "Please activate R statistics." Click **OK**. In the **R folder path** field, either type in the file path where R is located on your computer, or use the search button to browse to the R folder (Note: Select the folder corresponding to the version of R you have. Don't select the executable. For example, the file path may be C:\Program Files\R\R-2.15.2\). Click **Active**. A popup should appear saying "Successfully activated."

   i. Click **Verify** to check that all required packages are installed. A message will appear notifying you that all packages are installed, or, if they are not installed, MAD# will assist in installing them for you.

3) Below, you will see the same information that you entered in the very beginning of creating your project while in the **Project** tab. Confirm this information is correct. (Note: Here is where it is helpful to have written that description).

4) Click **Next>>** to proceed to the next tab, **Data organization**, where you will select the measured data that the simulations will be compared to for calculating likelihood.

Opening results from multiple runs

If you have already calculated likelihoods in several .xresult files, you may choose to compare results by plotting several priors and posteriors on the same axes. In order to follow this procedure, you must already have completed the steps in the subsequent sections in order to calculate likelihoods in the desired result files.

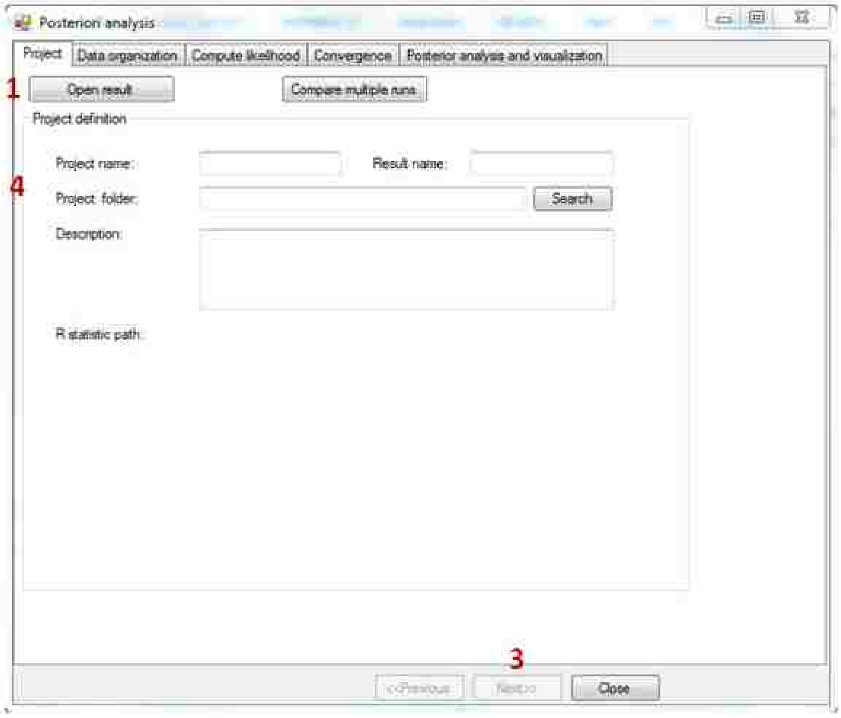
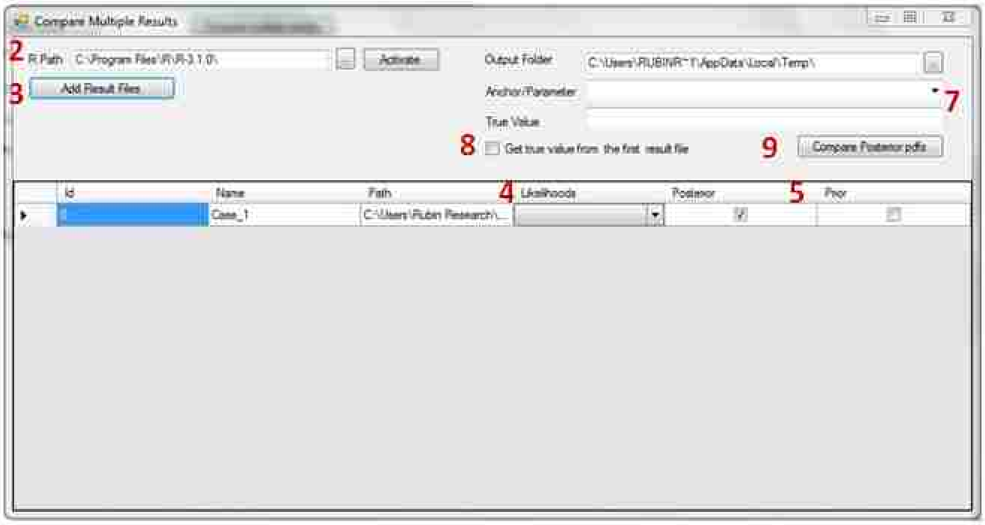


Figure A- 44: Post processing form

1) Click **Compare multiple runs**. A new window will appear.



R Path: C:\Program Files\R\R-3.1.0\

Output Folder: C:\Users\RUBINR~1\AppData\Local\Temp\

Id	Name	Path	Likelihoods	Posterior	Prior
	Case_1	C:\Users\Rubin Research\...		☑	□

Figure A- 45: Compare multiple results

2) First, verify the location of the R program folder, either by entering it into the **R Path** field or clicking the search button to locate it using windows explorer. Click **Activate**.

3) After activating R, the **Add Result Files** button will become active. Click to bring up a Windows explorer screen, in which you can select a .xresult file containing one set of results you wish to compare. Repeat this step for all .xresult files you wish to include.

4) As you add .xresult files, entries will appear in the table, with the Id, Name, and Path fields automatically filled. In the "Likelihoods" column, select the name of the likelihood vector that you wish to use to draw the posterior distribution for each result (note: the likelihood calculation is explained in the section **How do I calculate the likelihood?**). For a discussion on which likelihood vector to use to draw posteriors, see the section **How do I test for convergence?**

5) Use the checkboxes in the "Posterior" and "Prior" columns to indicate whether you wish to display the posterior distribution, the prior distribution, or both from each result. Note that you can add posteriors and priors from all results, but that adding more curves will result in the plot appearing more crowded.

6) MAD creates text files with X and Y data for each prior and posterior distribution drawn. To specify where these text files should be stored, enter a file path in the **Output Folder** field. By default, these files will be stored in the local Temp folder.

7) In the **Anchor/Parameter** dropdown menu, specify the anchor or random parameter for which you want to see distributions.

8) The **True Value** field will automatically populate from information stored in the .xresult file. If this information differs from file to file, and you wish to use the true value from the first .xresult file, check the box next to **Get true value from the first result file**.

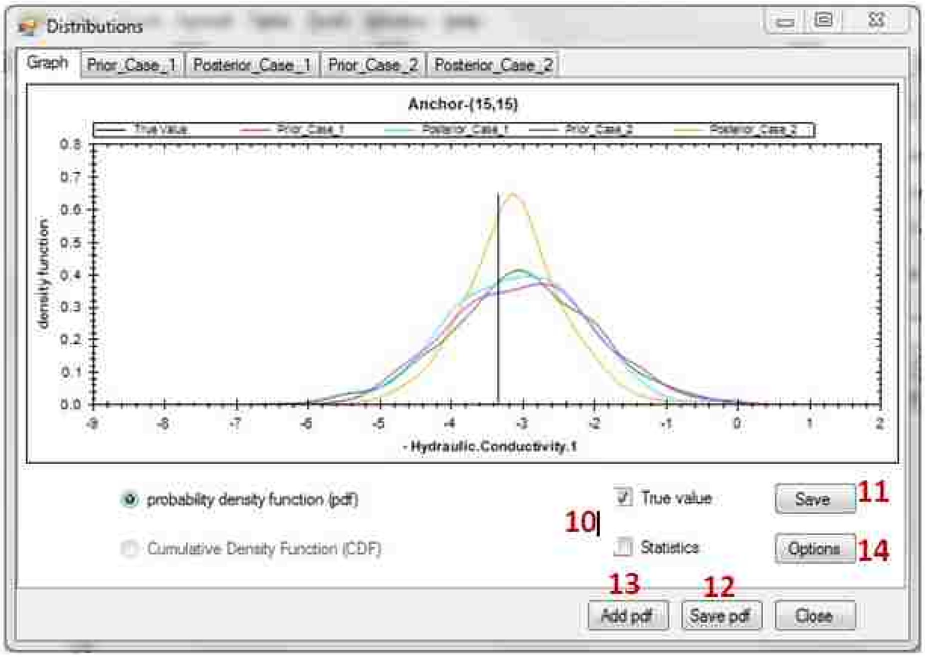


Figure A- 46: Posterior distributions comparison

9) Click **Compare Posterior PDFs**. A separate window will appear with the requested distributions displayed on the same axes.

10) You can display the true value (if applicable) and statistics about the distributions by checking the respective boxes.

11) You can save the entire plot as an image by clicking **Save.**

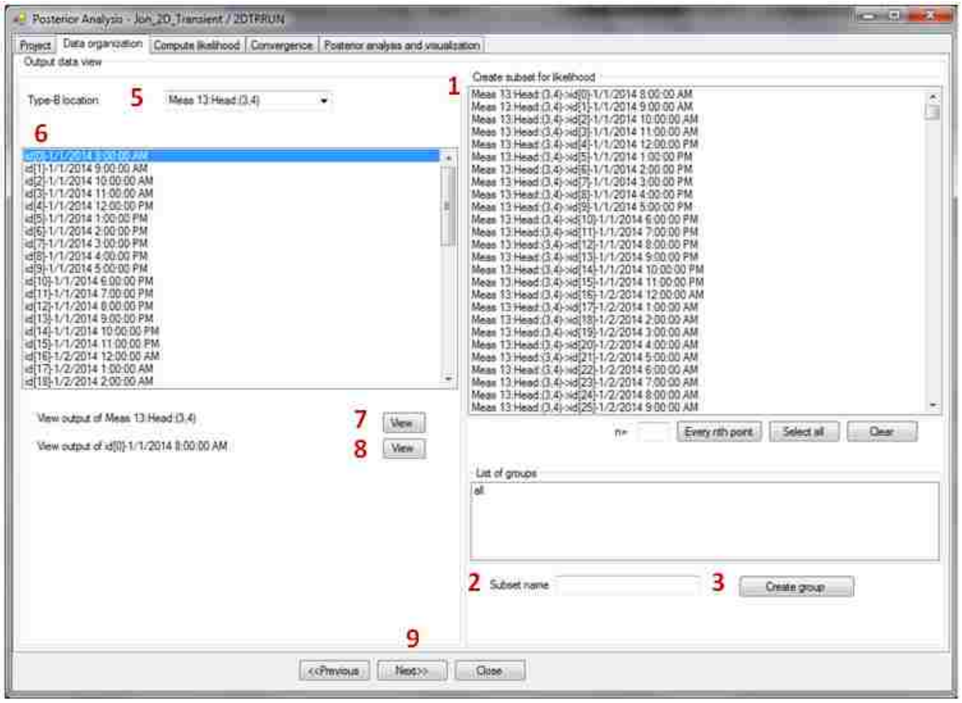
12) You can save any of the individual PDFs separately in a MAD# readable file by clicking **Save pdf**.

13) If you have a previously saved PDF that you'd like to add to the current plot (note that it must be in .mpdf format), you may do so by clicking **Add pdf**.

14) You can adjust additional plotting options by clicking **Options.**

A.13 How do I Pick the Data that the Simulations will be Compared to?

In this tab, you will see a list of the data (Type-B measurement values and/or aggregate data from those measurements) that you selected in the **Likelihood setup** tab before you ran the simulations. You will select which combinations of data from this list that you would like to use for comparing the simulation data to in order to calculate the likelihood (commonly this is will be the data stored). You are given the option of not using this entire list so that you may experiment with different combinations of your data, which is useful for comparing the condition value of different likelihood functions.



Type-B location: Meas 13:Head:(3,4)

```
id[0]-1/1/2014 8:00:00 AM
id[1]-1/1/2014 9:00:00 AM
id[2]-1/1/2014 10:00:00 AM
id[3]-1/1/2014 11:00:00 AM
id[4]-1/1/2014 12:00:00 PM
id[5]-1/1/2014 1:00:00 PM
id[6]-1/1/2014 2:00:00 PM
id[7]-1/1/2014 3:00:00 PM
id[8]-1/1/2014 4:00:00 PM
id[9]-1/1/2014 5:00:00 PM
id[10]-1/1/2014 6:00:00 PM
id[11]-1/1/2014 7:00:00 PM
id[12]-1/1/2014 8:00:00 PM
id[13]-1/1/2014 9:00:00 PM
id[14]-1/1/2014 10:00:00 PM
id[15]-1/1/2014 11:00:00 PM
id[16]-1/2/2014 12:00:00 AM
id[17]-1/2/2014 1:00:00 AM
id[18]-1/2/2014 2:00:00 AM
```

View output of Meas 13:Head:(3,4)

View output of id[0]-1/1/2014 8:00:00 AM

```
Meas 13:Head:(3,4)->id[0]-1/1/2014 8:00:00 AM
Meas 13:Head:(3,4)->id[1]-1/1/2014 9:00:00 AM
Meas 13:Head:(3,4)->id[2]-1/1/2014 10:00:00 AM
Meas 13:Head:(3,4)->id[3]-1/1/2014 11:00:00 AM
Meas 13:Head:(3,4)->id[4]-1/1/2014 12:00:00 PM
Meas 13:Head:(3,4)->id[5]-1/1/2014 1:00:00 PM
Meas 13:Head:(3,4)->id[6]-1/1/2014 2:00:00 PM
Meas 13:Head:(3,4)->id[7]-1/1/2014 3:00:00 PM
Meas 13:Head:(3,4)->id[8]-1/1/2014 4:00:00 PM
Meas 13:Head:(3,4)->id[9]-1/1/2014 5:00:00 PM
Meas 13:Head:(3,4)->id[10]-1/1/2014 6:00:00 PM
Meas 13:Head:(3,4)->id[11]-1/1/2014 7:00:00 PM
Meas 13:Head:(3,4)->id[12]-1/1/2014 8:00:00 PM
Meas 13:Head:(3,4)->id[13]-1/1/2014 9:00:00 PM
Meas 13:Head:(3,4)->id[14]-1/1/2014 10:00:00 PM
Meas 13:Head:(3,4)->id[15]-1/1/2014 11:00:00 PM
Meas 13:Head:(3,4)->id[16]-1/2/2014 12:00:00 AM
Meas 13:Head:(3,4)->id[17]-1/2/2014 1:00:00 AM
Meas 13:Head:(3,4)->id[18]-1/2/2014 2:00:00 AM
Meas 13:Head:(3,4)->id[19]-1/2/2014 3:00:00 AM
Meas 13:Head:(3,4)->id[20]-1/2/2014 4:00:00 AM
Meas 13:Head:(3,4)->id[21]-1/2/2014 5:00:00 AM
Meas 13:Head:(3,4)->id[22]-1/2/2014 6:00:00 AM
Meas 13:Head:(3,4)->id[23]-1/2/2014 7:00:00 AM
Meas 13:Head:(3,4)->id[24]-1/2/2014 8:00:00 AM
Meas 13:Head:(3,4)->id[25]-1/2/2014 9:00:00 AM
```

List of groups: all

Figure A- 47: Data output comparison

1) In the **Create subset for likelihood** list, there is listed all of the data (actual measurements and/or aggregate data) from all locations that you chose as potential data to use for

comparing to simulations. You may ctrl-click any combination of these data or use the **Select all** button to select the data for which you want to create a subset. You may also specify an interval for defining data to be included using the **Every nth point** button. Enter an integer to specify the interval you want, then click the **Every nth point** button to select the points in that interval. For example, if you want every 3rd data point, enter n = 3.

2) Enter a name in the **Subset name** box with which to identify this combination.

3) Click the **Create group** button.

4) Repeat steps 1-3 for every combination you wish to create.

5) If you do not wish to graphically view your simulation time-series, you may skip to step 9. If you do, click on the arrow for the dropdown list labeled **Type-B location**. Listed will be the different Type-B measurement locations. Select an entry that you would like to view.

6) In the box below that dropdown list, the data for the location you selected above will be displayed in either the form of actual data points or aggregate data. If you wish to view the distribution of values simulated for a datum, select the datum by clicking on it.

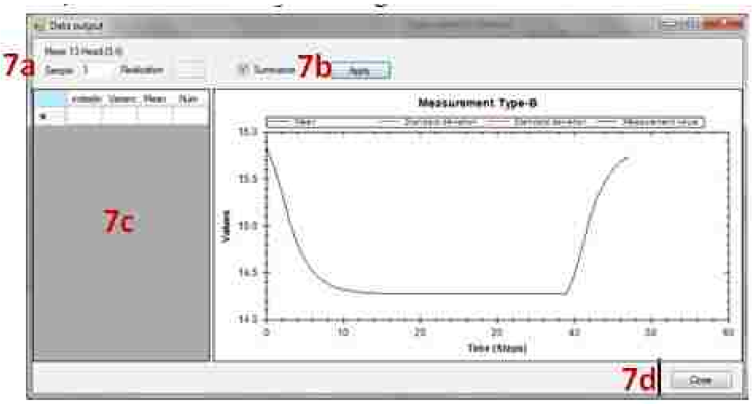


Figure A- 48: Output of a sample

7) [Transient projects only; Optional] To see the location's simulated time series, click on the first **View** button which will open a window.

   a. In this window, you can choose to view either one sample/realization pair by entering the specific sample and realization identifying numbers in the boxes respectively labeled **Sample** and **Realization**, or you may check off the **Summarize** checkbox and only enter the sample number to view a summary of all realizations of your sample.

   b. Click the **Apply** button to view your sample.

   c. Below there will appear a table of the sample time-series data. To the right of the table there will be a graph of your Type-B measurement, as well as the specified sample, evolving over time. If only one realization is chosen, then there will only be one curve. If all of the realizations for a sample are being summarized, then there will be three curves representing the sample (the mean and ± 1 standard deviation of all realizations) plus the actual measured time-series.

   d. Click **Close** when you are finished viewing your samples to return to the main tab.

   e. This feature will only work if **Save forward model data** was checked in the Simulations tab in preprocessing. If forward model data was not saved, a plot of the time series provided in the Type B measurement tab will be displayed.

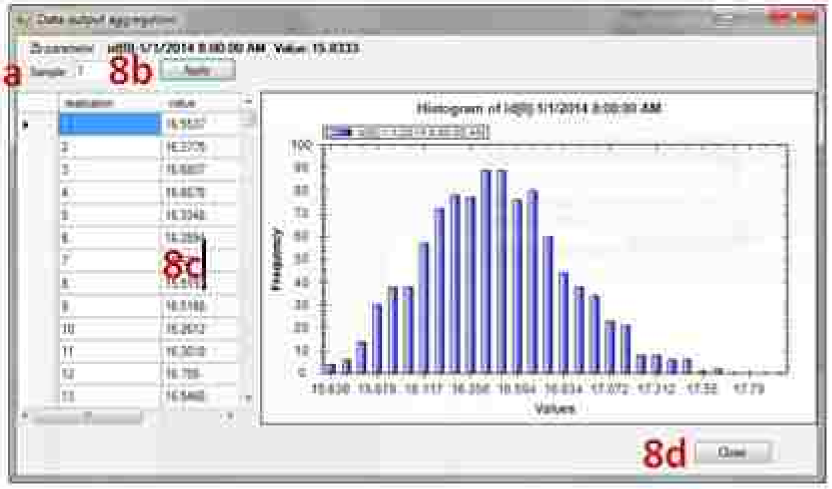
8) [Optional] To see the histogram of simulated values for a datum, click on the second **View** button which will open a window.

   a. In this window, you can choose which sample to view values for by typing in the sample number in the **Sample** field.

   b. Click the **Apply** button to view the histogram.

c. Below there will appear a table of the frequency of different simulated values. To the right of the table, a histogram will be plotted.

d. Click **Close** to return to the main tab.



Parameter h(0) 1/1/2014 8:00:00 AM Value: 15.8333

Sample 1

Realization	Value
1	15.9537
2	16.2776
3	15.9207
4	16.0676
5	16.3346
6	16.2554
7	[illegible]
8	[illegible]
9	16.5165
10	16.2012
11	16.3010
12	16.755
13	16.5460

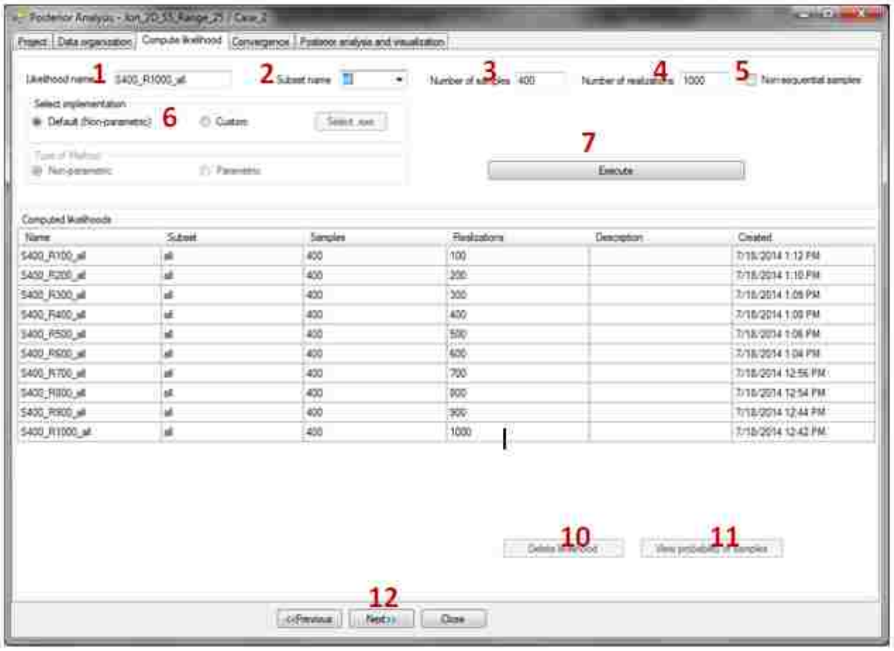
Figure A- 49: Output of a Type-B parameter in a sample

9) Click **Next>>** to proceed to the next tab, **Compute Likelihood**, where you conduct the likelihood computation process.

A.14 How do I Calculate the Likelihood?

The purpose of this tab is to compute the likelihood of your measured Type-B data being reproduced by a given configuration of anchor and structural model parameters. The purpose of generating multiple random fields per sample (anchor and structural model configuration) is to provide an ensemble of simulated Type-B data conditional on the sample to use for non-parametric kernel density estimation. A complete PDF is fit to the ensemble of simulated Type-B data and the probability of your measurement is evaluated at one point in this PDF. In this tab you will choose how many realizations to use (the size of the ensemble of simulated Type-B data), what subset of the Type-B data to use, and how many samples for which the likelihood will be calculated. Additionally, if simulation ensembles were generated on multiple machines or out of sequence on a single machine using the customize samples feature in the simulation tab, you can choose the IDs of multiple samples for which to calculate the likelihood using the non-sequential samples button. Note that varying the number of realizations used to populate the simulation ensemble is a necessary prerequisite to evaluating convergence in the next tab.

1) The **Likelihood name** box will be filled automatically by taking information from the subset name, number of samples, and number of realizations (specified in steps 2-4). A custom name can be provided, but it should not be entered until after specifying the subset name, number of samples, and number of realizations (steps 2-4), or it will be overwritten.



Likelihood name: S400_R1000_all

Subset name: all

Number of samples: 400

Number of realizations: 1000

Computed likelihoods

Name	Subset	Samples	Realizations	Description	Created
S400_R100_all	all	400	100		7/18/2014 1:12 PM
S400_R200_all	all	400	200		7/18/2014 1:10 PM
S400_R300_all	all	400	300		7/18/2014 1:09 PM
S400_R400_all	all	400	400		7/18/2014 1:08 PM
S400_R500_all	all	400	500		7/18/2014 1:06 PM
S400_R600_all	all	400	600		7/18/2014 1:04 PM
S400_R700_all	all	400	700		7/18/2014 12:56 PM
S400_R800_all	all	400	800		7/18/2014 12:54 PM
S400_R900_all	all	400	900		7/18/2014 12:44 PM
S400_R1000_all	all	400	1000		7/18/2014 12:42 PM

Figure A- 50: Calculate likelihood

2) From the Subset name dropdown list, select one of the data combinations that you designed in the **Data organization** tab.

3) Enter how many samples you wish to use for this particular experiment in the **Number of samples** box. Note: The maximum number of samples you may use is the number of samples you specified in the **Simulation** tab and actually created. By default this box will be filled with the maximum number of available samples.

4) Enter how many realizations you wish to use for this particular experiment in the **Number of realizations** box. Note: The maximum number of realizations you may use is the number of realizations you specified in the **Simulation** tab and actually created. By default this box will be filled with the maximum number of available realizations.

5) You have the option of not using the samples in a sequential order. For example, with the **Non- sequential samples** checkbox unchecked (the default) and n samples requested, the first n samples will be used. If checked, you will be asked in a subsequent step to select n number of samples to use.

6) You may also use a custom likelihood calculation method:

   a) Select the **Custom** radial button under **Select implementation**.

   b) Specify an executable for the likelihood calculation using the **Select .exe** button.

   c) Choose either the **Non-parametric** or **Parametric** radial button to specify the type of likelihood calculation used.

7) Click **Execute**.

8) If you checked the **Non-sequential samples** box:

   a) A separate window will appear listing the available samples. Click and drag, ctrl-click, and/or shift-click to create the subset you want to use in the calculation. Click **Accept** to begin the calculation.

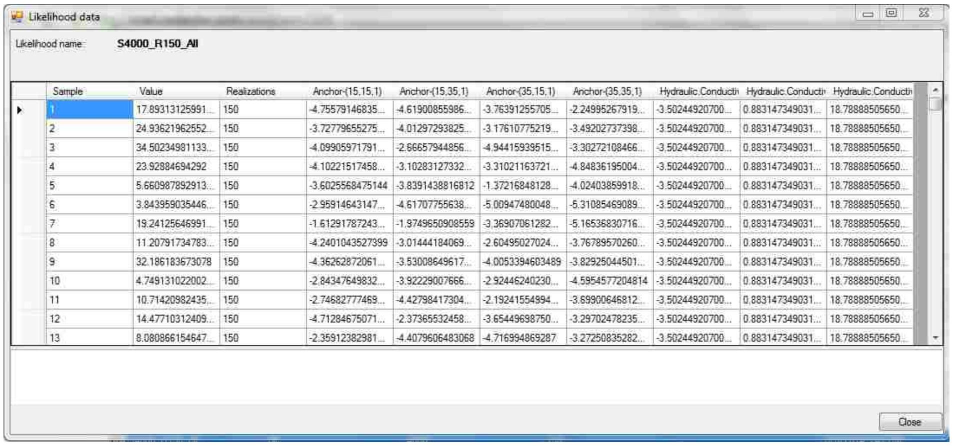
   b) MAD# will save the likelihood vector in the results folder. After the calculation completes, a separate window will appear showing the results of the calculation in the upper panel. Any information about the calculation process and any errors encountered will appear in the lower panel (“Log file”). The file path where the .txt file containing the likelihood vector was saved appears at the bottom of the window. Click **Close**.

9) Repeat steps 1-7 for each subset of data, sample count, and realization count combination you wish to view.

10) You can delete a likelihood experiment by selecting an experiment in the **Computed likelihoods** list then clicking **Delete likelihood**.

11) [Optional] To view a table of the likelihood value for each sample, you may click on a likelihood experiment in the **Computed likelihoods** list and then the **View probability of samples** button (Note: Expand the **value** column to see the entirety of the likelihood value; the exponent may not be visible). Each sample row will also display the prior values used in the realizations for all anchors and non-deterministic parameters.

12) Click **Next>>** to proceed to the next tab, **Convergence**, where you can determine if at the sample and realization counts you tried there are sufficient to converge the inferred likelihood function.



Likelihood name: **S4000_R150_All**

Sample	Value	Realizations	Anchor-(15,15,1)	Anchor-(15,35,1)	Anchor-(35,15,1)	Anchor-(35,35,1)	Hydraulic.Conducti	Hydraulic.Conducti	Hydraulic.Conducti
1	17.89313125991...	150	-4.75579146835...	-4.61900855986...	-3.76391255705...	-2.24995267919...	-3.50244920700...	0.883147349031...	18.78888505650...
2	24.93621962552...	150	-3.72779655275...	-4.01297293825...	-3.17610775219...	-3.49202737398...	-3.50244920700...	0.883147349031...	18.78888505650...
3	34.50234981133...	150	-4.09905971791...	-2.66657944856...	-4.94415939515...	-3.30272108466...	-3.50244920700...	0.883147349031...	18.78888505650...
4	23.92884694292	150	-4.10221517458...	-3.10283127332...	-3.31021163721...	-4.84836195004...	-3.50244920700...	0.883147349031...	18.78888505650...
5	5.660987892913...	150	-3.6025568475144	-3.8391438816812	-1.37216848128...	-4.02403859918...	-3.50244920700...	0.883147349031...	18.78888505650...
6	3.843959035446...	150	-2.95914643147...	-4.61707755638...	-5.00947480048...	-5.31085469089...	-3.50244920700...	0.883147349031...	18.78888505650...
7	19.24125646991...	150	-1.61291787243...	-1.9749650908559	-3.36907061282...	-5.16536830716...	-3.50244920700...	0.883147349031...	18.78888505650...
8	11.20791734783...	150	-4.2401043527399	-3.01444184069...	-2.60495027024...	-3.76789570260...	-3.50244920700...	0.883147349031...	18.78888505650...
9	32.186183673078	150	-4.36262872061...	-3.53008649617...	-4.0053394603489	-3.82925044501...	-3.50244920700...	0.883147349031...	18.78888505650...
10	4.749131022002...	150	-2.84347649832...	-3.92229007666...	-2.92446240230...	-4.5954577204814	-3.50244920700...	0.883147349031...	18.78888505650...
11	10.71420982435...	150	-2.74682777469...	-4.42798417304...	-2.19241554994...	-3.69900646812...	-3.50244920700...	0.883147349031...	18.78888505650...
12	14.47710312409...	150	-4.71284675071...	-2.37365532458...	-3.65449698750...	-3.29702478235...	-3.50244920700...	0.883147349031...	18.78888505650...
13	8.080866154647...	150	-2.35912382981...	-4.4079606483068	-4.716994869287	-3.27250835282...	-3.50244920700...	0.883147349031...	18.78888505650...

Figure A- 51: Likelihood vector and anchor values

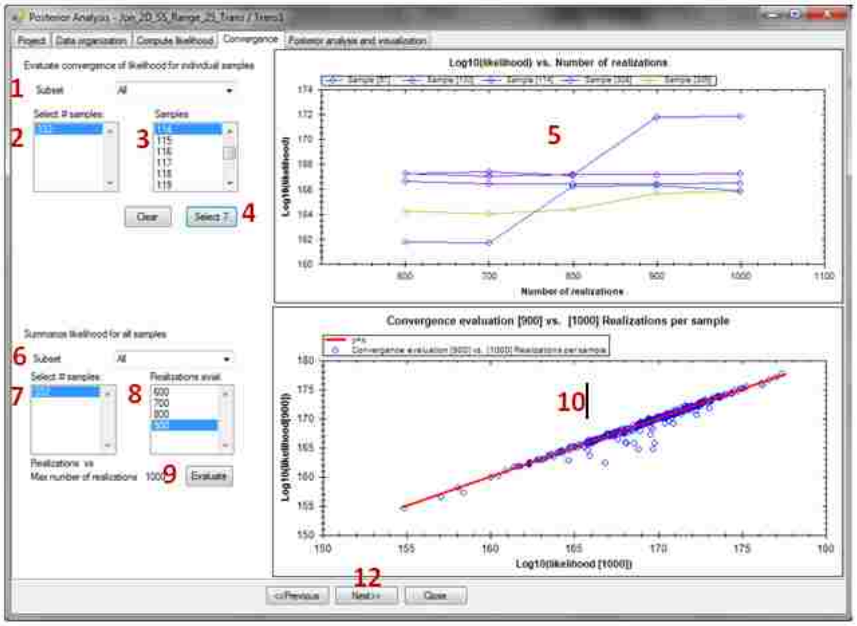
A.15 How do I Test for Convergence?

This tab is optional but highly recommended. The graphs generated will let you determine if the numbers of samples and realizations are sufficient to converge the inferred likelihood function. In this tab, based on the subsets of data used to calculate likelihoods in the previous tab, you will be able to visualize a few aspects of convergence. In the upper panel you will be able to plot the likelihood as inferred with different sizes of the simulated Type-B data ensemble. In the lower panel you will be able to plot a summary of the likelihood of all the samples inferred with the maximum number of realizations in the simulated Type-B data ensemble and the likelihood inferred on some smaller number of realizations. The summary plot is equivalent to taking all values of the likelihood shown in the upper panel at the rightmost point and plotting them against all the values of the likelihood at some point further left in the curves.

1) From the first **Subset** dropdown list, select the data combination that you wish to assess for convergence by comparing samples.

2) From the **Select # samples** list, select the number of samples you wish to assess.

3) From the **Samples** list, select the sample you wish to add to the graph. You can add up to seven samples, which will all be plotted on the same graph. Once a sample is selected, a plot should appear automatically in the upper panel.


Posterior Analysis - Jon_2D_SS_Range_25_Trans / Trans1
Project
Data organization
Compute likelihood
Convergence
Posterior analysis and visualization
Evaluate convergence of likelihood for individual samples
1
2
3
4
Clear
Select 7
Log10(likelihood) vs. Number of realizations
Sample [31]
Sample [113]
Sample [114]
Sample [304]
Sample [305]
5
174
172
170
168
166
164
162
160
Log10(likelihood)
600
700
800
900
1000
1100
Number of realizations
Summarize likelihood for all samples
6
7
8
9
Evaluate
Convergence evaluation [900] vs. [1000] Realizations per sample
y=x
Convergence evaluation [900] vs. [1000] Realizations per sample
10
180
175
170
165
160
155
150
Log10(likelihood[900])
150
155
160
165
170
175
180
Log10(likelihood [1000])
12
<<Previous
Next>>
Close

Subset: All

Samples: 114, 115, 116, 117, 118, 119

Subset: All

Realizations avail: 600, 700, 800, 900

Realizations vs Max number of realizations: 1000

Figure A- 52: Evaluate convergence

4) To have MAD# select seven samples at random from the list, click **Select 7**.

5) Examine the graph to the right. Plotted will be the log likelihoods of each samples you plotted as a function of number of realizations. With increasing number of realizations, each sample's line should become horizontal. Find the number of realizations from the x-axis that marks the beginning of the consistent horizontal behavior. Use this number of realizations in step 8.

6) From the second **Subset** dropdown list, select the data combination that you wish to assess for convergence by comparing different numbers of realizations.

7) From the **Select # samples** list, select the number of samples you wish to assess. (Note: This should be the same as in step 2 if assessing the same data).

8) From the **Realizations avail.** list, select the number of realizations you wish to compare to the maximum number of realizations.

9) Click **Evaluate**.

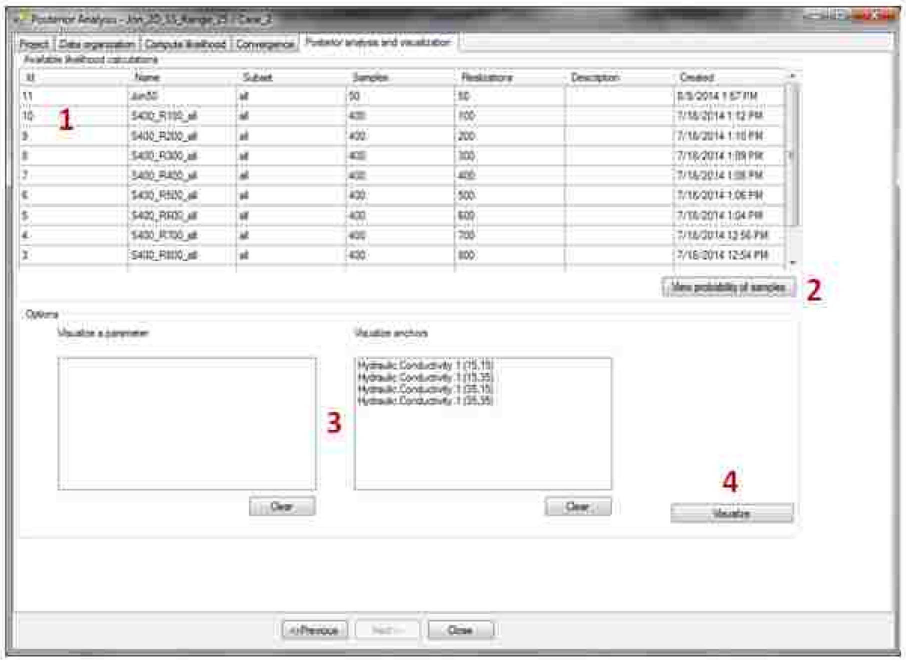
10) Examine the graph to the right. Plotted will be the log likelihoods of the maximum number of realizations on the x-axis and the log likelihoods of the current number of realizations (from step 8) on the y-axis. There will be a data point for each sample (from Step 7). The y=x line is also plotted. If your data points form a tight cluster with the y=x line, then the number of realizations are likely to provide adequate results.

11) If you are satisfied with your convergence results, continue to step 12. If you do not see that you have an adequate number of realizations, return to the simulation tab in the pre-processing block, use the same result name, increase the number of realizations per sample selection and hit execute again. You will be prompted that you are working in a previously existing result and asked if you want to continue. Select **Yes** to increase the available ensemble sizes and then revisit this convergence tab after calculating likelihoods with larger numbers of realizations in the **Compute likelihood** tab.

12) Click **Next>>** to proceed to the next and final tab, **Posterior analysis and visualization**, where you can view plots of the posterior compared to the prior for each of your likelihood experiments.

A.16 How do I View the Posterior?

The purpose of this last tab is to view graphs of the posterior PDFs of the random anchors and structural model parameters. In the graph, the prior is also plotted.


Posterior Analysis - Jun_20_15_Range_25 / Case_2
Project
Data organization
Compute likelihood
Convergence
Posterior analysis and visualization
Available likelihood calculations
1
View probability of samples
2
Options
Visualize a parameter
Visualize anchors
3
Clear
Clear
4
Visualize
<<Previous
Next>>
Close

Id	Name	Subset	Samples	Realizations	Description	Created
11	Jun50	all	50	50		8/5/2014 1:57 PM
10	S400_R100_all	all	400	100		7/16/2014 1:12 PM
9	S400_R200_all	all	400	200		7/16/2014 1:10 PM
8	S400_R300_all	all	400	300		7/16/2014 1:09 PM
7	S400_R400_all	all	400	400		7/16/2014 1:08 PM
6	S400_R500_all	all	400	500		7/16/2014 1:06 PM
5	S400_R600_all	all	400	600		7/16/2014 1:04 PM
4	S400_R700_all	all	400	700		7/16/2014 12:56 PM
3	S400_R800_all	all	400	800		7/16/2014 12:54 PM

Hydraulic Conductivity 1 (15,15)
Hydraulic Conductivity 1 (15,35)
Hydraulic Conductivity 1 (35,15)
Hydraulic Conductivity 1 (35,35)

Figure A- 53: Generate posterior

1) Select the likelihood experiment to use to update the prior and for which you would like to see a graph of the posterior PDFs.

2) [Optional] You may click the **View probability of samples** button to see the table of the likelihood value for each sample along with the prior data used for the realizations.

3) The **Visualize a parameter** panel will be populated with any available random structural parameters. Click one to view the posterior for a Type-A data structural parameter. The **Visualize anchors** panel will be populated with any available anchor locations. Click on one to view the posterior for an anchor. Note that you may only select one item per list, but that you may select an item from both lists at the same time. If you have a parameter and an

anchor selected, two different plots will appear with posteriors for the selected items.

4) Click **Visualize**. A pop-up window will appear with a graph. There are buttons for you to edit the graph (**Options**) and to save the graph (**Save**) (Note: The default file format for saving your posterior plot is an .emf which may not be compatible with your document processing software. You may change the file format in the save window). You may toggle between a probability distribution function and a cumulative distribution function with the respective radio buttons. If this project was synthetic and you provided true values in the Prior distribution tab, then you may click the **True value** checkbox to add the true value line to the graph. Check the **Statistics** checkbox to view the mean, variance, mode, and ratio of the prior and posterior distributions.

Data for the distributions can be accessed in spreadsheet form by clicking on the **Prior Data** and **Posterior Data** tabs, located above the plot area.

You can save any of the plotted distributions as a MAD# pdf file (which can be opened in any MAD visualization window) by clicking **Save pdf**. If you have another pdf saved already that you would like to add to the current plot, click **Add pdf** and locate the MAD# pdf file.

If you have a small number of samples, you may run into the problem that your posterior looks wider than your prior. If you have both structural parameters and anchors as random variables, and you created your prior samples file such that you have N anchor prior samples per structural parameter set, then you will need to request at least N+1 samples when executing MAD#. If you run less than N, then you are only using one structural parameter set and the smoothing function for generating the posterior distributions of the structural parameters can create a numerical artifact of a posterior wider than the bounds of its prior.

If the posterior does not appear, see step 2 on viewing your likelihood values. If the values are zero, this explains why you do not have a posterior. Zero likelihoods indicate that your realizations were not comparable to the measurements, and thus your priors were not compatible. See *Hou & Rubin, 2005* for an explanation of this compatibility. To resolve this issue, go back to the **Prior distribution** tab and provide different priors.

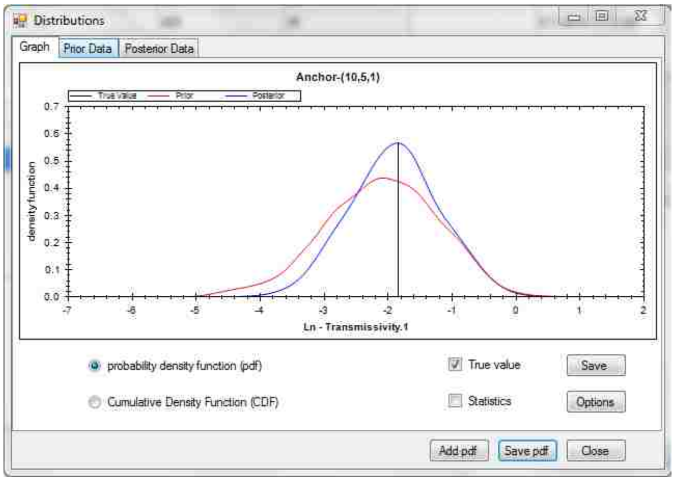


Figure A- 54: Prior and posterior distribution

5) Repeat step 1-4 for all the likelihood experiments you designed.

A.17 References

Carsel, R. F., R. S. Parrish. Developing joint probability distributions of soil water retention characteristics. *Water Resources Research*. 24(5). 1988.

Hou, Z., and Y. Rubin. On minimum relative entropy concepts and prior compatibility issues in vadose zone inverse and forward modeling, Water Resour. Res., 41(12), W12425, doi:10.1029/2005WR004082. 2005.

Over, M.W., Y. Yang, X. Chen, Y. Rubin. A strategy for improved computational efficiency of the method of anchored distributions. *Water Resources Research*. 2013.

Rubin, Y., X. Chen, H. Murakami, M. Hahn. A Bayesian approach for inverse modeling, data assimilation, and conditional simulation of spatial random fields. *Water Resources Research*. 46(10). 2010.

Yang, Y., M. Over, Y. Rubin. Strategic placement of localization devices (such as pilot points and anchors) in inverse modeling schemes. *Water Resources Research*. 48(8). 2012

APPENDIX B. MAD# PSEUDO CODE

B.1 Pre-processing Pseudo Code

This pseudo code shows the steps for configuring the pre-processing module in MAD#, where $f()$ is numerical model, $R()$ is the random field generator, (x, y) represents the domain, z_a is the Type-A data, z_b is the Type-B data, θ represents the structural parameters, ϑ_a represents the anchors, and $p(\theta, \vartheta_a| z_a)$ is the prior data.

Algorithm - pseudo code (Pre-processing)

```
Load f() -> Numerical model project
Select R() -> Random field generator
Get available Type-A and Type-B data <- f()
Define Domain y(x,y) <- f()
Select Target variable $z_a$ and inversion data $z_b$
Input available $z_a$ and $z_b$ data
Locate anchors $\vartheta_{xa}$
Define random variables Geostatistical model ($\theta$) <- R()
Input prior data of $p(\theta, \vartheta_a | z_a)$
Define $z_b$ vector
```

B.2 Processing Pseudo Code

The processing module in MAD# is summarized in a pseudo code, where $f()$ is the numerical model, $R()$ is the random field generator, $y(x,y)$ is a realization, z_a is the Type-A data, $\underline{z}_b$ is the Type-B data generated by the numerical model, θ represents the structural parameters, ϑ_a represents the anchors, n is the number of samples, and m is the number of realizations.

Pseudo code (Processing)

```
Execute Algorithm Pre-processing
Define Number of samples $n$
Define Number of realizations per sample $m$
Set $i,j = 0$
Set $\underline{z}_b[n,m]$
While (i < n)
    Set $y(x,y)[m]$ <- $R(z_{a,} \theta_I, \vartheta_{ai}, m)$
    While (j < m)
        $\underline{z}_b[i,j]$ <- $f(y(x,y)[j])$
        $j$++
    end while
    $i$++;
end while
```

B.3 Post-processing Pseudo Code

The post-processing module in MAD# is summarized in a pseudo code, where $L()$ is likelihood calculation tool, $Convergence()$ function represents the visual evaluation of the convergence, $F()$ is the likelihood function, z_a is the Type-A data, z_b is the Type-B data, $\underline{z}_b$ is the Type-B data generated by the numerical model, θ represents the structural parameters, ϑ_a represents the anchors, *n* is the number of samples, $p(\theta, \vartheta a \mid z_{a\,,}\, z_b)$ is the posterior distribution, $p(\theta, \vartheta_a \mid z_a)$ is the prior data, and $p(z_b \mid \theta, \vartheta a, z_a)$ is the likelihood .

Pseudo code (Post-Processing)

```
Execute  algorithm processing
Get Number of samples $n$
Set  $i = 0$
Set Likelihood[n]
While ($i < n$)
        $Likelihood[i] \leftarrow L(\underline{z}_b[i,*], z_b)$
       $i$++
end while
Set  $convergence \leftarrow Convergence\ (Likelihood[i])$
if ($convergence$ is false)
      Add realizations / samples using Algorithm 2
end if
$p(z_b \mid \theta, \vartheta a, z_a) \leftarrow F(Likelihood[])$
Show  posterior $p(\theta, \vartheta a \mid z_{a\,,} z_b) \leftarrow p(\theta, \vartheta_a \mid z_a) * p(z_b \mid \theta, \vartheta a, z_a)$
```

B.4 Levenberg-Marquardt Pseudo Code

The Levenberg-Marquardt algorithm is represented by the following pseudo code, where RFG is the random field generator, and FM is the forward model or numerical model.

```
Pseudo code  Levenberg-Marquardt  method
set  numIterations,  phi, threshold, parm, true_obs, sigma, num_obs, num_parm, μ
set   Jacobian[num_obs, num_parm]
set phi=0.5
For i=0; i< numIterations; i++
      if (phi>0)
        set  field= RFG(parm)
        set obs= FM(field)
        set error= true_obs- obs
        for (j=0; j< num_parm; j++)
           parm[j] += sigma
            set  field'= RFG(parm)
            set  obs'= FM(field')
             parm[j]-=sigma
             Jacobian[*,j]= (obs'-obs)/ sigma
         end for
     end if
  set A= Jacobian * Jacobian^T +  μI
  set b= Jacobian * error
  if ( |b| < threshold) break;
  Solve Ax=b
  if (|x| < threshold) break;
  set parm'= parm+x
  set field'= RFG(parm')
  set obs'= FM(field')
  phi = EvaluatePhi()
  if (phi>0)
      decrease $\mu$
      parm=parm'
  else
      increase $\mu$
  end if
end for
```