A Modular GIS-Based Software Architecture for Model Parameter Estimation using the Method of Anchored Distributions (MAD)

Yoram Rubin\(^2\), Daniel P. Ames\(^1\), Matthew Over\(^2\), Carlos Osorio-Murillo\(^1\)
\(^1\)Brigham Young University, Provo, Utah, USA
\(^2\)University of California Berkeley, Berkeley, California, USA

CUAHSI CyberSeminar 4/10/2013
Presentation Outline

• Brief project introduction and goals (Ames)
• Inverse modeling and the Method Anchored of Distributions (MAD) (Over)
• Live Demo of MAD# (Over + Osorio-Murillo)
• Summary (Ames)
Introduction: Inverse Modeling

• “Inverse Problem”
  – Estimate variables that *can’t be directly observed* using observable data.
  – E.g. “Given observed gravity, what is the Earth’s density?”

• Important but complex problem in hydrology
  – Difficult for every new researcher to learn from scratch
  – Requires people with a **wide range of knowledge**
  – Uncertainty Quantification is needed by decision makers

• Open framework needed
  – Greater adoption and application in multiple domains
  – Example of success at this approach in other domains e.g. CLM, HIS
  – Support education and training to teach the method to students
Overall Goal:
Open Source Model Inversion Framework

SPECIFIC REQUIREMENTS:

- Make advanced inverse modeling techniques more accessible to the broader scientific community.
- Follow open source approach in design, source code, issues-bug management.
- Build modular components that can be replaced by other better components as needed.
- Allow extensibility by 3rd parties to improve utility, longevity, and sustainability.
Method of Anchored Distributions

• General, Bayesian framework for conditional characterization of parametric uncertainty

• Yields complete PDFs for parameters, not just estimates

• Possible to condition on multiple data, which can be different type and/or scale
Governing Bayesian Proportionality

\[ f(\text{para.} | \text{data}) \propto f(\text{data} | \text{para.}) f(\text{para.}) \]

**Objective:** Obtain joint probability distribution of parameters conditional on data

**Implementation:** Define a prior, compute the likelihood
Overview of MAD Theory Presentation

- Description of parameter and data domains
- Monte Carlo procedure for likelihood inference
- See MAD in action!
Terminology of MAD

• Parameter domain is partitioned into:
  – Anchor parameters $\alpha$
  – Structural model parameters $\theta$

• Data domain is partitioned into:
  – Type-A data $z_a$
  – Type-B data $z_b$
The Role of Parameters

- Parameters $\alpha$ and $\theta$ describe the spatial distribution of the characterization target variable field $\tilde{Y}$.
  - Transmissivity field
  - Storativity field
  - Porosity field
Why the theory is called MAD

- Anchored distributions are PDFs of $\tilde{Y}$ at points locations in the domain
- Anchors are not physical devices, purely statistical
Why not put anchors everywhere?

- Every anchor is a dimension in the parameter domain

- Cost of Monte Carlo evaluation of PDFs grows \textit{nonlinearly} with dimensionality

- Also, in scarce data implementations can lead to ill-posed or underdetermined problems
Coming full circle...

- Because anchors cannot tractably be used exclusively in large domains, a structural model is used to “fill in the gaps” of the target variable field.
- Structural model needs to be random for likelihood inference.
Summarizing the parameter space

- Anchors $\alpha$ describe uncertainty in $\tilde{y}$ at specific locations (not the whole field $\tilde{Y}$)

- Structural model describes $\tilde{Y}$ everywhere besides anchors.

Spoiler alert: structural models must be compatible with anchors, i.e. reproduce values of $\tilde{y}$ at anchors
Two types of data

• Type-A directly related to $\tilde{Y}$
  – Provides information of $\tilde{Y}$ at points in the domain
  – Commonly forward model input
  – Again (like anchors) must be compatible with structural model

• Type-B indirectly related to $\tilde{Y}$
  – Provides information about $\tilde{Y}$ in regions of the domain, via ODE/PDE relationship
  – Commonly forward model output
Expanded formula

\[ f(\theta, \alpha | z_a, z_b) \propto f(z_b | \theta, \alpha, z_a) f(\theta, \alpha | z_a) \]

- **Posterior**
- **Likelihood**
- **Prior**

MAD Software Output → Inferred + User Specified
Inferring the likelihood

The likelihood is $f(z_b | \theta, \alpha, z_a)$ so our field realizations of $\tilde{Y}$ need to be conditional on structural parameters, anchors, and Type-A data, so the simulated Type-B data will also be conditional:
Nonparametric inference

• Collect simulated Type-B data (children) from family of realizations that have the same structural parameters, anchors, and Type-A data (parents)

• Infer PDF of the Type-B data

\[ f(z_b | \theta, \alpha, z_a) \]

• Evaluate the likelihood of the “parent”
Monte Carlo

- Combine the likelihood with the value of the prior to obtain a value proportional to the Bayesian posterior for the sample.

- Repeat the procedure for many samples in the parameter domain $\theta, \alpha$ to determine posterior PDFs.
Time to see it in action: an example

- Obtain posterior PDFs of geostatistical ($\theta$) model parameters and transmissivity ($T$) at anchor locations ($\alpha$)

- Condition on pressure head data ($z_b$) and a few direct measurements of $T$ ($z_a$)

- Generate random fields of $T$ using GSTAT
- Simulate $z_b$ using MODFLOW
Live Demo
Architecture
Managed Extension Framework (MEF) of .NET allows “your” software to easily host plugins.

- DotSpatial uses MEF to support geospatial extensions.
- MAD uses MEF to support new drivers.
MAD and the .NET Managed Extensibility Framework

What does MEF do?

- Discovery
- Composition
- Makes your application “pluggable”
Interfaces for custom “Drivers” allow for interchangeable geostatistical models, conditional simulators and forward models.
What We’ve Created

• HydroDesktop plugin (tightly integrated – not through a web system)

• Standalone desktop application (essentially the “desktop version” promised).
We created two things...
Planned Architecture Moving Forward

- “Core MAD Computational Engine”
  - Set of command line executables in C++
  - Fully cross platform
  - Will be the “back end” for desktop, web based, and high performance computing “front ends”

- Will meet the grant intention following a much better architecture design than initially planned.
A Modular GIS-Based Software Architecture for Model Parameter Estimation...

Plan to refactor like this...

MAD Desktop Application

MAD in HydroDesktop
Propose to refactor like this...

MAD Desktop Application & HydroDesktop Plugin
A Modular GIS-Based Software Architecture for Model Parameter Estimation …

Version 1.0 release would include desktop and plugin as is before refactoring with MODFLOW and HYDRUS driver.

MAD CORE

MAD Desktop Application & HydroDesktop Plugin

Plan to refactor like this...

Release soon... March 2013?
MAD Desktop becomes thin "wrapper" user interface to the core components.

Propose to refactor like this…

Web based version is a simple web form with educational materials.

High performance computing is supported by running the command line executables directly on a cloud resource.
MAD Open Source Web Portal

http://mad.codeplex.com
Summary Points

• We have created an open source extensible framework for inverse modeling.

• The modular implementation in .NET is intended to support long term sustainability of the project, meeting NSF goals for “sustainable software”.

• The (relatively) user-friendly nature of the software should encourage adoption of inverse modeling and the MAD technique by researchers and educators.