Title
Practical Improvement to and Application of Proper Orthogonal Decomposition Reduced Order Modeling to Experimental Design for Groundwater Monitoring Networks

Permalink
https://escholarship.org/uc/item/3pb974kn

Author
Ushijima, Timothy

Publication Date
2018

Peer reviewed|Thesis/dissertation
Practical Improvement to and Application of Proper Orthogonal Decomposition Reduced Order Modeling to Experimental Design for Groundwater Monitoring Networks

A dissertation submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy in Civil Engineering

by

Timothy Ushijima

2018
ABSTRACT OF THE DISSERTATION

Practical Improvement to and Application of Proper Orthogonal Decomposition Reduced Order Modeling to Experimental Design for Groundwater Monitoring Networks

by

Timothy Ushijima

Doctor of Philosophy in Civil Engineering

University of California, Los Angeles, 2018

Professor William W Yeh, Chair

Proper Orthogonal Decomposition (POD) is a method used to reduce the dimension of a highly discretized groundwater model. The reduced model is sometimes several orders of magnitudes smaller than the original model and can run several orders of magnitude faster. The key advantage of utilizing a POD reduced model is its ability to drastically reduce the computational burden of repeated model calls, which are required in Monte Carlo simulation, uncertainty analysis, and heuristically searched experimental design. Although POD has been applied to many areas of research, there continues to be room to improve its implementation. This dissertation consists of six chapters. After an introductory chapter, the second chapter discusses a method that can be used to improve the efficiency of constructing complex POD reduced models. The third through fifth chapters develops methodologies by which POD reduced models are used to solve the experimental design problem of optimizing a network of observation wells to gain information about the modeled aquifer. The final chapter offers some conclusions, discussions, and potential future research opportunities.
The dissertation of Timothy Ushijima is approved.

Steven Adam Margulis

Weng Kee Wong

Mekonnen Gebremichael

William W Yeh, Committee Chair

University of California, Los Angeles

2018
# Table of Contents

1. Introduction ........................................................................................................................................ 1

1.1. Groundwater modeling .................................................................................................................. 3

1.1.a. Proper Orthogonal Decomposition (POD) ............................................................................... 4

1.1.b. Fast Algorithm .............................................................................................................................. 6

1.2. Experimental Design ..................................................................................................................... 11

2. A Proposed Fast Algorithm to Construct the System Matrices for a Reduced-Order Groundwater Model ........................................................................................................................................ 16

2.1. Introduction ..................................................................................................................................... 17

2.1.a. Model Reduction ............................................................................................................................ 17

2.1.b. Proper Orthogonal Decomposition .............................................................................................. 18

2.2. Algorithm ..................................................................................................................................... 23

2.2.a. Basis ........................................................................................................................................... 23

2.2.b. Decomposition Step ...................................................................................................................... 29

2.2.c. Calculation Step ............................................................................................................................ 31

2.2.d. Application to Interpolated-Parameter Models .......................................................................... 32

2.3. Test Cases ..................................................................................................................................... 34

2.3.a. Two-Dimensional, Finite-Difference with Zoned-parameter Test Case ...................................... 34

2.3.b. Two-Dimensional, Finite-Difference with Interpolated-parameter Test Case ............................ 42

2.3.c. Large-Scale Two-Dimensional, Finite-Element Zoned Test Case .............................................. 50

2.4. Discussion and Conclusions ......................................................................................................... 54

2.5. Bibliography ................................................................................................................................. 58
3. Experimental design for estimating unknown groundwater pumping using genetic algorithm and reduced order model ................................................................. 65

3.1. Introduction .................................................................................................................. 67

3.1.a. Experimental Design .......................................................................................... 67

3.1.b. Confined Aquifer Groundwater-Flow Model ....................................................... 72

3.1.c. Proper Orthogonal Decomposition (POD) ......................................................... 74

3.1.d. Genetic Algorithm ............................................................................................. 77

3.2. Test Cases .................................................................................................................. 81

3.2.a. One-Dimensional Test Case .............................................................................. 81

3.2.b. Two-dimensional Test Case .............................................................................. 85

3.3. Conclusions .............................................................................................................. 94

3.4. Bibliography ........................................................................................................... 100

4. Experimental design for estimating unknown hydraulic conductivity in an aquifer using a genetic algorithm and reduced order model ........................................... 104

4.1. Introduction .............................................................................................................. 105

4.1.a. Background ........................................................................................................ 105

4.1.b. Confined Aquifer Groundwater Model ............................................................. 107

4.1.c. Proper Orthogonal Decomposition (POD) ....................................................... 109

4.1.d. Experimental Design .......................................................................................... 113

4.2. Methods ................................................................................................................... 118

4.2.a. Experimental Design ...................................................................................... 118

4.2.b. Genetic Algorithm ........................................................................................... 122

4.3. Test Cases ............................................................................................................... 124
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.a.</td>
<td>One-Dimensional Test Case</td>
<td>124</td>
</tr>
<tr>
<td>4.3.b.</td>
<td>Two-Dimensional Test Case</td>
<td>131</td>
</tr>
<tr>
<td>4.4.</td>
<td>Conclusions</td>
<td>143</td>
</tr>
<tr>
<td>4.5.</td>
<td>Bibliography</td>
<td>151</td>
</tr>
<tr>
<td>5.1.</td>
<td>Introduction</td>
<td>158</td>
</tr>
<tr>
<td>5.1.a.</td>
<td>Background</td>
<td>158</td>
</tr>
<tr>
<td>5.1.b.</td>
<td>Experimental Design</td>
<td>161</td>
</tr>
<tr>
<td>5.1.c.</td>
<td>Optimality Criteria</td>
<td>163</td>
</tr>
<tr>
<td>5.1.d.</td>
<td>Evaluation Challenges</td>
<td>165</td>
</tr>
<tr>
<td>5.2.</td>
<td>Test Cases</td>
<td>169</td>
</tr>
<tr>
<td>5.2.a.</td>
<td>Small-Scale, Two-Dimensional Test Case</td>
<td>169</td>
</tr>
<tr>
<td>5.2.b.</td>
<td>Large-Scale, Two-dimensional Test Case</td>
<td>178</td>
</tr>
<tr>
<td>5.3.</td>
<td>Discussion and Conclusions</td>
<td>187</td>
</tr>
<tr>
<td>5.4.</td>
<td>Bibliography</td>
<td>191</td>
</tr>
<tr>
<td>6.</td>
<td>Dissertation Conclusion, Discussions, and Future Research</td>
<td>197</td>
</tr>
<tr>
<td>7.</td>
<td>Complete Bibliography</td>
<td>201</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1. Two-dimensional, finite-difference zoned test case ................................................. 35
Figure 2. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the arithmetic mean .................................. 38
Figure 3. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the harmonic mean ........................................ 39
Figure 4. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the geometric mean .................................... 40
Figure 5. Two-dimensional, finite-difference with interpolated-parameter test case ............. 43
Figure 6. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the arithmetic mean .................................. 46
Figure 7. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the harmonic mean ........................................ 47
Figure 8. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the geometric mean .................................... 48
Figure 9. Comparison of the errors introduced by the Fast Algorithm and model reduction for full, interpolated-parameter models using various means .................................................. 49
Figure 10. Two-dimensional, finite-element test case ................................................................. 51
Figure 11. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 3 full model that uses the arithmetic mean ..................................... 54
Figure 12. Flowchart of GA ..................................................................................................... 80
Figure 13. 1-D Test Case Aquifer Setup ...................................................................................... 81
Figure 14. 2-D Test Case Hydraulic Zones .................................................................................. 85
Figure 15. 2-D Test Case Well Zones ................................................................. 88
Figure 16. Optimal Observation Well Locations .................................................. 89
Figure 17. Objective Score Value vs. Number of Observation Wells ...................... 91
Figure 18. Comparison of A and E-Optimal Observation Well Locations for Six and Eight Wells .................................................................................................................. 93
Figure 19. Flowchart of GA .............................................................................. 124
Figure 20. 1-D test case ...................................................................................... 125
Figure 21. Scores from the 1-D test case ............................................................. 127
Figure 22. Information vs. number of retained principal vectors ......................... 129
Figure 23. Estimated probability for a location to be an optimal location (1-D test case) .... 130
Figure 24. 2-D test case hydrologic zones ............................................................ 131
Figure 25. Norm of the error between the full and reduced model results at steady state .... 133
Figure 26. Percent error between the full and reduced model results at steady state ........ 134
Figure 27. 2-D test case experimental setup .......................................................... 135
Figure 28. Estimated probability for a location to be an optimal location for the 1, 2, 3, and 4 observation well networks (2-D Test Case) ......................................................... 136
Figure 29. Estimated probability for a location to be an optimal location for the 5, 6, 7, and 8 observation well networks (2-D Test Case) ........................................................................ 137
Figure 30. Estimated probability for a location to be an optimal location for the 9, 10, 11, and 12 observation well networks (2-D Test Case) ........................................................................ 138
Figure 31. Robust and HFO designs for 1, 2, 3, and 4 observation well networks (2-D Test Case) ............................................................................................................................. 139
Figure 32. Robust and HFO designs for 5, 6, 7, and 8 observation well networks (2-D Test Case) ................................................................. 140

Figure 33. Robust and HFO designs for 9, 10, 11, and 12 observation well networks (2-D Test Case) ................................................................. 141

Figure 34. Results from the 2-D test case ........................................................................................................ 142

Figure 35. Small-scale test case hydrologic zones .......................................................................................... 170

Figure 36. Small-scale test case experimental design setup ......................................................................... 171

Figure 37. A-, D-, and E-designs .................................................................................................................. 173

Figure 38. G- and I-designs ......................................................................................................................... 174

Figure 39. A-design efficiencies .................................................................................................................. 175

Figure 40. D-design efficiencies .................................................................................................................. 175

Figure 41. E-design efficiencies .................................................................................................................. 176

Figure 42. G-design efficiencies .................................................................................................................. 176

Figure 43. I-design efficiencies .................................................................................................................. 177

Figure 44. Large-scale test case hydrologic zones and pumping wells ....................................................... 179

Figure 45. Large-scale test case experimental design setup ......................................................................... 180

Figure 46. One-, two-, three-, and four-well designs .................................................................................. 181

Figure 47. Five-, six-, seven-, and eight-well designs .................................................................................. 182

Figure 48. Nine-, ten-, eleven-, and twelve-well designs .............................................................................. 183

Figure 49. A-design efficiencies .................................................................................................................. 184

Figure 50. D-design efficiencies .................................................................................................................. 184

Figure 51. E-design efficiencies .................................................................................................................. 185

Figure 52. G-design efficiencies .................................................................................................................. 185
Figure 53. I-design efficiencies........................................................................................................186
List of Tables

Table 1. Optimality Criteria ........................................................................................................... 13
Table 2. Equations for calculation of the mean ............................................................................. 36
Table 3. $\| \tilde{A}_r(k)_p - \tilde{A}_r(k)_r \|$ ..................................................................................... 36
Table 4. RMSE between the full and reduced model resulting from employing various means to characterize the border hydraulic conductivity ............................................................................... 41
Table 5. Computation time and FLOP count comparison for two-dimensional, zoned finite-difference test case .................................................................................................................. 42
Table 6. Computation time and FLOP count comparison for two-dimensional interpolated-parameter model test case ........................................................................................................... 44
Table 7. RMSD between model results ........................................................................................... 44
Table 8. Computation time and FLOP count comparison for the large scale, two-dimensional, finite-element test case ........................................................................................................... 52
Table 9. Model zonal parameters .................................................................................................. 52
Table 10. Comparison of the Trace of the Information matrices for the full and reduced models for the 1-D Test Case ......................................................................................................... 82
Table 11. Hydraulic Zone Properties ............................................................................................. 85
Table 12. Comparison of the Trace of the Information matrix for the full and reduced model for the 2-D Test Case ............................................................................................................. 86
Table 13. Optimization Results for the 2-D Test Case .................................................................. 89
Table 14. Number of model calls and observation well locations for the robust, HFO, and balanced designs for the 1-D Test Case ...................................................................................... 126
Table 15. 2-D test case known hydrologic parameters ................................................................... 132
Table 16. Optimality Criteria.............................................................................................................. 163
Table 17. Small-scale test case hydrologic properties............................................................................. 170
Acknowledgements

I am indebted to my advisor Dr. William W-G. Yeh for his guidance and patience to shepherd me through my Ph.D. studies. Without him, there is no doubt in my mind that I would not have been able to achieve the monumental goal of completing my Ph.D.

I would also like to thank my family especially my parents for encouraging and at times pushing me in this endeavor.

Portions of this dissertation were funded by the NSF under award EAR-1314422 and an AECOM endowment.

Chapter 2 contains the following reprinted journal article:


and retains the following copywrite:

© 2017 Made available under the CC-BY-NC-ND 4.0 license

http://creativecommons.org/licenses/by-nc-nd/4.0/
Chapter 3 contains the following reprinted journal article:


and retains the following copywrite:

© 2013 Made available under the CC-BY-NC-ND 4.0 license

[http://creativecommons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)

Chapter 4 contains the following reprinted journal article:


and retains the following copywrite:

© 2015 Made available under the CC-BY-NC-ND 4.0 license

[http://creativecommons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)

Chapter 5 contains the following submitted journal article:


and retains the following copywrite:

© 2018 Made available under the CC-BY-NC-ND 4.0 license

[http://creativecommons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)
VITA

Education

Master of Science in Civil and Environmental Engineering at University of California, Los Angeles (2011)

Bachelor of Science in Civil Engineering, California State University Northridge (2009)

Journal Publications


Conference Presentations

San Francisco, California.


Chapter 1

1. Introduction

As groundwater becomes an increasingly stressed natural resource, it is imperative that accurate, usable models be created to be able to best understand how to manage this natural resource. The dual requirements of accuracy and usability pose significant challenges to hydrographers. The drive to build highly accurate models tends to result in models becoming very complex and computationally expensive to run. This in turn can decrease the models’ usability since many modern techniques (e.g., Monte Carlo methods or heuristic searches) require running a model hundreds—if not thousands—of times. To balance these two often competing goals, a number of techniques have been developed, including building surrogate models whose results approximate the results from the original model but are much less computationally expensive to run (Antoulas, Sorensen, & Gugercin, 2001; Asher, Croke, Jakeman, & Peeters, 2015). These types of methods are referred to as the creation of a reduced model (the surrogate model) of a full model (the original model). Within this category of techniques falls the method of using Proper Orthogonal Decomposition (POD). POD projects the full model onto a sub-space of a greatly reduced dimension to create a reduced model that is computationally much less expensive to run than the full model while controlling the loss of accuracy accrued by this projection. Generally, this reduction is on the scale of several orders of magnitude (Siade, Putti, & Yeh, 2010). POD has been applied to varied fields of research (Cazemier, Verstappen, & Veldman, 1998; Kowalski & Jin, 2003; Willcox & Peraire, 2002) in addition to groundwater (McPhee & Yeh, 2008; Siade et al., 2010; Siade, Putti, & Yeh, 2012; Vermeulen, Heemink, & Te Stroet, 2004). Despite this excellent research, there are still areas of research to be explored and this
dissertation explores two of these areas. The first is a method to increase the usability of POD reduced models by improving the efficiency of constructing a reduced model. One of the downsides of traditional POD model reduction is that projecting the full model onto the reduced sub-space is a computationally expensive process. This is not an issue if it can be carried out off-line, but some studies require an online construction of the reduced model (e.g., a Bayesian inverse problem (Boyce & Yeh, 2014) or a Monte Carlo simulation (Pasetto, Guadagnini, & Putti, 2014)), thus necessitating an online projection of the full model onto the reduced sub-space. This drawback of traditional POD model reduction is rarely discussed in the literature.

The Discrete Empirical Interpolation Method (DEIM) (Chaturantabut & Sorensen, 2010) performs two model reductions (one of the column space and one of the row space) as compared to the single POD reduction of the column space to reduce the computational burden of the sub-space projection. While useful, the drawbacks of this method are first, it requires a second model reduction, increasing the error in the reduced model, and second, it still requires an online projection of the full model onto the reduced sub-space. This dissertation proposes a revolutionary algorithm that allows for the efficient online construction of POD reduced models without requiring additional model reductions or online projections. This algorithm is presented in Chapter 2.

The second area of research that this dissertation investigates is the practical application of using POD reduced models in the field of experimental design in the context of groundwater modeling. Chapter 3 presents an application of POD model reduction with regard to an experimental design investigating unknown forcing; Chapter 4 extends the application to unknown model parameters; and Chapter 5 employs this practical application to investigate the relationship between various optimality criteria in the context of groundwater modeling.
1.1. Groundwater modeling

The modeling of groundwater has its foundation in the experiments performed by Henry Darcy in 1856 to characterize the flow rate of a fluid through a porous medium. These experiments led to the development of Darcy’s Law which relates flow rate to change in pressure by means of the hydraulic conductivity of the porous medium. Darcy’s Law is given as

$$q = -K \nabla h$$

where $q$ [L/T] is the flow rate, $K$ [L/T] is the hydraulic conductivity, and $\nabla h$ [L/T] is the change in pressure. L denotes the length unit (meters, feet, etc.), and T denotes the time unit (days, hours, etc.). Subsequent studies led to the development of the following PDE to describe three-dimensional groundwater flow in a confined, anisotropic aquifer

$$\frac{\partial}{\partial x} \left( K_x \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial h}{\partial z} \right) - F = S_s \frac{\partial h}{\partial t}$$

(1)

with initial and boundary conditions

$$h(x, y, z) = f_1(x, y, z)$$
$$h(x, y, z, t) = f_2(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_1)$$
$$q_n(x, y, z, t) = f_3(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_2)$$

where $h$ is the hydraulic head [L]; $K_x$, $K_y$, and $K_z$ are the hydraulic conductivities in the $x$, $y$, and $z$ directions [L/T]; $S_s$ is the specific storage [L$^{-1}$]; $F$ is the specific volumetric pumping rate [T$^{-1}$]; $q_n$ is the specific discharge normal to the flux boundary ($\Gamma_2$) [T$^{-1}$]; $\Gamma_1$ is the fixed head
boundary; \( f_1, f_2, \) and \( f_3 \) are known functions (Bear, 1979). Without loss of generality, the state variable of the PDE may be transformed from head (h) to drawdown (s), where drawdown is defined as the difference between an initial head (H) and head at some later time (h). When the governing equation is written in terms of drawdown, the initial and Dirichlet boundary conditions are equal to zero. The governing PDE (Eq. (1)) may also be discretized through finite element or finite difference approximations to produce the following set of ODE’s:

\[
B \frac{ds}{dt} + As = q
\]  

(2)

where \( s \in R^{N_n} \) is a vector of drawdown values, \( A \in R^{N_n \times N_n} \) is the stiffness matrix, \( B \in R^{N_n \times N_n} \) is the mass matrix, \( q \in R^{N_n} \) is a vector of sinks (pumping, recharge, evaporation, etc.), and \( N_n \) is the total number of nodes onto which the aquifer is discretized. This set of ODE’s is referred to as the full model.

1.1.a. Proper Orthogonal Decomposition (POD)

POD is a method that projects a system of linear equations whose solution exists in a space \( V \in R^{N_n} \) onto a subspace \( \nu \in R^{n_p} \) where \( \text{Range}(\nu) = \text{Range}(V) \) and \( n_p \leq N_n \). It does this through the use of a projection matrix \( P \in R^{N_n \times n_p} \) that is formed by factorizing a collection of snapshots (solutions from the full model at specified times) to find a matrix whose columns form a basis set which span the column space of the snapshot set. Two of the commonly used factorizations are eigenvalue decomposition or singular value decomposition. Once this
projection matrix has been obtained, the state vector of the full model \( s \in R^{N_p} \) may be
projected onto the subspace \( \nu \) to find the reduced state vector \( r \in R^{n_p} \) such that

\[
s \approx Pr
\]

The Galerkin projection is then used to project the full model onto the subspace by pre-
multiplying the Eq. (2) by \( P^T \) (Siade et al., 2010):

\[
P^TBP \frac{dr}{dt} + P^TAPr = P^Tq
\]

If we let \( B = P^TBP, \ A = P^TAP, \) and \( q = P^Tq \), then this equation may be rewritten as

\[
\tilde{B} \frac{dr}{dt} + \tilde{A}r = \tilde{q}
\]

Eq. (5) is referred to as the reduced model. If \( n_p \ll N_p \), then Eq. (5) is much less
computationally expensive to solve than Eq. (2). Obviously, the accuracy of the reduced model is
dependent on how well the snapshot set represents the full dynamics of the model. Siade et al.
(2010) developed a method to approximate the optimal time snapshot set which will produce
reduced models with controllable error, and Pasetto et al. (2013) expanded this to include
parameter snapshots. By approximating the time derivative using a backward finite-difference
Eq. (5) may be solved at each time step:

\[
(\tilde{B} + \tilde{A})r = \tilde{Ar} = b
\]
1.1.b. Fast Algorithm

Despite the reduced model’s ability to be much less computationally expensive to run with controllable error, a major drawback to POD model reduction is the overhead cost of projecting the full model onto the reduced space (Eq. (4)). In some cases, this may be carried out offline, for example if the aquifer parameters (e.g., hydraulic conductivity) do not change. However, other cases (e.g., uncertainty analysis, experimental design) require the aquifer parameters to change during the simulation, thus requiring repeated online projections which may be infeasible. In the face of this challenge, rather than limiting the simulation abilities (e.g., assuming average parameter values, a-priori projecting a certain number of parameter values offline, etc.), we developed an algorithm (dubbed the Fast Algorithm) which allows for inexpensive, online calculations of Eq. (4). We do this by rewriting Eq. (2) as

$$B(s, \frac{ds}{dt}) + A(k)s = q$$

where $s$, and $k$ contain information on the nodal values of specific storage and hydraulic conductivity, respectively. For simplicity, we will assume the specific storage is constant, so we will ignore the mass matrix from the following discussion. However, the Fast Algorithm may be extended to include a variable specific storage. After applying POD, this equation becomes

$$\hat{A}(k)r = b$$

where

$$\hat{A}(k) = P^T B(s) P + P^T A(k) P$$

(7)
\( \hat{A}(k) \) is referred to as the reduced system matrix. A critical component of POD model reduction is calculating Eq. (7), but this projection is very costly with the FLOP (floating point operation) count being \( O\left(N_n^2 + n_p^3\right) \). When compared to the fact that the FLOP count of solving the reduced model is \( O(n_p^3) \), it is clear that the computational cost of Eq. (7) can easily be the dominate cost of a simulation. The Fast Algorithm addresses this challenge by finding a function \( \Phi(k) \) such that

\[
\hat{A}(k) = P^T A(k) P = \Phi(k)
\]

where \( \Phi(k) \) is much more efficient to solve than Eq. (7). Chapter 2 describes the Fast Algorithm and its mathematical basis in much greater detail, but in summary, the Fast Algorithm takes advantage of the fact that the stiffness matrix may be formed by a tri-, penta-, or septa-diagonal matrix with each row corresponding to a node in the model. Assuming a zoned, 1-dimensional, finite difference model with a constant distance between nodes, the matrix may be formed as a tri-diagonal with each row generally taking the form

\[
\frac{k_i}{\Delta x^2} \begin{pmatrix} 0 & \ldots & 1 & -2 & 1 & \ldots & 0 \end{pmatrix}
\]

where \( k_i \) is the hydraulic conductivity in the \( i^{th} \) zone and \( \Delta x \) is the distance between nodes. Nodes on boundaries between zones are special cases that have a similar but slightly different form. The Fast Algorithm takes advantage of this to individually project each zone to arrive at

\[
\hat{A}(k) \approx \sum_{i=1}^{n} k_i P^T A(z_i) P
\]
where \( n_z \) is the number of nodes, \( k_i \) is the hydraulic conductivity in the \( i^{th} \) zone, and \( A(z_i) \) is the sub-matrix of the stiffness matrix that represents the nodes in the \( i^{th} \) zone. The Fast Algorithm deals with the special cases to Eq. (8) by isolating them for their own projection. This allows the approximation in Eq. (9) to become

\[
\hat{A}(k) = \sum_{i=1}^{n_z} k_i A'(z_i) + \sum_{b=1}^{n_b} k_b A''(b_b) \tag{10}
\]

where \( A'(z_i) \) is the projection of the sub-matrix \( A(z_i) \), \( k_b \) is the hydraulic conductivity across the \( b^{th} \) boundary, \( n_b \) is the number of boundaries, and \( A''(b_b) \) is the projection of the sub-matrix of the stiffness matrix related to the \( b^{th} \) boundary. The Fast Algorithm finds these sub-matrices through the decomposition step detailed below.

1.1.2.1. Decomposition Step

1. Set \( k_i = k_0 \forall i \in k \)

2. Calculate \( A_{r_0} = P^T A(k_0) P \)

3. For each zone \( j \in \Omega \), where \( \Omega \) is the model space
   a. Set \( k_j := k_0 + \Delta k = k' \)
   b. Calculate \( \hat{A}_j = P^T A(k_j) P \), where \( k_j \) is the hydraulic conductivity vector with all values set to \( k_0 \) except the \( j^{th} \) value \( (k_j) \), which is set to \( k' \)
   c. Calculate \( A'(z_j) = \hat{A}_j - \hat{A}_0 \)
   d. Set \( k_j := k_0 \)
4. Calculate the correction factors

\[ c_1 = \frac{1}{k' + k_0 + \|k_0, k'\|} \]

\[ c_2 = k' - 2k_0 + \|k', k_0\| - |k' - k_0| \]

5. For each border \( b_k \in \Omega \), where \( b_k \) is the border between zones \( l \) and \( m \)

   a. Set \( k_i := k', k_m := k' \)

   b. Calculate \( \hat{A}_b = P^T A(b_k) P \), where \( K_{b_k} \) is the hydraulic conductivity vector with

   all values set to \( k_0 \) except the \( l \) and \( m \)th values \((k_i \text{ and } k_m)\), which are set to \( k' \)

   c. Calculate \( A''(b_k) = c_i \left( \hat{A}_b - A'(z_k) - A'(z_m) - \hat{A}_0 \right) \)

   d. Set \( k_i := k_0, k_m := k_0 \)

6. For each zone \( j \in \Omega \)

   a. Correct \( A'(z_j) := A'(z_j) - c_2 \left( \sum_{k=1}^{n} A''(b_k) \right) \)

   where \( b_k \) is a zone that borders on zone \( j \) and \( n_l \) is the number of zones that border on zone \( j \)

Once the Fast Algorithm has isolated the various sub-matrices through the application of the decomposition step, it is able construct a reduced system matrix for any desired parameter combination by applying the construction step as shown below.

1.1.2.2. Construction Step
1. For each border $b_k \in \Omega$
   
   a. Calculate $k_b = \|k_i, k_i\|$ 

2. Calculate 

   $$\Phi(k) = A(k) = \sum_{i=1}^{n_z} k_i A'(z_i) + \sum_{b=1}^{n_b} k_b \ell b''(b_b),$$

   where $k_i$ is the hydraulic conductivity in zone $i$, $n_z$ is the number of zones in the model, 
   
   $k_b$ is the hydraulic conductivity across the $b^{th}$ boundary between two zones in the model, 

   and $n_b$ is the number of boundaries between two zones.

Though demonstrated on a one-dimensional finite difference model, the Fast Algorithm is applicable without modification to finite element models, two- or three-dimensional models, and models with varying distances between nodes (or element size). It may also be applied to interpolated hydraulic conductivity fields with the caveat that, depending on how the field is interpolated, there may be no way to correct the approximation in Eq. (9). However, in those cases, by the nature of the model construction, the error will be small. Though the decomposition step is computationally expensive, requiring multiple projections, it is carried out offline and only the computationally inexpensive construction step must be carried out online to construct a new reduced system matrix. The FLOP count of Eq. (10) is $O(n_p^3)$ and is significantly cheaper than performing the projection in Eq. (7). This cost reduction and the fact that it is black-box (i.e., it may be applied to any groundwater modeling software as long as the system matrix may be extracted) vastly expands the applicability of POD model reduction by making it more efficient to utilize complex models.
1.2. Experimental Design

One of those areas in which complex models are required is experimental design. Chapters 3-5 contain detailed discussions of experimental design that are summarized here. Experimental design is the field of study that seeks to conduct experiments in an optimal fashion in order to gain the maximum amount of some type of information. During the 1920s, statistics became the primary means to define experimental design. As such, modern experimental design takes advantage of many statistical elements, such as the Jacobian matrix \( J_d \) which contains information on how observations of a system will vary with respect to changes in system parameters of interest (i.e., the observation sensitivities). The Jacobian matrix is defined as:

\[
J_d = \begin{pmatrix}
\frac{\partial o_1}{\partial \theta_1} & \cdots & \frac{\partial o_1}{\partial \theta_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial o_{n_{\text{obs}}}}{\partial \theta_1} & \cdots & \frac{\partial o_{n_{\text{obs}}}}{\partial \theta_N}
\end{pmatrix}
\]

where \( n_{\text{obs}} \) is the total number of observations taken (which may vary spatially and temporally), \( N \) is the total number of system parameters of interest, and \( \frac{\partial o_i}{\partial \theta_j} \) is the sensitivity of the \( i^{th} \) observation \( (o_i) \) to changes in the \( j^{th} \) parameter \( (\theta_j) \). A full Jacobian matrix would contain the sensitivities of an infinite number of observations to each system parameter, which is obviously impossible to obtain. The task of experimental design is:

1. Given a set of parameters of interest and (generally) a set number of possible observations
2. Distribute the observations to maximize the information gained about the system.

To do this, experimental design assumes a model and then collects the sensitivities of potential observations in the information matrix \( F \)

\[
F = J_d^T J_d W
\]

where \( W \) is a user-specified weighting matrix. Under the assumptions that: 1) a least-squares error criterion is used for parameter estimation; 2) the observation errors are uncorrelated with equal variance; and 3) \( W \) is the identity matrix, the information matrix defined in Eq. (12) is equivalent to the inverse of the covariance matrix of the estimated parameters \( C(\theta) \)

(Cleveland & Yeh, 1990; Kutner, Nachtsheim, & Li, 2004)

\[
C(\theta) = \\
\begin{pmatrix}
\sigma^2(\theta_1) & \sigma(\theta_1, \theta_2) & \cdots & \sigma(\theta_1, \theta_n) \\
\sigma(\theta_1, \theta_2) & \sigma^2(\theta_2) & \cdots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
\sigma(\theta_n, \theta_1) & \cdots & \sigma(\theta_n, \theta_{n-1}) & \sigma^2(\theta_n)
\end{pmatrix}
\]

A set of potential observations is referred to as a “design”, and once the information matrix for a design has been collected, experimental design measures the information matrix to determine how much information that design will collect. The optimal design is the one that obtains the maximum information, and a design’s “optimality” is a reference to how close it is to the optimal design. A design’s optimality is relative, so the measure used to concretely describe it is “efficiency”, which is defined as the ratio of a design’s score (how much information it would be able to gather) to the score of the optimal design.
In general, there are two types of information that may be gathered about a system. The first type is information regarding the system parameters $\theta$. Designs optimized to collect this type of information will be designed such that collected observations will be able to minimize the uncertainty in identifying system parameters (e.g., hydraulic conductivity or forcing). The second type of information that may be gathered is related to the system response $\Psi$. This information seeks to minimize the uncertainty of the predicted state of the system at key locations and times, or all locations, at all times. While theoretically gathering one type of information would result in a corresponding understanding of the other type of information, this is not always the case in practice because of scarcity of information. Depending on what type of information is being sought, different optimality criteria may be chosen. These criteria are different measures of a design’s score and are uniquely related to an objective function. The objective function defines the design that is optimal with respect to that criterion. Table 1 provides a list of some of the most commonly used optimality criteria, what type of information it seeks, its objective function, and a description of what an optimal design achieves.

**Table 1. Optimality Criteria**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Information Class</th>
<th>Objective Function $\left(\phi, (\omega)\right)$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\theta$</td>
<td>$\max_\omega \left(\text{trace}(F)\right)$</td>
<td>Maximizes total information</td>
</tr>
<tr>
<td>E</td>
<td>$\theta$</td>
<td>$\max_\omega \left(\min_i \lambda_i(F)\right)$</td>
<td>Maximizes unique information</td>
</tr>
<tr>
<td>D</td>
<td>$\theta$</td>
<td>$\max_{\omega} \det(F)$</td>
<td>Maximizes uncorrelated information</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>G</td>
<td>$\Psi$</td>
<td>$\min_{\omega} \max_i \left( j_{d,i} \left( F^{-1} \right) \left( j_{d,i} \right)^T \right)$</td>
<td>Minimizes the maximum prediction error</td>
</tr>
<tr>
<td>I</td>
<td>$\Psi$</td>
<td>$\min_{\omega} \left( \text{mean} \left( \forall i \left( j_{d,i} \left( F^{-1} \right) \left( j_{d,i} \right)^T \right) \right) \right)$</td>
<td>Minimizes the average prediction error</td>
</tr>
</tbody>
</table>

where $j_{d,i}$ is the $i^{th}$ row in $J_d$, $\lambda_i$ is the $i^{th}$ eigenvalue of $F$, $\omega$ is a design, and $J_d$ is the Jacobian for that $\omega$. Chapter 5 offers a more detailed discussion on how these are related to one another and how implementation challenges may be overcome. It should be noted that $A$-optimality as defined in Table 1 is technically maximum-information-optimality, not $A$-optimality. The true definition of $A$-optimality would be $\text{trace}(F^{-1})$ which does not equal $\frac{1}{\text{trace}(F)}$, however, it may be proved that the optimal maximum-information design is the optimal $A$-design. Therefore “A-optimality” is used because the term is more widely understood.

Regardless of what objective function is chosen, the experimental design problem may be formulated as the following optimization problem:

$$\text{Optimize} \quad \omega \in \Omega \quad (\text{Objective Function})$$

$$\text{s.t. Constraints}$$
where $\Omega$ is the set of all possible designs, and the constraints might include limitations on the design size (number of observations), location, or distribution of $\omega$. In the context of groundwater modeling, many times this optimization problem is non-convex, non-differentiable, and requires a combinatorial search to solve. Practically speaking, this search would be infeasible to perform with traditional mathematical programming techniques, given the hundreds, or thousands, of nodes in a realistically-scaled groundwater model. Heuristic searches can be a useful tool to overcome this challenge (Montas HJ, Mohtar RH, Hassan AE, 2000) as exemplified by Genetic Algorithms (GAs) which have been employed to solve groundwater experimental design problems in the past (Babbar-Sebens & Minsker, 2010; McPhee & Yeh, 2006a; Reed, Minsker, & Valocchi, 2000; Wu J, Zheng C, 2005). Numerous other types and modifications of heuristic searches exist, each with their own advantages and disadvantages. One limiting factor to utilizing heuristic searches is that, in general, they require coupling to a system model which may be called tens, if not hundreds, of thousands of times during the course of the search. This would be infeasible if the system model is a realistically-scaled groundwater model. To overcome this, the original system model may be replaced by a surrogate model that is less computationally expensive to call. Chapters 3-5 demonstrate how POD reduced models, with the Fast Algorithm improvement as applicable, fill this need.

The following chapters are reprints of papers submitted to various peer-reviewed journals detailing research in improving POD model reduction with the Fast Algorithm (Chapter 2, published in Advances in Water Resources), applying POD reduced models to experimental design (Chapters 3 and 4, published in Water Resources Research and Advances in Water Resources respectively), and understanding experimental design better in the context of groundwater modeling (Chapter 5, submitted to Water Resources Research).
Chapter 2

A Proposed Fast Algorithm to Construct the System Matrices for a Reduced-Order Groundwater Model (T.T. Ushijima & Yeh, 2017)

Abstract

Past research has demonstrated that a reduced-order model (ROM) can be two-to-three orders of magnitude smaller than the original model and run considerably faster with acceptable error. A standard method to construct the system matrices for a ROM is Proper Orthogonal Decomposition (POD), which projects the system matrices from the full model space onto a subspace whose range spans the full model space but has a much smaller dimension than the full model space. This projection can be prohibitively expensive to compute if it must be done repeatedly, as with a Monte Carlo simulation. We propose a Fast Algorithm to reduce the computational burden of constructing the system matrices for a parameterized, reduced-order groundwater model (i.e., one whose parameters are represented by zones or interpolation functions). The proposed algorithm decomposes the expensive system matrix projection into a set of simple scalar-matrix multiplications. This allows the algorithm to efficiently construct the system matrices of a POD reduced-order model at a significantly reduced computational cost compared with the standard projection-based method. The developed algorithm is applied to three test cases for demonstration purposes. The first test case is a small, two-dimensional, zoned-parameter, finite-difference model; the second test case is a small, two-dimensional, interpolated-parameter, finite-difference model; and the third test case is a realistically-scaled, two-dimensional, zoned-parameter, finite-element model. In each case, the algorithm is able to accurately and efficiently construct the system matrices of the reduced-order model.
2.1. Introduction

2.1.a. Model Reduction

In recent years, the trend of research has been to construct highly discretized simulation models that are computationally expensive for even one simulation run. At the same time, many techniques require repeated model calls, for example solving large-scale optimization problems, particularly those dealing with uncertainty. These competing needs affect not just groundwater research but all hydrology research fields, for example in the areas of planning and management (Baú & Mayer, 2006; Sreekanth, Moore, & Wolf, 2016) or monitoring network design (Luo, Wu, Yang, Qian, & Wu, 2016). Unfortunately, as groundwater models grow in complexity, so does the computational burden of running them. When only a few runs are required, this computational burden is not a major concern; however, when a groundwater model must be called repeatedly, for example in a Monte Carlo simulation, the computational burden may render a problem infeasible to solve because the model must be called tens if not hundreds of thousands of times (Babbar-Sebens & Minsker, 2010; Reed et al., 2000; T.T. Ushijima & Yeh, 2015; Timothy T. Ushijima & Yeh, 2013). This computational burden also leads to an inability of a heuristic search to find a near-optimal solution in a reasonable timeframe. This inability is one of the major obstacles to the general use of evolutionary optimization schemes (Maier et al., 2014). To alleviate the computational burden of running large-scale models, a number of methods have been proposed (Asher et al., 2015), including reduced-order models (ROMs). These low-dimensional, surrogate models seek to reproduce the results of complex models at a much reduced computational cost (Antoulas et al., 2001). In addition, the literature shows that ROMs can be two-to-three orders of magnitude smaller than the original model and run considerably faster with acceptable error. Projection-based methods are often used to construct
an ROM. Some of these methods project the groundwater model onto its null space to quantify uncertainty (Doherty & Christensen, 2011); others, such as Proper Orthogonal Decomposition (POD), project onto an orthonormal subspace that spans the model’s solution space (Asher et al., 2015). A final example is the Dynamic Emulation Modeling approach (Castelletti, Galelli, Ratto, Soncini-Sessa, & Young, 2012; Castelletti, Galelli, Restelli, & Soncini-Sessa, 2012), which projects onto a subspace that has some real-world significance.

2.1.b. Proper Orthogonal Decomposition

The theory behind solution space projection methods states that if the solutions from a groundwater model exist in some space \( V \in \mathbb{R}^{N_n} \), where \( N_n \) is the dimension of the original model, there exists some subspace \( \lambda \in \mathbb{R}^{n_p} \) such that \( \text{Range}(\lambda) = \text{Range}(V) \) and \( n_p \leq N_n \). If \( n_p \ll N_n \), and the original (often referred to as the full) groundwater model can be projected onto \( \lambda \), then solutions to the original groundwater model can be found with much less computational effort. POD has been studied extensively in the past (Cazemier et al., 1998; Kowalski & Jin, 2003; Willcox & Peraire, 2002) and many studies have demonstrated that a POD reduced-order groundwater model significantly decreases the cost of running the model (McPhee & Yeh, 2008; Siade et al., 2010). Examples of applications of POD include predictive groundwater models (Boyce & Yeh, 2014; McPhee & Yeh, 2006b; Pasetto et al., 2013; Siade et al., 2010; Vermeulen, Stroet, & Heemink, 2006), solving the inverse problem of parameter estimation (Liu, Zhou, Birkholzer, & Illman, 2013; Siade et al., 2012), and modeling of variable density flow and transport processes (Li, Chen, Hu, & Navon, 2013). These studies provide an excellent background on the development and application of POD to groundwater models, described briefly here.
The governing partial differential equation (PDE) for three-dimensional flow in a confined, anisotropic aquifer is:

\[
\frac{\partial}{\partial x} \left( K_x \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial h}{\partial z} \right) = S_x \frac{\partial h}{\partial t} + f
\]  

(13)

where \( K_i \) is the hydraulic conductivity along the \( i^{th} \) coordinate axes, \( h \) is the hydraulic head, \( S_y \) is the specific storage, \( t \) is time, and \( f \) is a collection of all the forcing terms (e.g., pumping or recharge) (Bear, 1979). Given Eq. (13), we apply finite-difference or finite-element approximations to this PDE to produce the following set of linear ordinary differential equations (ODEs):

\[
B \frac{dh}{dt} + A(k)h = q
\]

where \( B \in \mathbb{R}^{N_x \times N_x} \) is the mass matrix, \( A(k) \in \mathbb{R}^{N_x \times N_x} \) is the stiffness matrix, and \( q \in \mathbb{R}^{N_x} \) is a vector of the forcing terms. When we approximate the time derivative using a backward finite-difference, this model can be solved over time as:

\[
(B + A(k))h = \tilde{A}(k)h = \tilde{b}
\]  

(14)

where \( k \in \mathbb{R}^{N_x} \) is a vector of the hydraulic conductivity in each node of the finite-difference (or finite-element) approximation, the system matrix \( \tilde{A}(k) \in \mathbb{R}^{N_x \times N_x} \) depends on \( k \), and \( h \in V \). Eq. (14) is referred to as the full model and its system matrix depends on many properties (e.g., specific storage or temporal or spatial discretization). In this paper, we will focus on its dependence on \( k \).
POD begins by finding some matrix $P \in \mathbb{R}^{N \times n_p}$ whose columns form an orthonormal basis for $V$. The following multiplication is then performed:

$$P^T \hat{A}(k) Pr = P^T \hat{b}$$  \hspace{1cm} (15)

where $P^T \hat{A}(k) P \in \mathbb{R}^{n_p \times n_p}$ is the system matrix for the reduced model and $r \in \mathbb{R}^{n_p}$ is the state vector of the reduced model. When $n_p \ll N$, we are able to solve this reduced model (Eq. (15)) at a much reduced computational cost compared to the full model (Eq. (14)). The solutions from the reduced model $(r)$ can be projected back onto the full model space $(V)$ to approximate the solutions from the full model $(h)$:

$$h \approx Pr$$

The accuracy of this approximation depends on the quantity and quality of the information that is captured in the $P$ matrix. Excellent research has been published on ways to construct $P$ to lead to accurate approximations of $h$ within a user specified error tolerance (Pasetto et al., 2013; Siade et al., 2010). It has been shown that methods exist that allow us to construct $P$ in a way that leads to good approximations of $h$ independent of changes in model forcing (Siade et al., 2010) and/or model parameters (e.g., $k$) (Pasetto et al., 2013; Vermeulen et al., 2004). In general, these methods rely on taking $n_p$ realizations of the full model to capture the maximum information that can be gained about the full model.

The ability to construct a reduced model with an acceptable error allows us to solve previously infeasible types of problems (Boyce & Yeh, 2014; T.T. Ushijima & Yeh, 2015; Timothy T. Ushijima & Yeh, 2013). However, a major drawback of POD model reduction is the
computational burden of constructing the system matrix for the reduced model. As Eq. (15) shows, the system matrix for the reduced model \( (A_r(k)) \) can be constructed by calculating the following matrix multiplication:

\[
\tilde{A}_r(k) = P^T \tilde{A}(k) P
\]  

(16)

The FLOP (floating point operation) count of this matrix multiplication is \( O\left(N^n + n_p^3\right) \). Since we desire \( n_p \ll N_n \), the FLOP count is \( O\left(N^n\right) \). If we know \( k \), this construction could even be performed once; therefore, the computational burden is manageable if performed online (i.e., during a simulation). The construction even could be conducted offline (i.e., performed prior to the simulation) and stored, such that the only online computational burden is inputting the system matrix. Unfortunately, if the groundwater response needs to be simulated under different hydraulic conductivities (e.g., a Bayesian inverse problem (Boyce & Yeh, 2014), a Monte Carlo simulation (Pasetto et al., 2014), or worst-case scenario design (T.T. Ushijima & Yeh, 2015)), the simulation may require the online construction of the system matrix of the reduced model (referred to as the reduced system matrix) tens if not hundreds of thousands of times. Since the FLOP count of solving the reduced model is \( O\left(n_p^3\right) \), it is clear that the computational cost of constructing the reduced model dominates the computational cost of solving the reduced model and easily can be the dominant cost of the simulation.

This drawback of POD is rarely discussed in the literature. Most of the published literature focuses on the reduction in computational burden once the reduced model has been constructed, but seldom discusses the total computational cost (constructing and running the reduced model).
It is possible to employ certain methods to control the online computational cost; for example, we could limit the range of $k$ for online construction or construct $A_r(k)$ offline for some set values of $k$. Unfortunately, both of these methods are undesirable as they can limit the applicability of the simulation. One technique that has been developed to reduce the online computational burden of re-computing the reduced system matrices is the discrete empirical interpolation method (DEIM) (Chaturantabut & Sorensen, 2010). DEIM seeks to identify critical indices in the model space around which a new projection matrix can be developed that will reduce the row space of the model domain. When coupled with POD (which reduces the column space of the model domain), DEIM is able to make inexpensive online calculations and then interpolate the results over the full model domain, avoiding the expense of projecting the full system matrix. This technique has been applied to many studies in hydrology and groundwater, including those dealing with shallow water equations (Ştefănescu & Navon, 2013), flow through porous media (Ghasemi & Gildin, 2015), and solving the Navier-Stokes equations (Xiao et al., 2014). While this technique is useful, it requires an additional model reduction (DEIM) on top of the original model reduction (POD), each of which introduces additional error to the final reduced model.

In the following sections, we develop an algorithm that efficiently constructs the reduced system matrix online, eliminating the need to limit the range of $k$ or construct some pre-set $A_r(k)$ matrices offline. The innovation of this algorithm lies in its ability to construct the reduced system matrix external to the simulation model at a computational cost that is orders of magnitude less than the standard projection method. Because the algorithm is external to the simulation model, it can be applied to any existing groundwater model. The user does not have to understand how the original groundwater model is constructed or how the original system matrix
is formulated. In most instances, the algorithm is able to construct an exact reduced system matrix (i.e., not an approximation of Eq. (16)), and in all cases it is able to construct a close approximation.

2.2. Algorithm

2.2.a. Basis

We assume that the model parameter \( k \) is properly parameterized by either the zonation method or an interpolation method, and then seek to find some function \( \Phi(k) \) such that

\[
A_r(k) = \Phi(k)
\]  

(17)

where \( \Phi(k) \) can be solved more efficiently (i.e., solved at a much lower FLOP count) than Eq. (16) can be solved. Since \( A_r(k) \) is non-linear with respect to changes in \( k \), we use the first order Taylor series expansion to find a linear function that approximates \( A_r(k) \). As we will show below, we can predict the error resulting from this approximation therefore the algorithm calculates a linear correction term to add to the Taylor series expansion. The resulting equation \( \Phi(k) \) is linear and can be solved efficiently.

In the following sections, we first will explain the basis for the algorithm from which the general algorithm naturally will follow. We use a one-dimensional, confined, finite-difference groundwater model to explain the basic principles; however, the algorithm is applicable to two- or three-dimensional, finite-difference or finite-element models.

Using an implicit, finite-difference scheme at each time step, the following set of linear algebraic equations approximates the governing equation:
where $h_i$ is the head at the $i^{th}$ location and is the dependent variable, $q_i$ is the forcing in the $i^{th}$ location, $k_i$ is the hydraulic conductivity at the $i^{th}$ location, $\Delta t$ is the time discretization, $\Delta x$ is the spatial discretization, $S_s$ is the specific storage, $\|k_j, k_i\|$ is some norm between the hydraulic conductivity at the $i^{th}$ and $j^{th}$ neighboring locations, and $h'_i$ is the head at the $i^{th}$ location at the previous time step. Additional terms easily can be added to the model flow in additional dimensions. The first term on the left-hand side of this equation comes from the mass matrix $B$ and the other three terms come from the stiffness matrix $A(k)$. For illustration purposes, we focus our analysis on $k$ and for simplicity we ignore the contribution of the mass matrix from this point forward. To extend this algorithm to include the construction of the reduced mass matrix, one simply would apply the same algorithm to $S_s$.

While this algorithm can be applied to over-parameterized models (e.g., those in which each node or cell has an independent hydraulic conductivity), it is best applied to a model in which the model parameters are properly parameterized. We assume that the model parameters have been properly parameterized, as is done in solving the inverse problem of parameter identification, where parameterization is used to reduce the parameter dimension to a finite dimensional form in order to ensure a stable and unique solution of the inverse problem. In general, there are three methods that can be used to accomplish parameterization (Yeh, 2015). The first is the zonation method, in which the aquifer is divided into a number of zones and a constant parameter value is used to characterize each zone. The spatial distribution of the transmissivity is then represented...
by a finite number of constants (zonal transmissivity values). The second is the interpolation method, in which the parameter is approximated by an interpolation function. For example, if finite-elements are used as the interpolation function, the flow region of the aquifer is divided into a number of elements connected by nodes. The parameter then is approximated by a linear combination of the basis functions. Other interpolation schemes include spline, polynomial and kriging. The pilot point method is a technique that has been used in geostatistics for parameter interpolation (Certes & de Marsily, 1991; de Marsily, Lavedau, Boucher, & Fasanino, 1984; Doherty, 2003; Marsh LaVenue & Pickens, 1992; RamaRao, LaVenue, de Marsily, & Marietta, 1995). In this method, the parameter is kriged using sample points along with a set of pilot points whose values are determined by an inverse procedure. The third method of parameterization is a combination of zonation and interpolation (Tsai & Yeh, 2004). In our analysis, we assume that the model parameters have been properly parameterized by either the zonation method or an interpolation method. We first focus the algorithm’s application to zoned models and then discuss its application to interpolated models.

Under the assumption that we are ignoring the contribution of the mass matrix to the system matrix, we write the left-hand side of the full model in vector form as:

\[
\begin{pmatrix}
0 & \cdots & \frac{k_{i-1},k_i}{\Delta x^2} \\
& \ddots & \frac{k_i,k_{i+1}}{\Delta x^2} \\
& & \frac{k_{i+1},k_{i+2}}{\Delta x^2} \\
& & & \ddots \\
& & & & \frac{k_{n-1},k_n}{\Delta x^2} \\
& & & & & 0
\end{pmatrix}
\begin{pmatrix}
h_i \\
h_{i-1} \\
h_i \\
\vdots \\
\vdots \\
0
\end{pmatrix}
\] (18)
In zoned models, for node $i$ in the middle of a zone (i.e., where $k_i = \parallel k_{i-1}, k_i \parallel = \parallel k_i, k_{i+1} \parallel$), the corresponding row of $A(k)$ (the left-hand row vector in Eq. (18)) takes the following form:

$$
\begin{pmatrix}
0 & \cdots & \frac{k_i}{\Delta x^2} & -\frac{2k_i}{\Delta x^2} & \frac{k_i}{\Delta x^2} & \cdots & 0
\end{pmatrix}
$$

(19)

As noted before, this tri-diagonal matrix for one-dimensional flow easily can be expanded to models with higher dimensions. A two-dimensional model can be represented as a penta-diagonal matrix and a three-dimensional model as a hepta-diagonal matrix, each with coefficients similar to the one shown in Eq. (19).

If node $i$ lies on a boundary of a zone, one of the two following cases will hold:

1. $k_i = \parallel k_{i-1}, k_i \parallel \neq \parallel k_i, k_{i+1} \parallel$
2. $k_i = \parallel k_i, k_{i+1} \parallel \neq \parallel k_{i-1}, k_i \parallel$

Thus for two nodes that lie next to each other but in different zones, their corresponding rows of $A(k)$ can be represented as:

$$
\begin{pmatrix}
0 & \cdots & \frac{k_{zi}}{\Delta x^2} & -\frac{k_{zi}}{\Delta x^2} & \frac{k_{zi}}{\Delta x^2} & 0 & 0 & \cdots & 0
\end{pmatrix}
$$

$$
\begin{pmatrix}
0 & \cdots & 0 & \frac{k_{zi}}{\Delta x^2} & -\frac{k_{zi}}{\Delta x^2} & \frac{k_{zi}}{\Delta x^2} & 0 & \cdots & 0
\end{pmatrix}
$$

(20)

From Eqs. (19) and (20) we see that $A(k)$ is composed of two types of submatrices. The first type is dependent on the hydraulic conductivity in zone $i$ and can be expressed as:
\[
\hat{A}(k_i) = \begin{pmatrix}
0 & \cdots & 0 & \frac{k_i}{\Delta x^2} & -2\frac{k_i}{\Delta x^2} & \frac{k_i}{\Delta x^2} & \cdots & 0 & \cdots & 0 \\
0 & \cdots & 0 & \frac{k_i}{\Delta x^2} & -\frac{k_i}{\Delta x^2} & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \frac{k_i}{\Delta x^2} & 0 & \cdots & 0
\end{pmatrix}
\]

\[
= k_i \begin{pmatrix}
0 & \cdots & 0 & \frac{1}{\Delta x^2} & -2\frac{1}{\Delta x^2} & \frac{1}{\Delta x^2} & \cdots & 0 & \cdots & 0 \\
0 & \cdots & 0 & \frac{1}{\Delta x^2} & -\frac{1}{\Delta x^2} & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \frac{1}{\Delta x^2} & 0 & \cdots & 0
\end{pmatrix}
\]

where the first row represents a row in the full system matrix \((A(k))\) corresponding to a node in the middle of zone \(i\) (i.e., not on a boundary of zone \(i\)), the second row represents a row in \(A(k)\) corresponding to a node in zone \(i\) and bordering another zone, the third row represents a row in \(A(k)\) corresponding to a node in a zone other than \(i\) but bordering on zone \(i\), and \(A'(z_i)\) is the sensitivity of \(A(k)\) to changes in the hydraulic conductivity of zone \(i\).

The second type of submatrix depends on, what we term, the “border hydraulic conductivity” (i.e., the hydraulic conductivity across the border between zones \(i\) and \(j\)). It takes the form:
\[ \hat{A}(k_{i,j}) = \begin{pmatrix} 0 & \cdots & 0 & -\frac{\|k_i, k_j\|}{\Delta x^2} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & -\frac{\|k_i, k_j\|}{\Delta x^2} & 0 & \cdots & 0 \end{pmatrix} \]

\[ = \|k_i, k_j\| \begin{pmatrix} 0 & \cdots & 0 & -\frac{1}{\Delta x^2} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & -\frac{1}{\Delta x^2} & 0 & \cdots & 0 \end{pmatrix} \]

\[ = \|k_i, k_j\| A''(b_k) \]

where the two rows in \( \hat{A}(k_{i,j}) \) correspond to the second and third rows in \( \hat{A}(k_i) \) (i.e., a node in zone \( i \) that borders zone \( j \) and a node in zone \( j \) that borders zone \( i \)) and \( A''(b_k) \) is the sensitivity of \( A(k) \) to changes in the border hydraulic conductivity across border \( b_k \), which lies between zones \( i \) and \( j \). We refer to this submatrix as the “border sensitivity matrix.”

It is clear, then, that if we know how to decompose \( A(k) \) in this manner for some model, we can construct a new \( A(k) \) for any other \( k \) in a computationally efficient manner via:

\[ A(k) = \sum_{i=1}^{n_z} k_i A'(z_i) + \sum_{b=1}^{n_b} k_b A''(b_b) \]

\[ (23) \]

where \( k_i \) is the hydraulic conductivity in zone \( i \), \( n_z \) is the number of zones in the model, \( k_b \) is the hydraulic conductivity across the \( b^\text{th} \) boundary between two zones in the model, and \( n_b \) is the number of boundaries between two zones. Since we are trying to calculate \( P^T A(k) P \), we pre- and post-multiply Eq. (23) by \( P^T \) and \( P \), respectively, and after distributing the multiplication and applying the commutative property of scalar multiplication, we arrive at:
\[ P^T A(k)P = A_r(k) = \sum_{i=1}^{n_r} k_i P^T A'(z_i)P + \sum_{b=1}^{n_b} k_b P^T A''(b_{i,j})P \]

This approach can be replicated to find the sensitivity of the mass matrix \( B \) to changes in specific storage \( S_s \) or we can assume the mass matrix to be known, in which case \( B_r = P^T BP \) would only need to be calculated once. Under the assumption that the mass matrix is known, we can efficiently compute the system matrix of the reduced model (Eq. (16)) as:

\[ \tilde{A}_r(k) = \sum_{i=1}^{n_r} k_i P^T A'(z_i)P + \sum_{b=1}^{n_b} k_b P^T A''(b_{i,j})P + P^T BP \]

We then are faced with the task of deriving an algorithm that decomposes \( \tilde{A}_r(k) \), while treating the simulation as a black-box (i.e., without requiring the user to understand how the original \( \tilde{A}_r(k) \) was constructed).

### 2.2.b. Decomposition Step

The first step of the algorithm is to calculate the sensitivities of all the submatrices of \( A_r(k) \) by a first-order Taylor series expansion and difference approximation as follows:

\[
\frac{\partial f}{\partial \theta} \approx \frac{f(\theta + \Delta \theta) - f(\theta)}{\Delta \theta} \quad (24)
\]

where \( \frac{\partial f}{\partial \theta} \) is the change of function \( f(\theta) \) with respect to a change of parameter \( \theta \) and \( \Delta \theta \) is a small perturbation of \( \theta \). The algorithm evaluates the reduced system matrix \( \left( A_r(k) \right) \) for a baseline hydraulic conductivity \( (k_0) \), sequentially perturbs the hydraulic conductivity in each
zone to determine the zonal sensitivity matrices \((A'(z_i))\), and finally, sequentially perturbs the hydraulic conductivity in each pair of zones to determine the border sensitivity matrices \((A''(b_k))\). Since \(A_i(k)\) is non-linear with respect to changes in \(k\), Eq. (24) is insufficient to calculate \(\Phi(k)\); however, the algorithm calculates the correction factors necessary to add to Eq. (24) to correct for the non-linearity. The following algorithm describes the decomposition steps:

7. Set \(k_i = k_0 \forall i \in k\)

8. Calculate \(A_{r0} = P^TA(k_0)P\)

9. For each zone \(j \in \Omega\), where \(\Omega\) is the model space
   a. Set \(k_j := k_0 + \Delta k = k'\)
   b. Calculate \(A_{rj} = P^TA(k_j)P\), where \(k_j\) is the hydraulic conductivity vector with all values set to \(k_0\) except the \(j^{th}\) value \((k_j)\), which is set to \(k'\)
   c. Calculate \(A'(z_j) = A_{rj} - A_{r0}\)
   d. Set \(k_j := k_0\)

10. Calculate the correction factors

\[
c_1 = \frac{1}{\frac{k'}{k_0} + \|k_0, k'\|} \]

\[
c_2 = k' - 2k_0 + \|k', k_0\| - |k' - k_0| \]

11. For each border \(b_k \in \Omega\), where \(b_k\) is the border between zones \(l\) and \(m\)
a. Set \( k_i := k', k_m := k' \)

b. Calculate \( A_{rb} = P^T A(k_{b}) P \), where \( k_{b} \) is the hydraulic conductivity vector with all values set to \( k_0 \) except the \( l \) and \( m^{th} \) values \((k_i \text{ and } k_m)\), which are set to \( k' \)

c. Calculate \( A''(b_k) = c_1 (A_{rb} - A'(z_k) - A'(z_l) - A_{rb}) \)

d. Set \( k_i := k_0, k_m := k_0 \)

12. For each zone \( j \in \Omega \)

a. Correct \( A'(z_j) := A'(z_j) - c_2 \left( \sum_{k=1}^{n_j} A''(b_k) \right) \)

where \( b_k \) is a zone that borders on zone \( j \) and \( n_j \) is the number of zones that border on zone \( j \)

We note that the correction factors in Step 4 are not approximations. Rather, they are the true correction factors that correct the error introduced by taking a first order Taylor series expansion of \( A_r(k) \) and are derived directly from the algorithm.

2.2.c. Calculation Step

Once we have decomposed the system matrix we easily can construct a system matrix for any \( k \) by multiplying each zonal and border sensitivity matrix by its respective hydraulic conductivity and summing the results. The following algorithm describes the calculation steps:

3. For each border \( b_k \in \Omega \)

a. Calculate \( k_b = \|k_x, k_y\| \)
4. Calculate

\[ \Phi(k) = A(k) = \sum_{i=1}^{n_z} k_i A'(z_i) + \sum_{b=1}^{n_b} k_b A''(b_k) \] (25)

where \( k_i \) is the hydraulic conductivity in zone \( i \), \( n_z \) is the number of zones in the model,

\( k_b \) is the hydraulic conductivity across the \( b^{th} \) boundary between two zones in the model, and \( n_b \) is the number of boundaries between two zones.

Since each \( A'(z_i) \) and \( A''(b_k) \) has size \( n_p \times n_p \) the FLOP count of solving Eq. (25) is \( O(n_p^3) \).

We refer to the combination of the decomposition and the calculation algorithm as the “Fast Algorithm”. When we compare the cost of solving Eq. (25) \( O(n_p^3) \) to the cost of the standard projection method \( O(N_v^2) \) and the cost of solving the reduced model \( O(n_p^3) \), we recall that \( O(n_p^3) \ll O(N_v^2) \) and see that the Fast Algorithm is able to reduce the cost of the most expensive step so that it no longer dominates the total simulation cost.

2.2.d. Application to Interpolated-Parameter Models

While the Fast Algorithm is best applied to a zoned-parameter model, it also can be applied to an interpolated-parameter model. An interpolated-parameter model is one in which

\[ k_i = \sum_{j=1}^{n_o} \lambda_{ij} k_j \]
where $k_i$ is the hydraulic conductivity at the $i^{th}$ node, $k_j$ is the hydraulic conductivity at the $j^{th}$ data point (i.e., where the hydraulic conductivity is known or assumed), and $\lambda_{j,i}$ is the chosen basis function that relates $k_j$ to $k_i$. In such a model, there are no boundaries between zones of hydraulic conductivity; rather, there are smooth transitions where one node’s hydraulic conductivity is generally different but close in value to the hydraulic conductivities of all the neighboring nodes.

As noted before, while the Decomposition (2.b.) and Calculation (2.c.) procedures are designed to efficiently calculate Eq. (17) for a zoned model, the same procedures can be applied to an interpolated model. The accuracy of the algorithm may be slightly affected, however, because no correction factors have been found to correct the first order Taylor series approximation of an interpolated model (i.e., the Fast Algorithm is not always able to find a $\Phi(k)$ that satisfies Eq. (17)). This is a limitation in general, but it is not critical for two reasons. First, the Fast Algorithm is able to calculate Eq. (17) for models employing a linear (i.e., arithmetic) mean in Eq. (18) (e.g., finite-element models). Second, although the Fast Algorithm is not able to calculate Eq. (17) for models using non-linear norms, it will be able to accurately approximate it such that $A_r(k) \approx \Phi(k)$. We can make this claim because when dealing with various types of means (arithmetic, harmonic, etc.), for two numbers that are close together all the means are approximately equal; however, as the numbers diverge, the means also diverge from each other. Since an interpolated model does not have sharp interfaces between hydraulic conductivities, the coefficients in Eq. (18) will be approximately equal regardless of what norm is employed, something that is not necessarily the case with a zoned model. As a result, the error introduced
by employing a linear rather than a non-linear mean generally will be small and negligible compared to the error introduced by the model reduction.

2.3. Test Cases

After developing the algorithm, we test it against three different two-dimensional cases. The test cases were chosen to demonstrate the algorithm’s general applicability to groundwater modeling. The fact that two models (the first and third) utilize zoned-parameters and one (the second) uses interpolation to parameterize demonstrates the algorithm’s applicability to different parameterization schemes. Furthermore, the test cases demonstrate that the algorithm can be used with finite-difference models (the first and second) or finite-element models (the third). Finally, the test cases have vastly different scales. The first two test cases are small-scale models while the third test case is a large-scale (i.e., real-world-scaled) model based on a real-world model developed for an aquifer in the Oristano plain in west-central Sardinia, Italy (Cau, Lecca, Putti, & Paniconi, 2002; Siade et al., 2012). In the large-scale model, the horizontal dimensions, boundaries, and finite-element mesh are taken from the Oristano model, but the parameters, depth, and boundary conditions are synthetic. Because of the small number of nodes in their full models \( \left( N_n \right) \), the two small-scale test cases easily are solved and verified. However, the large-scale test case is solved by an existing model solver, called SAT2D (Paniconi & Putti, 1994). We treat SAT2D as a black-box model to test the proposed Fast Algorithm.

2.3.a. Two-Dimensional, Finite-Difference with Zoned-parameter Test Case

The first test case is a synthetic, two-dimensional, zoned, finite-difference model. Figure 1 shows the parameter zonation pattern and pumping well locations. (Figure 1).
Figure 1. Two-dimensional, finite-difference zoned test case

The test aquifer is 80 m x 75 m with a depth of 1 m and is discretized by the finite-difference scheme into cubic cells 1 m on each side, resulting in a model space of 5950 nodes (\(N_n\)). The aquifer is divided into four zones with zonal hydraulic conductivities as shown. \(S_s\) is set uniformly to 1 m\(^{-1}\) throughout the aquifer (a large \(S_s\) value is used to speed-up the time required to reach a steady state). Six pumping wells are placed in the flow region as shown. Constant head boundaries exist at both the left and right boundaries. The top and bottom boundaries are set as no-flow boundaries. The initial head is 0 m in the flow region (assuming 0 m is the top of the aquifer). We use finite-difference to discretize the aquifer and construct three models corresponding to three different methods of characterizing the hydraulic conductivity across the boundaries between the zones (harmonic, geometric, and arithmetic means). Table 2 shows how
the hydraulic conductivity for flow across the boundary of two zones (with hydraulic conductivities of \( K_1 \) and \( K_2 \), respectively) is calculated for each of these means.

\[
\text{Table 2. Equations for calculation of the mean}
\]

<table>
<thead>
<tr>
<th>Mean</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmonic</td>
<td>( \frac{2K_1K_2}{K_1 + K_2} )</td>
</tr>
<tr>
<td>Geometric</td>
<td>( \sqrt{K_1K_2} )</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>( \frac{K_1 + K_2}{2} )</td>
</tr>
</tbody>
</table>

For each of these full models, we use the methodology presented by Ushijima and Yeh (2015) to construct a reduced model that has a dimension of \( 10 \left ( n_p \right ) \), and evaluate matrix norm \( \left \| \right \| \) of the error between the reduced system matrix, calculated by projecting both the full system matrix \( \left ( \tilde{A}_r(k)_p \right ) \) as well as the reduced system matrix, calculated by the Fast Algorithm \( \left ( A_r(k)_F \right ) \) for each of the modeled means. Table 2 shows the error between the two reduced system matrices for each type of mean. As we see, the errors are negligible.

\[
\text{Table 3.} \left \| \tilde{A}_r(k)_p - A_r(k)_F \right \|
\]

<table>
<thead>
<tr>
<th>Harmonic</th>
<th>Geometric</th>
<th>Arithmetic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.58E-15</td>
<td>2.98E-15</td>
<td>2.73E-15</td>
</tr>
</tbody>
</table>
After verifying that the Fast Algorithm is able to find a $\Phi(k)$ that satisfies Eq. (17), we investigate the error that would be introduced if the Fast Algorithm were to employ a mean different from that employed by the full model (e.g., modeling border hydraulic conductivities with the harmonic mean in the full model but employing the arithmetic mean in the reduced model).

Figure 2–Figure 4 compare the errors introduced at steady state by the Fast Algorithm employing various means. In each figure, the upper left-hand sub-figure displays the steady state head in the aquifer as modeled by the full model employing the indicated mean, the upper right-hand sub-figure shows the error introduced by the Fast Algorithm employing the harmonic mean, the lower left-hand sub-figure presents the error introduced by the Fast Algorithm employing the geometric mean, and the lower right-hand sub-figure exhibits the error introduced by the Fast Algorithm employing the arithmetic mean. Figure 2 compares the errors if the full model employed the arithmetic mean, Figure 3 contrasts the errors if the full model used the harmonic mean, and Figure 4 assesses the errors if the full model used the geometric mean. As expected, in all cases the error introduced at steady state by the Fast Algorithm is minimized when the Fast Algorithm and full model employ the same mean; however, in all cases the error introduced by the Fast Algorithm employing a mean other than the mean used by the full model is virtually insignificant when compared with the error introduced by model reduction.
Figure 2. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the arithmetic mean.
Figure 3. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the harmonic mean.
Figure 4. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 1 full model that uses the geometric mean

We further analyze the introduced error by evaluating the error over time. Table 4 shows the Root Mean Square Error (RMSE) of head between the full model and the reduced model that results from employing various means to characterize the hydraulic conductivity across zone
boundaries (the error measurements were taken at the ten arbitrarily chosen time steps of 0.1, 0.3, 0.8, 2.5, 7.5, 21.5, 62, 180, 520, and 1500 days). In the table, the rows correspond to the mean the Fast Algorithm employed and the columns correspond to the mean the full model employed.

**Table 4. RMSE between the full and reduced model resulting from employing various means to characterize the border hydraulic conductivity**

<table>
<thead>
<tr>
<th>Reduced Model Mean</th>
<th>Full Model Mean</th>
<th>Geometric</th>
<th>Arithmetic</th>
<th>Harmonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric</td>
<td>6.56E-6 m</td>
<td>7.10E-6 m</td>
<td>3.13E-4 m</td>
<td></td>
</tr>
<tr>
<td>Arithmetic</td>
<td>1.62E-4 m</td>
<td>7.10E-6 m</td>
<td>8.50E-4 m</td>
<td></td>
</tr>
<tr>
<td>Harmonic</td>
<td>8.47E-5 m</td>
<td>7.10E-6 m</td>
<td>6.26E-6 m</td>
<td></td>
</tr>
</tbody>
</table>

As expected, the error is the smallest when the full model and reduced model employ the same mean; however, the error introduced by the Fast Algorithm employing a different mean is minimal.

After verifying that the Fast Algorithm is able to accurately reproduce the reduced model even when incorrect assumptions regarding the means are made, we then investigate the computational efficiency of the Fast Algorithm. Table 5 compares the computation time and Floating-Point Operation (FLOP) count required to construct a reduced system matrix by employing both the standard matrix projection method and the Fast Algorithm.
Table 5. Computation time and FLOP count comparison for two-dimensional, zoned finite-difference test case

<table>
<thead>
<tr>
<th>System matrix construction method</th>
<th>Computation Time (s)</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix Projection</td>
<td>0.002354</td>
<td>35,402,500</td>
</tr>
<tr>
<td>Fast Algorithm</td>
<td>0.000238</td>
<td>1,000</td>
</tr>
</tbody>
</table>

In both cases, the reduced system matrices were constructed in Python 3.3 running on an Intel i7-2.80 GHz processor. At this scale, the Fast Algorithm does not demonstrate a major computational time saving; however, there is a drastic reduction by orders of magnitude in the FLOP count. This reduction will be very significant when dealing with highly discretized models (i.e., models having tens or hundreds of thousands of nodes). After verifying the Fast Algorithm’s ability to satisfy Eq. (17), we create 500 aquifers with the same spatial zonation pattern but with different zonal hydraulic conductivities in order to evaluate the algorithm’s ability to reproduce reduced models independent of changes in $k$. The calculated $\| \tilde{A}_r (k)_p - \tilde{A}_r (k)_p \|$ for each of these 500 zonal hydraulic conductivity sets is at most $O(1 \times 10^{-20})$, which is less than the round-off error.

2.3.b. Two-Dimensional, Finite-Difference with Interpolated-parameter Test Case

To demonstrate the algorithm’s applicability to an interpolated-parameter model, we take the two-dimensional, finite-difference model from Section 3.a. and convert it from a zoned-parameter model to an interpolated-parameter model. Figure 5 shows the interpolated model in which there are nine basis points (locations of known hydraulic conductivity). The nodal hydraulic conductivity values are chosen at random and we interpolate the hydraulic conductivity field with linear, triangular, finite-element basis functions (Figure 5).
As with the zoned test case, we use three means (harmonic, arithmetic, and geometric) to characterize the hydraulic conductivity between nodes and employ Ushijima and Yeh’s methodology (2015) to construct reduced models for each by the standard projection method and the Fast Algorithm; each has a dimension of $10^p n_p$. Table 6 compares the FLOP count and computation time of the standard projected method and the Fast Algorithm for this interpolated-parameter model, constructed in Python 3.3 running on an Intel i7-2.80GHz processor.
Table 6. Computation time and FLOP count comparison for two-dimensional interpolated-parameter model test case

<table>
<thead>
<tr>
<th>System matrix construction method</th>
<th>Computation Time (s)</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix Projection</td>
<td>0.0008006</td>
<td>35,402,500</td>
</tr>
<tr>
<td>Fast Algorithm</td>
<td>0.0001675</td>
<td>1,000</td>
</tr>
</tbody>
</table>

As with the two-dimensional, zoned-parameter model test case, even though the computation time reduction is minor, the reduction of the number of required FLOPs is quite significant.

To test the accuracy of the Fast Algorithm constructed reduced model, in Table 7 we compare modeled head at 10 arbitrarily chosen simulation times (0.1, 0.3, 0.8, 2.5, 7.5, 21.5, 62, 180, 520, and 1500 days) and display the Root Mean Square Difference (RMSD) between the Fast Algorithm reduced model and full model, the Fast Algorithm and Projected reduced models, and the Projected reduced model and full model.

Table 7. RMSD between model results

<table>
<thead>
<tr>
<th></th>
<th>Projected vs. Full</th>
<th>Projected vs. Fast</th>
<th>Fast vs. Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Model Mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arithmetic</td>
<td>1.05E-04 m</td>
<td>1.24E-14 m</td>
<td>1.05E-04 m</td>
</tr>
<tr>
<td>Harmonic</td>
<td>1.05E-04 m</td>
<td>4.00E-07 m</td>
<td>1.05E-04 m</td>
</tr>
<tr>
<td>Geometric</td>
<td>1.05E-04 m</td>
<td>2.00E-07 m</td>
<td>1.05E-04 m</td>
</tr>
</tbody>
</table>

Note that RMSD is computed in the same manner as RMSE. We term it RMSD because when comparing the results of the Projected reduced model and the Fast Algorithm reduced model, we have no “true” result against which to compute an error. In the table, each row compares the
results for models employing the mean indicated in the left-most column, while each column compares two of these three respective models. In the first column, the projected-reduced model and full model comparison gives us our baseline of error introduced by the model reduction. The second column displays the comparison of the projected-reduced model and the Fast Algorithm-reduced model showing the additional error introduced by the Fast Algorithm assumptions. The final column compares the Fast Algorithm-reduced model and full model, giving us the combined total error introduced by the model reduction and Fast Algorithm assumptions when applied to an interpolated-parameter model. As expected, the error introduced by the Fast Algorithm assumptions is negligible for a full model employing the arithmetic mean and minor for full models employing the harmonic and geometric means. In all cases, the error introduced by the Fast Algorithm assumptions is orders of magnitude less than the error introduced by model reduction. Figure 6-Figure 8 compare the baseline error introduced by model reduction with the additional error introduced by the Fast Algorithm assumptions at the steady state condition. In each figure, the upper left-hand sub-figure shows the steady state head in the aquifer as modeled by the full model, the upper right-hand sub-figure displays the baseline error introduced by model reduction, the lower-right hand sub-figure presents the additional error introduced by the Fast Algorithm assumptions, and the lower left-hand sub-figure shows the total error introduced to this interpolated model by model reduction and the Fast Algorithm. Figure 6 compares the errors if the full model employs the arithmetic mean, Figure 7 presents the errors if the full model employs the harmonic mean, and Figure 8 displays the errors if the full model employs the geometric mean. As expected, Figure 6 (arithmetic mean) reveals that the error introduced by the Fast Algorithm is insignificant for this full model. Figure 7 and Figure 8 show that the error introduced by the Fast Algorithm is significant when compared with the error
introduced by model reduction (Projected Reduced vs. Fast Algorithm); however, the total error is insignificant relative to the full model (Projected Reduced vs. Full or Fast Algorithm vs. Full).

Figure 6. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the arithmetic mean.
Figure 7. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the harmonic mean.
Figure 8. Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 2 full model that uses the geometric mean.

To test the robustness of the Fast Algorithm, we construct 500 aquifers with the same spatial layout but different data point hydraulic conductivities. Since with an interpolated model we cannot find a $\Phi(k)$ that satisfies Eq. (17), we do not evaluate $\left\| \tilde{A}_r(k)_p - \tilde{A}_r(k)_r \right\|$. Rather, we compare the RMSE of head between the Fast Algorithm model and projected model with the RMSE of head between the projected and full model, allowing us to see the relative error.
introduced by the Fast Algorithm verses the model reduction. Figure 9 shows this RMSE comparison for ten arbitrarily chosen simulation times (0.1, 0.3, 0.8, 2.5, 7.5, 21.5, 62, 180, 520, and 1500 days), displaying the RMSE of head between a model built by the Fast Algorithm and one built using the standard projection method on the y-axis, and the RMSE of head between the full model and the standard projected model on the x-axis. In the figure, each color cluster corresponds to a different full model mean.

Figure 9. Comparison of the errors introduced by the Fast Algorithm and model reduction for full, interpolated-parameter models using various means
As the figure shows, the error introduced by the Fast Algorithm approximation is orders of magnitude less than the error introduced by model reduction and, in the case of the arithmetic mean, very close to the round-off error, as expected.

2.3.c. Large-Scale Two-Dimensional, Finite-Element Zoned Test Case

In the final test case, we demonstrate the algorithm’s applicability to a realistically-scaled, finite-element, zoned-parameter model. In this case, the model solver is treated as a black-box. The test case is an 11 km x 9 km, two-dimensional, synthetic aquifer based on a real world finite-element model of an aquifer in the Oristano plain in west-central Sardinia, Italy (Cau et al., 2002; Siade et al., 2012) (Figure 10). This test case adopts the two-dimensional, finite-element mesh from the Oristano model; but the parameters, boundary conditions, and model depth are synthetic.
The model has \( \sim 30,000 \) nodes \( \left( N_n \right) \) with seven hydraulic conductivity zones and 20 pumping wells (as shown) and is solved using SAT2D, a black-box, finite-element groundwater model solver. This is the same finite-element test case used by Ushijima and Yeh (2015). We follow their methodology to build a reduced model that has a dimension \( 10^3 \left( n_p \right) \). Table 8 shows the comparison of computational cost for constructing the reduced system matrix by the standard matrix projection method and the Fast Algorithm.
Table 8. Computation time and FLOP count comparison for the large scale, two-dimensional, finite-element test case

<table>
<thead>
<tr>
<th>System matrix construction method</th>
<th>Computation Time (s)</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix Projection</td>
<td>1.11</td>
<td>900,000,000</td>
</tr>
<tr>
<td>Fast Algorithm</td>
<td>&lt;10E-9</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

Both reduced system matrices were constructed in C++11 running on an Intel i7-2.80GHz processor. The computational runtime of the reduced model for a time period of 50 days is 0.027 s, while the runtime for the full model is 64 s. For this large-scale, real-world problem we clearly see the advantages of employing the Fast Algorithm as it is able to significantly reduce the computational cost of constructing the reduced system matrix.

As with the two small-scale test cases, we then verify the robustness of the Fast Algorithm. We randomly select 500 $k$'s for the seven hydraulic conductivity zones and construct reduced models for each. For each set of hydraulic conductivities, we measure the error between the system matrix constructed by both the standard projection method $\left(\bar{A}_r(k)_r\right)$ and the Fast Algorithm $\left(\bar{A}_r(k)_F\right)$. For each of the 500 sets, the $\left|\bar{A}_r(k)_r - \bar{A}_r(k)_F\right|$ is at most $O(1\times10^{-20})$, which less than the round-off error. We select one of these random $k$'s (shown in the second column of Table 9) to test the accuracy of the reduced model built by the Fast Algorithm (note that the specific storage and depth were randomly selected once and held constant for all 500 $k$'s).

Table 9. Model zonal parameters

<table>
<thead>
<tr>
<th>Zone</th>
<th>Hydraulic Conductivity</th>
<th>Specific</th>
<th>Depth</th>
</tr>
</thead>
</table>

52
The finite-element formulation only leads to a model that uses the arithmetic mean to characterize hydraulic conductivity between nodes with differing hydraulic conductivities, so only one set of comparisons of error in head is made, as Figure 11 shows. This results from the fact that when the full model employs the arithmetic mean (e.g., a finite-element model), the border sensitivity matrices calculated by the Fast Algorithm cancel each other out. For this reason, the error is independent of the mean used by the reduced model to characterize the border hydraulic conductivities. The upper left-hand sub-figure shows the steady state head in the aquifer as modeled by the full model employing the arithmetic mean, the upper right-hand sub-figure displays the error introduced by the Fast Algorithm employing the harmonic mean, the lower left-hand sub-figure exhibits the error introduced by the Fast Algorithm employing the geometric mean, and the lower right-hand sub-figure presents the error introduced by the Fast Algorithm employing the arithmetic mean. As the figure shows, in all cases the absolute error in the reduced model is very small and is the largest at maximum drawdown. Thus the relative error introduced by the model reduction is minimal. As discussed by Ushijima and Yeh (2015) for this

<table>
<thead>
<tr>
<th></th>
<th>(m/day)</th>
<th>Storage (m⁻¹)</th>
<th>(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13.188</td>
<td>1.2E⁻⁵</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>5.936</td>
<td>2.7E⁻⁵</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>1.986</td>
<td>1.0E⁻⁵</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>7.947</td>
<td>1.7E⁻⁵</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>13.502</td>
<td>2.1E⁻⁵</td>
<td>96</td>
</tr>
<tr>
<td>6</td>
<td>11.982</td>
<td>3.4E⁻⁵</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>3.54</td>
<td>1E⁻⁵</td>
<td>104</td>
</tr>
</tbody>
</table>
reduced model, the error could be minimized by building a reduced model with a larger
dimension (i.e., $n_p > 103$).

![Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 3 full model that uses the arithmetic mean](image)

**Figure 11.** Comparison of the error at steady state resulting from assuming various means when reducing the Test Case 3 full model that uses the arithmetic mean

Since in this case the Fast Algorithm only produces one reduced model regardless of assumption of mean, the resulting RMSE is 0.0107 m for all three reduced models.

**2.4. Discussion and Conclusions**

We developed an efficient algorithm, called the Fast Algorithm, to calculate the system matrix for a POD reduced model. The Fast Algorithm extracts the necessary information from a full
model without having to understand the original full model; that is, the original full model is treated as a black-box. The key advantages of this algorithm are that it drastically reduces the computational cost of constructing the reduced system matrix and it easily can be implemented for any simulation model in which a large number of model calls with varying parameters is required, such as Monte Carlo simulations or heuristic searches.

In order to extract the necessary information, the algorithm uses results from the full model to derive sensitivity matrices that show how the system matrix of the reduced model would vary if various model parameters of interest (e.g., \( k \)) are changed. Because these sensitivity matrices are independent of changes to these model parameters of interest they only need to be derived once and then can be used to build new system matrices through the performance of a set of inexpensive scalar-matrix multiplications. By comparison, the standard method of constructing the system matrix of a reduced model requires performing multiple, expensive matrix-matrix multiplications.

The advantages of the Fast Algorithm are evident by comparing the FLOP count of the Fast Algorithm \( O(n_p^3) \) against the FLOP count for the standard projection method \( O(N_n^2) \). We have concluded that anytime the Fast Algorithm is applied \( O(n_p^3) < O(N_n^2) \). The reason is that, in model reduction, \( n_p \) is mainly controlled by the number of model parameters. While there is no exact relationship, the larger the number of model parameters, the larger the \( n_p \) we will need to construct an accurate reduced model. To justify the computation cost of first reducing the model and the resulting loss of accuracy, it is required that \( n_p << N_n \) so that a significant decrease in model runtime can be achieved. From the relationship of the FLOP counts of the Fast
Algorithm and standard projected method we certainly can make the claim that the Fast Algorithm is only useful when \( n_p \leq N_n^{2/3} \). Now, we should note that the \( O(N_n^2) \) FLOP count is the cost of a blind matrix multiplication of Eq. (16), and if we exploit the sparsity of the full system matrix, the FLOP count of calculating Eq. (16) is \( O(N_n^2 n_p^2) \). Using this FLOP count, we would argue that the Fast Algorithm is only useful when \( n_p \leq N_n \); however, we already have established the requirement that \( n_p \leq N_n^{2/3} \). Furthermore, since the Fast Algorithm only has a practical application in the context of model reduction, we can claim that \( n_p \ll N_n \) in all cases that the Fast Algorithm would be applied, therefore \( O(n_p^3) < O(N_n^2) \) and \( O(n_p^3) < O(N_n^2 n_p^2) \).

We demonstrated the Fast Algorithm by applying it to three different test cases. The first and second test cases were small-scale, finite-difference models. We used these test cases to demonstrate the Fast Algorithm’s applicability to both zoned- and interpolated-parameter models. In these test cases we demonstrated that the Fast Algorithm is able to reproduce the reduced model for a zoned model and make accurate estimates of the reduced model for interpolated models. We also verified that the error in model results introduced by making incorrect assumptions about the mean used in the finite-difference scheme is minimal compared to the error introduced by the model reduction. In the third test case we demonstrated the Fast Algorithm on a real-world-scaled, zoned-parameter, finite-element model. This test case reveals that the Fast Algorithm can be implemented with a finite-element model solved on a black-box groundwater model solver. This test case further demonstrated that with a finite-element model, there are no assumptions that need to be made about the mean used in the finite-element scheme. In each test case the Fast Algorithm was able to construct accurate reduced models at much
reduced computational cost in terms of both FLOPs and actual computation time. This is especially significant since the standard projection method takes advantage of the scarcity of the full system matrix when calculating Eq. (17). With each test case we verified that the Fast Algorithm is able to efficiently construct accurate reduced models independent of changes in model parameters. This ability allows the Fast Algorithm to be useful for increasing overall efficiency of methods, such as worst-case scenario optimization or Bayesian analysis, that require repeated model calls with various model parameters.
2.5. Bibliography


Chapter 3

Experimental design for estimating unknown groundwater pumping using genetic algorithm and reduced order model (Timothy T. Ushijima & Yeh, 2013)

Abstract

An optimal experimental design algorithm is developed to select locations for a network of observation wells that provides maximum information about unknown groundwater pumping in a confined, anisotropic aquifer. The design employs a maximal information criterion that chooses, among competing designs, the design that maximizes the sum of squared sensitivities while conforming to specified design constraints. The formulated optimization problem is non-convex and contains integer variables necessitating a combinatorial search. Given a realistic large-scale model, the size of the combinatorial search required can make the problem difficult, if not impossible, to solve using traditional mathematical programming techniques. Genetic algorithms (GAs) can be used to perform the global search; however, because a GA requires a large number of calls to a groundwater model, the formulated optimization problem still may be infeasible to solve. As a result, Proper Orthogonal Decomposition (POD) is applied to the groundwater model to reduce its dimensionality. The information matrix in the full model space then can be searched without solving the full model. Results from a small-scale test case show identical optimal solutions among the GA, integer programming, and exhaustive search methods. This demonstrates the GA’s ability to determine the optimal solution. Additionally, the results show that a GA with POD model reduction is several orders of magnitude faster in finding the optimal solution than a GA using the full model. The proposed experimental design algorithm is
applied to a realistic, two-dimensional, large-scale groundwater problem. The GA converged to a solution for this large-scale problem.
3.1. Introduction

Unknown forcing in an aquifer system can have drastic effects on the reliability of the results from a groundwater model. Therefore, construction of an accurate and useful groundwater model requires the accurate estimation of these forcing parameters. Unknown forcing parameters may include recharge, leakage, or evaporation loss, but in general the most common and significant forcing comes from unknown pumping rates. Unknown pumping may result from private wells that do not report pumping or from pumping wells where it is suspected that the reported pumping rates are incorrect. An inverse model designed to estimate unknown pumping requires observed data, the most important of which are observations of head (groundwater level). However, taking observations is expensive, time consuming, or difficult to obtain, particularly if aquifer parameters vary spatially. Consequently, inverse modeling always faces an observation scarcity problem. The practical application of the experimental design problem, then, is to design an observation network conforming to a specified set of constraints (such as allowable budget or spatial or temporal constraints) that will provide the maximum amount of information about the unknown forcing of interest.

3.1.a. Experimental Design

Experimental design covers the set of problems that involves performing experiments (in our case taking measurements) in a way that gains the maximum amount of useful information subject to a set of constraints. In practice we generally set a limit on the number of feasible experiments and then apply some scheme to distribute those experiments to maximize the amount of useful information gained. We then consider the relationship between the number of experiments and amount of useful information received by varying the limit on the number of experiments allowed. This description immediately raises some questions: What is useful
information? How do we quantify the amount of information gained? Answering these questions has been the topic of much research spanning diverse fields of study such as engineering, statistics, biology, and medicine, to name a few.

Since the 1920’s statistical methods have been established as the primary means for answering these questions. For example, we find useful information in the Jacobian matrix \( J_d \), defined as

\[
J_d = \begin{pmatrix}
\frac{\partial o_1}{\partial \theta_1} & \cdots & \frac{\partial o_1}{\partial \theta_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial o_{n_{obs}}}{\partial \theta_1} & \cdots & \frac{\partial o_{n_{obs}}}{\partial \theta_N}
\end{pmatrix}
\]

(26)

where \( n_{obs} \) is the total number of observations taken, which may vary spatially and temporally; \( N \) is the total number of parameters of interest; and \( \frac{\partial o_i}{\partial \theta_j} \) \((i = 1, \ldots, n_{obs}; j = 1, \ldots, N)\) is the sensitivity of the \( i^{th} \) observation \( (o_i) \) to changes in the \( j^{th} \) parameter \( (\theta_j) \). There are three methods which can be used to calculate the sensitivities of a system: (1) the parameter perturbation method, (2) the sensitivity equation method, and (3) the adjoint state method (Yeh, 1986). For convenience, we choose the parameter perturbation method as follows:

\[
\frac{\partial o_i}{\partial \theta_j} \approx \frac{\Delta o_i}{\Delta \theta_j} = \frac{o_i(\theta + \Delta \theta_j) - o_i(\theta)}{\Delta \theta_j}
\]

(27)

where \( \theta \) is a vector of the estimated parameter values, \( o_i(\theta) \) is the \( i^{th} \) model simulated value using the parameter values in \( \theta \), \( \Delta \theta_j \) is a small increment of the \( j^{th} \) parameter (called the
perturbation of \( \theta_j \), and \( \Delta \theta_j \) is a vector in which all element are zero except for the \( j^{th} \) element which is equal to \( \Delta \theta_j \) (Poeter, Hill, Banta, Mehl, & Chirstensen, 2005). Using the parameter perturbation method to estimate sensitivities for general models requires \((N+1)\) model calls (a baseline model solution and \( N \) model runs to estimate the sensitivities) (Yeh, 1986). In general, there are two requirements for making accurate estimates of \( \frac{\partial o_i}{\partial \theta_j} \) through the use of Eq. (27).

First, \( \Delta \theta_j \ll \theta \) and second, \( \theta \) must be close to the true parameter values. Note that given an infinite budget, observations can be taken at infinitesimally fine spatial and temporal resolutions. However, in the real world this is never the case; consequently, another way to formulate the experimental design is by addressing the problem of finding the elements of this theoretical infinitely large Jacobian matrix that should be included in the analysis. The second question mentioned above then arises: How do we quantify the amount of information contained in a particular Jacobian matrix?

To answer this question, we adopt the concept of the information matrix \( (I) \) from statistics as defined here:

\[
I = J_d^T * J_d * W
\]

where \( W \) is a user specified weighting matrix. Under the assumptions that a least-squares error criterion is used for parameter estimation, the observation errors are uncorrelated with equal variance, and \( W \) is the identity matrix. The information matrix defined in Eq. (28) is equivalent to the inverse of the covariance matrix of the estimated parameters (Cleveland & Yeh, 1990; Kutner et al., 2004). Given the definition in Eq. (28), the amount of information contained in any
given information matrix may be quantified by a number of methods. Commonly used methods are A-optimality, which seeks to minimize the trace of the covariance matrix; D-optimality, which seeks to minimize the determinant of the covariance matrix (Steinberg & Hunter, 1984); and E-optimality, which seeks to minimize the maximum eigenvalue of the covariance matrix (Steinberg & Hunter, 1984). Under the assumptions outlined above (i.e., that the information matrix is equivalent to the inverse of the covariance matrix of the estimated parameters), the A- and D- optimality criteria seek to maximize the trace and the determinant of the information matrix, respectively. In very general terms, A-optimality seeks to obtain the largest amount of information possible, while D-optimality seeks to balance the amount of information gained while minimizing the amount of covariance between observations. In the past, optimal design problems in the context of groundwater modeling have used both of these optimality criteria. A-optimality has been used for designing optimal observation networks for confined aquifer parameter estimation (Hsu & Yeh, 1989), transport parameter estimation (Cleveland & Yeh, 1990), and unconfined aquifer parameter estimation (Altmann-Dieses, Schloder, Bock, & Richter, 2002). D-optimality has been used for designing optimal pumping tests for parameter estimation (Nishikawa & Yeh, 1989), an optimal multi-objective observation network for parameter estimation and model discrimination (Knopman & Voss, 1989), and an optimal observation network for dispersion parameters (Catania & Paladino, 2009). Studies also have been performed to not only find the optimal design but to evaluate whether the amount of information gathered is useful for achieving some objective (McCarthy & Yeh, 1990) or if the information received is sufficient (Chang, Sun, & Yeh, 2005).

The concept of experimental design has been applied extensively to groundwater modeling; however, many studies have faced difficulty when solving for the optimal observation network
because of the combinatorial search required. In a realistic, highly discretized, large-scale groundwater model, referred to in this paper as the full model, there may be tens or hundreds of thousands of nodes. Because of this, the dimensionality of the search quickly becomes so large that it becomes impossible to solve through mathematical programming techniques such as integer programming or the Simplex method with relaxation. Consequently, other methods are required to solve this optimization problem. Throughout the years many different methods have been developed to solve large-scale optimization problems that cannot be solved or are difficult to solve through traditional mathematical techniques because of issues relating to the problems’ dimensionality, non-convexity, or non-differentiability. Genetic Algorithms (GAs) are one such set of techniques. GAs employ methods based on the concepts of evolution and survival of the fittest to search the feasible space for the optimal solution to a general optimization problem (Mitchell, 1998). In this study, the GA is built around a base GA code (GAlib) developed at MIT that has many built-in features suitable for use in the particular problem under consideration, such as support for real number genomes, easy adaptation, and the option to implement various “flavors” of GA (Wall, 1995).

GAs have been used in the past with various optimality criterion to develop optimal observation networks (Babbar-Sebens & Minsker, 2010; McPhee & Yeh, 2006b; Reed et al., 2000). However, many of these studies were challenged by the fact that GAs do not address the computational burden of the original model. In the combinatorial search examined, we found two available options. The first option requires the storage and accessing of an inordinately large amount of data resulting from the groundwater model’s spatial and temporal dimensions. Alternately, to avoid this burden, the second option requires calling the groundwater model numerous times as the GA evolves. The disadvantage of the second option is that when coupled
with any groundwater model with realistic spatial and temporal dimensions, this approach will be prohibitively slow because a GA may need to call the groundwater model hundreds or even thousands of times before the termination criterion (e.g., convergence) is met.

To address the issue of computation time, we apply Proper Orthogonal Decomposition (POD) to the groundwater model to reduce the model space (spatial dimension) and thus the computational burden of calling the model. POD has been shown to have the ability to reduce the dimension of a groundwater model by several orders of magnitude while maintaining over 99% accuracy (McPhee & Yeh, 2008). Siade et al. (2010) demonstrated that by applying POD, a groundwater model that originally contained over 200,000 spatial nodes could be reduced to a model containing only 10 spatial nodes, resulting in an approximately 1,000 times increase in speed of solving the model. Note that the temporal dimension remains untouched. However, with such a large decrease in the spatial dimension the time dimension becomes trivial. As a result of applying POD, calling the reduced model even thousands of times becomes inconsequential.

3.1.b. Confined Aquifer Groundwater-Flow Model

Three dimensional groundwater flow in a confined, anisotropic aquifer with pumping is described by the following PDE (Bear, 1979):

\[
\frac{\partial}{\partial x} \left( K_x \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial h}{\partial z} \right) - q = S_x \frac{\partial h}{\partial t}
\]  

(29)

with initial and boundary conditions
\[ h(x, y, z) = f_1(x, y, z) \]
\[ h(x, y, z, t) = f_2(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_1) \]
\[ q_n(x, y, z, t) = f_3(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_2) \]

where \( h \) is the hydraulic head [L]; \( K_x, K_y, \) and \( K_z \) are the hydraulic conductivities in the \( x, y, \) and \( z \) directions [L/T]; \( S_s \) is the specific storage [L\(^{-1}\)]; \( q \) is the specific volumetric pumping rate [T\(^{-1}\)]; \( q_n \) is the specific discharge normal to the flux boundary \((\Gamma_2)\) [T\(^{-1}\)]; \( \Gamma_1 \) is the fixed head boundary; \( f_1, f_2, \) and \( f_3 \) are known functions; L denotes the length unit (meters, feet, etc.); and T denotes the time unit (days, hours, etc.). Note that the above equation holds for any length and time unit as long as the chosen units are consistent.

Without loss of generality, we can change the state variable from head \((h)\) to drawdown \((s)\).

Drawdown is defined as the difference between the initial head before pumping \((H)\) and the head after pumping \((h)\) (i.e., \( s = H - h \)). After this linear transformation, the initial and Dirichlet boundary conditions are equal to zero. Following this change of state variable, the elements of the Jacobian matrix become \( \frac{\partial s_i}{\partial \theta_j} \) and the PDE of the governing equation can be discretized through finite difference approximations into a system of linear ODE’s:

\[ B \frac{ds_i}{dt} + As_i = q_i, \tag{31} \]

where \( s_i \) is a vector of dimension \( N_n \) of drawdown values at time \( t \), \( N_n \) is the total number of nodes in the discretized model, \( A \in \mathbb{R}^{N_n \times N_n} \) is the stiffness matrix, \( B \in \mathbb{R}^{N_n \times N_n} \) is the mass matrix, and \( q_i \in \mathbb{R}^{N_n} \) is a vector of sinks (pumping, recharge, evaporation, etc.) at time \( t \). Eq. (31)
is referred to as the full model. In most cases of interest matrices $A$ and $B$ are large, sparse, and positive definite (Siade et al., 2010) (note that, throughout the text, a bold face letter indicates a matrix or vector while a non-bold face letter indicates a scalar. For example, $s_t$ indicates the vector of all drawdown values at time $t$ while $s_{i,t}$ denotes the drawdown in the $i^{th}$ node at time $t$).

Though Eq. (31) is solved in terms of drawdown, head values ($h$) easily can be calculated by reversing the linear transformation $s = H - h$. POD is then applied to Eq. (31), obtaining a reduced system of equations:

$$B \frac{\text{d}\tilde{r}_t}{\text{d}t} + A r_t = q,$$

such that $\text{dim}(r_t) \leq \text{dim}(s_t)$ (Siade et al., 2010). However, for reduced models to be useful, $\text{dim}(r_t) \ll \text{dim}(s_t)$.

3.1.c. Proper Orthogonal Decomposition (POD)

Proper Orthogonal Decomposition has been a topic of much research (Cazemier et al., 1998; Kowalski & Jin, 2003; Willcox & Peraire, 2002). In addition, good summaries of applying POD to a groundwater-flow model exist in the literature (McPhee & Yeh, 2008; Siade et al., 2010, 2012; Vermeulen et al., 2004).

POD is based on the idea that a system of linear equations of dimension $N_n$ may be projected into a subspace of dimension $n_p$ such that $n_p \leq N_n$ and $\hat{s}_t = Pr_t$, where $\hat{s}_t \in \mathbb{R}^{N_n}$ is the approximation of the original state vector ($s_t$), $P \in \mathbb{R}^{N_n \times n_p}$ is the projection operator given by a matrix whose
columns form an orthonormal basis spanning the subspace \( V \subset \mathbb{R}^N \), and \( r_t \in \mathbb{R}^n \) is the state vector of the reduced space (Siade et al., 2010). This approximation can be expressed as

\[
\hat{s}_t = s_{nat,t} + \sum_{k=1}^{n_r} p_k r_{k,t}
\]

where \( r_{k,t} \) denotes the \( k^{th} \) element of \( r_t \); \( p_k \) is the \( k^{th} \) column of \( P \); and \( s_{nat,t} \) is the steady state condition of the drawdown without forcing (Vermeulen et al., 2004). Note that in general \( s_{nat,t} = 0 \) \( \forall t \). We construct the matrix \( P \) by running the full model and taking observations, sometimes called snapshots, of drawdown at all nodes in the full model. We then use these observations to create an orthonormal basis of a reduced model space that approximates the full model space (Siade et al., 2010). The reduced model’s performance is related to the quality of the snapshots (how well the snapshots represent the dynamics of the system). Siade et al. (2010) developed a method for finding an approximation of the optimal snapshot times, which is used to find the optimal snapshot set in this application. This set is then stored in \( X \in \mathbb{R}^{N_s \times n_r} \), where \( N_s \) is the number of nodes (or ODE’s) in the full model and \( n_r \) is the total number of snapshots taken. A number of methods then can be employed to construct the orthonormal basis of the reduced space (Shlizerman, Ding, Williams, & Kutz, 2012; Siade et al., 2010). We choose Singular Value Decomposition (SVD) for its computational simplicity compared to eigenvalue decomposition. To construct \( P \), we perform SVD on \( X \) such that

\[
X = P\Sigma V^T
\]

where \( P \) and \( V \) contain the left and right singular vectors of \( X \), respectively, and \( \Sigma \) is a diagonal matrix containing the singular values of \( X \). It can be shown that left singular vectors span the
same subspace as the principal vectors of $X$; therefore, to be consistent with terminology used in previous works, we refer to $P$ as a matrix containing the principal vectors of $X$. We then perform Principal Component Analysis (PCA) on the matrix $P$ to eliminate insignificant principal vectors (McPhee & Yeh, 2008; Vermeulen et al., 2004) so that the final projection matrix explains over 99.99% of the variance of the full model. The procedure for constructing $P$ can be summarized as follows:

1. For each pumping well (or a set of pumping wells whose ratio of pumping rates relative to each other is held fixed), perform a full model run with the well(s) set at a constant pumping rate, collect snapshots, and store them in the matrix $X$.

2. Perform SVD on $X$ such that $X = P \Sigma V^T$.

3. Eliminate insignificant principal vectors from $P$ such that $P \in \mathbb{R}^{N \times n_p}$, where $n_p$ is the number of principal components kept after PCA.

After constructing $P$, we use the Galerkin projection to reduce the full model by employing the following equation (Siade et al., 2010):

$$P^TBP \frac{dr_i}{dt} + P^TAPr_i = P^Tq_i$$ (33)

If we let $B = P^TBP$, $A = P^TAP$, and $q = P^Tq$, Eq. (33) takes on the form of Eq. (32) and becomes

$$B \frac{dr_i}{dt} + Ar_i = q_i,$$
with \( r_t \in \mathbb{R}^n \) (Siade et al., 2010). We refer to this formulation (Eq. (32)) as the reduced model. The solution to this set of ODE’s then can be approximated through a stable time stepping technique such as Implicit Euler or Crank-Nicolson or solved analytically through matrix exponential (Siade et al., 2010).

3.1.d. Genetic Algorithm

The GA searches the entire feasible space through the use of real number genomes, overlapping populations (making it a steady-state GA), and elitism. The GA indexes all possible observation well locations in the full model space (referred to as \( \Omega \)) and creates a population with individuals that have some feasible combination of observation well locations \( (\omega \subset \Omega) \). The GA evaluates the fitness (i.e., how well a particular individual optimizes the objective function) of these individuals by calling the reduced model to estimate the Jacobian matrix \( (J_d) \) and the information matrix \( (I) \) (Eq. (26) and Eq. (28), respectively). Maximizing the trace of \( I \), referred to as A-optimality in this paper, is chosen as the criterion for selecting the optimal individual. Thus the reduced model evaluates the following fitness function for each individual:

\[
z = \text{Tr}(J_d^T * J_d * W)
\]

It is important to show how the reduced model within the GA estimates the elements within the Jacobian matrix \( \left( \frac{\partial s_j}{\partial \theta_i} \right) \) to evaluate the fitness criterion of a particular design. Achieving a good approximation of the Jacobian requires some a-priori knowledge about the parameter values, which, in general, can be difficult to acquire. Since we want to gather information on unknown pumping rates, obtaining a-priori knowledge is not an issue because the estimated aquifer
sensitivities relate to pumping \( (q_i) \). Due to the linear relationship between changes in pumping \( (q_i) \) and changes in drawdown \( (s) \) (Eq. (31)), no a-priori knowledge of pumping is required to make accurate approximations of sensitivities in the Jacobian matrix \( \left( \frac{\partial s_i}{\partial q_{ij}} \right) \). In fact, we could use any values for \( q_i \) and \( \Delta q_{ij} \) in Eq. (27) and make accurate approximations. For simplicity, the reduced model then assumes \( q_i = 0 \ \forall i \); this results in the drawdown also being zero. Thus

\[
\frac{\partial s_i}{\partial q_{ij}} \approx \frac{s_i(\Delta q_{ij})}{\Delta q_{ij}}.
\]

where \( \Delta q_{ij} \) is the perturbation of the \( j^{th} \) pumping rate and \( s_i(\Delta q_{ij}) \) is the \( i^{th} \) simulated drawdown using \( \Delta q_{ij} \).

As a result, for each individual in a population, the GA passes the reduced model \( P \subset P \) (the rows of \( P \) corresponding to the elements in \( \omega \)). Then the reduced model calculates the reduced solution \( (R) \) at the observation times and makes a projection onto the full space such that

\[
J_{d_{ij}} = \frac{(PR)_{i,j}}{\Delta q_{ij}} \quad (35)
\]

By doing this, the GA evaluates the fitness of each individual (Eq. (34)) entirely in the reduced space, leading to vast a reduction in computational burden.

The following optimization problem shows the formulation for the GA:
\[
\text{max } \text{Tr}(J_d^T J_d W)
\]
\[
\text{s.t. } \sum_{i=\text{zone } j} x_i \leq 1 \ \forall j
\]
\[
\sum_{i=1}^{N_j} x_i \leq n_{\text{obswell}}
\]

where \(n_{\text{obswell}}\) is the total number of allowable observation wells; \(\text{zone } j\) indicates all the nodes in the \(j^{th}\) zone; \(N_j\) is the total number of nodes in the model; and \(x \in \mathbb{R}^{N_x}\) contains 0 and 1 binary variables (equal to 1 if an observation is taken at that node and 0 otherwise). In this study we chose \(W\) to be the identity matrix of size \(n_q\), where \(n_q\) is the total number of wells under consideration, and \(J_d\) is calculated via the reduced model. The GA achieves convergence when there is no deviation from the best solution and the best population solutions for \(800 \times (n_{\text{obswell}})\) iterations (i.e., convergence of solution). A flowchart of the GA with POD reduced model is shown in Figure 12.
Figure 12. Flowchart of GA
3.2. Test Cases

3.2.a. One-Dimensional Test Case

We developed a synthetic experimental setup similar to the ones used in previous papers related to POD reduction in groundwater modeling (McPhee & Yeh, 2006b; Siade et al., 2010) to validate the GA’s potential for finding the globally optimal experimental design (the setup is shown in Figure 13).

![Figure 13. 1-D Test Case Aquifer Setup](image)

The test aquifer is 100m long with a depth of 1m and a width of 1m and is divided into cubic cells that are 1m on each side. The specific storage \(S_s\) is \(\frac{1}{m}\), the hydraulic conductivity \(K\) is 15 \(\frac{m}{day}\) in nodes 1 through 50 (Zone 1) and 5 \(\frac{m}{day}\) in nodes 51 through 100 (Zone 2). The time step \(\Delta t\) is set to 0.1 days, the aquifer is modeled for 100 days, and \(s_{nat,j} = 0 \ \forall t\). Observations are taken arbitrarily at 0.5, 1, 3, 5, 10, 15, 25, 40, 55, and 90 days. At node 51 one pumping well pumps water from the aquifer continually throughout the simulation period at an unknown rate. Using SAT2D (Paniconi & Putti, 1994) the fully constructed model consists of a system of 303 equations because SAT2D requires three rows of 101 nodes to create a finite element mesh to
simulate the aquifer. Only the 101 nodes from the middle row are treated as “real” locations; the rest are “virtual” locations used to simulate the aquifer. After applying POD, the resulting reduced model contains only seven nodes and thus seven equations. When run on an Intel Core 2.40 GHz i5 CPU, the full model takes three seconds to complete one run while the reduced model completes one run in 0.012 seconds, an increase in speed of three orders of magnitude with virtually identical results as seen in Table 1.

Table 10. Comparison of the Trace of the Information matrices for the full and reduced models for the 1-D Test Case

<table>
<thead>
<tr>
<th>Full model Trace</th>
<th>Reduced model Trace</th>
<th>Absolute Error</th>
<th>Average Error</th>
<th>Maximum Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>478.964</td>
<td>458.921</td>
<td>20.043</td>
<td>0.0200</td>
<td>0.0042</td>
</tr>
</tbody>
</table>

Table 1 compares the trace of the full information matrix \( I \) obtained from the full model with that obtained from the reduced model. \( I \) contains information from all the nodes that may be used as observation well locations, at all observation times. Note that the full information matrix from the reduced model is obtained by first computing the reduced solution and then projecting the resulting Jacobian matrix onto the full space by using Eq. (35). Finally, Eq. (28) is used to calculate \( I \). While the absolute error seen in Table 10 may seem large, we note that the trace of the information matrix is the sum of the squared sensitivities at potential observations to changes in pumping. Accordingly, for this full information matrix, there are 101 potential locations, ten observation times, and one pumping well, yielding a total of 1,010 potential observations. Calculating error per location leads to the conclusion that for any one observation, the average error in the amount of information is only 0.020. In addition, we can see that the maximum
relative error, which is the maximum error for one well at one time, is less than half of one percent. Considering that the error in the information matrix includes the summation of squared errors from the full model space (101 nodes), we can conclude that the error in the information matrix as a result of model reduction is negligible.

For the experimental design problem the 101 nodes under consideration for potential observation well locations are divided into two zones, one containing nodes 1 through 50 and the other containing nodes 51 through 101. Two observation wells are allowed to be placed in the aquifer with the constraint that both cannot be located in the same zone. We then construct the full Jacobian matrix \( \mathbf{J} \) (Eq. (26)) for this aquifer, containing sensitivities for all the potential observation well nodes at all the simulated times. We then apply three methods to search for the globally optimal observation well locations. First, we perform an exhaustive search in which the fitness function (Eq. (34)) for all possible observation well location combinations (at the specified observation times) is evaluated. We then find the combination that results in the maximum feasible objective function value. Second, we formulate the experimental design problem as an integer programming problem solving for the optimal observation well locations. This optimization problem is shown as:

\[
\begin{align*}
\max_{\mathbf{x}} & \quad f^T \mathbf{x} \\
\text{s.t.} & \quad 0 \leq \mathbf{x} \leq 1 \\
& \quad \sum_{i=\text{zone } j} x_i \leq 1 \quad \forall \\
& \quad \sum_{i=1}^{N_i} x_i \leq n_{\text{obswell}}
\end{align*}
\]

where
$$f_i = \sum_{r=1}^{n_{t_{\text{mes}}}} \sum_{j=1}^{n_{q}} J_d(i + T_r \ast N_n, j)^2$$

and where \( n_{t_{\text{mes}}} \) is the number of observation times; \( n_q \) is the number of pumping wells in the aquifer; \( N_n \) is the total number of nodes in the aquifer; \( n_{\text{obswell}} \) is the total number of allowable observation wells; \( T \in \mathbb{R}^{n_{t_{\text{mes}}}} \) is a vector containing the observation times; \( J_d(i, j) \) is the \( i, j^\text{th} \) element of \( J_d \); \( \text{zone } j \) indicates all the nodes in the \( j^\text{th} \) zone; \( f \in \mathbb{R}^{N_n} \); and \( x \in \mathbb{R}^{N_n} \). Finally, we apply the GA method to the formulated problem. The GA coupled with the reduced model takes 47 seconds and 6,336 model calls to reach its termination criterion (i.e., convergence of solution). Experiments then are run comparing the results of coupling the full model to the GA in the place of the reduced model. It is found that the GA coupled with the full model converges to the same solution as the GA coupled with the reduced model and with a similar number of required model calls. It is found after comparing the results from the three distinct methods (GA, exhaustive search, and integer programming) that each reaches the same conclusion: given the 1-D test aquifer and pumping well setup the optimal locations for the observation wells are at nodes 51 and 50. Although this test case may seem trivial, it is a good test of the GA’s abilities. For instance, in this test case, an intuitive solution would be to place the observation wells as close as possible to the pumping well. One then can compare this intuitive solution to the mathematically optimal design and see that they are the same. Finally, it is evident that without any prior knowledge or built-in intelligence related to the problem, the GA was able to identify this optimal solution. This lends confidence to the hypothesis that in a large scale model, the GA is a valid search method for seeking the globally optimal solution. By observing the number of model calls necessary for the GA to achieve convergence, and the fact that the number of
required model calls appears independent of which model (full or reduced) is used, we easily can see the advantage of employing a reduced model rather than the full model.

3.2.b. Two-dimensional Test Case

After completing the 1-D test case, we construct a 2-D test case to assess the GA on a large-scale, real-world-sized model. The 2-D horizontal model is based on a model developed to simulate the groundwater flow in a confined aquifer in the Oristano plain in west-central Sardinia, Italy (Cau et al., 2002; Siade et al., 2012). We assume that the aquifer is surrounded on all sides by Dirichlet boundaries \( s_{nat,t} = 0 \ \forall t \) and the aquifer is divided into seven hydrologic zones, as seen in Figure 14, with zonal properties shown in Table 11.

![Figure 14. 2-D Test Case Hydraulic Zones](image)

<table>
<thead>
<tr>
<th>Zone</th>
<th>( K_x ) (m/day)</th>
<th>( K_y ) (m/day)</th>
<th>( S_x ) (1/m)</th>
<th>Depth (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1.20E-05</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 11. Hydraulic Zone Properties
In SAT2D (Paniconi & Putti, 1994) the fully constructed model contains 29,197 nodes; through POD the full model is reduced to a model containing 109 nodes while capturing over 99% of the variance of the snapshot set. In numerical experiments on an Intel Core 2.40 GHz i5 CPU, the full model takes 67 seconds to complete a single model run while the reduced model takes 25.4 seconds to complete twenty model runs, 25 seconds of which are required for a onetime read-in of the reduced model parameters. In other words, ignoring the one-time cost of reading-in the reduced model parameters, the reduced model only takes approximately 0.02 seconds to complete a model run. As we would expect, some loss of information results from the model reduction; accordingly, the solution from the reduced model does not perfectly match the solution from the full model. Table 12 shows a comparison of the trace of the full information matrices ($I$) (containing information from all the nodes at all the observation times—0.5, 1, 3, and 5 days which, as in the 1-D case, were chosen arbitrarily) obtained from the full model and from a projection of the solution from the reduced model onto the full space.

Table 12. Comparison of the Trace of the Information matrix for the full and reduced model for the 2-D Test Case

<table>
<thead>
<tr>
<th></th>
<th>Full model</th>
<th>Reduced</th>
<th>Absolute</th>
<th>Average</th>
<th>Maximum Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>2.70E-05</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>1.00E-05</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>15</td>
<td>1.70E-05</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>12</td>
<td>2.10E-05</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
<td>3.40E-05</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>11</td>
<td>1.00E-05</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td>model Trace</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>---------</td>
<td>---------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>12259.61</td>
<td>14729.45</td>
<td>2469.839</td>
<td>0.0011</td>
<td>0.0562</td>
<td></td>
</tr>
</tbody>
</table>

It should be noted that, unlike the 1-D test case, we have no “virtual” locations; as a result, all the nodes from the model are included in this full information matrix. As with the 1-D test case, although it might seem that the absolute error is very large, when considered in relation to the total number of observations (in this case 29,197 locations at four times for 20 wells for a total of 2,335,760 observations) the average error in information for any one observation is 0.0011. The maximum relative error, which shows the maximum error in received information for any single well from the full model at a single time step compared to the reduced model, is also fairly small. Considering that the absolute error in the information matrix includes a summation of squared errors from the full model space (29,197 nodes), we see that this amount of lost information in the model reduction is negligible. Taking all this into account, we decide that the trade off in accuracy vs. computational cost was acceptable.

For the experimental design problem, we divide the aquifer into 22 arbitrary zones (see Figure 15) with some zones designated to allow only for pumping wells and others designated to allow only for observation wells. In a real-world scenario the zones that allow for pumping might be private properties such as farms, while the zones that allow for observation wells could be public land in which the municipality or government agency could place observation wells. Zones 1, 2, 4, 8, 12, and 16 through 22 are designated for observation wells while zones 3, 5, 6, 7, 9, 10, 11, 13, 14, and 15 are designated for pumping wells. After designating the well zones, 20 pumping wells are placed throughout the allowable zones as shown in Figure 15.
As with the 1-D test case, each observation zone is allowed to contain at most one observation well. With this zonation, there are 10,490 feasible potential observation locations in the model. Accordingly, this is the dimension of the optimization problem. A number of experimental designs are then investigated. First a network with a single observation well is optimized and then networks with incrementally more (up to 12) observation wells are optimized. We then compare the results from different experimental designs. As noted above, for all scenarios observations are taken at 0.5, 1, 3, and 5 days.

The results of the optimization with the GA are shown in Figure 16 and Table 13.
Figure 16. Optimal Observation Well Locations

Table 13. Optimization Results for the 2-D Test Case

<table>
<thead>
<tr>
<th></th>
<th>One observation well</th>
<th>Two observation wells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model calls</td>
<td>13878</td>
<td>19880</td>
</tr>
<tr>
<td>Objective Score</td>
<td>1.53E-06</td>
<td>2.67E-06</td>
</tr>
<tr>
<td>Three observation wells</td>
<td>Four observation wells</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>Model calls</td>
<td>40039</td>
<td></td>
</tr>
<tr>
<td>Objective Score</td>
<td>3.75E-06</td>
<td></td>
</tr>
<tr>
<td>Five observation wells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model calls</td>
<td>16417</td>
<td></td>
</tr>
<tr>
<td>Objective Score</td>
<td>5.02E-06</td>
<td></td>
</tr>
<tr>
<td>Seven observation wells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model calls</td>
<td>25707</td>
<td></td>
</tr>
<tr>
<td>Objective Score</td>
<td>6.18E-06</td>
<td></td>
</tr>
<tr>
<td>Nine observation wells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model calls</td>
<td>32222</td>
<td></td>
</tr>
<tr>
<td>Objective Score</td>
<td>7.08E-06</td>
<td></td>
</tr>
<tr>
<td>Eleven observation wells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model calls</td>
<td>20551</td>
<td></td>
</tr>
<tr>
<td>Objective Score</td>
<td>7.78E-06</td>
<td></td>
</tr>
</tbody>
</table>

From Table 13 we can begin to draw some conclusions about this experimental design problem.

From observing the number of model calls required to achieve convergence, we immediately see the advantage of coupling the GA with a POD reduced model rather than the full model. As seen in Table 13 and Figure 17, when an additional observation well is added, the amount of information (and the corresponding objective function score) increases, as expected. However, the marginal amount of information gained from adding an additional observation well decreases as the total number of observation wells increases. Figure 17 shows this trend continuing until almost no additional information stands to be gained from adding the 12th observation well.
compared to having only 11 observation wells. It should be noted that there may be multiple designs with the same number of observation wells that are all A-optimal.

![Fitness Score vs. Number of Observation Wells](image)

**Figure 17. Objective Score Value vs. Number of Observation Wells**

We observe that the GA tends to group the observation wells close together and in areas marked by high concentrations of pumping wells (Figure 16). This is not unexpected and makes physical sense since the A-optimality criterion seeks to find a design that produces the most information regardless of whether that information already has been obtained. This is supported by the results obtained, which indicate that the design based on the A-optimality selects observation wells in the areas that are most sensitive to drawdown (i.e., the areas marked by high concentrations of pumping wells) and ignores areas that are least sensitive to drawdown. As a consequence, these A-optimal networks may produce observations with large covariance between observations and insufficient information about the wells located away from the areas marked by high concentrations of pumping wells. Because of this, it has been argued that other optimality criteria should be used—for example, E-optimality. An E-optimal design would
maximize the minimum eigenvalue of the information matrix (Yeh, 1992) and thus gather the most information possible from the pumping well from which it gets the least amount of information. This would probably result in more spread among the observation well locations and a different network pattern for each design. The E-optimality criterion as well as other optimality criteria could be easily implemented with a slight modification of the objective function in the GA. The equation below shows the objective function for E-optimality:

\[
\max \left( \min_i \left( \lambda_i \right) \right)
\]

where \( \lambda_i \) is the \( i^{th} \) eigenvalue of \( I \). Figure 18 shows a comparison of the A- and E-optimal observation networks for six and eight locations.
Figure 18. Comparison of A and E-Optimal Observation Well Locations for Six and Eight Wells

We notice that in the E-optimal networks the observation wells are more spread out and there are more significant differences in location patterns compared to the A-optimal design. The results make physical sense when we consider the definition of E-optimality. Though these results are interesting, the particulars of the optimal solution (either A or E) to which the GA converge are not of great interest. The true result of interest is that the GA coupled with the reduced model is able to achieve convergence with a realistic, large-scale model. This is something that would not be feasible to achieve if the GA were coupled with the full model due to the computational requirements of the large number of model calls required.
3.3. Conclusions

This paper presented a methodology that applies POD model reduction to experimental design. Though the full model may contain tens of thousands of nodes, through POD the reduced model contains only tens or hundreds of nodes resulting in a drastic savings in computational time of several orders of magnitude for each model call. We developed reduced models that run three orders of magnitude faster than the full models. Each reduced model produced a sufficiently accurate estimation of the Jacobian matrix of the full model. We also presented a methodology for solving the combinatorial search for the optimal observation well network by using a genetic algorithm (GA) to systematically search over the feasible set of observation wells. It does this by calling the reduced model to obtain information about the aquifer and then using that information to converge toward a globally optimal solution. The algorithm was constructed in such a way that the model calls required by GA are entirely carried-out in the reduced space. This is a major contribution of the proposed methodology in that it is able to search the information matrix in the full model space without incurring the cost of solving the full model. Thus the algorithm presented eliminates the need to store and access prohibitively large amounts of data or make an infeasible number of time-consuming full model runs.

We first verified the methodology on a small 1-D test case in which it was feasible to both perform an exhaustive search and solve an integer programming problem to obtain the optimal network design. A GA was then applied to verify that the algorithm would converge toward the globally optimal network and show that GA could be a valid search method in large-scale models where it is infeasible to use either an exhaustive search or integer programming. In this case the GA did converge to the global optimum after requiring over 6000 model calls, verifying its ability to search the feasible space. This large number of model calls demonstrated the need to
replace the full model with a reduced model. After verifying the results of the GA with the 1-D test case, we applied the algorithm to a large-scale, real-world-sized model. The full model contained 29,197 nodes and was reduced through POD to a model containing only 109 nodes. The solution from the reduced model was not a perfect match to the solution from the full model. However, this is expected with a full model of this size and a dimensionality reduction of several orders of magnitude. Despite the error, the reduced model was able to provide an acceptably accurate representation of the behavior of the full model. Table 13 shows the number of model calls required to achieve convergence, demonstrating that computational cost is of great importance in the GA. It is obvious that the cost associated with constructing the reduced model is trivial compared to the number of model calls needed in the GA.

As shown in Figure 16, the observation well locations obtained from the A-optimal design tend to group together in the area where the pumping wells are most concentrated. This design can lead to either large covariance among the observations taken by these networks or insufficient data about the aquifer as a whole. Figure 18 shows that the E-optimal design results in a different network of observation wells in which the observation wells are spread-out across the aquifer system. Regardless of whether A-optimality is the best choice for the optimality criterion in this or other aquifer experimental designs, the results show that the GA with the POD presented here is a valid search method for any problem-specific objective. This combined approach is an improvement over simply employing the GA coupled with a full model or attempting to formulate a mathematical optimization problem for the full model, which may be infeasible to solve.
In future work, we could research the effects of worst-case scenario pumping (i.e., given an “optimal” observation network, what pumping would result with the least amount of information) and apply experimental design to determine an observation network that would best handle a worst-case scenario. Also, we could apply the GA with POD methodology to a groundwater model that is reduced in a way that is not only forcing-independent but also parameter-independent. By doing this, we could then extend the experimental design problem to one that seeks to gather information not just about estimating the unknown forcing but also the unknown aquifer parameters.

Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \in \mathbb{R}^{N_x \times N_x}$</td>
<td>Stiffness matrix for the full model</td>
</tr>
<tr>
<td>$A \in \mathbb{R}^{n_x \times n_p}$</td>
<td>Stiffness matrix for the reduced model</td>
</tr>
<tr>
<td>$B \in \mathbb{R}^{N_x \times N_x}$</td>
<td>Mass matrix for the full model</td>
</tr>
<tr>
<td>$B \in \mathbb{R}^{n_x \times n_p}$</td>
<td>Mass matrix for the reduced model</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Simulation time step</td>
</tr>
<tr>
<td>$\Delta q_j$</td>
<td>Perturbation of the $j^{th}$ pumping rate</td>
</tr>
<tr>
<td>$\Delta q_{t_j} \in \mathbb{R}^{N_x}$</td>
<td>Vector used to perturb the $j^{th}$ element of $q$,</td>
</tr>
<tr>
<td>$\Delta \theta_j$</td>
<td>Perturbation of the $j^{th}$ parameter</td>
</tr>
<tr>
<td>$\Delta \theta_j \in \mathbb{R}^{N}$</td>
<td>Vector of zeros except the $j^{th}$ element which is equal to $\Delta \theta_j$</td>
</tr>
<tr>
<td>$f \in \mathbb{R}^{N_x}$</td>
<td>Vector used to calculate the trace $I$ in the integer programming problem</td>
</tr>
<tr>
<td>$f_1, f_2, f_3$</td>
<td>Known functions describing initial and boundary conditions of an aquifer</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$\Gamma_1$</td>
<td>Fixed head boundary</td>
</tr>
<tr>
<td>$\Gamma_2$</td>
<td>Flux boundary</td>
</tr>
<tr>
<td>$H \in \mathbb{R}^{(N_w \times n_p)}$</td>
<td>Vector of initial head values</td>
</tr>
<tr>
<td>$h \in \mathbb{R}^{(N_a \times n_p)}$</td>
<td>Vector of head values</td>
</tr>
<tr>
<td>$I \in \mathbb{R}^{n_e \times n_q}$</td>
<td>Full information matrix calculated by $J_d^T J_d W$</td>
</tr>
<tr>
<td>$I \in \mathbb{R}^{n_e \times n_q}$</td>
<td>Information matrix calculated by $J_d^T J_d W$</td>
</tr>
<tr>
<td>$J_d \in \mathbb{R}^{N_a \times n_q}$</td>
<td>Full Jacobian matrix (sensitivities of all $N_a$ nodes, at all times, to all wells)</td>
</tr>
<tr>
<td>$J_d \in \mathbb{R}^{n_{obs} \times n_q}$</td>
<td>Jacobian matrix containing only the observations of interest to all wells</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Hydraulic conductivity in the $i^{th}$ direction</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>$i^{th}$ eigenvalue of $I$</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number of parameters of interest</td>
</tr>
<tr>
<td>$n_{loc}$</td>
<td>Total number of feasible observation well locations</td>
</tr>
<tr>
<td>$N_a$</td>
<td>Number of nodes in the full model</td>
</tr>
<tr>
<td>$n_{obs}$</td>
<td>Total number of observations taken</td>
</tr>
<tr>
<td>$n_{obs_well}$</td>
<td>Maximum number of allowable observation wells</td>
</tr>
<tr>
<td>$n_p$</td>
<td>Number of principal components used in the reduced model</td>
</tr>
<tr>
<td>$n_q$</td>
<td>Number of pumping wells in the aquifer</td>
</tr>
<tr>
<td>$n_{sp}$</td>
<td>Number of snapshots taken for each pumping well to build the reduced model</td>
</tr>
<tr>
<td>$P \in \mathbb{R}^{N_a \times n_p}$</td>
<td>Full projection matrix</td>
</tr>
<tr>
<td>$P \in \mathbb{R}^{n_{obs} \times n_p}$</td>
<td>Matrix of rows of $P$ corresponding to some $\omega$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\Omega \in \mathbb{R}^{N_x}$</td>
<td>Vector containing the index of all feasible observation well locations</td>
</tr>
<tr>
<td>$\omega \in \mathbb{R}^{n_{obs}}$</td>
<td>Vector contacting some feasible set of observation well locations ($\omega \subset \Omega$)</td>
</tr>
<tr>
<td>$o_i$</td>
<td>$i^{th}$ observation</td>
</tr>
<tr>
<td>$o_i(\theta)$</td>
<td>$i^{th}$ model simulated value using the parameter values in $\theta$</td>
</tr>
<tr>
<td>$Q$</td>
<td>Specific volumetric pumping rate</td>
</tr>
<tr>
<td>$q_n$</td>
<td>Specific discharge normal to the flux boundary ($\Gamma_2$)</td>
</tr>
<tr>
<td>$q \in \mathbb{R}^{n_x}$</td>
<td>Vector of sinks at time $t$ for the full model</td>
</tr>
<tr>
<td>$q \in \mathbb{R}^{n_y}$</td>
<td>Vector of sinks at time $t$ for the reduced model</td>
</tr>
<tr>
<td>$R \in \mathbb{R}^{n_x \times n_y}$</td>
<td>Matrix containing the reduced solution at the observation times</td>
</tr>
<tr>
<td>$r \in \mathbb{R}^{n_y}$</td>
<td>Vector of the reduced solution at time $t$</td>
</tr>
<tr>
<td>$s \in \mathbb{R}^{N_x \times n_y}$</td>
<td>Vector of drawdown values</td>
</tr>
<tr>
<td>$\Sigma \in \mathbb{R}^{n_y \times n_y}$</td>
<td>Diagonal matrix containing the singular values of $X$</td>
</tr>
<tr>
<td>$s_{nat,t} \in \mathbb{R}^{N_x}$</td>
<td>Vector of natural system dynamics at time $t$</td>
</tr>
<tr>
<td>$S_s$</td>
<td>Specific storage</td>
</tr>
<tr>
<td>$s \in \mathbb{R}^{N_x}$</td>
<td>Vector of drawdown values at time $t$</td>
</tr>
<tr>
<td>$\hat{s} \in \mathbb{R}^{N_x}$</td>
<td>Vector of the approximation of $s$</td>
</tr>
<tr>
<td>$s_i(q_i)$</td>
<td>$i^{th}$ simulated drawdown using $q_i$</td>
</tr>
<tr>
<td>$t$</td>
<td>Time index</td>
</tr>
<tr>
<td>$\theta \in \mathbb{R}^N$</td>
<td>Vector of nominal parameter values</td>
</tr>
<tr>
<td>$U \in \mathbb{R}^{N_x \times n_y}$</td>
<td>Matrix containing the left singular vectors of $X$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$V$</td>
<td>Subspace spanned by the columns of $P$</td>
</tr>
<tr>
<td>$V \in \mathbb{R}^{n_p \times n_p}$</td>
<td>Matrix containing the right singular vectors of $X$</td>
</tr>
<tr>
<td>$W \in \mathbb{R}^{n_q \times n_q}$</td>
<td>Some user specified weighting matrix used in calculating $I$</td>
</tr>
<tr>
<td>$X \in \mathbb{R}^{N_x \times n_p}$</td>
<td>Matrix containing all collected snapshots</td>
</tr>
<tr>
<td>$x \in \mathbb{R}^{N_x}$</td>
<td>Vector of binary variables indicating if a node has an observation well</td>
</tr>
<tr>
<td>$zone \ j$</td>
<td>All the nodes in the $j^{th}$ zone</td>
</tr>
</tbody>
</table>
3.4. Bibliography


Experimental design for estimating unknown hydraulic conductivity in an aquifer using a genetic algorithm and reduced order model (T.T. Ushijima & Yeh, 2015)

Abstract

We develop an experimental design algorithm to select locations for a network of observation wells that provide the maximum robust information about unknown hydraulic conductivity in a confined, anisotropic aquifer. Since the information that a design provides is dependent on an aquifer’s hydraulic conductivity, a robust design is one that provides the maximum information in the worst-case scenario. The design can be formulated as a max-min optimization problem. The problem is generally non-convex, non-differentiable, and contains integer variables. We use a Genetic Algorithm (GA) to perform the combinatorial search. We employ proper orthogonal decomposition (POD) to reduce the dimension of the groundwater model, thereby reducing the computational burden posed by employing a GA. The GA algorithm exhaustively searches for the robust design across a set of hydraulic conductivities and finds an approximate design (called the High Frequency Observation Well Design) through a Monte Carlo-type search. The results from a small-scale 1-D test case validate the proposed methodology. We then apply the methodology to a realistically-scaled 2-D test case.
4.1. Introduction

4.1.a. Background

Hydraulic conductivity is one of the most important parameters to take into account when modeling an aquifer. Unknown hydraulic conductivities can have drastic effects on the reliability of groundwater model results. Estimating these unknown hydraulic conductivities through inverse modeling is an important area of research in groundwater modeling. Solving the inverse problem requires field observations of head. However, collecting good observations is expensive, time consuming, and difficult, particularly if aquifer parameters vary spatially. As a result, inverse modeling always faces an observation scarcity problem. Given an infinite budget, we could take observations at infinitesimally fine spatial and temporal resolutions; however, in the real world there are budget constraints on the number of experiments (observations) that can be conducted. In general, the goal of optimal experimental design for parameter estimation is to select the observation locations and sampling frequency such that a specified criterion is optimized subject to a set of constraints. The constraints frequently encountered are cost, reliability of the estimated parameters, and time and duration of the experiments. In groundwater modeling, if one assumes that observations are taken from the beginning of the pumping test to the end, the experimental design problem is simplified to the determination of a network of observation locations. The most commonly used criterion for optimal experimental design is the maximization of a measure of the information matrix. This formulation generally lends itself to a combinatorial optimization problem that is nonlinear and non-convex. We then can determine the tradeoff between number of experiments and amount of useful information collected by varying the number of experiments and evaluating the amount of useful information received for each experiment. While the concept of experimental design for parameter estimation in the
context of groundwater modeling has been investigated extensively, the combinatorial search required to solve for the optimal observation network of observations has challenged many studies. A realistic, highly discretized, large-scale groundwater model, referred to in this paper as the full model, may have tens or hundreds of thousands of nodes and a resulting equivalent number of equations. As a consequence, the dimension of the combinatorial search quickly becomes so large that it is infeasible to solve through mathematical programming techniques such as mixed integer nonlinear programming. Thus other methods are required to solve this optimization problem. Genetic Algorithms (GAs) are one of a number of methods that have been developed over the years to solve large-scale optimization problems that are difficult or impossible to solve through traditional mathematical programming techniques. GAs do not require the calculation of derivatives and also can deal with discontinuous functions. While GAs have been used by a number of studies to design optimal observation networks using various optimality criteria (Babbar-Sebens & Minsker, 2010; McPhee & Yeh, 2006b; Reed et al., 2000), many of these studies were challenged by the fact that GAs require a large number of model calls. Thus for a highly discretized groundwater model, using a GA to design an optimal observation network may be computationally inefficient, even infeasible. To overcome this, we apply a proper orthogonal decomposition (POD) model reduction to the groundwater model to reduce its spatial dimension. POD is an effective model reduction technique that maintains the physics of the groundwater model. In many instances it has been shown that the reduced model is orders of magnitudes smaller than the original full model and runs 1,000 times faster. Since the parameters themselves are to be estimated, we develop a parameter-independent reduced model for the optimal experimental design. By doing this, we are able to reduce an inordinately large-scale, combinatorial optimization problem to a manageable size.
Ushijima and Yeh (2013) applied POD model reduction to develop an algorithm for experimental design that seeks information related to unknown pumping rates. The current study expands on this previous study by applying POD model reduction to develop an algorithm for experimental design that seeks information related to unknown model parameters. While similar, these two problems are different. Because the response of the groundwater system is linear with respect to changes in pumping, the sensitivity coefficients are invariant and only need to be calculated once. On the other hand, the response of the groundwater system to changes in hydrologic parameters (e.g., hydraulic conductivity) is nonlinear and dependent on the unknown parameters. This makes the current study much more challenging and difficult.

The major contribution of this paper is to demonstrate the applicability of using reduced order models to ensure that solving the underlying combinatorial optimization problem is computationally feasible. In other words, our proposed method allows us to solve realistically-scaled, optimal experimental design for parameter estimation problems in groundwater modeling that were not possible to solve in the past.

4.1.b. Confined Aquifer Groundwater Model

The governing equation for three-dimensional groundwater flow in a confined, anisotropic aquifer, can be expressed by the following partial differential equation (PDE) (Bear, 1979):

\[
\frac{\partial}{\partial x} \left( K_x \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( K_y \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial h}{\partial z} \right) - F = S \frac{\partial h}{\partial t}
\]  

(36)

with initial and boundary conditions
\begin{align*}
h(x, y, z) &= f_1(x, y, z) \\
h(x, y, z, t) &= f_2(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_1) \\
q_n(x, y, z, t) &= f_3(x, y, z, t), \quad (x, y, z, t) \in (\Gamma_2)
\end{align*}

where \( h \) is the hydraulic head [L]; \( K_x, K_y, \) and \( K_z \) are the hydraulic conductivities in the \( x, y, \) and \( z \) directions [L/T]; \( S_s \) is the specific storage [L\(^{-1}\)]; \( F \) is the specific volumetric pumping rate [T\(^{-1}\)]; \( q_n \) is the specific discharge normal to the flux boundary \( (\Gamma_2) \) [T\(^{-1}\)]; \( \Gamma_1 \) is the fixed head boundary; \( f_1, f_2, \) and \( f_3 \) are known functions; \( L \) denotes the length unit (meters, feet, etc.); and \( T \) denotes the time unit (days, hours, etc.). Without loss of generality we can change the state variable from head \( (h) \) to drawdown \( (s) \). Drawdown is defined as the difference between the initial head \( (H) \) and head at some time \( (h) \) (i.e., \( s = H - h \)). This linear transformation converts the initial and Dirichlet boundary conditions to zero. The governing PDE (Eq. (36)) then can be discretized through finite difference (or finite element) approximations into a system of linear ordinary differential equations (ODEs):

\[ B \frac{ds}{dt} + As = q \]  

(37)

where \( s \in \mathbb{R}^{N_n} \) is a vector of drawdown values, \( A \in \mathbb{R}^{N_n \times N_n} \) is the stiffness matrix, \( B \in \mathbb{R}^{N_n \times N_n} \) is the mass matrix, \( q \in \mathbb{R}^{N_n} \) is a vector of sinks (pumping, recharge, evaporation, etc.), and \( N_n \) is the total number of nodes the aquifer is discretized into. Eq. (37) is referred to as the full model. By approximating the time derivative and assuming that all model parameters except hydraulic conductivity are fixed, we can write this linear set of ODEs as a set of linear equations:

\[ A(k)s = b \]  

(38)
where $A(k) \in \mathbb{R}^{N_x \times N_t}$ contains the model parameters as well as initial and boundary conditions, $b \in \mathbb{R}^{N_x}$ contains the forcing, and $k$ is a vector of hydraulic conductivities. Because of the dimension of $A(k)$, solving (38) is very computationally expensive. To overcome this, we use Proper Orthogonal Decomposition (POD) to reduce the dimension of the groundwater model and thus alleviate the computation burden of solving the full model.

4.1.c. Proper Orthogonal Decomposition (POD)

Many studies have focused on POD model reduction (Cazemier et al., 1998; Kowalski & Jin, 2003; Willcox & Peraire, 2002) as well as its use in reducing groundwater models (McPhee & Yeh, 2008; Pasetto et al., 2013; Siade et al., 2010, 2012; Vermeulen et al., 2004). POD projects a system of linear equations of dimension $N_n$ onto a subspace of dimension $n_p$ such that $n_p \leq N_n$ and $\hat{s} = Pr$, where $\hat{s} \in \mathbb{R}^{N_x}$ is the approximation of the original state vector $(s)$, $P \in \mathbb{R}^{N_x \times n_p}$ is a the projection operator given by a matrix whose columns form an orthogonal basis spanning the subspace $V \subset \mathbb{R}^{N_x}$, and $r \in \mathbb{R}^{n_p}$ is a vector of the reduced solution (Siade et al., 2010). We construct the matrix $P$ by collecting snapshots of the drawdown from the full model at various times and storing the snapshots in the matrix $X \in \mathbb{R}^{N_x \times n_sp}$ where $n_sp$ is the number of snapshots taken. We use the method developed by Siade et al. (2010) to approximate the optimal times to take snapshots so that the model accurately represents the dynamics of the original system. Vermeulen et al. (2004) showed that taking snapshots at all combinations of hydraulic conductivity results in a parameter-independent reduced model guaranteeing that for any $k$, the error $(e = Pr - s)$ will be less than a specified tolerance level $\tau$. However, this combinatorial search would require an inordinate number of full model calls. To minimize the number of model
calls, we use a greedy algorithm like the one proposed by Grepl and Patera (2005) to choose hydraulic conductivity combinations with which to take snapshots based on the residual 

\( r_e = A(k)Pr - b \). Note that the norm of the residual \( r_e \) can be computed completely in the reduced space without running the full model. It has been shown that the residual \( r_e \) can be used to estimate the true error \( e \), which is defined as the difference between the solution of the full model and solution of the reduced model (Grepl & Patera, 2005; Haasdonk & Ohlberger, 2011; Hasenauer, Löhning, Khammash, & Allgöwer, 2012; Rovas, Machiels, & Maday, 2005).

However, Hasenauer et al. (2012) pointed out that in many cases \( \| r_e \|_2 \gg \| e \|_2 \). Hinze and Kunkel (2012) developed a method to linearly scale the residual given a one-dimensional parameter space. Pasetto et al. (2013) extended this concept by employing a non-linear scaling factor such that

\[
\frac{d}{\| r_e \|_2} = \| e \|_2
\]  

(39)

where \( d \) is the scaling factor. This scaled residual is then used to choose hydraulic conductivity combinations. Choosing hydraulic conductivity combinations based on the norm of the error allows us to estimate the lower bound of the accuracy of the reduced model and the reduced model’s ability to approximate the Jacobian matrix. We use the algorithm proposed by Pasetto et al. (2013) to select hydraulic conductivity combinations to construct a parameter-independent reduced model. The algorithm combines a greedy algorithm for snapshot selection in the parameter space and an approximation of the optimal distribution of the snapshots in time. Using a residual-based estimation of the error associated with the reduced model allows a considerable reduction of the number of full model calls required for the computation of the principal vectors.
Assuming we know what snapshots to collect and store in \( X \), we perform singular value decomposition (SVD) on \( X \) such that

\[
X = P \Sigma V^T
\]

where \( P \in R^{N_x \times n_p} \) and \( V \in R^{n_p \times n_p} \) contain the left and right singular vectors of \( X \), respectively, and \( \Sigma \in R^{n_p \times n_p} \) is a diagonal matrix containing the singular values of \( X \) (Shlizerman et al., 2012). In previous work \( P \) was constructed by finding the eigenvectors of \( XX^T \)

\[
(XX^T = QAQ^{-1})
\]

and retaining the \( n_p \) principal vectors of \( Q \) (Siade et al., 2010) where \( Q \in R^{N_x \times n_p} \) is a matrix containing the eigenvectors of \( X \) and \( A \in R^{N_x \times N_a} \) is a matrix containing the eigenvalues of \( X \). Since \( \text{Range}(P) = \text{Range}(Q) \), to be consistent with previous work we refer to \( P \) as a matrix containing the principal vectors of \( X \). We then use the Galerkin projection to reduce the full model (Eq. (37)) as follows (Siade et al., 2010):

\[
P^TBP \frac{dr}{dt} + P^TAPr = P^Tq
\]

Alternatively, we can reduce the model by projecting Eq. (38) as follows:

\[
P^TA(k)Pr = P^Tb
\]

Eq. (40) can be transformed into Eq. (41) by approximating the time derivative in Eq. (40). If we let \( P^TA(k)P = \tilde{A}(k) \) and \( P^Tb = \tilde{b} \), Eq. (40) becomes

\[
\tilde{A}(k)r = \tilde{b}
\]
We refer to Eq. (42) as the reduced model. Unfortunately, we do not know a-priori what snapshots to collect, so the algorithm proposed by Pasetto et al. (2013) is used iteratively to collect snapshots and build the \( P \) matrix. The procedure for constructing \( P \) is summarized as follows:

1. Select a hydraulic conductivity combination \( (k) \), run the full model, collect the snapshots at the times according to the algorithm proposed by Siade et al. (2010) and store them in \( \chi \subseteq X \), and perform SVD on \( \chi \) such that \( \chi = \hat{P}\Sigma V^T \), where \( \hat{P} \subseteq P \).
   
   a. If \( P \) does not exist, create \( P \) from the first principal vector of \( \hat{P} \) and construct a reduced model.

2. Calculate \( \| e \|_2 \)
   
   a. While \( \| e \|_2 \geq \tau \), augment \( P \) with sequential principal vectors from \( \hat{P} \) and construct a new reduced model.

3. Calculate \( d \| r_e \| \)

4. If \( d \| r_e \| \geq \tau \), return to step 1 and select the next \( k \) in the parameter space at which to take snapshots by solving the following greedy search (Pasetto et al., 2013):

   \[
   \arg \max_k d \| r_e \|
   \]

   Else if \( d \| r_e \| < \tau \), output \( P \in R^{N_x \times n_y} \) and exit.

McPhee and Yeh (2008) showed that POD could reduce a groundwater model by several orders of magnitude while sustaining a loss in accuracy of less than one percent. Siade et al. (2010)
demonstrated that a groundwater model containing over 200,000 nodes and thus over 200,000 equations could be reduced to a model with less than 10 equations through the application of POD. The reduced model runs approximately 1,000 times faster than the full model with acceptable error. By replacing the full, computationally expensive model with a reduced, computationally inexpensive model, we are able to feasibly use a GA to search for the optimal network design.

4.1.d. Experimental Design

Now we have a set of methods that allow us to feasibly search for the network design that gives us the most useful information. However, we still are faced with two questions: What is useful information? How do we quantify the amount of information gained? Answering these questions has been the topic of much research spanning diverse fields of study. Many of them share common elements.

Since the 1920s statistical methods have been accepted as the primary means to answer these questions. For example, the Jacobian matrix \( J_d \) is defined as

\[
J_d = \begin{pmatrix}
\frac{\partial o_1}{\partial \theta_1} & \cdots & \frac{\partial o_i}{\partial \theta_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial o_{n_{obs}}}{\partial \theta_1} & \cdots & \frac{\partial o_{n_{obs}}}{\partial \theta_N}
\end{pmatrix}
\]  

(43)

where \( n_{obs} \) is the total number of observations taken, which may vary spatially and temporally; \( N \) is the total number of parameters of interest; and \( \frac{\partial o_i}{\partial \theta_j} \) \( (i = 1, \ldots, n_{obs}; j = 1, \ldots, N) \) is the sensitivity.
of the $i^{th}$ observation ($o_i$) to changes in the $j^{th}$ parameter ($\theta_j$) that contains information, expressed by the sensitivities, about the system of interest. These sensitivities provide useful information, as they show how the system responds to changes in the parameters of interest.

Three methods have been proposed over the years to calculate the sensitivities of a system: (1) the parameter perturbation method, (2) the sensitivity equation method, and (3) the adjoint state method (Yeh, 1986). A convenient method to use is the perturbation method described as

$$\frac{\partial o_i}{\partial \theta_j} \approx \frac{o_i(\theta + \Delta \theta_j) - o_i(\theta)}{\Delta \theta_j}$$

(44)

where $\theta$ is a vector of the estimated parameter values, $o_i(\theta)$ is the $i^{th}$ model simulated value using the parameter values in $\theta$, $\Delta \theta_j$ is a small increment of the $j^{th}$ parameter (called the perturbation of $\theta_j$), and $\Delta \theta_j$ is a vector in which all elements are zero except for the $j^{th}$ element, which is equal to $\Delta \theta_j$ (Poeter et al., 2005). Using the parameter perturbation method to estimate sensitivities for general models requires $(N+1)$ model calls (a baseline model solution and $N$ model runs to estimate the sensitivities) (Yeh, 1986). In general, there are two requirements for making accurate estimates of $\frac{\partial o_i}{\partial \theta_j}$ through the use of Eq. (44). First, $\Delta \theta_j \ll \theta$ and second, $\theta$ must be close to the true parameter values. As noted, in the real world we cannot take observations at infinitesimally fine spatial and temporal resolutions; consequently, we can think of the experimental design problem as a problem of finding which elements of this theoretically, infinitely large Jacobian matrix should be included in the analysis. This answers the first question (What is useful information?) raised above.
To answer the second question (How do we quantify the amount of information gained?) we adopt the concept of the information matrix \( \mathbf{I} \) from statistics as follows:

\[
\mathbf{I} = \mathbf{J}_d^T \times \mathbf{J}_d \times \mathbf{W}
\]  

(45)

where \( \mathbf{W} \) is a user-specified weighting matrix. Under the assumptions that 1) a least-squares error criterion is used for parameter estimation; 2) the observation errors are uncorrelated with equal variance; and 3) \( \mathbf{W} \) is the identity matrix, the information matrix defined in Eq. (45) is equivalent to the inverse of the covariance matrix of the estimated parameters \( \mathbf{C}(\theta) \) (Cleveland & Yeh, 1990; Kutner et al., 2004). Eq. (46) shows the covariance matrix. This matrix contains the variance of the estimated parameter values \( \sigma^2(\theta_i) \) and the covariance of estimated parameters to each other \( \sigma(\theta_i, \theta_j) \).

\[
\mathbf{C}(\theta) = \begin{bmatrix}
\sigma^2(\theta_1) & \sigma(\theta_1, \theta_2) & \cdots & \sigma(\theta_1, \theta_n) \\
\sigma(\theta_2, \theta_1) & \sigma^2(\theta_2) & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
\sigma(\theta_n, \theta_1) & \cdots & \sigma(\theta_n, \theta_{n-1}) & \sigma^2(\theta_n)
\end{bmatrix}
\]  

(46)

We could use any number of methods to quantify the amount of information contained in a given covariance or information matrix. Commonly used methods include measuring the trace (A-optimality), determinant (D-optimality), or minimum eigenvalue (E-optimality) of the covariance matrix (Steinberg & Hunter, 1984). The most commonly used optimality criterion for groundwater studies is the maximal information criterion (Cleveland & Yeh, 1990; Hsu & Yeh, 1989), which measures the trace of the information matrix. Under the assumptions mentioned above, (i.e., the information matrix is equivalent to the inverse of the covariance matrix of the
estimated parameters), a design is said to be D-optimal if it maximizes the determinant of the information matrix, while a design is said to be E-optimal if it maximizes the minimum eigenvalue of the information matrix. It can be shown that a design that is A-optimal (minimizes the trace of the covariance matrix) also maximizes the trace of the information matrix and the information received, thus making that design maximal information-optimal. We may include a set of constraints in the experimental design to constrain the feasible solution space. In the past A, D, and maximal information-optimality have been used to solve experimental design problems in the context of groundwater modeling. A-optimality has been used for unconfined aquifer parameter estimation (Altmann-Dieses et al., 2002). D-optimality has been used for designing an optimal multi-objective observation network for parameter estimation and model discrimination (Knopman & Voss, 1989); for optimal pumping tests for parameter estimation (Nishikawa & Yeh, 1989); and for optimal observation network design for dispersion parameters (Catania & Paladino, 2009). Maximal information-optimality has been used for optimal observation networks for confined aquifer parameter estimation (Hsu & Yeh, 1989) and for transport parameter estimation (Cleveland & Yeh, 1990). Cleveland and Yeh (1991) pointed out that it is advantageous to use maximal information optimality over A- or D-optimality because the maximal information criterion can be computed directly in a straightforward manner without matrix inversion. McCarthy and Yeh (1990) performed a study not only to find the optimal design but also evaluate if the amount of information gathered is useful for achieving some objective. Chang et al. (2005) studied data sufficiency for an optimal network design.

Since the goal of our experimental design is to gain information about the unknown hydraulic conductivities, the Jacobian matrix is the matrix of sensitivities of drawdown observations to changes in hydraulic conductivity. Thus,
where the elements in the Jacobian matrix \( \frac{\partial s_i}{\partial K_j} \) are approximated:

\[
\frac{\partial s_i}{\partial K_j} \approx \frac{s_i(k + \Delta k_j) - s(k)}{\Delta K_j}
\]

where \( s_i(k) \) is the \( i^{th} \) model simulated observation (drawdown) using the estimated hydraulic conductivities in \( k \in \mathbb{R}^{n_z} \), \( \Delta K_j \) is the perturbation of the \( j^{th} \) hydraulic conductivity, \( \Delta k_j \in \mathbb{R}^{n_z} \) is a vector perturbing the \( j^{th} \) hydraulic conductivity, and \( \frac{\partial s_i}{\partial K_j} \) is the sensitivity of the \( i^{th} \) drawdown to a change in the \( j^{th} \) hydraulic conductivity. As noted before, to make accurate approximations of \( J_d \), \( \Delta k_j \ll k \) and \( k (K_1, \ldots, K_N) \) must be close to the true parameter values. The first requirement is fairly simple to satisfy. However, meeting the second requires some \textit{a-priori} knowledge, which may not be available. If we cannot acquire this knowledge \textit{a-priori} we could acquire it through a sequential experimental design (Hsu & Yeh, 1989). Unfortunately, this method can be too costly and time consuming to be feasible, particularly in the context of groundwater. We also could place an \textit{a-priori} probability distribution on the parameters to derive an \textit{a-posteriori} information matrix (Dietrich, Raatz, & Hesselbach, 2010). A third option would be to perform the optimization using the meta-Jacobian matrix as opposed to the standard Jacobian matrix (de Brauwere et al., 2009). The standard Jacobian matrix contains sensitivities of
observations to changes in hydraulic conductivity at a set value of hydraulic conductivity. On the other hand, we form the meta-Jacobian matrix by taking a number of standard Jacobians matrices formed at various values of hydraulic conductivities and augmenting all the matrices together. This optimization would give us a design that would be potentially sub-optimal for all values of hydraulic conductivity but would provide a near-optimal amount of information across the widest range of values. Unfortunately, this design cannot set a lower limit on the amount of information obtained in the worst-case scenario.

4.2. Methods
4.2.a. Experimental Design

We choose maximal information-optimality (maximizing the trace of $I$) as the fitness criterion for this optimization study. The optimization employs discrete decision variables whose range spans the nodes of the finite element mesh used to model the aquifer. Thus the observation locations chosen will fall on the nodes of that mesh. Furthermore, we define discrete pumping and observation zones in the aquifer. These zones define areas of the aquifer in which observations wells or pumping wells may be placed. These two zonation types are physically related to each other as they cannot overlap but are distinct from the aquifer’s hydrologic zones that define areas of uniform hydrologic properties (e.g., hydraulic conductivity, specific storage, depth, etc.). The constraints applied to this optimization are as follows: (1) at most one observation well may be placed in an observation zone; and (2) the total number of observation wells cannot exceed some upper limit. This yields the experimental design formulation
\begin{align}
\max_{\omega \in \Omega} Tr\left( J_d^T \times J_d \times W \right) \\
\text{s.t.} \quad \sum_{i=\text{zone } j} x_i \leq 1 \quad \forall j \\
\sum_{i=1}^{N_i} x_i \leq n_{\text{obswell}}
\end{align}

(49)

where $n_{\text{obswell}}$ is the total number of allowable observation wells, $\text{zone } j$ indicates all the nodes in the $j^{th}$ observation zone, $N_n$ is the total number of nodes in the model, $\omega \in R^{n_{\text{obswell}}}$ is some feasible set of observation well locations, $\Omega$ is the space of all feasible sets of observation well locations, and $x \in R^{N_n}$ contains binary variables (equal to 1 if an observation is taken at that node and 0 otherwise). In this study we choose $W$ to be the identity matrix of size $n_z$, where $n_z$ is the total number of hydrologic zones with unknown hydraulic conductivity. Rather than depending on acquiring some a-priori knowledge, performing a sequential design, or employing a meta-Jacobian, we choose to perform a robust experimental design that can be described by the following max-min formulation (Sun & Yeh, 2007):

\begin{align}
\max_{\omega \in \Omega} \left( \min_{k \in \hat{K}} \Tr \left( J_d^T \times J_d \times W \right) \right) \\
\text{s.t.} \quad \sum_{i=\text{zone } j} x_i \leq 1 \quad \forall j \\
\sum_{i=1}^{N_i} x_i \leq n_{\text{obswell}}
\end{align}

(50)

where $\hat{K}$ is the space of feasible hydraulic conductivities. In other words, the robust design is the one that provides the maximum amount of information in the worst-case scenario. We assume that the zonation pattern of the aquifer already has been determined from prior geological information. To solve this optimization problem, we first discretize the zonal hydraulic...
conductivity into a number of increments. Specifically, for each of the \( n_z \) hydrologic zones we
discretize the feasible range for its hydraulic conductivity into \( n_l \) increments. We then form all
possible combinations of hydraulic conductivity in all hydrologic zones, resulting in a total of
\( n_z n_l \) combinations. We will refer to this set of combinations as \( K \in \mathbb{R}^{n_z \times n_l} \) and solve the
optimization problem over this set. In the robust experimental design, minimization is obtained
by the proper choice of \( k \subset K \). This yields the following robust experimental design
formulation:

\[
\max_{\omega \in \Omega} \left( \min_{k \subset K} \operatorname{Tr} \left( J_{d}^T \times J_{d} \times W \right) \right)
\]

\[
\text{s.t. } \sum_{i=\text{zone } j} x_i \leq 1 \ \forall j
\]

\[
\sum_{i=1}^{N_z} x_i \leq n_{\text{obswell}}
\]

Note that Eq. (50) optimizes over the full, continuous hydraulic conductivity space while Eq.
(51) optimizes over a discretized hydraulic conductivity space. This max-min combinatorial
optimization is nonlinear, non-convex, and extremely difficult to solve. To solve this problem,
we use the following systematic search procedure:

1. Discretize \( \hat{K} \) into \( n_z n_l \) parameter combinations (i.e., combinations of zonal hydraulic
   conductivities) to produce \( K \in \mathbb{R}^{n_z \times n_l} \).

2. For each combination, use Eq. (48) to approximate the elements of the Jacobian matrix;
solve Eq. (49), which searches for some observation network \( (\omega) \) that is an optimal
observation network \( (\tilde{\omega}_i) \) for the given parameter combination; and perform an
exhaustive search over the remaining $n_i^{n_i} - 1$ parameter combinations to identify the worst-case scenario (i.e., the combination for which $\tilde{\omega}_i$ provides the least amount of information). The score (i.e., the value of the objective function for a particular $\tilde{\omega}_i$) associated with the worst-case scenario is referred to as the robust score.

3. After completing step 2 for all parameter combinations, choose the robust design by finding the $\tilde{\omega}_i$ that has the maximum worst-case score.

In addition to finding the robust experimental design, we also find what we term the High Frequency Observation Well (HFO) design, described as follows. If we let $\tilde{\omega}_i^{(j)}$ be the $j^{th}$ optimal location in $\tilde{\omega}_i$ and $\hat{\omega}$ be the set of all $\tilde{\omega}_i$ such that

$$\hat{\omega} = \bigcup_{i=1}^{n_i} \tilde{\omega}_i$$

then the HFO is the $n_{obswell} \tilde{\omega}_i^{(j)} \subset \hat{\omega}$ that occurs with the highest frequency. In other words, we take the set of all $\tilde{\omega}$’s and find the observation locations that occur most frequently in that set. We then create a contour plot of the frequency (scaled to sum to 1) with which some $\tilde{\omega}_i^{(j)}$ occurs in $\hat{\omega}$. This gives us a map that estimates the probability distribution of optimal locations. In a way this is similar to a Monte Carlo approach to estimating the probability of a location residing in the optimal set $\omega^*$, assuming a uniform prior of hydraulic conductivity where $\omega^*$ is the optimal solution to a specific experimental design given the true aquifer parameters. Doing this allows us to take a probabilistic rather than deterministic approach to placing observation wells. This approach could have some real-world advantages. For instance, if it is found after the
optimization is completed that one of the optimal locations cannot be used and the optimization cannot be re-run, the determination of which feasible location should be used as a replacement can be made based on the plot of the estimated probability distribution.

4.2.b. Genetic Algorithm

Wall (1995) developed a base GA code (GAlib) that supports many features that lend themselves to solving the particular problem at hand. These features include support for real number genomes, adaptability, and options that allow the implementation of various “flavors” of GA. From this base code, we developed a GA that searches the feasible region by using real number genomes. To do this, we index all possible observation well locations in the full model space (this index is referred to as $\Omega$) and the GA creates populations comprised of individuals that represent feasible combinations of observation well locations ($\omega \subset \Omega$). For each individual in a population, the GA calls the reduced model to approximate the Jacobian ($J_a$) and Information ($I$) matrices (Eq. (47) and Eq. (45), respectively) to evaluate the individual’s fitness (i.e., how well an individual optimizes the objective function). To calculate Eq. (48) in the reduced space, the GA passes $MP$ to the reduced model, where $M \in R^{n_{well} \times N_s}$ is an observation matrix in which the $i^{th}$ row vector contains a 1 in the column associated with the $i^{th}$ element in $\omega$ and 0s elsewhere. The reduced model calculates the reduced solution ($R(k)$) at the observation times and solves. Eq. (48) in the reduced space through the following equation:

$$
J_{a,i} = \frac{\partial s_i}{\partial K_j} \approx \left( \frac{\hat{PR}(k + \Delta k_j) - \hat{PR}(k)}{\Delta K_j} \right)_{i,j}
$$

(53)
By calculating the fitness of each individual (i.e., how close the solution is to the solution of Eq. (49)) in the reduced space, we achieve a drastic reduction in computation time.

After evaluating the fitness of the individuals in a population, the GA uses mutations and crossovers to solve the optimization problem of Eq. (49). The GA remembers the elite score (i.e., the best score achieved across all generations) and declares convergence when there is no deviation between the elite score and the best current generation score for 300 generations (i.e., convergence of solution). This process then is repeated exhaustively for all $k \in K$ to solve the robust experimental design problem (Eq. (51)). Ushijima and Yeh (2013) showed that for a fixed $k$, it is possible to solve the experimental design problem of selecting optimal observation locations using a GA that calls a reduced model to estimate the Jacobian matrix. A flowchart of the GA with the POD reduced model appears in Figure 19.
4.3. Test Cases

4.3.a. One-Dimensional Test Case

We develop a synthetic, one-dimensional test cast to validate our algorithm (see Figure 20). The setup is similar to ones developed for previous studies related to POD reduced groundwater models (McPhee & Yeh, 2006b; Siade et al., 2010). The aquifer is 100 m long with a depth and width of 1 m. The aquifer is discretized into cubic cells 1 m per side. The specific storage \( S_s \) is
$1m^{-1}$ throughout the aquifer (we use a large value of $S$, here to speed-up the solution time to reach a steady state). Two wells pumping at the same rate ($1m^3/day$) are placed in the aquifer at node 31 and node 69 and the aquifer is divided into two hydrologic zones with unknown hydraulic conductivity (designated “Unknown Zone 1” and “Unknown Zone 2”), as seen in Figure 20.

![Figure 20. 1-D test case](image)

The upper and lower bounds on hydraulic conductivity throughout the aquifer are $20m/day$ and $0.1m/day$, respectively. When modeled in SAT2D (Paniconi & Putti, 1994) the full model consists of 303 nodes (as a result of the finite element mesh used to model the aquifer). We reduce this full model to a parameter-independent reduced model containing only 25 equations. On an Intel Core 2.40 GHz i5 CPU the full model takes 14 seconds to complete a run, while the reduced model takes 0.004 seconds to complete a run.

For the experimental design problem, we divide the aquifer into two pumping zones and four observation zones as shown in Figure 20. Zones 2 and 5 are designated as pumping zones while Zones 1, 3, 4, and 6 are designated as observation zones. We discretize the hydraulic conductivity space into three increments ($0.1m/day$, $10.05m/day$, and $20m/day$), yielding $2^3 = 9$ combinations of hydraulic conductivities (corresponding to nine different $\omega_i$’s). The optimal
sampling frequency can be determined through experimental design. However, for this study we take observations from the beginning to the end of the experiments. We let $\Delta k_j = 0.01k_j \forall j$ and observations are taken at 0.5, 1.5, 3, 5, 9, 15, 25, 40, 60, and 100 days. We investigate four experimental designs designated One, Two, Three, and Four observation well networks, respectively. Table 14 shows the number of model calls required for the algorithm to achieve convergence and the observation well locations chosen for the robust and HFO designs in the algorithm. In addition, Table 14 lists the observation well locations for a balanced design (i.e., a possible $a$-priori best guess). After finding the robust, HFO, and balanced designs we compare the results in three ways: the robust score (maximum-minimum score), mean score, and maximum score as shown in Figure 21.

Table 14. Number of model calls and observation well locations for the robust, HFO, and balanced designs for the 1-D Test Case

<table>
<thead>
<tr>
<th>1 observation well network</th>
<th>2 observation well network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model calls</td>
<td>HFO</td>
</tr>
<tr>
<td>141,210</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td>3 observation well network</td>
<td>4 observation well network</td>
</tr>
<tr>
<td>Model calls</td>
<td>HFO</td>
</tr>
<tr>
<td>218,202</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>
These metrics provide insight into which of the designs would be expected to perform better in the worst case, best case, and on average. It should be noted that the results are very closely tied to the specific experiment setup. In the setup here, as we see, the scores for each of the three metrics are fairly close. As expected by the definition, the robust design outperforms the other designs in robustness. However, in mean score and max score the robust design is sometimes outperformed by the other designs. The balanced design approximates the scores of the optimized designs fairly well but never attains the highest score.

Figure 21. Scores from the 1-D test case
After obtaining the optimal designs, we investigate the inaccuracy introduced to the optimal
design by replacing the full model with the reduced model in this 1-D test case. We test the
robustness of the information from the reduced model by systematically increasing the error
tolerance level ($\tau$) used during reduced model construction. This produces less accurate reduced
models; that is, the reduced models with fewer principal vectors. We found that with a one-
principal vector reduced model, the optimized robust and HFO design locations are very close to
the locations selected by the 25-principal vector reduced model. In addition, when using a two-
principal vector reduced model, the optimized designs select the same locations as the 25-
principal vector reduced model. This makes sense, as the first few principal vectors capture the
dominating characteristics of the full model. For this 1-D case, it appears that the design is not
sensitive to the changes in $\tau$. However, information is lost as the number of retained principal
vectors is decreased. Figure 22 shows the trace of the information matrix obtained by the robust
and HFO designs as well as the reduced model error (RMSE) as a function of the number of
retained principal vectors.
Figure 22. Information vs. number of retained principal vectors

As expected, the amount of information obtained by the HFO and robust designs both increase as a function of increasing number of retained principal vectors. The outliers to this trend are artifacts of the model reduction. It appears that the information loss plateaus around 10 principal vectors. This implies that there is virtually no information lost by reducing the number of principal vectors from 25 to 10, though this is highly model-dependent.
Figure 23. Estimated probability for a location to be an optimal location (1-D test case)

Figure 23 shows an approximation of the probability of a location serving as an optimal location (i.e., being a part of $\omega^*$). We achieve this by taking the frequency at which a location is an optimal location and normalizing the frequencies for a particular experimental design so that the frequencies sum to one. From this figure we see that, given the specific experimental setup, the observation wells tend to be located as close to the pumping wells as possible with a fairly balanced distribution around each pumping well. The results of the optimization are consistent with expected results. We deduce the expected results by considering that the optimality criteria chosen will cause the algorithm to choose nodes in the observation zones that are as close as possible to the pumping wells. Because of this, the nodes near the center of the aquifer will not
be chosen until the algorithm is forced to include them because of the constraints placed on the optimization, particularly the constraint that no more than one observation well can be placed in any observation zone. Based on the hydraulic conductivity variation used, we would expect a balanced distribution around each pumping well. Imbalances in estimated probabilities for the expected closest locations are the result of random noise as the algorithm chooses between equally likely locations. A larger sample size would conceal this random noise.

4.3.b. Two-Dimensional Test Case

We construct a 2-D horizontal test case to assess the algorithm on a large-scale, real-world-sized model. The model simulates the groundwater flow in a confined aquifer of the Oristano plain in west-central Sardinia, Italy (Cau et al., 2002; Siade et al., 2012). We assume the aquifer is surrounded on all sides by Dirichlet boundaries and is divided into seven hydrologic zones with unknown hydraulic conductivities, as seen in Figure 24.

![Figure 24. 2-D test case hydrologic zones](image-url)
The assumed properties of the hydrologic zones appear in Table 15.

Table 15. 2-D test case known hydrologic parameters

<table>
<thead>
<tr>
<th>Zone</th>
<th>$S_s$ (1/m)</th>
<th>Depth (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.20E-05</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>2.70E-05</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>1.00E-05</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>1.70E-05</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>2.10E-05</td>
<td>96</td>
</tr>
<tr>
<td>6</td>
<td>3.40E-05</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>1.00E-05</td>
<td>104</td>
</tr>
</tbody>
</table>

Twenty pumping wells are scattered throughout the aquifer (see Figure 27), all extracting water from the aquifer at the same rate $(1,000 m/day)$. Using SAT2D (Paniconi & Putti, 1994), the full model contains 29,197 nodes and 57,888 elements. Using the POD methodology described earlier, we construct a parameter-independent reduced model that contains only 113 equations. There is always some error associated with model reduction. To verify that the error for the reduced model is acceptable we randomly select 500 hydraulic conductivity combinations from a uniform distribution $\sim U(0.1 m/day, 20 m/day)$, run the full and reduced models for each combination, and compare the results. For each combination, we calculate and sort the norm of the error and the norm of the relative error between the full and reduced models at steady state in descending order based on the norm of the error. Figure 25 and Figure 26 show the error exceedance curves between the full and reduced models. These figures are similar to breakthrough curves and can be used to see how many parameter combinations have a worse
error than some specified cutoff. For example, Figure 25 shows that the maximum norm of the error at steady state for any parameter combination is 0.004 and the norm of the error is greater than 0.001 (the error tolerance used in the algorithm to construct the reduced model) for only 20 of the 500 parameter combinations sampled. Figure 26 shows the relative error (i.e., the norm of the error divided by the magnitude of the drawdown); we can see that the error introduced by model reduction is insignificant for all 500 combinations.

Figure 25. Norm of the error between the full and reduced model results at steady state
Figure 26. Percent error between the full and reduced model results at steady state

When run on an Intel Core 2.40 GHz i5 CPU, the full model takes 64 seconds to complete a run while the reduced model takes only 0.027 seconds to complete a run.

For the experimental design problem we divide the aquifer into 22 arbitrary pumping/observation zones. Pumping wells exist in Zones 3, 5, 6, 7, 9, 10, 11, 13, 14, and 15. These zones are designated as pumping zones and the rest (Zones 1, 2, 4, 8, 12, and 16 through 22) are designated as observation zones. For practicality, we overlay a coarse grid on the aquifer with a spacing of 500m and only allow observation wells to be placed on the grid nodes that fall within an observation zone. After performing this, we find 328 feasible observation well locations. Sun and Yeh (2007) demonstrated that considering only the upper and lower bounds of the hydraulic conductivity will provide sufficient information to develop a robust experimental design. Therefore, when solving the max-min experimental design problem, we only consider the upper and lower bounds of the hydraulic conductivity (0.1m/day and 20m/day), yielding a total of
$2^7 = 128$ hydraulic conductivity combinations. The number of feasible observation well locations and hydraulic conductivity combinations define the dimension of the experimental design problem. The experimental design problem setup appears in Figure 27.

![Figure 27. 2-D test case experimental setup](image)

We let $\Delta k_j = 0.01k_j \ \forall j$, take observations at 0.1, 0.5, 1, 5, 10, and 50 days and investigate a number of experimental designs. First, we optimize a single observation well network, then we optimize a two-observation well network, and so on until finally we optimize a twelve-observation well network. The results are shown in Figure 28 - Figure 33.
Figure 28. Estimated probability for a location to be an optimal location for the 1, 2, 3, and 4 observation well networks (2-D Test Case)
Figure 29. Estimated probability for a location to be an optimal location for the 5, 6, 7, and 8 observation well networks (2-D Test Case)
Figure 30. Estimated probability for a location to be an optimal location for the 9, 10, 11, and 12 observation well networks (2-D Test Case)
Figure 31. Robust and HFO designs for 1, 2, 3, and 4 observation well networks (2-D Test Case)
Figure 32. Robust and HFO designs for 5, 6, 7, and 8 observation well networks (2-D Test Case)
Figure 33. Robust and HFO designs for 9, 10, 11, and 12 observation well networks (2-D Test Case)
In Figure 28-Figure 30, contours represent the frequency at which a particular location is chosen as a $\tilde{\omega}_i^{(j)}$ (the $j^{th}$ optimal location in $\tilde{\omega}_i$). This frequency is scaled to add-up to one, giving us an estimated probability that a particular location will be a $\tilde{\omega}_i^{(j)}$. As we see in the figures, as the dimension of the optimization problem (the number of observation wells being optimized) increases, the number of unique locations chosen as a $\tilde{\omega}_i^{(j)}$ grows. However, the highest frequency locations tend to remain in the same general area of the aquifer. We observe in the figures that all $\tilde{\omega}_i^{(j)}$ seem to show a higher preference for the observation zones in the upper left corner of the aquifer and a lower preference for the observation zones in the lower center of the
aquifer. It is interesting to point-out that when we optimize the twelve observation well network almost every feasible location is chosen as a $\tilde{\omega}_j^{(j)}$ at least once. Figure 31-Figure 33 show the HFO designs (x) and robust designs (○). From the figures, one sees that the HFO designs tend to be fairly close to the robust designs. However, only in one observation well network are the two designs the same. We compare the scores of the HFO and robust designs in the same way as in the 1-D test case -- the max-min (robust) score, the max score, and the mean score. Again, as with the 1-D test case, the results are tied closely to the specific experiment setup. For the 2-D test case, Figure 34 shows that the robust design outperforms the HFO design in robust score (as expected) and, for most cases, in max score. On the other hand, we see that for most cases the HFO design outperforms the robust design in mean score. In all cases, however, the scores are fairly close. As with the 1-D test case, the figures that are perhaps most interesting are the ones that show the contour plots estimating the probability of a location being an optimal location. Figure 34 also shows the number of model calls required by the algorithm to achieve convergence. As expected, in general as optimization complexity increases (i.e., larger observation well networks), the number of model calls required to achieve convergence increases. Since this is a heuristic algorithm, this is not an absolute requirement; thus we observe that for some optimizations, the algorithm converges faster than for less complex optimizations.

4.4. Conclusions

We have developed a robust experimental design algorithm that can be used to select the locations for a network of observation wells that provide maximum robust information about the unknown hydraulic conductivities. The algorithm maximizes information in the worst-case scenario. The formulated combinatorial, max-min problem is nonlinear and non-convex with mixed integer variables. These characteristics make it very difficult, if not impossible, to solve a
real-world-sized problem using traditional mathematical programming techniques, such as mixed integer nonlinear programming. This paper’s novelty is demonstrating that a genetic algorithm (GA) coupled with a parameter-independent reduced model is able to overcome this difficulty.

The algorithm accomplishes this by taking some feasible range of hydraulic conductivity for each of the $n_z$ hydrologic zones with unknown hydraulic conductivity, discretizing this range into $n_l$ increments, and forming all $n_l^{n_z}$ possible combinations of these zonal hydraulic conductivities. Then, for a given number of observation wells $n_{obswell}$, a groundwater model approximates the Jacobian and information matrices to find $n_l^{n_z}$ solutions $\left(\hat{\omega}_{i=1,...,n_l^{n_z}}\right)$, each of which maximizes the information criterion for a given parameter combination. The algorithm scores each feasible location with the number of times that location is selected as part of a $\hat{\omega}_i$ and creates a contour map of the score of each feasible location (scaled to sum to one). This approximates the probability of a feasible location serving as an optimal location (i.e., being a part of $\omega^*$). The $n_{obswell}$ locations with the highest scores are classified as the High Frequency Observation Well (HFO) design. The algorithm adds this design to the original set of designs $\left(\hat{\omega}\right)$ to produce a final set $\left(\hat{\omega}_{+1}\right)$ with $n_l^{n_z} + 1$ designs. A groundwater model then is called to evaluate the optimality (i.e., fitness score) of all designs in $\hat{\omega}_{+1}$ for all $n_l^{n_z}$ parameter combinations. The algorithm records the robust, maximum, and mean scores for each design and classifies the design with the maximum-minimum score as the robust design. As we have shown, this algorithm requires an inordinately large number of groundwater model calls, rendering calling a full groundwater model infeasible. To overcome this, the algorithm calls a parameter-independent reduced groundwater model. The reduced model runs several orders of magnitude
faster than the full model. This allows the problem to be solved in the reduced space, drastically reducing the computational burden of the algorithm.

We first tested our algorithm on a small 1-D experimental setup to test its convergence and then ran it on a realistically-sized 2-D experimental setup. The parameter-independent model reduction algorithm reduced the 303-equation, 1-D full model to a 25-equation reduced model and the 29,197-equation full 2-D model to a 113-equation reduced model. In both cases, because of model reduction, the algorithm achieved convergence in a reasonable amount of time.

For the two experimental setups, the HFO design performs similarly, but slightly worse than the robust design and sometimes outperforms the robust design in max and mean score. We find the estimated probability of a feasible location being an optimal location by plotting the frequency (scaled to sum to one) that a particular feasible location is selected as a $\hat{\omega}_i^{(j)}$ (the $j^{th}$ optimal location in $\hat{\omega}_i$). Executing this algorithm for the 1-D case took 761,817 model calls and 142,728,056 model calls for the 2-D case. Given that we saw a time improvement of several orders of magnitude with both the 1-D and 2-D test cases, the advantage of solving the problem in the reduced space is obvious. For comparative perspective, if we were to use a GA coupled to the full 2-D model at 64 seconds per model call it would take 9E9 seconds, or 289.7 years, to achieve convergence. Clearly this is completely infeasible, but by replacing the full model with the reduced model we are able to solve this formerly infeasible experimental design problem.

For both the 1-D and 2-D cases the algorithm converged to robust and HFO designs that had similar scores. As expected, the robust design always performed at least as well as the HFO design in robust scoring. On the other hand, there seemed to be no real pattern controlling which design performed better with the mean or maximum scores. We expect that the maximum and
mean scores are very sensitive to the experimental setup. Perhaps even more interesting than these results, the algorithm allows us to approximate the probability of a feasible location serving as an optimal location for both test cases. Along with solving for the robust and HFO designs, this approximation also would be infeasible without coupling the reduced model to the algorithm. The results show that the GA coupled with a parameter-independent reduced model is a valid search method to search for the optimal solution to the computationally expensive max-min optimization required in robust experimental design. Along the way, the algorithm can collect additional data, such as finding the HFO design and mapping an approximation of the probability of a location being an optimal location.

In future work we could research experimental designs that seek information about both unknown parameters and unknown pumping rates. We also could apply parameter and pumping-independent reduced modeling to study the data sufficiency problem of designing an optimal pump test and observation network. Another application that may be of interest would be coupling a reduced model to a conjunctive use optimization model, either with known or unknown parameters.

**Notation**

\[
\begin{align*}
A &\in \mathbb{R}^{N_x \times N_u} \quad \text{Stiffness matrix for the full model} \\
\tilde{A} &\in \mathbb{R}^{N_p \times N_p} \quad \text{Stiffness matrix for the reduced model} \\
A(k) &\in \mathbb{R}^{N_x \times N_u} \quad \text{Matrix representing the system of linear equations in the full model} \\
\tilde{A}(k) &\in \mathbb{R}^{n_p \times n_p} \quad \text{Matrix representing the system of linear equations in the reduced model} \\
B &\in \mathbb{R}^{N_x \times N_u} \quad \text{Mass matrix for the full model}
\end{align*}
\]
$B \in R_{n_p \times n_p}$ Mass matrix for the reduced model

$b \in R_{N_a}$ Vector describing the forcing of the full model

$\tilde{b} \in R_{N_a}$ Vector describing the forcing of the reduced model

$\chi$ Matrix of snapshots collected at some hydraulic conductivity $k$

d Scaling factor for the residual

$F$ Specific volumetric pumping rate

$\Delta K_j$ Perturbation of the $j^{th}$ hydraulic conductivity

$\Delta k_j \in R^n$ Vector perturbing the $j^{th}$ hydraulic conductivity

$\Delta o_i$ Change in the $i^{th}$ observation

$\Delta \theta_j$ Perturbation of the $j^{th}$ parameter

$\Delta \theta_j \in R^N$ Vector perturbing the $j^{th}$ parameter

$f_1, f_2, f_3$ Known functions describing initial and boundary conditions of an aquifer

$e$ Error between the full and reduced model

$\Gamma_1$ Fixed head boundary

$\Gamma_2$ Flux boundary

$H \in R_{N_s \times n_p}$ Vector of initial head values

$h \in R_{N_s \times n_p}$ Vector of head values

HFO High frequency observation well design

$I \in R_{n_q \times n_q}$ Information matrix calculated by $J_d^T * J_d * W$

$J_d \in R_{n_{obs} \times n_q}$ Jacobian matrix containing only the observations of interest to all wells
\[ J_{d_{i,j}} \] \( i, j \)th element in \( J_d \)

\( K \in \mathbb{R}^{n_x \times n_u} \) Matrix of columns storing all parameter combinations to be searched

\( K_i \) Hydraulic conductivity in the \( i^{th} \) direction

\( \hat{K} \) The space of feasible hydraulic conductivities

\( k \in \mathbb{R}^{n_z} \) Vector of hydraulic conductivities

\( M \in \mathbb{R}^{n_{obs} \times N_x} \) Observation matrix isolating the rows of \( P \) corresponding to some \( \omega \)

\( A \in \mathbb{R}^{N_x \times N_u} \) Matrix containing the eigenvalues of \( X \)

\( L \) Length unit (meters, feet, etc.)

\( N \) Total number of parameters of interest

\( n_l \) Number of parameter increments

\( N_n \) Number of nodes in the full model

\( n_{obs} \) Total number of observations taken

\( n_{obs\text{well}} \) Maximum number of allowable observation wells

\( n_p \) Number of principal components used in the reduced model

\( n_{sp} \) Number of snapshots taken for each pumping well to build the reduced model

\( n_z \) Number of hydrologic zones with unknown hydraulic conductivity

\( P \in \mathbb{R}^{N_x \times n_p} \) Full projection matrix

\( \hat{P} \in \mathbb{R}^{N_x \times n_p} \) Principle vectors of a particular set of snapshots

\( \text{PDE} \) Partial differential equation

\( \Omega \) The space of all feasible sets of observation well locations

\( \omega \in \mathbb{R}^{n_{obs\text{well}}} \) Vector contacting some feasible set of observation well locations (\( \omega \subset \Omega \))
\[ \hat{\omega} \in R^{n_{obs} \times n_{x}} \text{ The set of all } \hat{\omega}_i \]
\[ \hat{\omega}_{i+1} \in R^{n_{obs} \times n_{x} + 1} \text{ } \hat{\omega} \text{ plus the HFO design} \]
\[ \omega^* \in R^{n_{obs} \times n_{x}} \text{ The optimal experimental design solution given the true aquifer parameters} \]
\[ \hat{\omega}_i \in R^{n_{obs} \times n_{x}} \text{ } i^{th} \text{ solution output by the GA} \]
\[ \hat{\omega}_i^{(j)} \text{ } j^{th} \text{ location in } \hat{\omega}_i \]

ODE  Ordinary differential equation

\[ o_i \text{ } i^{th} \text{ observation} \]
\[ o_j(\theta) \text{ } i^{th} \text{ model simulated value using the parameter values in } \theta \]

\[ Q \text{ Matrix containing the eigenvectors of } X \]

\[ q \in R^{N_x} \text{ Vector of sinks for the full model} \]
\[ q_n \text{ Specific discharge normal to the flux boundary } (\Gamma_2) \]

\[ R(k) \in R^{n_p \times n_{obs}} \text{ Matrix containing the reduced solution at the observation times} \]
\[ r \in R^{n_p} \text{ Vector of the reduced solution at time } t \]
\[ r_e \text{ The residual between the full and reduced model} \]

\[ s \in R^{N_x \times n_p} \text{ Vector of drawdown values} \]
\[ s_i \text{ } i^{th} \text{ drawdown value} \]

\[ \Sigma \in R^{n_p \times n_p} \text{ Diagonal matrix containing the singular values of } X \]
\[ S_s \text{ Specific storage} \]
\[ \hat{s}_i \in R^{N_x} \text{ Vector of the approximation of } s_i \]

SVD  Singular value decomposition
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>Time unit (days, hours, etc.)</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Error tolerance</td>
</tr>
<tr>
<td>( \theta \in \mathbb{R}^N )</td>
<td>Vector of nominal parameter values</td>
</tr>
<tr>
<td>( \theta_j )</td>
<td>( j^{th} ) parameter</td>
</tr>
<tr>
<td>( V \subset \mathbb{R}^{N_x} )</td>
<td>Subspace spanned by the columns of ( P )</td>
</tr>
<tr>
<td>( V \in \mathbb{R}^{n_p \times n_p} )</td>
<td>Matrix containing the right singular vectors of ( X )</td>
</tr>
<tr>
<td>( W \in \mathbb{R}^{n_r \times n_q} )</td>
<td>Some user specified weighting matrix used in calculating ( I )</td>
</tr>
<tr>
<td>( X \in \mathbb{R}^{N_s \times n_p} )</td>
<td>Matrix containing all collected snapshots</td>
</tr>
<tr>
<td>( x \in \mathbb{R}^{N_x} )</td>
<td>Vector of binary variables indicating if a node has an observation well</td>
</tr>
<tr>
<td>( x_i )</td>
<td>( i^{th} ) element in ( x )</td>
</tr>
<tr>
<td>( zone \ j )</td>
<td>All the nodes in the ( j^{th} ) observation zone</td>
</tr>
</tbody>
</table>
4.5. Bibliography


estimation for parametrized dynamical systems by offline/online decomposition.


http://doi.org/10.1080/13873954.2010.514703


http://doi.org/10.1029/WR025i005p01025


Chapter 5

Comparative Study of Employing Various Optimality Criteria for Experimental Design in Groundwater Modeling (Timothy T. Ushijima & Yeh, 2018)

Abstract

We examine the relationships among designs that result from employing various optimality criteria to an experimental design problem in groundwater modeling. The relationships among designs are measured by each design’s “efficiencies”. Efficiency differs from simple optimization in that it measures the ratio of a design’s score to the best possible score, both with respect to some optimality criterion. We apply this concept to an experimental design problem that seeks to select a network of observation wells for sampling in order to estimate the unknown aquifer parameters. We consider the five classical A-, D-, E-, I- and G- optimality criteria. Since finding an optimal observation well network requires solving a combinatorial optimization problem that is non-convex, non-differentiable, and contains integer variables, we use Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to perform the search. To facilitate the search, GA and PSO are coupled with a reduced groundwater model to perform the large number of model calls. We use Proper Orthogonal Decomposition (POD) to reduce a highly discretized groundwater model such that the dimension of the reduced model is orders of magnitude smaller than the original full model and runs considerably faster. We demonstrate the relationships among optimality criteria on two test cases. One is a small synthetic aquifer and the other is a large-scale groundwater model.
5.1. Introduction

5.1.a. Background

Estimating unknown hydrologic properties through inverse modeling is an important area of research in all areas of hydrology, including groundwater modeling. In order to solve an inverse model, we must have field observations (generally of the hydraulic head). To do this, we must face the problem that collecting good observations is expensive, time consuming, and difficult, particularly if the observations vary spatially. In the ideal world, we would be able to make observations at infinitesimally fine resolutions in the temporal and spatial domains; however, in the real world we are not able to do this because the resolution at which we can make observations is limited by technology and budget. The goal of optimal experimental design is to select observations (these can vary both spatially and/or temporally) such that some specified criterion is optimized, subject to a set of constraints. Cost, reliability of estimated parameters, and timing or duration of sampling are common constraints placed on an experimental design problem. There are two general approaches to select the observations and they are not mutually exclusive. One approach uses sequential observations; i.e., observing the real system, updating the model, and then using the updated model to select the next round of observations. The benefit of this method is that the model continually is being updated with new information. This approach allows for such methods as the Ensemble Kalman filter to be employed in innovative ways (Man, Zhang, Li, Zeng, & Wu, 2016). Downsides to this method include the real-world cost in time and repeated sampling and the reality that some optimality criteria do not lend themselves as well to systematic sampling as others (e.g., A-optimality with its additive nature lends itself more to sequential sampling than does D-optimality, which considers all observations, referred to as the observation network, as a whole). The alternate approach to
sequential sampling is to design the entire network a-priori and then collect observations. The benefits of this approach include a reduction in real-world time and installation costs and the ability to design the network as a whole rather than one at a time (which can lead to sub-optimal results). Of course the downside of this method is that it is much more dependent on a-priori knowledge than sequential sampling. This dependence generally leads to a compromise design, for example a “mode” design (Jin, Ranjithan, & Mahinthakumar, 2014) or a “worst-case” design (T.T. Ushijima & Yeh, 2015). Of course, concepts from these two frameworks can be combined to take advantage of each method’s benefits while minimizing the associated drawbacks (Wöhling, Geiges, & Nowak, 2016). For purposes of this study, the second approach (a-priori designing the complete network) is used; however, there is no reason the concepts could not be applied to a sequential sampling study.

In general, the design of an observation network is formulated as an optimization problem whose optimality criterion is some measure of the useful information contained in a design’s information matrix. The formulation usually lends itself to a combinatorial optimization problem that is non-linear and non-convex. In the context of groundwater modeling, much research has attempted to overcome the challenges posed by these non-linear, non-convex, combinatorial optimization problems. Heuristic searches such as Genetic Algorithms (GAs) or Particle Swarm Optimization (PSO) searches have demonstrated their ability to search-out the solution to large-scale optimization problems that are difficult or impossible to solve with traditional mathematical programming techniques. GAs have been applied effectively to experimental design in groundwater modeling (Babbar-Sebens & Minsker, 2010; McPhee & Yeh, 2006b; Reed et al., 2000). Other research has shown that model reduction techniques, such as Proper Orthogonal Decomposition (POD), can be used to reduce the dimension of a highly discretized
groundwater model (T.T. Ushijima & Yeh, 2015; Timothy T. Ushijima & Yeh, 2013). The reduced model, which is orders of magnitudes smaller than the original full model, can then be coupled to a heuristic search to perform a combinatorial search in the reduced space. With these tools in hand, it is possible to apply experimental design to many problems that were previously intractable. These tools also allow us to investigate how to best apply experimental design to a groundwater problem. The primary question that must be answered is which optimality criteria one should use to design the optimal observation network. It is of course possible to employ multi-objective optimization (e.g., coupling a classic optimality criterion with a metric of the distance between observations (Jin et al., 2014)). However, many times a single objective optimization is used, with A- or D- optimality the most common optimality criterion. A- and D-optimality criteria have been used in many studies (Altmann-Dieses et al., 2002; Catania & Paladino, 2009; Cleveland & Yeh, 1991; Hsu & Yeh, 1989; Knopman & Voss, 1989; Nishikawa & Yeh, 1989) and each of these two optimality criteria offer different benefits and drawbacks. A-optimality is computationally simple and stable as well as easy to measure. It can be used to determine areas of an aquifer that are sensitive to individual parameters of interest but cannot distinguish new information from what already has been collected (i.e., the network may produce highly correlated data). On the other hand D- (and E-) optimality consider the joint confidence ellipsoid (Nordqvist, 2000) but can be difficult to measure. Sciortino et al. (2002) compared A- and D-optimality with respect to parameter estimation and demonstrated that D-optimal designs will tend to produce parameter estimates with lower uncertainty, but A-optimal designs may produce more accurate parameter estimates. This discrepancy highlights the fact that the “best” answer might be the result of a compromise among optimality criteria rather than a single result. Finding this best answer would require performing a multi-objective optimization, something
that would be prohibitively expensive. The challenge, then, is to find the best-feasible solution (i.e., one that is not prohibitively expensive to find). To do this, one must look deeper than a simple least-error framework and consider how the optimality criteria relate to one another (i.e., how optimal they are with respect to each other).

In his seminal work on experimental design, Kiefer (1959) argued that a design that is optimal in one measure will be close to optimal with respect to other measures. While other fields have sought to verify this assumption and to quantify “close”, the field of groundwater research has yet to address this fundamental question. In this study, we provide insight into this question by discussing the general properties of different criteria and comparing the relationships among the results.

5.1.b. Experimental Design

Experimental design encompasses the methods that seek to predict and minimize some measure of error of a proposed model. The goal of an experimental design study is to produce a sampling scheme (referred to as a design) such that the information provided by that design will allow the construction of the most accurate model possible. In general, the proposed model is a linear model of the form

\[
s = J_d \theta
\]

(54)

where \( s \in R^m \) is the state vector, \( \theta \in R^n \) is the vector of model parameters, and \( J_d \in R^{m \times n} \) is the sensitivity matrix (referred to as the Jacobian matrix) containing elements as shown below:
where \( s_i \) is the \( i^{th} \) element in \( s \), \( \theta_j \) is the \( j^{th} \) element in \( \theta \), and \( \frac{\partial s_i}{\partial \theta_j} \) is the sensitivity of \( s_i \) with respect to changes in \( \theta_j \). Eq. (54) is a surrogate model for some system model – in our case, a groundwater model. To set-up the experimental design problem we form the elements of \( J_d \) from the groundwater model we seek to gain information about. Three different methods can be used to calculate the sensitivities in \( J_d \): (1) the parameter perturbation method, (2) the sensitivity equation method, or (3) the adjoint state method (Yeh, 1986). The parameter perturbation method has the following general form:

\[
\frac{\partial s_i}{\partial \theta_j} \approx \frac{s_i(\theta + \Delta \theta_j) - s_i(\theta)}{\Delta \theta_j}
\]

where \( \Delta \theta_j \) is a small change of the the \( j^{th} \) parameter (called the perturbation of \( \theta_j \)) (Poeter et al., 2005). There are two general requirements for using Eq. (56) to make accurate estimates of \( \frac{\partial s_i}{\partial \theta_j} \).

First, \( \theta \) must be close to the true parameter values, and second, \( \Delta \theta_j \ll \theta_j \).
5.1.c. Optimality Criteria

Before formulating the experimental design problem, we need to determine what information we want to obtain about our proposed model, as this will dictate how the problem is formulated.

There are two general classes of information we can obtain about our proposed model:

1. Information about the model parameters ($\theta$)
2. Information about the model’s predictions of the state vector ($s$)

Based on what information we seek to obtain, we choose an appropriate optimality criterion. From classical experimental design, there are five commonly employed optimality criteria (Steinberg & Hunter, 1984).

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Information Class</th>
<th>Objective Function ($\Phi, (\omega)$)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\theta$</td>
<td>$\max_{\omega} \left( \text{trace}(F) \right)$</td>
<td>Maximizes total information</td>
</tr>
<tr>
<td>E</td>
<td>$\theta$</td>
<td>$\max_{\omega} \left( \min \lambda_i (F) \right)$</td>
<td>Maximizes unique information</td>
</tr>
<tr>
<td>D</td>
<td>$\theta$</td>
<td>$\max_{\omega} \left( \text{det}(F) \right)$</td>
<td>Maximizes uncorrelated information</td>
</tr>
</tbody>
</table>
\[
F = J_d^T J_d
\]  
(57)

\(j_{d,i}\) is the \(i^{th}\) row in \(J_d\), \(\lambda_i\) is the \(i^{th}\) eigenvalue of \(F\), \(\omega\) is some sampling strategy, and \(J_d\) is formed from that \(\omega\). Under the assumptions that 1) a least-squares error criterion is used for parameter estimation and 2) the observation errors are uncorrelated and have equal variance, \(F\) is the inverse of the covariance matrix of estimated parameters (Cleveland & Yeh, 1990; Kutner et al., 2004). Since E- and D-optimality are traditionally defined in relation to the covariance matrix, the formulations shown in Table 16 lead to scores that are the inverse of the scores produced by the traditional formulation. This is not the case for the criterion labeled A-optimality. The formulation labeled A-optimality is the maximum information criterion that is not equivalent to A-optimality. However, it can be proven that the design that is maximum information-optimal is also A-optimal and vice-versa. For this reason, we use the maximum information criterion but refer to it as A-optimality. For G- and I-optimality, the range of \(i\) may be limited (i.e., the minimization can be carried-out over some subset of \(s\) that is of particular interest or over all the elements in \(s\)). In past experimental design research, A- and D-optimality
have been used to estimate unknown aquifer parameters (Altmann-Dieses et al., 2002; Knopman & Voss, 1989) and design observation well networks (Catania & Paladino, 2009; Knopman & Voss, 1989; T.T. Ushijima & Yeh, 2015; Timothy T. Ushijima & Yeh, 2013), and D-optimality has been used to design optimal pumping tests (Nishikawa & Yeh, 1989). G- and I-optimality have not been applied widely to groundwater modeling. Once an appropriate optimality criterion has been chosen, we formulate the experimental design problem as the following optimization problem:

\[
\text{Optimize} \quad \omega \in \Omega \quad (\text{Objective Function}) \\
\text{s.t. Constraints}
\]

where \( \Omega \) is the set of all possible sampling locations and the constraints include limitations on the size (number of observation wells), location, or distribution of \( \omega \).

5.1.d. Evaluation Challenges

In general, for small-scale problems, A-optimal designs may be found through linear programming, and D-, E-, and I-optimal designs may be found through convex optimization. G-optimal designs, however, are not linear or convex and are generally difficult to find even for small-scale problems. In the context of experimental design problems for developing an optimal observation well network for a groundwater model, solving this optimization problem is even harder. Often these problems are non-convex, non-differentiable, and require a combinatorial search to solve. For a realistically-scaled groundwater model, the combinatorial search would be infeasible to solve with traditional mathematical programming techniques because of the hundreds or thousands of nodes in the model. Heuristic algorithms can be used to overcome this
challenge (Montas HJ, Mohtar RH, Hassan AE, 2000). An example of such an algorithm would be Genetic Algorithms (Babbar-Sebens & Minsker, 2010; McPhee & Yeh, 2006b; Reed et al., 2000; Wu J, Zheng C, 2005), which utilize breeding and mutation to search a solution space. An alternate heuristic search commonly used for other optimizations is the Particle Swarm Optimization (PSO). PSOs search the solution space by deploying sets of particles that utilize position and velocities to update the best candidate solution (Kennedy & Eberhart, 1995). Each type of heuristic search presents its own unique advantages and disadvantages. Utilizing velocities allows PSOs to converge quickly but at the cost of an increased likelihood of becoming trapped in a local optimum. On the other hand, a GA sacrifices speed to take advantage of the randomness of its mutations to attempt to avoid being trapped in local optima (He, Xu, Yang, & Liao, 2014). Since heuristics searches are not guaranteed to converge to the global optimum they also will not necessarily converge to the same local optimum, even among repeated runs. Despite this, by design, heuristic searches converge toward the global optimum and thus should produce similar results. Heuristic searches also introduce the challenge that an inordinately large number of model calls are required, possibly rendering the search ineffective. However, past research has demonstrated that this can be overcome through methods such as model reduction (T.T. Ushijima & Yeh, 2015; Timothy T. Ushijima & Yeh, 2013).

From a computational standpoint, evaluating the A- or E-optimality of some \( \omega \) is straightforward; however, evaluating the D-, G-, or I-optimality of some \( \omega \) presents computational challenges. Computing the determinant \( (\det(\bullet)) \) of a general matrix will be unstable for a nearly singular matrix (i.e., \( \det(\bullet) \approx 0 \)). For this reason utilizing the determinant in an algorithm is discouraged (Anderson et al., 1999). A method that can be used to stabilize the
calculation of the determinant takes advantage of the fact that the determinant of a matrix is the product of all the eigenvalues of that matrix. One could include a heuristic to declare the determinant equal to zero if a small enough eigenvalue is observed. An additional challenge that optimal observation network heuristic searches face when utilizing D-optimality is that the objective function surface will be very flat, making it very difficult to find a good search direction. A way to overcome this challenge is to artificially exaggerate the objective function surface features to assist the algorithm to find a good search direction. We do this by taking note that for a \( n \times n \) matrix \( A \)

\[
\det(cA) = c^n \det(A)
\]

where \( c \) is a constant. We take advantage of this by pre-multiplying the \( F \) by a constant \( c \) such that

\[
O\left(\det(cF)\right) \approx 1
\]

where \( O(\bullet) \) is the order of magnitude. By doing this, small fluctuations in the objective function surface are magnified so that they are not missed by a heuristic search due to round-off error. Once the heuristic algorithm has searched-out the optimal solution, \( c^n \) then may be divided-out to find the true value of \( \det(F) \).

The computational challenge of computing G- or I-optimality is related to the numerical stability of computing D-optimality. As Table 16 shows, computing G- or I-optimality following the formal definition requires computing a matrix inverse \( (F^{-1}) \). Since
\[ F^{-1} = \frac{1}{\det(F)} \text{adj}(F) \]

computing the matrix inverse will be unstable for nearly singular matrices. To overcome this challenge, we note Eq. (57) implies

\[ F^{-1} = J_d^{-1} \left( J_d^T \right)^{-1} = J_d^{-1} \left( J_d^{-1} \right)^T \]

Furthermore we note that if we apply singular value decomposition to \( J_d \) such that

\[ J_d = U \Sigma V^T \]

where \( U \) is a matrix containing the left singular vectors, \( \Sigma \) is a diagonal matrix containing the singular values \((\sigma_i)\), and \( V \) is a matrix containing the right singular values; then

\[ J_d^{-1} = V \Sigma^{-1} U^T \]

where \( \Sigma^{-1} \) is a diagonal matrix containing the inverses of the singular values \( \left(\frac{1}{\sigma_i}\right) \). We then recognize

\[ j_{d,i} \left( F^{-1} \right) \left( j_{d,i} \right)^T = j_{d,i} \left( V \Sigma^{-1} U^T U \Sigma^{-1} V \right) \left( j_{d,i} \right)^T = \left( j_{d,i} V \right) \Sigma^{-2} \left( j_{d,i} V \right)^T \]
where $\Sigma^{-2}$ is a diagonal matrix containing the squares of the inverses of the singular values $\left(\frac{1}{\sigma_i}\right)$. Thus by employing a singular value decomposition of $J_d$ we are able to avoid computing $F^{-1}$ and are able to evaluate the G- and I-optimality of some $\omega$ in a stable manner.

The final evaluation challenge we face in this study is that the objective function values (scores) only have meaning in a relative sense (how much larger or smaller is the score than the score of some other design). In addition, there is no way to compare the scores directly, so the concept of design efficiency is adopted to compare a design’s performance across measurements (Wong, 1994). We define a design’s efficiency as

$$E_\star = \frac{\Phi_{\star}(\omega)}{\Phi_{\star}(\omega^\star)}$$

(58)

where $\omega^\star$ is the best-possible design with respect to $\Phi_{\star}(\omega)$. With the efficiency measure, we are able to determine how optimal some $\omega^\star$ is with respect to a different $\Phi_{\star}(\omega)$ (e.g., how well an A-optimal design approximates an E-optimal design). This ratio allows us to see, for example, how much E-information is contained within an A-optimal design. With these tools in hand, we are able to carry-out the comparative study.

5.2. Test Cases

5.2.a. Small-Scale, Two-Dimensional Test Case

We develop a synthetic, two-dimensional test case to demonstrate our study. The synthetic aquifer is confined, has dimensions of $25m \times 35m$ and is divided into three hydrologic zones (see Figure 35) with varying hydrologic properties. Table 16 shows the hydrologic properties of
the aquifer. Note that an unusually large value of specific storage is assumed to speed-up the simulation (i.e., decrease the time required to reach steady state). Constant-head conditions are imposed on the north and south boundaries and no-flow conditions apply to the east and west boundaries.

![Figure 35. Small-scale test case hydrologic zones](image)

**Table 17. Small-scale test case hydrologic properties**

<table>
<thead>
<tr>
<th>Zone</th>
<th>Specific Storage (m$^{-1}$)</th>
<th>Depth (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>22</td>
</tr>
</tbody>
</table>
We assume the hydraulic conductivities in this aquifer are unknown, so the experimental design seeks to gain information on the hydraulic conductivities in each of the zones. We divide this aquifer into nine observation/pumping zones (Figure 36). Zones 1-6 are set as observation zones and zones 7-9 are set as pumping zones. One set of constraints that we place on the experimental design is that at most one observation well can be placed in an observation zone and no observation wells can be placed in a pumping zone.

![Figure 36. Small-scale test case experimental design setup](image)

Since $J_d$ depends on hydraulic conductivity, we perform a robust experimental design for the worst case scenario (T.T. Ushijima & Yeh, 2015). We assume the hydraulic conductivity in each of the three hydrologic zones is in the range $[0.1 \text{ m/day}, 20.0 \text{ m/day}]$. We follow the technique outlined by Ushijima and Yeh (2015) and solve the experimental design over the set formed by discretizing the range of each zone into three levels, then forming all combinations of levels of zonal hydraulic conductivity. In addition to the constraints on number and location of observation wells with respect to the observation/pumping zones, we explore how the results vary with respect to changes in the total number of observation wells by constraining the total.
number of observation wells, then varying that constraint over the range $[1, 6]$. In order to observe how differences in heuristic searches may affect the results, for each of the optimality criteria list in Table 16 we use both a GA (Wall, 1995) and a PSO (Pedersen, 2008) to find the robust one-observation-well network (one-well design) up through the six-observation-well network (six-well design). Though these results are tied closely to the specific experimental design setup, thus precluding absolute general statements, we can make some interesting observations.

First, as would be expected, the GA and PSO converged to similar but different results in terms of both spatial distributions of the observation wells and objective function score. In general the scores were within $O(1\%)$ of each other but for this test case, the PSO outperformed the GA (i.e., produced more optimal results) for smaller designs (i.e., those with smaller observation well networks) but the GA outperformed the PSO for larger designs. From these two sets of results, the best (based on objective function score) designs were compiled into one set and are shown in Figure 37-Figure 38.
Figure 37. A-, D-, and E-designs
Figure 38. G- and I-designs

From Figure 37 and Figure 38 we can see that the observation well spatial distribution for the various designs are consistent with what would be expected from each criterion. The A-designs (i.e., the designs to which the GA and PSO converged over the A-solution space) grouped their observation wells close to pumping wells regardless of whether or not the information gained would be redundant; the D-designs tended to spread-out the observation wells the most, consistent with minimizing covariance; and the E-designs spread-out the observation wells evenly, consistent with maximizing unique information. It is a little harder to predict how I- and G-designs will behave but one would expect the resulting designs to be similar spatially and we
see this in the results. After plotting-out the observation well locations, we graphed the efficiencies for these designs in Figure 39-Figure 43.

![Figure 39. A-design efficiencies](image)

![Figure 40. D-design efficiencies](image)
Figure 41. E-design efficiencies

Figure 42. G-design efficiencies
In this test case, we note that all the designs are fairly A-optimal (i.e., have high A-efficiencies) but for the most part, the A-designs did not have high non-criterion efficiencies. As might be expected, the E- and D-designs are fairly optimal with respect to each other, as are the G- and I-designs. Taking a higher-level view, we note that the prediction-designs (G- and I-designs) are more optimal with respect to the other prediction-design than they are to parameter-designs (A-, E-, and D-optimality). The same is seen for the parameter-designs, as they are more optimal with respect to the other parameter-designs than they are with respect to the prediction-designs. While analyzing the results, the general weakness of heuristic searches becomes apparent, as both the GA and PSO converge to sub-optimal solutions. This becomes obvious when the non-criterion efficiencies are calculated and it is clear that the GA-one-, three-, and four-well D-designs and the four-well G-design had E-efficiencies greater than 100% (i.e., these designs are more E-optimal than the E-designs of the same size). In addition, the five- and six-well G-designs have I-efficiencies greater than 100%. We thus conclude that the GA became trapped in sub-optimal
solutions while searching the E- and I-solution spaces. Based on efficiency, we also conclude that the PSO was trapped in a sub-optimal solution while searching for the five-well I-design. Although neither the GA nor the PSO are able to converge to the best observed location, both five-well designs were close approximations of the best observed location such that the I-efficiencies of the five-well G-design were 103% and 107%, respectively. Based on the “best” criterion, the GA-five-well I-design was included in the best set (Figure 42).

5.2.b. Large-Scale, Two-dimensional Test Case

After comparing the results of the small-scale test case, we develop a large-scale test case based on a real-world groundwater model. We take the finite-element mesh from a groundwater model constructed for an aquifer in the Oristano plain in west-central Sardinia, Italy (Cau et al., 2002; Siade et al., 2012) and apply synthetic parameter zones and boundary conditions to the model. The test case aquifer is 11 km x 9 km and the finite-element mesh has 29,197 nodes, 57,888 elements, seven parameter zones, 20 pumping wells, and is surrounded by constant head boundaries (Figure 44). This is a highly discretized, realistic groundwater model.
This is the same test case used by Ushijima and Yeh (2013) to test their robust experimental design algorithm. Following their methodology, we construct a reduced model that has a size $n_p$ of 103, and a dimension reduction of two orders of magnitude. After constructing the reduced model, we divide the aquifer into 22 observation/pumping zones and for practicality, overlay a course observation grid that has a discretization of 400m (Figure 45).
We define a feasible observation location as a location that falls within an observation zone (i.e., an observation/pumping zone without a pumping well) and on a node of the observation grid. After imposing these constraints, we find that there are 531 feasible observation locations. We then seek information on the hydraulic conductivity with the experimental design, so we set-up a robust experimental design following the methodology of Ushijima and Yeh (2015). Previous research by Sun and Yeh (2007) demonstrated that for a robust experimental design for hydraulic conductivity, it is sufficient to consider only the upper and lower bounds of the hydraulic conductivity, so we assume upper and lower bounds of $0.1 \text{m/day}$ and $20\text{m/day}$ for the hydraulic conductivity in all parameter zones and consider all possible combinations of zonal hydraulic conductivity. As with the first test case, we use both a GA and a PSO to search for the solution to the experimental design problem. In this case, we develop one- through twelve-well
designs for each of the optimality criterion listed Table 16. These two sets of designs are combined to form a best set whose locations are shown in Figure 46-Figure 48.

Figure 46. One-, two-, three-, and four-well designs
Figure 47. Five-, six-, seven-, and eight-well designs
Again, as with the first test case, absolute general statements are hard to make since these results are tied to this specific experimental design, but some interesting results are seen. Most importantly, the general trends we observe in the small-scale test case again appear in this realistically-scaled test case; for example, with regards to the spatial distribution of the different designs the A-designs tend to group the observation wells close to the area of the aquifer that has the highest concentration of pumping wells, while the other criteria tend to spread the observation wells throughout the aquifer. In terms of efficiencies (graphed in Figure 49-
53), the designs for the realistically-scaled test case also behave similarly to the designs for the small test case.

**Figure 49. A-design efficiencies**

**Figure 50. D-design efficiencies**
Figure 51. E-design efficiencies

Figure 52. G-design efficiencies
Figure 53. I-design efficiencies

As with the small test case, all the designs are fairly A-optimal but the A-designs are not non-criterion-optimal. Though less obvious than in the small-scale test case, the E- and D-designs are relatively optimal with respect to each other. In general, the prediction-designs (G- and I-designs) are more optimal with respect to each other than with respect to the parameter-designs (A-, D-, and E-designs), while similarly the parameter-designs are more optimal with respect to each other than with respect to the prediction-designs. One interesting feature in these results that, while present, is not as noticeable in the small-scale results is that for the smaller designs, there are many instances in which the D-, G-, and I-efficiencies are, or virtually are, equal to zero, indicating that these designs captured minimal information about these criteria. This behavior makes sense, since it could be difficult for a small number of observation wells to obtain sufficient information to characterize a large aquifer, particularly if they are not distributed to do so.
While analyzing the results, we again observe that the PSO tends to outperform (with respect to objective function score) the GA for the smaller designs (i.e., those with smaller observation well networks) while the GA outperforms the PSO for the larger designs. In addition, both the GA and PSO again fail to search-out all the best-possible locations. For the GA, the five-well I-design is outperformed (with respect to I-optimality) by the five-well G-design, and the two- and seven-well E-designs are outperformed by the G- and D-designs of the same size, respectively. Similarly, the PSO converges to a three-well G-design that has an I-efficiency greater than 100%.

5.3. Discussion and Conclusions

In this paper, we analyzed the A-, E-, D-, G-, and I-optimality criteria and demonstrated their applicability to a groundwater experimental design problem. The analysis compares the properties of the different optimality criteria, including what information each seeks, as well as challenges that could be faced in implementing the criteria in an experimental design problem. The largest challenge is the numerical instability encountered when computing the objective function for some of the criteria, specifically D-, G-, and I-optimality. To overcome this difficulty, we suggested techniques that could be used to stabilize the computation of these objective functions. Following the analysis of the objective criteria, we showed how heuristic searches coupled with POD-reduced models could be used to search the solution space of a groundwater experimental design problem for optimal designs (with respect to the various criteria). Finally, we employed the concept of optimal efficiency to compare the relative performance of the various “optimal” designs.
While, as noted before, the results are test case-specific, some general conclusions and observations can be drawn. One observation is that while it is generally accepted that a design that is optimal with respect to one optimality criterion will be fairly optimal with respect to other optimality criteria. However, the test case results demonstrate that this is not always the case. This assumption seems to be fairly valid when the optimality criteria are seeking the same type of information (i.e., information about model parameters or model predictions) but not when the optimality criteria are seeking different types of information. For our test cases, among the criteria seeking information about model parameters, we observe that the D- and E-designs are fairly optimal with respect to each other (i.e., had high efficiencies) and exhibit fairly high A-efficiencies; however the A-designs produce very low D- and E-efficiencies. These results make sense in light of the fact that A-designs seek the maximum amount of information, while D- and E-designs seek the highest quality information (uncorrelated or unique data, respectively). The maximum amount of information may be highly correlated or not unique, leading to low D- or E-efficiencies; however, high quality information should contain a large amount of information, leading to a high A-efficiency. The test case results show that the optimal designs for model prediction (G and I) do not have high model parameter efficiencies. This might be expected, as different model parameter values could produce similar predictions. However, it was surprising to see that optimal designs for model parameters (A, D, or E) did not necessarily produce high model prediction efficiencies, as one might expect a model with parameter values close to the true values would produce good model predictions.

In this study we developed a reduced model using POD and coupled the reduced model with GA and PSO to solve the combinatorial optimization problem for experimental design. As previous research on heurist searches shows, the downside of heuristic searches is that there is no way to
prove that a location in the solution space is the global optimum; rather, it shows just that it is more optimal than all visited locations. This leads to the risk that the search would converge to a sub-optimal solution, particularly when the solution space is difficult to search. In the context of experimental design, the solution spaces could have large areas whose score is very close to the computer’s round-off error threshold (particularly with the D- and E-optimality criteria), leading to difficulties for the algorithms in determining search directions. In addition, these types of solution spaces tend to have very spiky optimal regions, making it easy for the algorithm to bypass the global optimum. An optimality criterion such as G-optimality, which minimizes a maximum, may have a very discontinuous solution space that could lead to the algorithm getting stuck on one side or the other of a “cliff”. In terms of heuristic searches, the A-optimality criterion presents the most searchable solution space, as its formulation (a summation) lends itself to a relatively smooth and well-behaved solution space. In this study, we see evidence that both the GA and PSO got stuck in sub-optimal solutions, as many GA- and PSO-designs had non-criterion efficiencies greater than 100% – something that would be impossible had the searches converged to the global optima. While this is of course undesirable, many times the sub-optimal solution appeared to be a good estimate of the best-possible solution. For example, in the case shown in Figure 42, the I-optimality of the five-well G-design is 102%. This inspires confidence that in general, a GA or a PSO will find at least good approximate solutions to experimental design problems.

While the study presented here is by no means exhaustive, we can see that prior to the application of an experimental design problem it is imperative to have a clear understanding of what information is being sought so an appropriate optimality criterion may be chosen. In addition, the computation complexities faced by an experimental design problem must be
understood and steps must be taken to remedy them. If heuristic searches are used to overcome some of the computational complexities, their own computational complexities also must be dealt with. In addition, the choice of search must be made wisely, as not all searches perform equally well. Despite these challenges heuristic searches, particularly when coupled with reduced models, may be the only option to solve realistically-scaled experimental design problems. In the future, this study could be expanded to include other heuristic searches such as randomized PSOs (He et al., 2014) or Cuckoo searches (Yang & Suash Deb, 2009). Alternate design methods also could be explored, such as methods that identify critical design points within a groundwater model. Examples include the Greedy Algorithm proposed by Boyce et al. (2015), the discrete empirical interpolation method (DEIM) (Stanko, Boyce, & Yeh, 2016), or extension to a Bayesian formulation (Zhang, Zeng, Chen, Chen, & Wu, 2015).
5.4. Bibliography


http://doi.org/10.1109/NABIC.2009.5393690

http://doi.org/10.1029/WR022i002p00095

http://doi.org/10.1002/2014WR015740
Chapter 6

Dissertation Conclusion, Discussions, and Future Research

These studies demonstrate the applicability of utilizing POD reduced modeling in groundwater modeling. The Fast Algorithm extends the usefulness of POD reduced models to complex models that would normally be infeasible to reduce because of the cost associated with repeated re-projections of the reduced system matrix. The Fast Algorithm takes advantage of the general form of the full system matrices and the properties of matrix operations to create a series of normalized projections that can be scaled and summed in a manner that is much more computationally efficient than re-projecting the full system matrix onto the reduced space. This allows reduced system matrices to be formed online at will, without incurring the dominating cost of utilizing matrix multiplication. The algorithm is able to accurately construct reduced system matrices for finite-element or finite-difference models with zoned or arithmetically-averaged interpolated parameter fields. Additionally, it runs as a “black-box” add-on requiring only that the full model output a full system matrix and that the user also must indicate what norm is used to model parameter values across zone boundaries. Despite its advances, the algorithm does have shortcomings. If the norm is not provided or is different from the norm used in the full model, the algorithm is not guaranteed to be able to construct an exact reduced system matrix. The algorithm is also unable to correct for non-arithmetic norms when building reduced system matrices for interpolated parameter fields. In all these cases, the algorithm assumes an arithmetic mean. Though error introduced, if this assumption is not true, will be minor (generally, much smaller than the error introduced by model reduction) these shortcomings leave room for improvement. Potential improvements to this algorithm include deriving correction
factors for non-arithmetic-interpolated fields and developing a computationally inexpensive method to detect the full model norm.

These improvements would allow the Fast Algorithm to further improve POD model reduction and increase POD model reduction’s usefulness in situations requiring large numbers of model runs. One such situation arises when using heuristic searches for experimental design. Previously, the heuristic search and its associated large number of required model runs presented a serious obstacle. However, as the studies in this dissertation show, POD model reduction opens up new avenues of research that were previously computationally infeasible. The application of experimental design to gather information on system forcing and parameters is not novel, but the POD reduced models allow these experimental designs to be carried out in a much more efficient manner than was previously possible with similar studies. This decrease in computational cost allows much more complex and larger-scale experimental designs to be carried out. One immediate illustration of this is the comparative analysis from the final study in this dissertation. This high-level look at experimental design in groundwater would be incredibly difficult without POD model reduction as it requires completing multiple individual experimental designs, each of which requires their own heuristic search. High-level studies are important to be able to gain a better understanding of a method and can reveal, among other things, potential challenges, benefits, and best practices. For example, the comparative analysis showed that to achieve the desired results, one must put careful consideration toward the end goal before applying experimental design. The analysis also highlighted the fact that while useful, heuristic searches may be limited by computational precision and self-imposed limitations on search completeness (i.e., how much of a combinatorial search it is allowed to perform).
Notwithstanding the improvements that utilizing POD reduced models introduce, these experimental design studies would benefit from further development. As these studies suffered from common shortcomings of heuristic searches, one area of development would be in the use of improved search algorithms. This dissertation touched on this matter briefly by using both a standard GA and standard PSO while carrying out a comparative analysis of optimality criteria. This area should be expanded to include improved GAs and PSOs and other state-of-the-art heuristic algorithms. Another area of development would be in the construction of the designs themselves. Future research should further explore classic single objective designs and compare them to multi-objective designs. Bayesian approaches have also been shown to have promise (Zhang et al., 2015) and should be studied further. Alternatives to heuristic searches should also be explored, for example, using a Greedy Algorithm (Boyce et al., 2015) or capturing information from the Discrete Empirical Interpolation Method (Stanko et al., 2016).

Lastly, an important area of development would be to further decrease the computational cost of the heuristic search. The POD reduced models allow the execution of more complex studies requiring larger numbers of model calls but these studies still require limitations to place feasible bounds on the search space. Loosening or removing these limitations would improve the ability to search out or verify global optima. The Fast Algorithm assists in this area by reducing the cost of POD model reduction, thus allowing more computational time to be devoted to the search rather than model overhead. Another means of decreasing the cost to run the reduced model would be to increase the accuracy of the model reduction—in other words, producing a smaller but just as accurate reduced model. Potential approaches could include improving on the snapshot selection techniques possibly by incorporating Independent Components (ICs) rather than only Principle Components (PCs). The advantage of ICs is that they isolate the sources in a
signal (i.e., what is forcing variations in the snapshot set) rather than only uncorrelated information as do PCs. This could be used to inform the collection of a more optimal snapshot set. A major disadvantage of ICs is that thus far, there are no stable methods to calculate them as there are for PCs.

While there continues to be opportunities for further study, this dissertation presents research that advances the field of groundwater hydrology research. Taken as a whole, this dissertation progresses a methodology that has been demonstrated to be able to open up previously intractable areas of study. The Fast Algorithm improves the computational efficiency of POD reduced models and the experimental design studies demonstrate how POD reduced models may be used in conjunction with heuristic searches to design observation networks in order to gather information on system forcing and parameters. These experimental design studies serve to not only broaden the understanding of the application of experimental design to groundwater hydrology research, but also to present the concept of surrogate modeling, something that may be used in other areas of groundwater hydrology research.
7. Complete Bibliography


http://doi.org/10.1016/0309-1708(91)90040-U


He, Y., Xu, Q., Yang, S., & Liao, L. (2014). Reservoir flood control operation based on chaotic


http://doi.org/10.1093/imanum/dri044


