Reduced order parameter estimation using quasilinearization and quadratic programming

Adam J. Siade,[1] Mario Putti,[2] and William W.-G. Yeh[1]

Received 30 September 2011; revised 24 March 2012; accepted 18 April 2012; published 1 June 2012.

[1] The ability of a particular model to accurately predict how a system responds to forcing is predicated on various model parameters that must be appropriately identified. There are many algorithms whose purpose is to solve this inverse problem, which is often computationally intensive. In this study, we propose a new algorithm that significantly reduces the computational burden associated with parameter identification. The algorithm is an extension of the quasilinearization approach where the governing system of differential equations is linearized with respect to the parameters. The resulting inverse problem therefore becomes a linear regression or quadratic programming problem (QP) for minimizing the sum of squared residuals; the solution becomes an update on the parameter set. This process of linearization and regression is repeated until convergence takes place. This algorithm has not received much attention, as the QPs can become quite large, often infeasible for real-world systems. To alleviate this drawback, proper orthogonal decomposition is applied to reduce the size of the linearized model, thereby reducing the computational burden of solving each QP. In fact, this study shows that the snapshots need only be calculated once at the very beginning of the algorithm, after which no further calculations of the reduced-model subspace are required. The proposed algorithm therefore only requires one linearized full-model run per parameter at the first iteration followed by a series of reduced-order QPs. The method is applied to a groundwater model with about 30,000 computation nodes where as many as 15 zones of hydraulic conductivity are estimated.

Citation: Siade, A. J., M. Putti, and W. W.-G. Yeh (2012), Reduced order parameter estimation using quasilinearization and quadratic programming, *Water Resour. Res.*, *48*, W06502, doi:10.1029/2011WR011471.

1. Introduction

[2] Simulating the dynamics of real-world groundwater systems requires the use of accurate numerical models. Even though these models may be based on the underlying physical processes of a system, intrinsic model parameters must be identified in order for the model response to be sufficiently accurate. A multitude of algorithms exist whose purpose is to adjust the parameter values of a model such that the model output matches its associated measured values as closely as possible. This type of problem is commonly referred to as the inverse problem. *Yeh* [1986], *Sun* [1994], and *Oliver and Chen* [2011] provide comprehensive reviews on the inverse problem as it applies to groundwater hydrology. Currently, the most popular methods are based on the output error criterion, where a starting estimate of the parameter vector is updated such that the norm of the difference between observed states and their corresponding model-predicted values is minimized. *Cooley* [1985] provides a comparison of four different nonlinear regression methods of parameter identification; the most efficient methods were found to be the Marquardt [*Marquardt*, 1963] and quasilinearization [*Yeh and Tauxe*, 1971a, 1971b] methods. Some current, popular software include PEST [*Doherty*, 2002] and UCODE [*Poeter et al.*, 2005]. These software applications employ algorithms that are largely based on the Gauss-Marquardt-Levenberg methods [*Levenberg*, 1944; *Marquardt*, 1963].

[3] Methods based on the output error criterion require a significant number of model runs in order to evaluate parameter updates from one iteration to the next. Therefore, the computational demand associated with a forward run of the numerical model has a large impact on the overall CPU requirement of the parameter estimation algorithm. Reducing the computational demand associated with the numerical model can significantly reduce the computational demand of the parameter estimation algorithm. A method known as proper orthogonal decomposition (POD) has become very popular recently for achieving significant model-order reduction [*Cazemier et al.*, 1998; *Willcox and Peraire*, 2002; *Vermeulen et al.*, 2004; *McPhee and Yeh*, 2008; *Siade et al.*, 2010]. This method of model reduction essentially projects the original full-model solution from the space of functions where it resides into a subspace

[1]Department of Civil and Environmental Engineering, University of California, Los Angeles, California, USA.
[2]Department of Mathematical Methods and Models for Scientific Applications, University of Padua, Padua, Italy.

Corresponding author: W. W.-G Yeh, Department of Civil and Environmental Engineering, University of California, 405 Hilgard Ave., Los Angeles, CA 90095, USA. (williamy@seas.ucla.edu).

0043-1397/12/2011WR011471

(e.g., via Galerkin projection) generated from only a few model runs, such that the number of equations that need to be solved is greatly reduced. The resulting loss in accuracy remains small or controllable. This is achieved by developing a specific set of basis functions such that time-varying linear combinations of these basis functions can adequately approximate the original full-model solution for all times and with any forcing. Since the number of POD basis functions is much smaller than the number of computational nodes, the magnitude of model reduction can be very significant. *Siade et al.* [2010] reduced a basin-scale groundwater model, using POD, resulting in a reduced model that ran approximately 1000 times faster than its corresponding original full model with a negligible loss in accuracy.

[4] The accuracy of the reduced model via POD is dependent on the quality of the basis functions that span the reduced model subspace. *Siade et al.* [2010] provides a methodology for evaluating a good set of basis functions when considering changing values of forcing, e.g., groundwater extraction/injection rates. However, when one is changing the values of the parameters, such as hydraulic conductivity, these basis functions begin to lose accuracy. This loss of accuracy is due to the nonlinear relationship between model parameters and model states. This presents a problem for reduced-order parameter estimation, which requires iterative updates of the parameter values. *Park et al.* [1998] and *Vermeulen et al.* [2005] present methodologies for dealing with this issue. In both articles, the authors use the method of snapshots to develop the basis functions that span the reduced model space. Snapshots are samples of the original full-model state variable at specified instants in time. A snapshot set is collected for each well, individually, given a constant unit forcing and a specific set of parameter values. Snapshot sets are collected over a specific range of parameter values that adequately capture parameter variability around their current estimates. Throughout the parameter estimation algorithm, the current estimate of the parameters may "move" outside this range, requiring the re-evaluation of the reduced model using a new range of parameter values. However, many snapshot sets are needed in order to adequately capture all possible combinations of parameter ranges each time the reduced model is evaluated. For example, a snapshot set for each extraction/injection well is needed when one of the parameters is at the upper end of its range and the others are at their lower ends. Additional snapshot sets are required for each of these combinations at each extraction/injection well. In particular, in the case of one well and two parameters, four snapshot sets are needed; in the case of two wells and three parameters, 16 snapshot sets are needed, etc. Additionally, snapshot sets may be required for parameter values within their ranges rather than at the upper and lower bounds only. Each snapshot set requires an original full-model run. Therefore, for highly parameterized systems with a large number of extraction wells, the computational gain of the model reduction is overcome by the computational burden of developing snapshot sets.

[5] In this paper, we propose a methodology that no longer requires the development of a "moving" parameter range when developing snapshots. The reduced model must be developed once only; the resulting basis functions are accurate for the entire parameter estimation procedure. The parameters under investigation are zonal hydraulic conductivity values. The parameter estimation procedure employed is based on quasilinearization and quadratic programming. *Bellman and Kalaba* [1965] originally developed quasilinearization for parameter identification in a system of nonlinear ordinary differential equations. It involves solving a series of linearized initial value problems such that the sequence of solutions converges to the solution of the original nonlinear problem. *Yeh and Tauxe* [1971a, 1971b] applied quasilinearization to parameter estimation in groundwater modeling while *Park et al.* [1998] applied it to flow reactor modeling. *Yeh* [1975] combined quasilinearization and quadratic programming for parameter estimation in a partial differential equation. The algorithm essentially consisted of solving a series of sequential quadratic programming (QP) problems. However, in practice this algorithm suffers from the fact that each QP problem is so large that the computational burden of solving it is near the same magnitude as that of current Gauss-Newton type approaches. In this study, we show that POD model reduction can dramatically reduce the computational requirement of the individual QP problems, resulting in a drastic increase in overall inversion efficiency. The method requires the evaluation of one snapshot set for each hydraulic conductivity zone in order to build the reduced model. Snapshots are collected from the linearized full model (where changes in conductivity become the forcing term) rather than the original full model (where groundwater extraction/injection is the forcing term). The proposed method can handle highly parameterized systems with a large number of extraction/injection wells and still achieve significant reductions in CPU time.

2. Confined Aquifer Groundwater Flow Model

[6] The following partial differential equation (PDE) describes two-dimensional groundwater flow for a confined, anisotropic aquifer with pumping [*Bear*, 1979]:

$$\frac{\partial}{\partial x}\left(bK_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(bK_y\frac{\partial h}{\partial y}\right)-q-bS_s\frac{\partial h}{\partial t}=0, \qquad (1)$$

with initial and boundary conditions:

$$h(x,y,0)=h_0(x,y);$$
$$h(x,y,t)=h_d(x,y,t), \qquad (x,y,t)\in(\Gamma_1);$$
$$\left(bK_x\frac{\partial h}{\partial x}n_x+bK_y\frac{\partial h}{\partial y}n_y\right)(x,y,t)=q_n(x,y,t), \qquad (x,y,t)\in(\Gamma_2),$$

where h is the hydraulic head (L); $K_x(x,y), K_y(x,y)$ are spatially varying hydraulic conductivities in the x and y directions, respectively (L/T); S_s is the specific storage (L^{-1}); $b(x,y)$ is the thickness of the aquifer (L); q is the specific volumetric pumping rate (LT^{-1}); Γ_2 is the flux boundary; Γ_1 is the fixed head boundary; and h_0, h_d, and q_n are known functions. For simplicity and without loss of generality, we assume isotropic behavior of the aquifer, i.e., $K_x = K_y = k(x,y)$.

[7] The application of the superposition principle to equation (1) followed by spatial discretization of the resulting PDE (e.g., by finite differences, finite elements, etc.)

yields a system of linear ordinary differential equations (ODEs) for the drawdown, $\mathbf{s}$:

$$\mathbf{B}\frac{d\mathbf{s}}{dt} = \mathbf{A}\mathbf{s} + \mathbf{q} = \mathbf{f}(\mathbf{s}, \mathbf{k}), \tag{2}$$

where $\mathbf{A}$ is the $n \times n$ stiffness matrix; $\mathbf{B}$ is the $n \times n$ mass matrix; $\mathbf{s}$ and $\mathbf{q}$ are the n-dimensional vectors of nodal (cell) drawdowns and source/sinks, respectively; $\mathbf{f}(\mathbf{s}, \mathbf{k}) = \mathbf{A}\mathbf{s} + \mathbf{q}$ is a vector-valued function depending on drawdown $\mathbf{s}$ and hydraulic conductivity $\mathbf{k}$; and n is the number (generally very large) of spatial computational nodes (cells). The n_z-dimensional vector $\mathbf{k}$ represents the spatially varying hydraulic conductivity, which is assumed to be discretized into n_z material zones. Drawdown is the difference between the initial head and the head after pumping, i.e., $s = H - h$, where H is the initial head (e.g., steady state or the natural system dynamics). In the majority of cases of practical interest, matrices $\mathbf{A}$ and $\mathbf{B}$ are large, sparse, symmetric, and positive definite. Upon the application of an implicit Euler scheme, equation (2) can be approximated as

$$\left(\mathbf{A} - \frac{1}{\Delta t_j}\mathbf{B}\right)\mathbf{s}_j = -\frac{1}{\Delta t_j}\mathbf{B}\mathbf{s}_{j-1} - \mathbf{q}_j, \tag{3}$$

where $\mathbf{s}_j$ and $\mathbf{q}_j$ are vectors of nodal drawdown and extraction rate values at time j, respectively; and Δt_j is the length of the j-th time step.

3. Groundwater Parameter Estimation Via Quasilinearization and Quadratic Programming

3.1. Problem Formulation

[8] The primary goal of parameter estimation is to identify the parameter vector that minimizes some norm of the residuals, i.e., differences between the model-predicted state variable(s) and those observed in the field. The most commonly used objective for this class of problems is minimizing the sum of the squared residuals. The parameter vector of interest for this study consists of the zonal hydraulic conductivity values, $\mathbf{k}$. The state vector of interest consists of the nodal (cell) drawdown values, $\mathbf{s}$. Note that generally, matrix $\mathbf{A}$ in equations (2) and (3) is the only term that explicitly contains the vector $\mathbf{k}$. Therefore, the general problem statement can be written as

$$\begin{aligned} &\min_{\mathbf{k}} \sum_{j=1}^{n_t}\sum_{i=1}^{n_j}\left(s_{\Re_j(i),j} - s^*_{i,j}\right)^2, \\ &\text{subject to :} \\ &\left(\mathbf{A} - \frac{1}{\Delta t_j}\mathbf{B}\right)\mathbf{s}_j = -\frac{1}{\Delta t_j}\mathbf{B}\mathbf{s}_{j-1} - \mathbf{q}_j \qquad j = 1, \ldots, n_t \\ &\mathbf{k}_{\min} \le \mathbf{k} \le \mathbf{k}_{\max}, \end{aligned} \tag{4}$$

where n_t is the number of time steps; n_j is the number of observation locations at time step j; $s_{\Re_j(i),j}$ and $s^*_{i,j}$ are the drawdown values, for measurement location i and time j, predicted by the model and observed in the field, respectively; and $\Re_j(i)$ maps the position of the appropriate computational node (cell) to its corresponding i-th observation at time j.

3.2. Quasilinearization and Quadratic Programming

[9] The method of quasilinearization and quadratic programming, as it applies to parameter estimation, consists essentially of solving a series of quadratic programming (QP) subproblems such that the solution to these problems converges to the solution of the original nonlinear inverse problem. *Yeh* [1975] provides a methodology in which the governing equation is linearized about the current estimate of the parameter vector and the state vector using a Taylor series expansion. This linearized equation replaces the original governing equation in the least squares parameter estimation problem resulting in a QP problem. The solution to this QP problem then becomes the current estimate for a new Taylor series expansion resulting in a new QP. This process is repeated in the linearized system until the solution converges to the solution of the original nonlinear least squares problem.

[10] Consider the governing equations after spatial discretization and before temporal discretization, i.e., the system of ordinary differential equations (ODEs) in equation (2). Applying a Taylor series expansion about some current estimate of the drawdown, $\mathbf{s}^m$, and hydraulic conductivity, $\mathbf{k}^m$, retaining up to the first order terms only, results in the following:

$$\begin{aligned} \mathbf{B}\frac{d\mathbf{s}^{m+1}}{dt} &= \mathbf{f}(\mathbf{s}^m, \mathbf{k}^m) + \nabla_{\mathbf{s}}\mathbf{f}(\mathbf{s}^m, \mathbf{k}^m)(\mathbf{s}^{m+1} - \mathbf{s}^m) \\ &\quad + \nabla_{\mathbf{k}}\mathbf{f}(\mathbf{s}^m, \mathbf{k}^m)(\mathbf{k}^{m+1} - \mathbf{k}^m), \end{aligned} \tag{5}$$

where $\mathbf{s}^{m+1}$ is an approximation of the drawdown, given some new parameter vector, $\mathbf{k}^{m+1}$. The Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{s}$ is

$$\nabla_{\mathbf{s}}\mathbf{f}(\mathbf{s}^m, \mathbf{k}^m) = \mathbf{A}^m,$$

where $\mathbf{A}^m$ is the $\mathbf{A}$ matrix in equation (2) composed of the current estimate of hydraulic conductivity, $\mathbf{k}^m$. The Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{k}$ can be approximated numerically via finite difference as

$$[\nabla_{\mathbf{k}}\mathbf{f}^m]_{i,j} = \frac{df_i^m}{dk_j} \approx \frac{f_i(k_j^m + \Delta k_j) - f_i(k_j^m)}{\Delta k_j} = [\mathbf{D}^m]_{i,j},$$

where Δk_j is some relatively small increment of hydraulic conductivity for element j of $\mathbf{k}^m$, and $\mathbf{D}^m \approx \nabla_{\mathbf{k}}\mathbf{f}^m$ contains the current estimate of $\mathbf{s}^m$. Substituting these Jacobian matrices into equation (5) results in the following equation:

$$\mathbf{B}\frac{d\mathbf{s}^{m+1}}{dt} = \mathbf{A}^m\mathbf{s}^m + \mathbf{q} + \mathbf{A}^m(\mathbf{s}^{m+1} - \mathbf{s}^m) + \mathbf{D}^m(\mathbf{k}^{m+1} - \mathbf{k}^m). \tag{6}$$

[11] This equation can be rewritten such that it has the same form as the governing equation for groundwater flow (equation (2)), with an additional forcing term associated with changes in hydraulic conductivity:

$$\mathbf{B}\frac{d\mathbf{s}^{m+1}}{dt} = \mathbf{A}^m\mathbf{s}^{m+1} + \mathbf{D}^m(\mathbf{k}^{m+1} - \mathbf{k}^m) + \mathbf{q}. \tag{7}$$

[12] At the m-th iteration, the implicit Euler scheme can be used to approximate the solution to equation (7) in time as follows:

$$\left(-\mathbf{A}^m + \frac{1}{\Delta t_j}\mathbf{B}\right)\mathbf{s}_j^{m+1} = \frac{1}{\Delta t_j}\mathbf{B}\mathbf{s}_{j-1}^{m+1} + \mathbf{D}_j^m(\mathbf{k}^{m+1} - \mathbf{k}^m) + \mathbf{q}_j. \quad (8)$$

[13] Note that $\mathbf{D}^m$ is a function of time because it contains the current estimate $\mathbf{s}^m$, which is a function of time. The QP subproblem now can be solved using the linearized equations of the governing ODEs (equations (7) and (8)), yielding the following algorithm:

Algorithm QQP:
Given the initial feasible estimates $\mathbf{k}^0$ and $\mathbf{s}^0$:
For $m = 0, 1, \ldots,$ until convergence, do:
find $\mathbf{k}^{m+1}$ such that:

$$\min_{\mathbf{k}^{m+1}} \sum_{j=1}^{n_t}\sum_{i=1}^{n_j}\left(s^{m+1}_{\hat{\mathbf{R}}_j(i),j} - s^*_{i,j}\right)^2$$

subject to:

$$\left(-\mathbf{A}^m + \frac{1}{\Delta t_j}\mathbf{B}\right)\mathbf{s}_j^{m+1} = +\frac{1}{\Delta t_j}\mathbf{B}\mathbf{s}_{j-1}^{m+1} + \mathbf{D}_j^m(\mathbf{k}^{m+1} - \mathbf{k}^m) + \mathbf{q}_j$$

$$j = 1, \ldots, n_t$$

$$\mathbf{k}_{\min} \le \mathbf{k}^{m+1} \le \mathbf{k}_{\max}. \quad (9)$$

[14] We can consider convergence achieved if the change in conductivity is small, i.e., $|\mathbf{k}^{m+1} - \mathbf{k}^m| < \tau$, where τ is a predefined tolerance, or if an insignificant change in the objective of equation (9) is observed. For convex programming problems, this process will converge to the global optimum of the original nonlinear inverse problem (equation (4)). However, it is important to note that, in general, this algorithm only guarantees convergence to a local optimum for nonconvex problems [*Bellman and Kalaba*, 1965].

[15] The algorithm presented thus far has not been used widely in practice due to the difficulty of solving each successive QP problem for real-world large-scale models. Depending on the QP algorithm employed, the successive QP problems (equation (9)) can be highly computationally demanding and become impractical or even infeasible. Therefore, Gauss-Marquardt-Levenberg methods (e.g., PEST [*Doherty*, 2002] and UCODE [*Poeter et al.*, 2005]) have become more popular for solving inverse problems in groundwater flow. However, the linearization of the governing equations does, in fact, result in a linear model whose order (i.e., number of equations) can be reduced significantly with the application of modern POD technology. With a much smaller set of equations in the constraint set, the optimization problem (equation (9)) can be solved very efficiently resulting in a more efficient and tractable overall estimation procedure.

3.3. Model Reduction Via Proper Orthogonal Decomposition (POD)

[16] In order to apply POD as accurately as possible to the linearized equations (equation (6)), the natural system dynamics must be removed [*Siade et al.*, 2010; *Vermeulen et al.*, 2004]. In other words, the model must be one in which the state variable remains at rest, i.e., zero everywhere, unless some forcing is applied. This is naturally true for models of drawdown where the superposition principle is used to remove the natural system dynamics from the governing equations of groundwater flow. However, in the case of parameter estimation, we are not interested in optimizing pumping rates. Here we are interested in the manner in which drawdown changes given a change in the hydraulic conductivity distribution. Accordingly, we must develop a linearized model that relates changes in drawdown, δ_s, with changes in hydraulic conductivity, δ_k. This model must remain at rest, i.e., $\delta_s = \mathbf{0}$, when there is no forcing, i.e., $\delta_k = \mathbf{0}$. We obtain such a model by rewriting equation (6), using the superposition principle, to yield

$$\mathbf{B}\frac{d\delta_s}{dt} = \mathbf{A}^m\delta_s + \mathbf{D}^m\delta_k, \quad (10)$$

where $\delta_s = \mathbf{s}^{m+1} - \mathbf{s}^m$ and $\delta_k = \mathbf{k}^{m+1} - \mathbf{k}^m$. This equation has the same general form as equation (2); however, the state variable is now δ_s and the forcing term is now $\mathbf{D}^m\delta_k$. Hence, the linearized model relates changes in drawdown (δ_s) to perturbations in zonal hydraulic conductivity values (δ_k).

[17] We use POD to approximate the linearized full-model solution (equation (10)) via a reduced model developed by means of the method of snapshots. A snapshot vector, $\overline{\delta}_s$, is the linearized full-model solution for all spatial nodes at some instant in time. We can approximate the state vector in space and time as a linear combination of linearized full-model snapshots:

$$\delta_s(t) \approx \hat{\delta}_s(t) = \sum_{i=1}^{n_s}\overline{\delta}_{s_i}w_i(t),$$

where $w_i(t)$ are some unknown weighting functions and n_s is the number of snapshots considered. This approximation can be written more generally in matrix form as

$$\hat{\delta}_s(t) = \sum_{i=1}^{n_s}\mathbf{u}_i\delta_{s_{r_i}}(t) = \mathbf{U}\delta_{s_r}(t),$$

where $\mathbf{U}$ is the $n \times n_s$ matrix of spatial basis functions (which can differ from the snapshot vectors) that span the reduced model space and δ_{s_r} is the vector of weighting functions which is considered the state vector in the reduced model space. The optimal matrix, $\mathbf{U}$, given some snapshot set can be determined using principal component analysis (PCA). To achieve this aim, the following eigenvalue problem is solved [*Siade et al.*, 2010; *McPhee and Yeh*, 2008; *Vermeulen et al.*, 2004]:

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\Lambda\mathbf{U}^T,$$

where matrix $\mathbf{X}$ is formed by the normalized snapshot vectors as columns; matrix $\mathbf{U}$ contains the principal vectors (eigenvectors) as columns; and Λ is a diagonal matrix

whose diagonal elements consist of the eigenvalues of each principal vector. The spectral decomposition of $\mathbf{X}\mathbf{X}^T$ (an $n \times n$ matrix) is obtained by calculating the eigenpairs of $\mathbf{X}^T\mathbf{X}$ (an $n_s \times n_s$ matrix), which are related to the eigenpairs of $\mathbf{X}\mathbf{X}^T$. Each eigenvalue represents the "amount" of variability captured by the corresponding principal vector (i.e., eigenvector). Using this quantitative measure, insignificant principal vectors can be identified and discarded. First, the normalized eigenvalues are obtained from

$$\phi_i = \frac{\lambda_i}{\sum_{j=1}^{n_s} \lambda_j}.$$

[18] These eigenvalues are organized in decreasing order and summed until $\sum_{i=1}^{n_p} \phi_i \geq \Phi$, where Φ is user-specified [*Siade et al.*, 2010; *McPhee and Yeh*, 2008; *Vermeulen et al.*, 2004]. The choice of Φ depends on the desired level of accuracy of the reduced model; larger Φ's result in larger but very accurate reduced models (more principal vectors) and smaller Φ's result in smaller but less accurate reduced models (less principal vectors). The principal vectors corresponding to the top n_p normalized eigenvalues are retained for building the reduced model and the remaining $n_s - n_p$ principal vectors are discarded. As a result the original $n \times n_s$ matrix, $\mathbf{U}$, is replaced by a smaller $n \times n_p$ matrix, $\mathbf{P}$, containing these n_p eigenvectors.

[19] The Galerkin projection of the linearized full-model equations onto the subspace spanned by the n_p eigenvectors yields the system of n_p, linearly independent, ODEs and n_p unknowns formulated as

$$\mathbf{P}^T\mathbf{B}\mathbf{P}\frac{d\delta_{s_r}}{dt} = \mathbf{P}^T\mathbf{A}^m\mathbf{P}\delta_{s_r} + \mathbf{P}^T\mathbf{D}^m\delta_k.$$

[20] Letting $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{B}\mathbf{P}, \tilde{\mathbf{A}}^m = \mathbf{P}^T\mathbf{A}^m\mathbf{P}$ and $\tilde{\mathbf{D}}^m = \mathbf{P}^T\mathbf{D}^m$, we obtain:

$$\tilde{\mathbf{B}}\frac{d\delta_{s_r}}{dt} = \tilde{\mathbf{A}}^m\delta_{s_r} + \tilde{\mathbf{D}}^m\delta_k.$$

[21] This reduced system of ODEs can be solved by any stable time-stepping technique, such as implicit Euler:

$$\left(-\tilde{\mathbf{A}}^m + \frac{1}{\Delta t}\tilde{\mathbf{B}}\right)\delta_{s_r,j} = \frac{1}{\Delta t}\tilde{\mathbf{B}}\delta_{s_r,j-1} + \tilde{\mathbf{D}}_j^m\delta_k. \tag{11}$$

[22] However, because of equation (11)'s drastically reduced size ($n_p \ll n$), often by several orders of magnitude, the system also can be solved very efficiently by analytical methods via matrix exponential [*Bellman*, 1960; *Willis*, 1979].

3.4. Reduced Order Quadratic Programming Formulation

[23] The reduced linearized model now can be used to solve each successive quadratic programming problem. Noting that $s_{i,j}^{m+1} = s_{i,j}^m + \mathbf{p}_i^T\delta_{s_r,j}$, substituting equation (11) into equation (9) results in the following QP problem:

$$\min_{\mathbf{k}^{m+1}} \sum_{j=1}^{n_t}\sum_{i=1}^{n_j}\left(\mathbf{p}_{\mathfrak{K}_j(i)}^T\delta_{s_r,j} + s_{\mathfrak{K}_j(i),j}^m - s_{i,j}^*\right)^2$$

subject to :

$$\left(-\tilde{\mathbf{A}}m + \frac{1}{\Delta t_j}\tilde{\mathbf{B}}\right)\delta_{s_r,j} = \frac{1}{\Delta t}\tilde{\mathbf{B}}\delta_{s_r,j-1} + \tilde{\mathbf{D}}_j^m\delta_k \tag{12}$$

$$j = 1, \ldots, n_t$$

$$\delta_k = \mathbf{k}^{m+1} - \mathbf{k}^m$$

$$\mathbf{k}_{\min} \leq \mathbf{k}^{m+1} \leq \mathbf{k}_{\max},$$

where $\mathbf{p}_{\mathfrak{K}_j(i)}$ is the $\mathfrak{K}_j(i)^{\text{th}}$ row of the $\mathbf{P}$ matrix. The constraint set has therefore been reduced from $(n * n_t)$ to $(n_p * n_t)$ equations (not including the upper and lower bounds on parameters), where $n_p \ll n$. It is important to note that from one QP problem to the next (i.e., between outer iterations of the overall algorithm) the principal vectors must be re-evaluated in order for the reduced model to achieve the greatest accuracy. This is a result of changing values of hydraulic conductivity between successive QPs (i.e., $\mathbf{A}$ is a function of $\mathbf{k}$). The reduced basis (i.e., principal vectors) requires one linearized full-model call per parameter to evaluate, which is exactly the same number of original full-model calls required by the Gauss-Levenberg-Marquardt method (e.g., PEST and UCODE) to evaluate the corresponding Jacobian matrix. However, as we show in this study, if the principal vectors are evaluated using a reasonable initial guess, subsequent inaccuracies produced by the reduced models throughout the algorithm become negligible. It is also important to note that the limitation on the number of parameters to be estimated is no different from that of a quadratic programming problem, which can be quite large.

4. One-Dimensional Test Case

[24] In this section, we apply the method proposed in this study to a one-dimensional test problem to illustrate the algorithm mechanics. This test problem is similar to that explored in *Siade et al.* [2010] and is displayed in Figure 1. The model has 101, equally spaced, computational nodes and has Dirichlet boundary conditions at nodes 1 and 101 of 0 m. There is a pumping well at node 51, with a constant extraction rate of 1 $m^3\ d^{-1}$. There are two equally sized zones of hydraulic conductivity (5.0 and 15.0 m d^{-1}, respectively) and a specific storage of $10^{-5}\ m^{-1}$ is used throughout.

[25] First we investigated the performance of POD model reduction as applied to the linearized model (equation (10)). Figure 2 illustrates the solution to the original full model along with the results of both the linearized full model and linearized reduced model given a change of hydraulic conductivity in each zone of 1.0 m d^{-1} (the initial values are those displayed in Figure 1). For this test case, the errors between the linearized full model and the linearized reduced model are on the same order of magnitude as those shown for the one-dimensional original full model by *Siade et al.* [2010]. We, therefore, conclude that the solutions obtained from the linearized full model and the linearized reduced model are essentially identical.

[26] The parameter estimation algorithm we present was applied to this one-dimensional test case. The algorithm

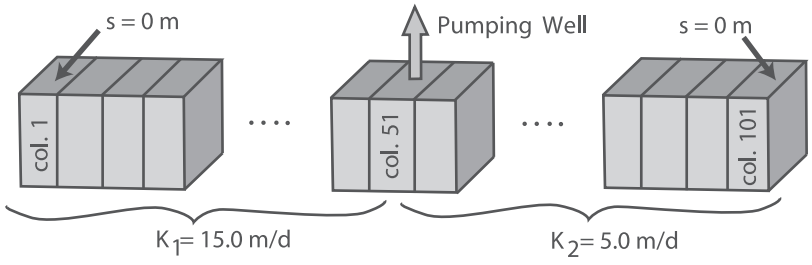

Figure 1. One-dimensional groundwater flow model.

was tested under three scenarios: using the linearized full model only, using POD model reduction of the linearized model where the reduced basis is updated at each iteration of the quasilinearization procedure, and using POD model reduction for the linearized model without updating the reduced basis, i.e., the reduced basis is determined using the initial values or the initial "guess" of hydraulic conductivity and never updated. A flowchart of the overall algorithm employed in this study is shown in Figure 3. We generated observations of drawdown using the true parameter values of $k_1 = 15.0$ m d^{-1} and $k_2 = 5.0$ m d^{-1} and recorded at each time step at nine locations ($x = 20, 30, 40, 45, 51, 56, 61, 71, 81$ m). The observations were also corrupted with normally distributed random noise to test the effects of measurement noise. Ten snapshots were selected optimally using the exponential function presented by *Siade et al.* [2010]. Figure 4 illustrates the convergence results for these three scenarios using an initial guess of $k_1 = 0.1$ m d^{-1} and $k_2 = 0.1$ m d^{-1}. The rate of convergence for the three scenarios was nearly identical until the least squares objective had fallen below 10^{-6}. After this point, the algorithm using scenarios 1 and 2 proceeded to converge superlinearly, whereas the algorithm employing scenario 3 appeared to converge linearly. Significant differences in convergence rates between scenarios 1 and 2 were not observed until the least squares objective was very small. However, if a different initial estimate of hydraulic conductivity were used, the relative rates of convergence for the three scenarios could be quite different; "good" initial guesses could result in very little difference in convergence between the three scenarios. In the case where measurement noise was added, there were negligible differences between the convergence rates of the three scenarios. This is due to the fact that errors associated with measurement noise dominated those associated with model order reduction via POD.

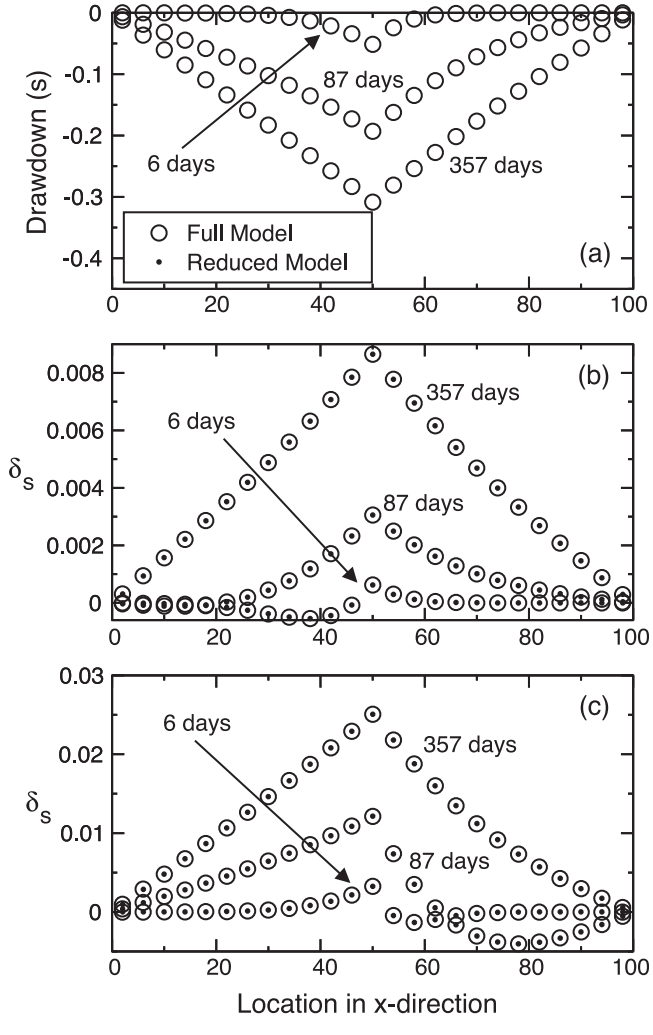

Figure 2. One-dimensional test model results for (a) the full, original flow model, (b) the linearized full and reduced models with k_1 increased by 1.0 m d^{-1}, and (c) the linearized full and reduced models with k_2 increased by 1.0 m d^{-1}.

[27] Aside from convergence rates, it is also important to evaluate whether or not the algorithm actually converges (oscillatory behavior is possible for scenario 3) or if it converges to the *global* optimum. Table 1 lists convergence results for a series of initial estimates of hydraulic conductivity (without measurement noise); the algorithm was considered to have converged when the least squares objective had fallen below 10^{-16} or diverged when more than 80 iterations were realized without convergence. The upper and lower bounds for each parameter were set to 10^3 m d^{-1} and 10^{-8} m d^{-1}, respectively. Although scenario 3 has some potential for oscillation, the initial guess must be very poor for nonconvergence to occur. In all cases tested, including those listed in Table 1, the global optimum was achieved; however, this cannot be guaranteed in the presence of insensitive and/or correlated parameters.

5. Two-Dimensional Application: Oristano, Italy

[28] The algorithm presented in this study was used to solve the inverse problem for a two-dimensional representation of a groundwater flow model in Oristano, Italy. The plain of Oristano is located in west-central Sardinia. The morphology of the territory is predominantly flat surrounded by the Monti Ferru and the Monti Arci hills on the

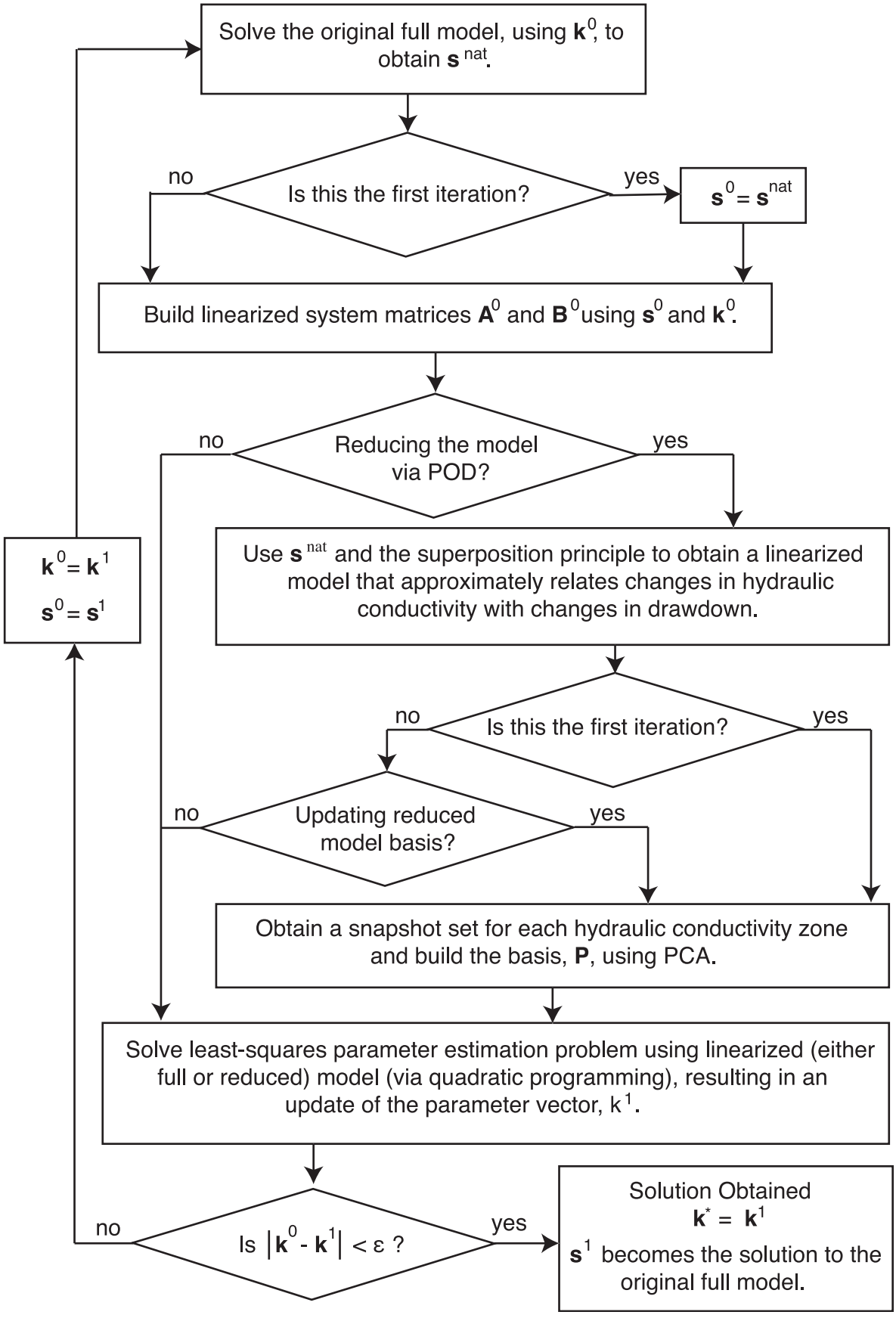

Figure 3. Flowchart of the algorithm presented in this study for all three scenarios.

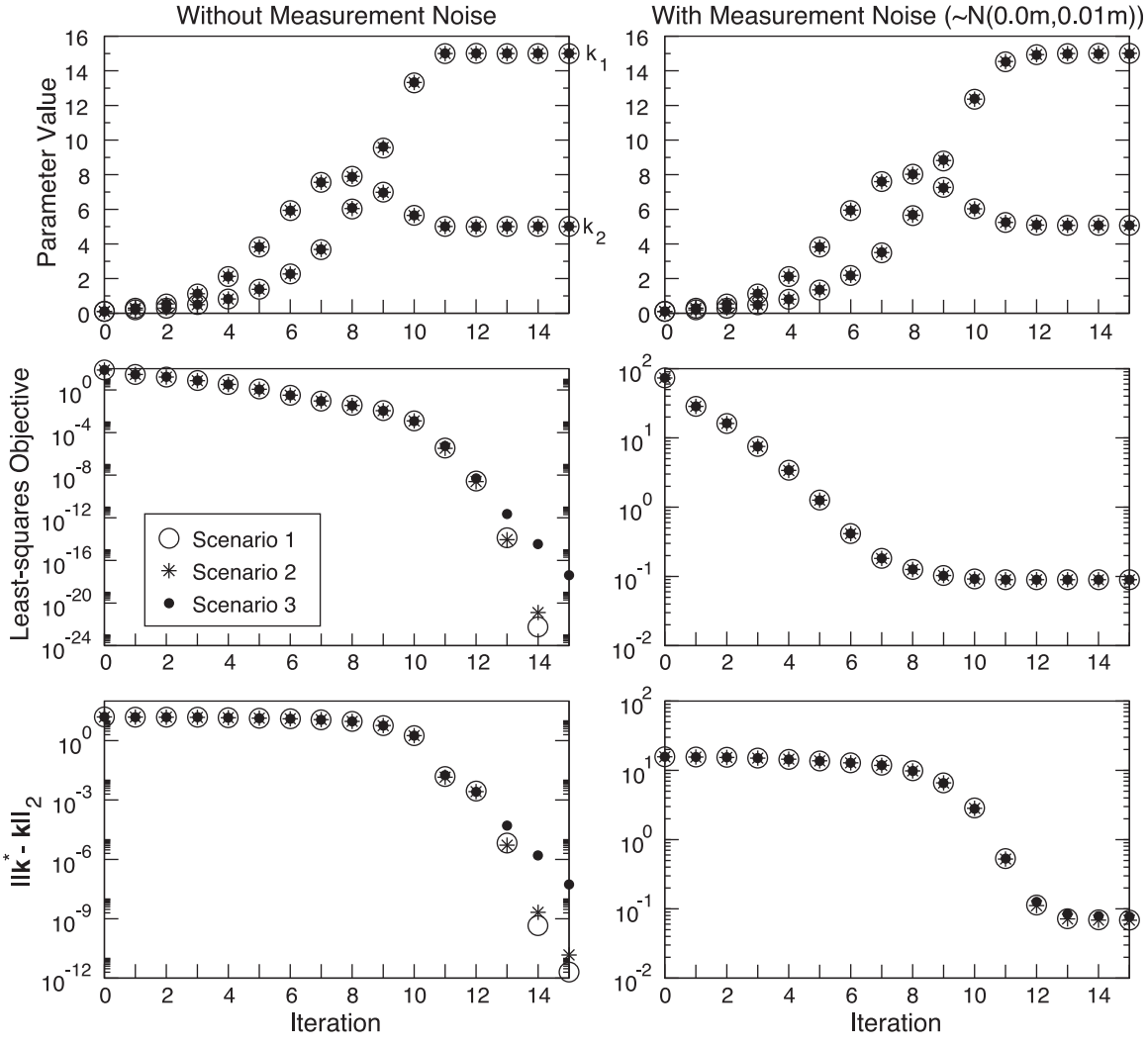

Figure 4. Convergence results for the one-dimensional test case, with and without measurement noise, using an initial estimate for hydraulic conductivity of $k_1 = 0.1$ and $k_2 = 0.1$ (note that the vertical scales may differ).

east and by the sea on the west. A heavily exploited multiaquifer system provides the source of water for agricultural and industrial uses. Approximately 25,000 wells are estimated to exist in the study region, although the number of wells that are actively withdrawing water from the aquifer system for agricultural and industrial uses is not known. Of these, 500 wells (some located also in the confined aquifer) have been monitored during recent years. The thickness of the multiaquifer system ranges from a minimum of ~28 m close to the hills and a maximum of ~218.5 m close to the sea. Three major units can be identified: a phreatic aquifer with an average thickness of ~12 m, a confining layer with an average thickness of ~4 m, and a confined aquifer with an average thickness of ~110 m [*Cau et al.*, 2002]. In this study, we concentrate on the confined aquifer, where most of the groundwater withdrawals take place. Since we are mainly interested in testing the algorithm proposed in this study for models with a large, realistic number of computational nodes, synthetic, simplified data was assumed for pumping, recharge, and boundary conditions.

[29] The numerical model for groundwater flow is discretized using the finite element method. The model contains 29,197 nodes and 57,888 elements and contains local grid refinement in 11 regions (Figure 5). For the purposes of this study, the model forcing was simplified such that there are six extraction well clusters (with a constant rate of 5000 m^3 d^{-1} each) and the entire outer boundary of the

Table 1. Convergence Statistics for the One-Dimensional Test Case Using Various Initial Estimates of the Parameters

Initial Estimate		Converge Scenario?			Iterations Required for Scenario		
k_1	k_2	1	2	3	1	2	3
0.1	0.1	Yes	Yes	Yes	14	14	15
10	10	Yes	Yes	Yes	6	6	6
17	17	Yes	Yes	Yes	12	12	12
100	100	Yes	Yes	Yes	18	18	16
0.15	50	Yes	Yes	No	16	16	–
50	0.15	Yes	Yes	Yes	9	9	17

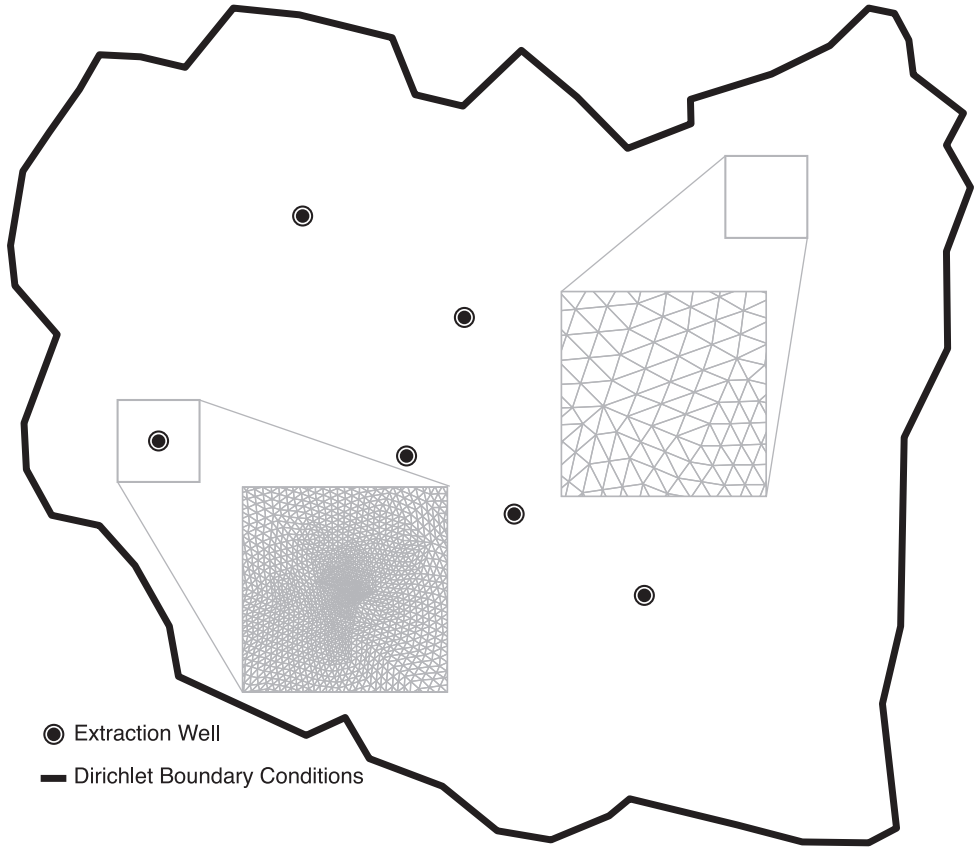

Figure 5. The model grid used as a two-dimensional "slice" of the Oristano, Italy model.

model is represented with Dirichlet boundary conditions (Figure 5). Specific storage is assumed constant throughout the model at 10^{-5} m^{-1}. Three different zonation patterns containing three, seven, and 15 zones, respectively, were considered for hydraulic conductivity (Figure 6). The "true" values of hydraulic conductivity (i.e., those used to generate the observations) are listed in Table 2 for each of the three zonation patterns. The initial values of each zone

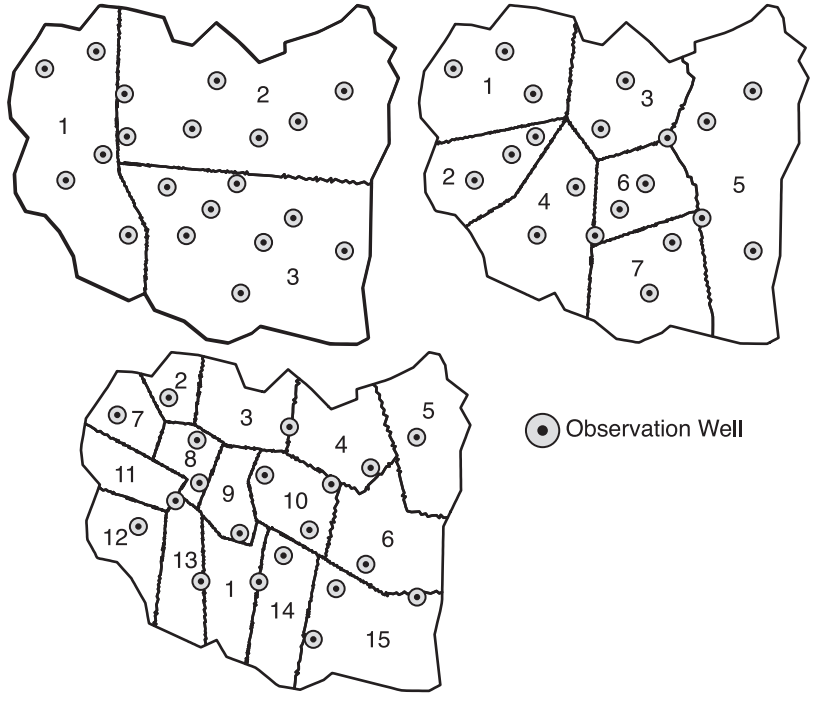

Figure 6. Zonation patterns used for the Oristano, Italy model.

Table 2. True Values of Hydraulic Conductivity by Zone for the Three Zonation Patterns Considered (m d^{-1})

	Zonation Pattern		
Zone	Three Zone	Seven Zone	15 Zone
1	15.0	15.0	15.0
2	5.0	5.0	5.0
3	7.0	7.0	7.0
4		12.0	12.0
5		3.0	3.0
6		20.0	20.0
7		10.0	10.0
8			2.0
9			9.0
10			18.0
11			21.5
12			4.3
13			0.5
14			16.1
15			1.0

were set to 1.0 m d^{-1}. The same three algorithm scenarios used in the one-dimensional test case also were employed in the Oristano model. Using the true parameter values, drawdown observations were generated at the locations shown in Figure 6 for each time step.

[30] Figure 7 shows the comparison between the linearized full model and linearized reduced model for the three-zone hydraulic conductivity distribution (Figure 6). The differences between the linearized full and linearized reduced model are indistinguishable and differences at the locations of the wells (maximum drawdown) are usually <0.0001 m. The impact of changing the hydraulic conductivity value of a zone is largely dependent on whether or not significant forcing resides in that zone. For example, in Figure 7, changing the hydraulic conductivity in zone 1 resulted in large changes in drawdown at the locations of the wells within zone 1; the same is true for zones 2 and 3.

[31] The three scenarios listed in section 4 were conducted for the Oristano model using the procedure outlined

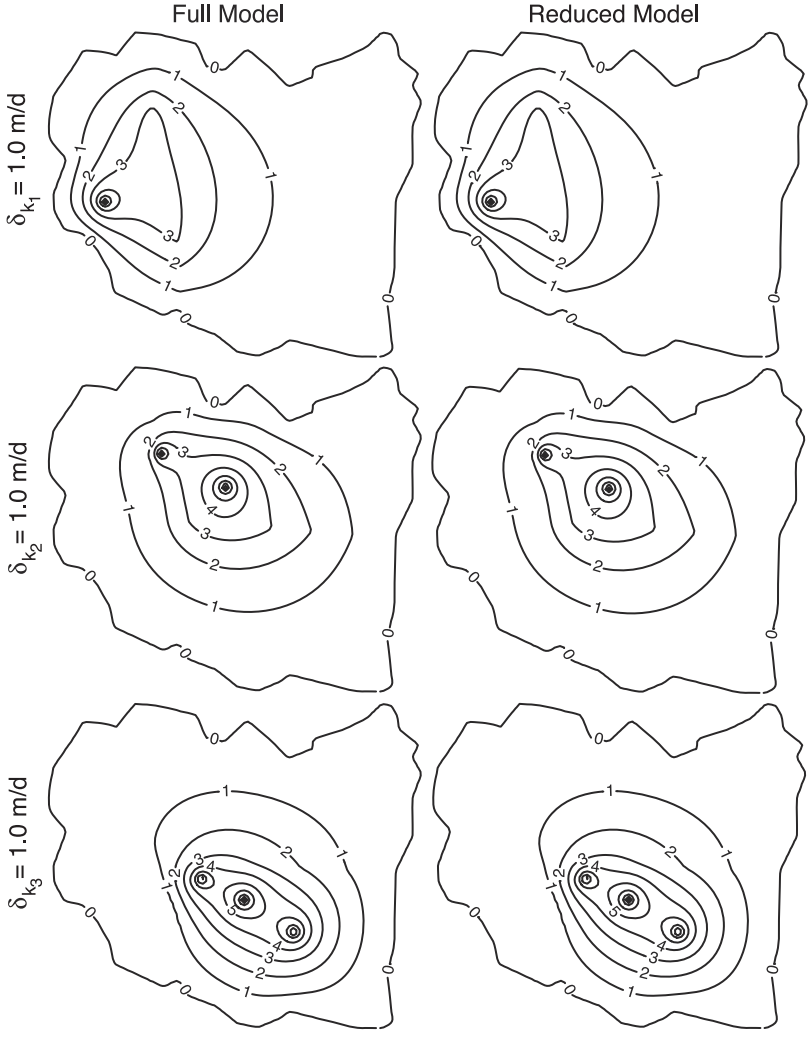

Figure 7. Changes in drawdown (m) given a unit change in hydraulic conductivity (1.0 m d^{-1}) for both the linearized full and reduced models.

in Figure 3. The effects of measurement noise also were explored; random Gaussian noise ($\sim N[0.0\text{ m}, 0.1\text{ m}]$) was added to the observation data. Figure 8 shows the convergence of the three-zone model with and without measurement noise. Similar to the convergence results of the one-dimensional test case, all three scenarios converge similarly (without noise) until the objective fell below 10^{-6}, where scenarios 2 and 3 began to deviate from scenario 1. However, with measurement noise, the least squares objective cannot fall below ~20.0; therefore, there was no significant difference in convergence between the three scenarios. This is more realistic because, in practice, there always will be noise present in measured data.

[32] The convergence results of the seven- and 15-zone models are shown in Figure 9. For the seven-zone case, the convergence of the three scenarios is similar to that of the three-zone Oristano model and the one-dimensional test model. However, the convergence of Scenarios 1 and 2 for the 15-zone case decreases around the fifth iteration. This is due to the fact that since the hydraulic conductivity of zone 7 is relatively insensitive to the least squares objective, this value reaches its lower bound on the first iteration. Therefore, the remaining zonal values must be adjusted to "compensate" for this constraint. Here we observe the consequences of over-parameterization and nonuniqueness, i.e., nearly the same objective function value is achieved with entirely different parameter values (Figure 10). Because of the combination of the insensitivity of zone 7 and the presence of nonuniqueness, the impacts associated with the lower bound constraint are not noticeable until the ninth iteration, when the objective starts to become small. At this point, the algorithm convergence is reduced until the hydraulic conductivity of zone 7 gets close to the true value, after which superlinear convergence is obtained. The convergence of the hydraulic conductivities of zones 2, 3, and 7, as shown in Figure 10, confirms these observations.

[33] The results associated with scenario 3 do not demonstrate the same impacts on convergence resulting from the parameters being constrained by their lower bounds. This is likely due to the fact that the Jacobian matrices (section 3.3) are projected onto a subspace that is determined using the initial guess of hydraulic conductivity, which does not contain parameters residing on their lower bounds, resulting in more robust or better-conditioned Jacobian matrices. In other words, the proposed algorithm associated with scenario 3 seems to overcome the problems of overparameterization and nonuniqueness present in the 15-zone test case. This is not true for scenario 2 because the basis functions, and hence the reduced-model subspace, are updated at each iteration using the current parameter values. So, at the second iteration, the basis functions are reconstructed using snapshots generated from a parameter set that is constrained by its lower bound. In summary, the results of scenario 3 may indicate a potentially unforeseen advantage associated with the proposed algorithm for overparameterized systems, in addition to computational considerations (of

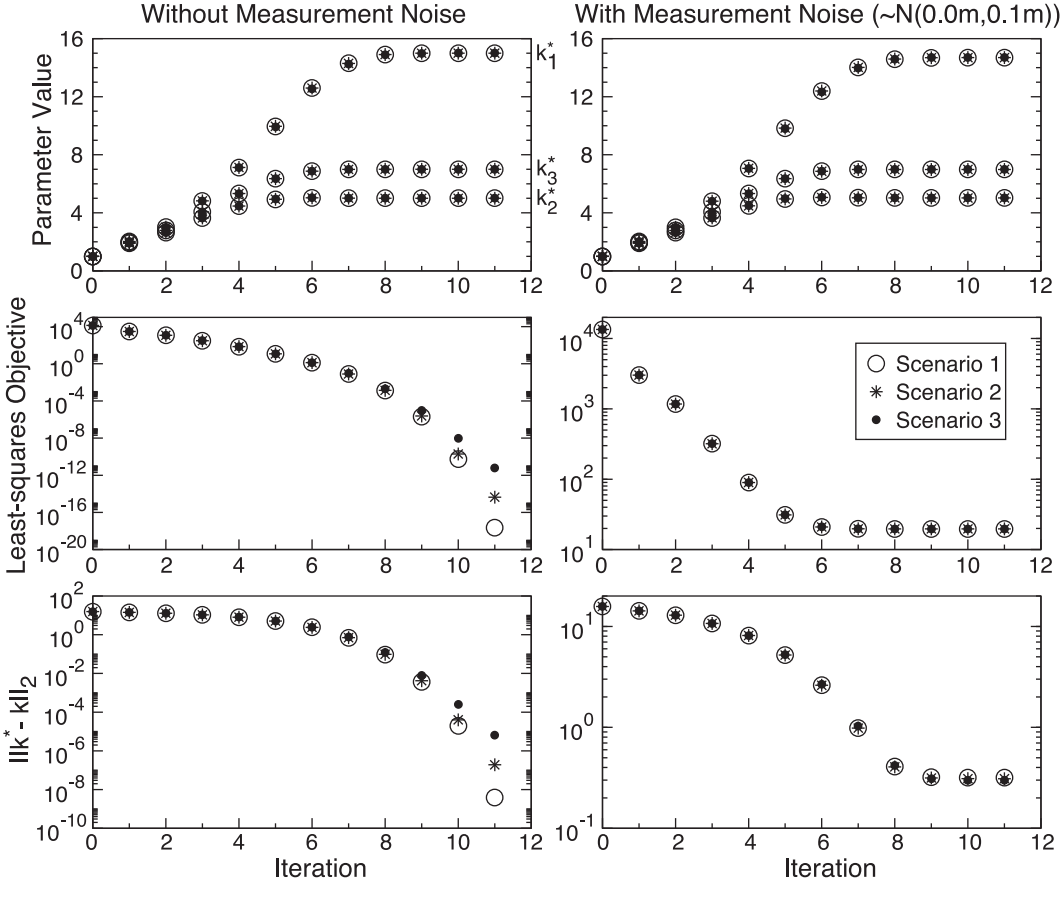

Figure 8. Convergence results for the three-zone case with and without measurement noise.

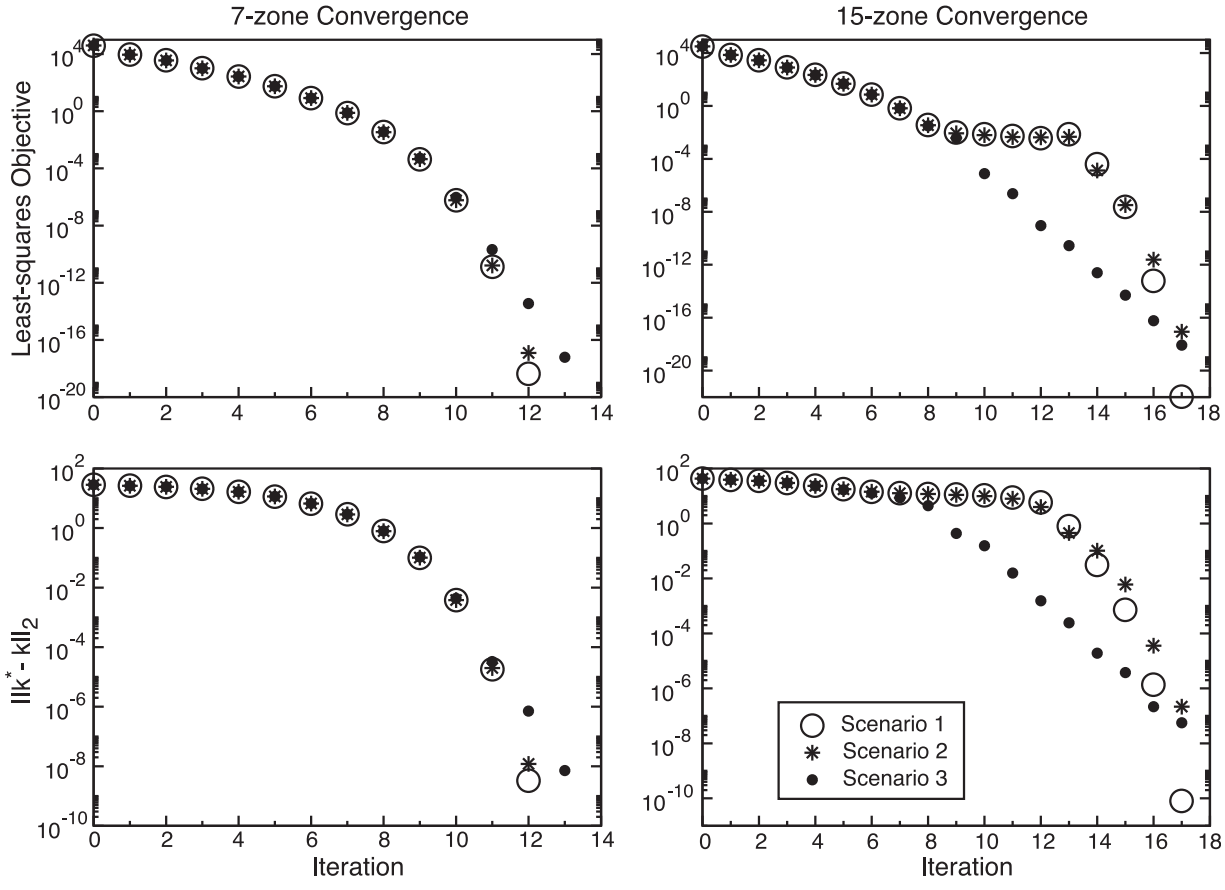

Figure 9. Convergence results for the seven- and 15-zone models.

which is the focus of this study), and is a subject of further research.

[34] Comparing, quantitatively, the computational improvements associated with the proposed method is not straightforward. As shown previously, each scenario may require a different number of iterations to achieve convergence. Additionally, each scenario may arrive at different parameter values throughout the inverse problem, which, in turn, may lead to different computational costs associated with iterative QP solvers. However, some comparisons are required. The linearized full model for the 15-zone case, at iteration 1, required ~24.5 s of run time on an iMac quad-core computer with an INTEL 2.93 GHz i7 processor equipped with 12 GB RAM, 256 KB L2 cache (per core), 8 MB L3 cache, with no hyperthreading and no automatic parallelization using a GNU GFortran compiler with O3 automatic optimization. The same simulation with the linearized reduced model required ~10.3 s. The vast majority of the principal vectors were retained for this reduced-model simulation; truncation of principal vectors will yield drastic reductions in the reduced-model run time. For example, if $\Phi = 0.99$ (section 3.3), only 50 principal vectors are retained on iteration 1 of the 15 zone case. The eigenvalue decomposition required very little effort (~1.4 s). Solving the QP problem at iteration 1 using the linearized full and reduced versions of the quasilinearization procedure (not including snapshot generation) required 37.7 and 15.35 min, respectively. This comparison will be different for each iteration. The overall inverse problem (15-zone case) for scenarios 1, 2 and 3 required 20 h 13 min, 7 h 12 min, and 3 h 51 min, respectively, yielding a speedup of more than 5 times for the method proposed in this study. Truncating principal vectors, exploiting the small size of the QPs, and more efficient data storage will likely reduce the computational expense even further.

6. Discussion and Conclusions

[35] The computational burden associated with solving the problem of parameter estimation is dependent on both the number of times the model under investigation must be called as well as the computational expense associated with calling this model. We have developed a new technique for solving the inverse problem in which the computational burden of solving the model is dramatically reduced. The method proposed is an extension of the quasilinearization technique where the governing system of differential equations is linearized with respect to the parameters, resulting in a least squares regression problem or a quadratic programming (QP) problem. The solution becomes an update on the parameter set. This process is then repeated until convergence takes place. Applying the proper orthogonal decomposition (POD) method drastically reduces the computational burden associated with these regression problems. This is achieved by reducing the dimensionality of the linearized flow model embedded in the QP problem to be solved at each iteration.

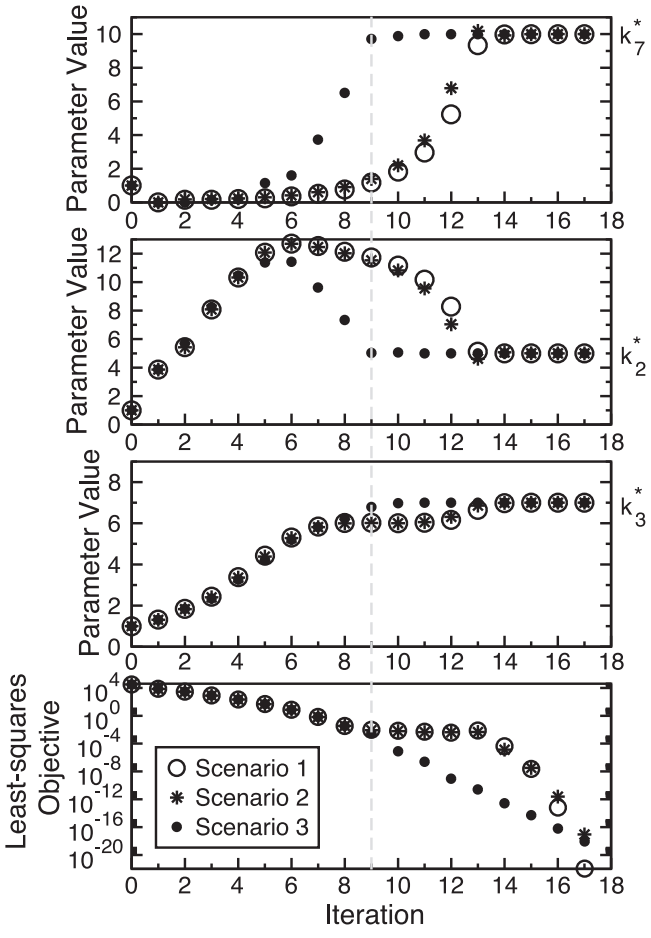

Figure 10. Convergence behavior of hydraulic conductivity in zones 2, 3, and 7 and the objective function for the 15-zone case. The dashed gray line indicates the iteration in which the objective function for scenario 3 diverges from that of scenarios 1 and 2.

[36] The proposed algorithm was used to solve the inverse problem for confined groundwater flow models. First, a one-dimensional test case was used to illustrate the algorithm mechanics. The methodology then was applied to a two-dimensional, finely discretized version of a real model in the Oristano region of the island of Sardinia, Italy. The results obtained from numerical experiments indicate that the convergence of the quasilinearization scheme was nearly identical for the linearized full model and the POD reduced model derived from the linearized full model. This suggests that the proposed method may be feasible for solving real-world large-scale inverse problems (i.e., problems with a large number of computational nodes).

[37] The implications associated with the reduced basis updates also were explored. Through simulation, the results have shown that this basis, i.e., the reduced model subspace, does not need to be updated between successive iterations of the quasilinearization procedure, so long as the initial estimate of the parameter values is within the region of convergence, which may be slightly smaller than that of the algorithm without POD model reduction. Removing the need for updating the basis between iterations drastically reduces the computational burden of solving the inverse problem since the snapshot set needs to be developed only once, at the first iteration. In other words, the algorithm proceeds by calling the linearized full model once per parameter at the first iteration only. Then the algorithm continues such that, at each iteration, the original full model is called once and a reduced-order QP problem is solved.

[38] Our numerical experiments indicate that without updating the reduced basis between iterations, the parameter estimation process may even become more stable and efficient. For the 15-zone case of the Oristano model, one of the parameter values was constrained by its lower bound at the first iteration. This was likely a result of this parameter being insensitive to the observation data. As a result, only the reduced-order algorithm for which the reduced basis was *not* updated at every iteration continued to converge superlinearly. This is likely the result of the subspace projection being based on the initial estimate and not on a parameter vector containing a parameter at its lower bound. This phenomenon of added stability to the inverse problem is beyond the scope of this paper and is a topic of further research.

[39] **Acknowledgments.** This material is based on work supported by NSF under award EAR-0910507 and ARO under award W911NF-10-l-0124. We would like to thank the three anonymous reviewers for their in-depth and constructive reviews.

References

Bear, J. (1979), *Hydraulics of Groundwater*, 592 pp., McGraw-Hill, N. Y.

Bellman, R. (1960), *Introduction to Matrix Analysis*, 328 pp., McGraw-Hill, N. Y.

Bellman, R. E., and R. E. Kalaba (1965), *Quasilinearization and Nonlinear Boundary-Value Problems*, 206 pp., Elsevier, N. Y.

Cau, P. L., G. Lecca, M. Putti, and C. Paniconi (2002), The influence of a confining layer on saltwater intrusion and surface recharge and groundwater extraction conditions, in *Computational Methods in Water Resources, Developments in Water Resources*, vol. 1, edited by S. M. Hassanizadeh et al., pp. 493–500, Elsevier, Amsterdam, Netherlands.

Cazemier, W., R. Verstappen, and A. Veldman (1998), Proper orthogonal decomposition and low-dimensional models for driven cavity flows, *Phys. Fluids*, *10*(7), 1685–1699.

Cooley, R. L. (1985), A comparison of several methods of solving nonlinear regression groundwater flow problems, *Water Resour. Res.*, *21*(10), 1525–1538.

Doherty, J. (2002), *PEST: Model-independent parameter estimation*, vol. 4, 279 pp., Watermark Numerical Computing, Brisbane, Queensland, Australia.

Levenberg, K. (1944), A method for the solution of certain non-linear problems in least squares, *Q. J. Appl. Math.*, *2*, 164–168.

Marquardt, D. W. (1963), An algorithm for least-squares estimation of nonlinear parameters, *J. Soc. Indust. Appl. Math.*, *11*(2), 431–441.

McPhee, J., and W. W.-G. Yeh (2008), Groundwater management using model reduction via empirical orthogonal functions, *J. Water Resour. Plann. Manage.*, *134*(2), 161–170.

Oliver, D. S., and Y. Chen (2011), Recent progress on reservoir history matching: A review, *Comput. Geophys.*, *15*(1), 185–221, doi:10.1007/s10596-010-9194-2.

Park, H. M., T. H. Kim, and D. H. Cho (1998), Estimation of parameters in flow reactors using the Karhunen-Loève decomposition, *Comput. Chem. Eng.*, *23*, 109–123.

Poeter, E. P., M. C. Hill, E. R. Banta, S. Mehl, and S. Christensen (2005), UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation, *U.S. Geological Survey Techniques Methods*, *6-A11*, 283 pp.

Siade, A. J., M. Putti, and W. W.-G. Yeh (2010), Snapshot selection for groundwater model reduction using proper orthogonal decomposition, *Water Resour. Res.*, *46*, W08539, doi:10.1029/2009WR008792.

Sun, N.-Z. (1994), *Inverse Problems in Groundwater Modeling*, 352 pp., Kluwer Acad., N. Y.

Vermeulen, P. T. M., A. W. Heemink, and C. T. Stroet (2004), Reduced models for linear groundwater flow models using empirical orthogonal functions, *Adv. Water Resour.*, *27*, 57–69.

Vermeulen, P. T. M., A. W. Heemink, and J. R. Valstar (2005), Inverse modeling of groundwater flow using model reduction, *Water Resour. Res.*, *41*, W06003, doi:10.1029/2004WR003698.

Willcox, K., and J. Peraire (2002), Balanced model reduction via the proper orthogonal decomposition, *AIAA J.*, *40*(11), 2323–2330.

Willis, R. (1979), A planning model for the management of groundwater quality, *Water Resour. Res.*, *15*(6), 1305–1312.

Yeh, W. W.-G. (1975), Optimal identification of parameters in an inhomogeneous medium with quadratic programming, *Soc. Pet. Eng. J.*, *15*(5), 371–375.

Yeh, W. W.-G. (1986), Review of parameter identification procedures in groundwater hydrology: The inverse problem, *Water Resour. Res.*, *22*(2), 95–108.

Yeh, W. W.-G., and G. W. Tauxe (1971a), Quasilinearization and the identification of aquifer parameters, *Water Resour. Res.*, *7*(2), 375–381.

Yeh, W. W.-G., and G. W. Tauxe (1971b), Optimal identification aquifer diffusivity using quasilinearization, *Water Resour. Res.*, *7*(4), 955–962.