Advances in Water Resources 102 (2017) 68–83

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

A proposed Fast algorithm to construct the system matrices for a reduced-order groundwater model

Timothy T. Ushijima, William W-G. Yeh*

Department of Civil and Environmental Engineering, University of California, Los Angeles, CA, 90095, USA

ARTICLE INFO

Article history:
Received 20 March 2016
Revised 13 January 2017
Accepted 23 January 2017
Available online 31 January 2017

Keywords:
Model reduction
Proper orthogonal decomposition
Groundwater modeling
Efficient system matrix construction

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.

1. Introduction

1.1. 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 and Mayer, 2006; Sreekanth et al., 2016) or monitoring network design (Luo et al., 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 and Minsker, 2010; Reed et al., 2000; Ushijima and Yeh, 2015, 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 and 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

* Corresponding author.
E-mail addresses: tim.ushijima@gmail.com (T.T. Ushijima), williamy@seas.ucla.edu (W.W-G. Yeh).

http://dx.doi.org/10.1016/j.advwatres.2017.01.010

approach (Castelletti et al., 2012a, 2012b), which projects onto a subspace that has some real-world significance.

1.2. Proper orthogonal decomposition

The theory behind solution space projection methods states that if the solutions from a groundwater model exist in some space $V \in R^{N_n}$, where N_n is the dimension of the original model, there exists some subspace $\lambda \in R^{n_p}$ such that $Range(\lambda) = Range(V)$ and $n_p \le N_n$. If $n_p \ll N_n$, and the original (often referred to as the full) groundwater model can be projected ontoλ, 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 and Jin, 2003; Willcox and Peraire, 2002) and many studies have demonstrated that a POD reduced-order groundwater model significantly decreases the cost of running the model (McPhee and Yeh, 2008; Siade et al., 2010). Examples of applications of POD include predictive groundwater models (Boyce and Yeh, 2014; McPhee and Yeh, 2006; Pasetto et al., 2013; Siade et al., 2010; Vermeulen et al., 2006), solving the inverse problem of parameter estimation (Liu et al., 2013; Siade et al., 2012), and modeling of variable density flow and transport processes (Li et al., 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:

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial h}{\partial z}\right) = S_s \frac{\partial h}{\partial t} + f \quad (1)$$

where K_i is the hydraulic conductivity along the i^{th} coordinate axes, h is the hydraulic head, S_s 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. (1), we apply finite-difference or finite-element approximations to this PDE to produce the following set of linear ordinary differential equations (ODEs):

$$\mathbf{B}\frac{d\mathbf{h}}{dt} + \mathbf{A}(\mathbf{k})\mathbf{h} = \mathbf{q}$$

where $\mathbf{B} \in R^{N_n x N_n}$ is the mass matrix, $\mathbf{A}(\mathbf{k}) \in R^{N_n x N_n}$ is the stiffness matrix, and $\mathbf{q} \in R^{N_n}$ 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:

$$(\mathbf{B} + \mathbf{A}(\mathbf{k}))\mathbf{h} = \tilde{\mathbf{A}}(\mathbf{k})\mathbf{h} = \tilde{\mathbf{b}} \quad (2)$$

where $\mathbf{k} \in R^{N_n}$ is a vector of the hydraulic conductivity in each node of the finite-difference (or finite-element) approximation, the system matrix $\tilde{\mathbf{A}}(\mathbf{k}) \in R^{N_n x N_n}$ depends on $\mathbf{k}$, and $\mathbf{h} \in V$. Eq. (2) 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 $\mathbf{k}$.

POD begins by finding some matrix $\mathbf{P} \in R^{N_n x n_p}$ whose columns form an orthonormal basis for V. The following multiplication is then performed:

$$\mathbf{P}^T\tilde{\mathbf{A}}(\mathbf{k})\mathbf{P}\mathbf{r} = \mathbf{P}^T\tilde{\mathbf{b}} \quad (3)$$

where $\mathbf{P}^T\tilde{\mathbf{A}}(\mathbf{k})\mathbf{P} \in R^{n_p x n_p}$ is the system matrix for the reduced model and $\mathbf{r} \in R^{n_p}$ is the state vector of the reduced model. When $n_p \ll N_n$, we are able to solve this reduced model (Eq. (3)) at a much reduced computational cost compared to the full model (Eq. (2)). The solutions from the reduced model ($\mathbf{r}$) can be projected back onto the full model space (V) to approximate the solutions from the full model ($\mathbf{h}$):

$$\mathbf{h} \approx \mathbf{P}\mathbf{r}$$

The accuracy of this approximation depends on the quantity and quality of the information that is captured in the $\mathbf{P}$ matrix. Excellent research has been published on ways to construct $\mathbf{P}$ to lead to accurate approximations of $\mathbf{h}$ 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 $\mathbf{P}$ in a way that leads to good approximations of $\mathbf{h}$ independent of changes in model forcing (Siade et al., 2010) and/or model parameters (e.g. $\mathbf{k}$) (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 and Yeh, 2014; Ushijima and Yeh, 2015, 2013). However, a major drawback of POD model reduction is the computational burden of constructing the system matrix for the reduced model. As Eq. (3) shows, the system matrix for the reduced model ($\mathbf{A}_r(\mathbf{k})$) can be constructed by calculating the following matrix multiplication:

$$\tilde{\mathbf{A}}_r(\mathbf{k}) = \mathbf{P}^T\tilde{\mathbf{A}}(\mathbf{k})\mathbf{P} \quad (4)$$

The FLOP (floating point operation) count of this matrix multiplication is $O(N_n^2 + n_p^3)$. Since we desire $n_p << N_n$, the FLOP count is $O(N_n^2)$. If we know $\mathbf{k}$, 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 and Yeh, 2014), a Monte Carlo simulation (Pasetto et al., 2014), or worst-case scenario design (Ushijima and 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 $O(n_p^3)$, 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 $\mathbf{k}$ for online construction or construct $\mathbf{A}_r(\mathbf{k})$ offline for some set values of $\mathbf{k}$. 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 and 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 (Ştefãnescu and Navon, 2013), flow through porous media (Ghasemi and 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.