Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 292 (2015) 122–137
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Global–local nonlinear model reduction for flows in heterogeneous porous media

Manal Alotaibi[a,b], Victor M. Calo[b,c], Yalchin Efendiev[a,b,*], Juan Galvis[d], Mehdi Ghommem[b]

[a] *Department of Mathematics & Institute for Scientific Computation (ISC), Texas A&M University, College Station, TX, USA*
[b] *Center for Numerical Porous Media (NumPor), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*
[c] *Applied Mathematics & Computational Science and Earth Sciences & Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*
[d] *Departamento de Matemáticas, Universidad Nacional de Colombia, Carrera 45 No 26-85 - Edificio Uriel Gutierréz, Bogotá D.C., Colombia*

Available online 24 November 2014

Dedicated to Mary Wheeler on the occasion of her 75th birthday anniversary

Abstract

In this paper, we combine discrete empirical interpolation techniques, global mode decomposition methods, and local multiscale methods, such as the Generalized Multiscale Finite Element Method (GMsFEM), to reduce the computational complexity associated with nonlinear flows in highly-heterogeneous porous media. To solve the nonlinear governing equations, we employ the GMsFEM to represent the solution on a coarse grid with multiscale basis functions and apply proper orthogonal decomposition on a coarse grid. Computing the GMsFEM solution involves calculating the residual and the Jacobian on a fine grid. As such, we use local and global empirical interpolation concepts to circumvent performing these computations on the fine grid. The resulting reduced-order approach significantly reduces the flow problem size while accurately capturing the behavior of fully-resolved solutions. We consider several numerical examples of nonlinear multiscale partial differential equations that are numerically integrated using fully-implicit time marching schemes to demonstrate the capability of the proposed model reduction approach to speed up simulations of nonlinear flows in high-contrast porous media.

Keywords: Generalized multiscale finite element method; Nonlinear PDEs; Heterogeneous porous media; Discrete empirical interpolation; Proper orthogonal decomposition

* Corresponding author at: Department of Mathematics & Institute for Scientific Computation (ISC), Texas A&M University, College Station, TX, USA. Tel.: +1 979 458 0836; fax: +1 979 862 4190.
E-mail address: efendiev@math.tamu.edu (Y. Efendiev).

http://dx.doi.org/10.1016/j.cma.2014.10.034
0045-7825/

1. Introduction

Nonlinear partial differential equations (PDEs), with multiple scales and/or high contrast in media properties, represent a class of problems with many relevant engineering and scientific applications in porous media. Solving these equations using iterative methods, such as Newton iterations, requires updating the numerical solution of a large system of equations at each iteration using the previous iterate results. Due to the extensive computational requirements resulting from the disparity of scales and nonlinearity, computing fine-grid solutions becomes prohibitively expensive. Moreover, these type of problems often involve coefficients that exhibit high-contrast and heterogeneous distributions. For example, when modeling subsurface flows the underlying permeability field is often represented as a high-contrast coefficient in the pressure equation. This complicates the simulation of a large number of configurations for design purposes. As such, we develop simplified reduced-order models to speed up simulations of nonlinear flows in porous media within a prescribed accuracy. In this paper, we present a coupled approach for solving nonlinear PDEs that combines local and global model-reduction techniques and the Discrete Empirical Interpolation Method (DEIM).

Proper Orthogonal Decomposition (POD) is one of the best known global model-reduction methods. The main purpose of this technique is to reduce the dimension of the dynamical system by projecting the high-dimensional system into a lower-dimensional manifold using a set of orthonormal basis functions (POD modes) constructed from a sequence of snapshots [1–5]. In addition to order reduction, this technique constitutes a powerful mode decomposition technique for extracting the most energetic structures from a linear or nonlinear dynamical process [1–4,6–12].

Coarse-grid computational models are often preferred because of the cost of solving the systems arising in the approximation of the nonlinear flow equation on the fine grid. Some accurate reduced-order methods have been introduced and used in various applications, such as Galerkin multiscale finite elements (e.g., [13–18]), mixed multiscale finite element methods (e.g., [19–22]), the multiscale finite volume method (see, e.g., [23]), and mortar multiscale methods (see e.g., [24,25]), where Mary Wheeler and her collaborators have made significant contributions. The main idea of the these methods is to construct coarse basis functions that approximate the solution on a coarse grid. Multiscale methods can be considered as local model reduction techniques that approximate the solution on a coarse grid for arbitrary coarse-level inputs. In this paper we apply the enriched coarse space construction from the Generalized Multiscale Finite Element Method (GMsFEM) as an effective tool for local model reduction [15,26–28].

Combining of the aforementioned local and global model-reduction schemes has been used for linear problems [18,26]. A significant reduction in the computational complexity when solving linear parabolic PDE in [18] has been achieved by combining the concepts of (GMsFEM) and (POD) and/or Dynamic Mode Decomposition (DMD). In [26], balanced truncation is used to perform global model reduction and is efficiently combined with the local model reduction tools introduced in [15]. More recently, local and global multiscale methods are combined to derive reduced-order models for nonlinear flows in high-contrast porous media. In [29], the proposed multiscale empirical interpolation method for solving nonlinear multiscale PDEs uses GMsFEM to represent the coarse-scale solution. To avoid performing fine-grid computations, the discrete empirical interpolation method introduced in [30] was used to approximate the nonlinear functions at selected points in each coarse region and then a multiscale proper orthogonal decomposition technique is used to find an appropriate interpolation vector. Although, the numerical results presented in [29] proved the applicability of the presented method, the reduction is limited by the full cost of the evaluation of the projected nonlinear function. When dealing with reduced-order models of nonlinear systems obtained by projecting the governing equations onto a subspace spanned by the POD modes, the evaluation of the projected nonlinear term is costly since it depends on the full dimension of the original system.

In this paper, our main contribution is to circumvent this issue by employing DEIM to approximate the nonlinear functions locally (at selected points in each coarse region) at the offline stage and globally (at selected points in the domain) at the online stage. For this reason, we refer to our method as global–local nonlinear approach. The numerical results presented in this paper show that the proposed method enables significant reduction in the computational cost associated with constructing projection-based reduced-order models. In addition to the model reduction, the proposed approach allows us to improve the reduced-order solutions in different ways. For instance, increasing the number of local and global points used at the offline and online stages, respectively, leads to better approximation (see Example 4.2). Also, using several offline parameter inputs (Example 4.3) improves the reduced-order solutions.

The remainder of the paper is organized as follows. In Section 2 we introduce and describe the model problem, the discrete empirical interpolation method, and generalized multiscale finite element method. The presented global–local

DEIM approach is then discussed in Section 3. Numerical results are presented in Section 4 and conclusions in Section 5.

2. Preliminaries

2.1. Model problem

We consider a time-dependent nonlinear flow governed by the following parabolic partial differential equation

$$\frac{\partial u}{\partial t} - \nabla \cdot (\kappa(x; u, \mu)\nabla u) = h(x) \quad \text{in } \Omega, \tag{1}$$

with some boundary conditions. The variable $u = u(t, x; \mu)$ denotes the pressure, Ω is a bounded domain, h is a forcing term, and in our case the permeability field represented by $\kappa(x; u, \mu)$ is a nonlinear function. Here, $\frac{\partial}{\partial t}$ is the time derivative and μ represents a parameter.

2.2. Discrete empirical interpolation method (DEIM)

We approximate with the Discrete Empirical Interpolation Method (DEIM) [30] local and global nonlinear functions. DEIM is based on approximating a nonlinear function by means of an interpolatory projection of a few selected snapshots of the function. The idea is to represent a function over the domain while using empirical snapshots and information at some locations (or components).

We briefly review DEIM as presented in [30]. Let $f(\tau) \in \mathbb{R}^n$ denotes a nonlinear function where $\tau \in \mathbb{R}^{n_s}$. Here, in general, n_s can be different from n. In a reduced-order modeling, τ has a reduced representation

$$\tau = \sum_{i=1}^{l} \alpha_i \zeta_i$$

where $l \ll n_s$. This leads us to look for an approximation of $f(\tau)$ at a reduced cost. To perform a reduced order approximation of $f(\tau)$, we first define a reduced dimensional space for $f(\tau)$. That is, we would like to find m basis vectors (where m is much smaller than n), $\psi_1, \ldots, \psi_m$, such that we can write

$$f(\tau) \approx \Psi d(\tau), \tag{2}$$

where $\Psi = (\psi_1, \ldots, \psi_m) \in \mathbb{R}^{n \times m}$.

The goal of DEIM is to find $d(\tau)$ using only a few rows of (2). In general, one can define $d(\tau)$'s using m rows of (2) and invert a reduced system to compute $d(\tau)$. This can be formalized using the matrix P

$$\mathrm{P} = [e_{\wp_1}, \ldots, e_{\wp_m}] \in \mathbb{R}^{n \times m},$$

where $e_{\wp_i} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^n$ is the $\wp_i^{\text{th}}$ column of the identity matrix $\mathrm{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \ldots, m$. Multiplying Eq. (2) by P^T and assuming that the matrix $\mathrm{P}^T \Psi$ is nonsingular, we obtain

$$f(\tau) \approx \tilde{f}(\tau) = \Psi d(\tau) = \Psi(\mathrm{P}^T \Psi)^{-1} \mathrm{P}^T f(\tau). \tag{3}$$

To summarize, approximating the nonlinear function $f(\tau)$, as given by Eq. (3), requires the following:

- Computing the projection basis $\Psi = (\psi_1, \ldots, \psi_m)$;
- Identifying the indices $\{\wp_1, \ldots, \wp_m\}$.

To determine the projection basis $\Psi = (\psi_1, \ldots, \psi_m)$, we collect function evaluations in an $n \times n_s$ matrix $\mathrm{F} = [f(\tau_1), \ldots, f(\tau_{n_s})]$ and employ POD to select the most energetic modes. This selection uses the eigenvalue decomposition of the square matrix $\mathrm{F}^T\mathrm{F}$ (left singular values) and forms the important modes using the dominant eigenvalues. These modes are used as the projection basis in the approximation given by Eq. (2). In Eq. (3), the term $\Psi(\mathrm{P}^T \Psi)^{-1} \in \mathbb{R}^{n \times m}$ is computed once and stored. The $d(\tau)$ is computed using the values of the function $f(\tau)$ at m points with the indices $\wp_1, \ldots, \wp_m$ identified using the following DEIM algorithm.

DEIM Algorithm [30]:

Input: The projection basis matrix $\Psi = (\psi_1, \ldots, \psi_m)$ obtained by applying POD on a sequence of n_s function evaluations.

Output: The interpolation indices $\overrightarrow{\wp} = (\wp_1, \ldots, \wp_m)^T$

1: Set $[|\rho|, \wp_1] = \max\{|\psi_1|\}$

2: Set $\Psi = [\psi_1]$, $P = [e_{\wp_1}]$, and $\overrightarrow{\wp} = (\wp_1)$

3: for $k = 2, \ldots, m$ do

– Solve $(P^T \Psi)w = P^T \psi_k$ for some w.

– Compute $r = \psi_k - \Psi w$

– Compute $[|\rho|, \wp_k] = \max\{|r|\}$

– Set $\Psi = [\Psi\ \psi_k]$, $P = [P\ e_{\wp_k}]$, and $\overrightarrow{\wp} = \begin{pmatrix} \overrightarrow{\wp} \\ \wp_k \end{pmatrix}$

end for

The computational saving is due to the resulting fewer evaluations of $f(\tau)$. This shows the advantage of using DEIM algorithm in our proposed reduction method. However, applying the DEIM algorithm to reduce the computational cost of the nonlinear function requires additional computations in the offline stage, which will be discussed in Section 3.2. These algorithms are successful if the nonlinear functions admit low dimensional approximations.

2.3. Generalized multiscale finite element method (GMsFEM)

Below we summarize the offline/online computational procedure in the following steps:

1. Offline computations:
   - 1.0. Generation coarse grid.
   - 1.1. Construction of snapshot space used to compute the offline space.
   - 1.2. Construction of a low dimensional space by performing dimension reduction in the space of local snapshots.
2. Online computations:
   - 2.1. For each input parameter set, compute multiscale basis functions.
   - 2.2. Solution of a coarse-grid problem for given forcing term and boundary conditions.

In the offline computation, we first construct a snapshot space $V_{\text{snap}}^{\omega_i}$. Constructing the snapshot space may involve solving various local problems for different choices of input parameters or different fine-grid representations of the solution in each coarse region. We denote each snapshot vector (listing the solution at each node in the domain) using a single index and create the following matrix

$$\Phi_{\text{snap}} = \left[\phi_1^{\text{snap}}, \ldots, \phi_{M_{\text{snap}}}^{\text{snap}}\right],$$

where ϕ_j^{snap} denotes the snapshots and M_{snap} denotes the total number of functions to keep in the local snapshot matrix construction.

In order to construct an offline space V_{off}, we reduce the dimension of the snapshot space using an auxiliary spectral decomposition. The main objective is to use the offline space to efficiently (and accurately) construct a set of multiscale basis functions to be used in the online stage. More precisely, we build a snapshot subspace that can approximate with sufficient accuracy any element of the original snapshot space. The quality of the approximation is determined in the sense defined via auxiliary bilinear forms. At the offline stage, the bilinear forms are chosen to be *parameter-independent* (through nonlinearity), such that there is no need to reconstruct the offline space for each ν value, where ν is assumed to be a parameter that represents u and μ_κ in $\kappa(x, u, \mu_\kappa)$. To construct the offline space, we use the average of the parameters over the coarse region ω_i in $\overline{\kappa}(x, \nu)$ while keeping the spatial variations. That is, ν represents both the average of u and μ. We consider the following eigenvalue problem in the space of snapshots,

$$A^{\text{off}} \Phi_k^{\text{off}} = \lambda_k^{\text{off}} S^{\text{off}} \Phi_k^{\text{off}}, \tag{4}$$

where

$$A^{\text{off}} = [a_{mn}^{\text{off}}] = \int_{\omega_i} \overline{\kappa}(x, \nu) \nabla \phi_m^{\text{snap}} \cdot \nabla \phi_n^{\text{snap}} = \Phi_{\text{snap}}^T \overline{A}\, \Phi_{\text{snap}},$$

$$S^{\text{off}} = [s^{\text{off}}_{mn}] = \int_{\omega_i} \overline{\widetilde{\kappa}}(x,\nu)\phi^{\text{snap}}_m \phi^{\text{snap}}_n = \Phi^T_{\text{snap}} \overline{S}\, \Phi_{\text{snap}}.$$

The coefficients $\overline{\kappa}(x,\nu)$ and $\overline{\widetilde{\kappa}}(x,\nu)$ are parameter-averaged coefficients (see [31]). The $\overline{A}$ denotes a fine-scale matrix, except that parameter-averaged coefficients are used in its construction. The fine-scale stiffness matrix A is constructed by integrating only on ω_i

$$A = [a_{mn}] = \int_{\omega_i} \kappa(x,u,\mu)\nabla\phi^{\text{snap}}_m \cdot \nabla\phi^{\text{snap}}_n. \tag{5}$$

To generate the offline space, we then choose the smallest M_{off} eigenvalues from Eq. (4) and form the corresponding eigenvectors in the respective space of snapshots by setting $\phi^{\text{off}}_k = \sum_j \Phi^{\text{off}}_{kj}\phi^{\text{snap}}_j$ (for $k = 1,\ldots,M_{\text{off}}$), where Φ^{off}_{kj} are the coordinates of the vector Φ^{off}_k. We then create the offline matrices

$$\Phi_{\text{off}} = \left[\phi^{\text{off}}_1, \ldots, \phi^{\text{off}}_{M_{\text{off}}}\right]$$

to be used in the online space construction.

The online coarse space is used within the finite element framework to solve the original global problem, where continuous Galerkin multiscale basis functions are used to compute the global solution. In particular, we seek a subspace of the respective offline space such that it can approximate well any element of the offline space in an appropriate metric. At the online stage, the bilinear forms are chosen to be *parameter dependent*. The following eigenvalue problems are posed in the reduced offline space:

$$A^{\text{on}}(\nu)\Phi^{\text{on}}_k = \lambda^{\text{on}}_k S^{\text{on}}(\nu)\Phi^{\text{on}}_k, \tag{6}$$

where

$$A^{\text{on}}(\nu) = [a^{\text{on}}(\nu)_{mn}] = \int_{\omega_i} \kappa(x,\nu)\nabla\phi^{\text{off}}_m \cdot \nabla\phi^{\text{off}}_n = \Phi^T_{\text{off}} A(\nu)\Phi_{\text{off}},$$
$$S^{\text{on}}(\nu) = [s^{\text{on}}(\nu)_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x,\nu)\phi^{\text{off}}_m \phi^{\text{off}}_n = \Phi^T_{\text{off}} S(\nu)\Phi_{\text{off}},$$

and $\kappa(x,\nu)$ and $\widetilde{\kappa}(x,\nu)$ are now parameter dependent. To generate the online space, we then choose the smallest M_{on} eigenvalues from (6) and form the corresponding eigenvectors in the offline space by setting $\phi^{\text{on}}_k = \sum_j \Phi^{\text{on}}_{kj}\phi^{\text{off}}_j$ (for $k = 1,\ldots,M_{\text{on}}$), where Φ^{on}_{kj} are the coordinates of the vector Φ^{on}_k. If $\kappa(x,u) = k_0(x)b(u)$, then one can use the parameter-independent case of GMsFEM. In this case, there is no need to construct an online space (i.e., the online space is the same as the offline space). From now on, we denote the online space basis functions by ϕ_i.

3. Global–local nonlinear model reduction

3.1. Local multiscale model reduction

The finite element discretization of (1) yields a system of ordinary differential equations given by

$$\mathrm{M}\dot{\mathrm{U}} + \mathrm{F}(\mathrm{U}) = \mathrm{H}, \tag{7}$$

where

$$\mathrm{U} = \begin{pmatrix} u_1 & u_2 & \cdots & u_{N_f} \end{pmatrix}$$

is the vector collecting the pressure values at all nodes in the local domain and H is the right-hand-side vector obtained by discretization. Using the offline basis functions, we can write (in discrete form)

$$\kappa(x,u,\mu) = \sum_{q=1}^{Q} \kappa_q(x) b_q(u,\mu). \tag{8}$$

This results in

$$\mathrm{F}(\mathrm{U},\mu) = \sum_{q=1}^{Q} \mathrm{A}_q \Lambda_1^q(\mathrm{U},\mu)\mathrm{U},$$

where we have

$$\mathrm{A}_q := [\mathrm{a}_{ij}^q] = \int_\Omega \kappa_q \nabla\phi_i^0 \cdot \nabla\phi_j^0, \qquad \mathrm{M} := [\mathrm{m}_{ij}] = \int_\Omega \phi_i^0\phi_j^0, \qquad \mathrm{H} := [\mathrm{h}_i] = \int_\Omega \phi_i^0 h,$$
$$\Lambda_1^q(\mathrm{U},\mu) = \mathrm{diag}\left(b_q(u_1,\mu) \quad b_q(u_2,\mu) \quad \cdots \quad b_q(u_{N_f},\mu)\right),$$

and ϕ_i^0 are piecewise linear basis functions defined on a fine triangulation of Ω.

Employing the backward Euler scheme for the time marching process, we obtain

$$\mathrm{U}^{n+1} + \Delta t\ \mathrm{M}^{-1}\mathrm{F}(\mathrm{U}^{n+1}) = \mathrm{U}^n + \Delta t\ \mathrm{M}^{-1}\mathrm{H}, \tag{9}$$

where Δt is the time-step size and the superscript n refers to the temporal level of the solution. The residue is defined as:

$$\mathrm{R}(\mathrm{U}^{n+1}) = \mathrm{U}^{n+1} - \mathrm{U}^n + \Delta t\ \mathrm{M}^{-1}\mathrm{F}(\mathrm{U}^{n+1}) - \Delta t\ \mathrm{M}^{-1}\mathrm{H} \tag{10}$$

with derivative (Jacobian)

$$\begin{aligned}\mathrm{J}(\mathrm{U}^{n+1}) &= D\mathrm{R}(\mathrm{U}^{n+1}) = I + \Delta t\ \mathrm{M}^{-1} D\mathrm{F}(\mathrm{U}^{n+1}) \\ &= I + \sum_{q=1}^{Q} \Delta t\ \mathrm{M}^{-1}\mathrm{A}_q \Lambda_1^q(\mathrm{U}^{n+1}) + \sum_{q=1}^{Q} \Delta t\ \mathrm{M}^{-1}\mathrm{A}_q \Lambda_2^q(\mathrm{U}^{n+1}),\end{aligned} \tag{11}$$

where

$$\Lambda_2^q(\mathrm{U},\mu) = \mathrm{diag}\left(u_1\frac{\partial b_q(u_1,\mu)}{\partial u} \quad u_2\frac{\partial b_q(u_2,\mu)}{\partial u} \quad \cdots \quad u_{N_f}\frac{\partial b_q(u_{N_f},\mu)}{\partial u}\right),$$

and D is the multi-variate gradient operator defined as $[D\mathrm{R}(\mathrm{U})]_{ij} = \partial R_i/\partial U_j$. The scheme involves, at each time step, the following iterations

$$\begin{aligned}&\mathrm{J}(\mathrm{U}_{(k)}^{n+1})\Delta\mathrm{U}_{(k)}^{n+1} = -\left(\mathrm{U}_{(k)}^{n+1} - \mathrm{U}^n + \Delta t\ \mathrm{M}^{-1}\mathrm{F}(\mathrm{U}_{(k)}^{n+1}) - \Delta t\ \mathrm{M}^{-1}\mathrm{H}\right) \\ &\mathrm{U}_{(k+1)}^{n+1} = \mathrm{U}_{(k)}^{n+1} + \Delta\mathrm{U}_{(k)}^{n+1},\end{aligned}$$

where the initial guess is $\mathrm{U}_{(0)}^{n+1} = \mathrm{U}^n$ and k is the iteration counter. The above iterations are repeatedly applied until $\|\Delta\mathrm{U}_{(k)}^{n+1}\|$ is less than a specific tolerance.

In our simulations, we use $Q = 1$ in (8) as our focus is on localized multiscale interpolation of nonlinear functionals that arise in discretization of multiscale PDEs. With this choice, we do not need to compute the online multiscale space (i.e., the online space is the same as the offline space).

We use the solution expansion (i.e., $u = \Phi z$) and employ the multiscale framework to obtain a set of N_c ordinary differential equations that constitute a reduced-order model; that is,

$$\dot{z} = -(\Phi^T\mathrm{M}\Phi)^{-1}\Phi^T\mathrm{F}(\Phi z) + (\Phi^T\mathrm{M}\Phi)^{-1}\Phi^T\mathrm{H}. \tag{12}$$

Thus, the original problem with N_f degrees of freedom is reduced to a dynamical system with N_c dimensions where $N_c \ll N_f$.

The nonlinear term $(\Phi^T\mathrm{M}\Phi)^{-1}\Phi^T\mathrm{F}(\Phi z)$ in the reduced-order model, given by Eq. (12), has a computational complexity that depends on the dimension of the full system N_f. As such, solving the reduced system still requires extensive computational resources and time. To reduce this computational requirement, we use multiscale DEIM as described in the previous section.

To solve the reduced system, we employ the backward Euler scheme; that is,

$$z^{n+1} + \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{F}}(z^{n+1}) = z^n + \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{H}}, \tag{13}$$

where $\widetilde{\mathrm{M}} = \Phi^T \mathrm{M}\Phi$, $\widetilde{\mathrm{F}}(z) = \Phi^T \mathrm{F}(\Phi z)$, and $\widetilde{\mathrm{H}} = \Phi^T \mathrm{H}$. We let

$$\widetilde{\mathrm{R}}(z^{n+1}) = z^{n+1} - z^n + \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{F}}(z^{n+1}) - \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{H}} \tag{14}$$

with derivative

$$\begin{aligned}\widetilde{\mathrm{J}}(z^{n+1}) &= D\widetilde{\mathrm{R}}(z^{n+1}) = I + \Delta t\ \widetilde{\mathrm{M}}^{-1} D\widetilde{\mathrm{F}}(z^{n+1}) \\ &= I + \sum_{q=1}^{Q} \Delta t\ \widetilde{\mathrm{M}}^{-1}\Phi^T \mathrm{A}_q \Lambda_1^q(\Phi z^{n+1})\Phi + \sum_{q=1}^{Q} \Delta t\ \widetilde{\mathrm{M}}^{-1}\Phi^T \mathrm{A}_q \Lambda_2^q(\Phi z^{n+1})\Phi.\end{aligned}$$

The scheme involves, at each time step, the following iterations

$$\widetilde{\mathrm{J}}(z_{(k)}^{n+1})\Delta z_{(k)}^{n+1} = -\left(z_{(k)}^{n+1} - z^n + \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{F}}(z_{(k)}^{n+1}) - \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{H}}\right) \tag{15}$$

$$z_{(k+1)}^{n+1} = z_{(k)}^{n+1} + \Delta z_{(k)}^{n+1}, \tag{16}$$

where the initial guess is $z_{(0)}^{n+1} = z^n$. The above iterations are repeated until $\|\Delta z_{(k)}^{n+1}\|$ is less than a specific tolerance. Furthermore, we use multiscale DEIM to approximate the nonlinear functions that appear in the residual $\widetilde{\mathrm{R}}$ and the Jacobian $\widetilde{\mathrm{J}}$ to reduce the number of function evaluations.

3.2. Global–local nonlinear model reduction approach

We denote the offline parameters by θ^{off} which include samples of the right-hand side $h(x)$ denoted by h_i^{off}, samples of μ denoted by μ_i^{off}, and samples of initial conditions denoted by $U_{0,i}^{\mathrm{off}}$. Similarly, the online parameter set is denoted by θ^{on} and includes the online source term h^{on}, the online μ (μ^{on}), and the online initial conditions U_0^{on}. We follow a global–local nonlinear model reduction approach that includes the following steps:

- **Offline stage**

  The offline stage includes the following steps:
  - Consider the offline parameters set $\theta^{\mathrm{off}} = \{\theta_i^{\mathrm{off}}\} = \{h_i^{\mathrm{off}}, \mu_i^{\mathrm{off}}, U_{0,i}^{\mathrm{off}}\}$.
  - Use θ_i^{off} to define the fine-scale stiffness and mass matrices, source terms and multiscale basis functions.
  - Compute the local snapshots of the nonlinear functions and use DEIM algorithm, as described in the previous section, to set the local DEIM basis functions and local DEIM points (L_0^{local}).
  - Generate snapshots of the coarse-grid solutions using local DEIM.
  - Record N_t instantaneous solutions (usually referred as snapshots) using coarse-grid approximations from the above step and collect them in a snapshot matrix as:

    $$\mathbf{Z}^{N_t} = \{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \ldots, \mathbf{Z}_{N_t}\} \tag{17}$$

    where N_t is the number of snapshots and N_c is the size of the column vectors $\mathbf{Z}_i$.
  - Compute the POD modes and use these modes to approximate the solution field on the coarse grid. As such, we assume an expansion in terms of the modes ψ_i; that is, we let

    $$z(x,t) \approx \tilde{z}(x,t) = \sum_{i=1}^{N_r} \alpha_i(t)\psi_i(x) \tag{18}$$

    or in a matrix form

    $$\mathbf{Z}^n \approx \tilde{\mathbf{Z}}^n = \Psi\alpha^n \tag{19}$$

    where $\Psi = \begin{pmatrix}\psi_1 & \cdots & \psi_{N_r}\end{pmatrix}$.

- **Online stage**

  The online stage includes the following steps:
  - Given online $\theta^{\text{on}} = \{h^{\text{on}}, \mu^{\text{on}}, U_0^{\text{on}}\}$.
  - Use the solution expansion given by (18) and project the governing equation of the coarse-scale problem onto the space formed by the modes to obtain a set of N_r ordinary differential equations that constitute a reduced-order model; that is,
  $$\dot{\alpha} = -(\underbrace{\Psi^T \Phi^T \mathbf{M} \Phi \Psi}_{N_r \times N_r})^{-1} \underbrace{\Psi^T}_{N_r \times N_c} \underbrace{\Phi^T}_{N_c \times N_f} \underbrace{\mathbf{F}(\Phi\Psi\alpha)}_{N_f \times 1} + (\Psi^T \Phi^T \mathbf{M} \Phi \Psi)^{-1} \Psi^T \Phi^T \mathbf{H}. \tag{20}$$
  - Employ Newton's method to solve the above reduced system. The Newton scheme involves at each time step the following iteration. We need to solve the linear system
  $$\widehat{\mathrm{J}}(\alpha_{(k)}^{n+1}) \Delta\alpha_{(k)}^{n+1} = -\Big(\alpha_{(k)}^{n+1} - \alpha^n + \Delta t\, \widehat{\mathrm{M}}^{-1} \widehat{\mathrm{F}}(\alpha_{(k)}^{n+1}) - \Delta t\, \widehat{\mathrm{M}}^{-1} \widehat{\mathrm{H}}\Big) \tag{21}$$
  where
  $$\widehat{\mathrm{M}} = \Psi^T \Phi^T \mathbf{M} \Phi \Psi = \Psi^T \tilde{\mathbf{M}} \Psi, \qquad \widehat{\mathrm{H}} = \Psi^T \Phi^T \mathbf{H} = \Psi^T \tilde{\mathbf{H}}, \qquad \widehat{\mathrm{F}} = \Psi^T \Phi^T \mathbf{F} = \Psi^T \tilde{\mathbf{F}}.$$
  Then
  $$\alpha_{(k+1)}^{n+1} = \alpha_{(k)}^{n+1} - (\widehat{\mathrm{J}}(\alpha_{(k)}^{n+1}))^{-1} \Big(\alpha_{(k)}^{n+1} - \alpha^n + \Delta t\, \widehat{\mathrm{M}}^{-1} \widehat{\mathrm{F}}(\alpha_{(k)}^{n+1}) - \Delta t\, \widehat{\mathrm{M}}^{-1} \widehat{\mathrm{H}}\Big).$$
  Thus, the original problem with N_f degrees of freedom is reduced to a dynamical system with N_r dimensions where $N_r \ll N_c \ll N_f$.
  - Use global DEIM to approximate the nonlinear functions that appear in the residual and Jacobian. To do so, we write the nonlinear function $\mathbf{F}(\Phi\Psi\alpha)$ in Eq. (20) as
  $$\mathbf{F}(\Phi\Psi\alpha) \approx \Psi^* d, \tag{22}$$
  where $\Psi^* = [\psi_1^*, \ldots, \psi_{L_0^{\text{global}}}^*]$ is the matrix of the global DEIM basis functions $\{\psi_i^*\}_{i=1}^{L_0^{\text{global}}}$. These functions are constructed using the snapshots of the nonlinear function $\mathbf{F}(\Phi z)$ computed offline and employ the POD technique to select the most energetic modes (see Section 2.2). The coefficient vector d is computed using the values of the function $\mathbf{F}$ at L_0^{global} global points.
  - Use the solution expansion given by (18) in terms of POD modes to approximate the coarse-scale solution and then use the operator matrix Φ to downscale the approximate solution and evaluate the flow field on the fine grid.

4. Numerical results

In this section, we use representative numerical examples to illustrate the applicability of the proposed global–local nonlinear model reduction approach for solving nonlinear multiscale partial differential equations. Before presenting the individual examples, we describe the computational domain used in constructing the GMsFEM basis functions. This computation is performed during the offline stage. We discretize with linear finite elements a nonlinear PDE posed on the computational domain $D = [0, 1] \times [0, 1]$. For constructing the coarse grid, we divide $[0, 1] \times [0, 1]$ into 10×10 squares. Each square is divided further into 10×10 squares each of which is divided into two triangles. Thus, the mesh size is $1/100$ for the fine mesh and $1/10$ for the coarse one. The fine-scale finite element vectors introduced in this section are defined on this fine grid. The fine-grid representation of a coarse-scale vector z is given by Φz, which is a fine-grid vector.

In the following numerical examples, we consider (1) with specified boundary and initial conditions, where the permeability coefficient and the forcing term are given by

$$\kappa(x; u, \mu) = \kappa_q(x) b_q(u, \mu) \quad \text{and} \quad h(x) = 1 + \sin(2\pi x_1) \sin(2\pi x_2).$$

Here, κ_q represents the permeability field with high-conductivity channels as shown in Fig. 1 and $b_q(u, \mu)$ is defined later for each example. We use the GMsFEM along with the Newton method to discretize (1). Furthermore, we employ the local multiscale DEIM in the offline stage and the global multiscale DEIM in the online stage to approximate the nonlinear functions that arise in the residual and the Jacobian.

Using the fine-scale stiffness matrix A that corresponds to (1), as defined in (5), we introduce the relative energy error as

$$\|E\|_{\mathrm{A}} = \sqrt{\frac{(\mathrm{U} - \widetilde{\mathrm{U}})^T \mathrm{A} (\mathrm{U} - \widetilde{\mathrm{U}})}{\mathrm{U}^T \mathrm{A} \mathrm{U}}}. \tag{23}$$

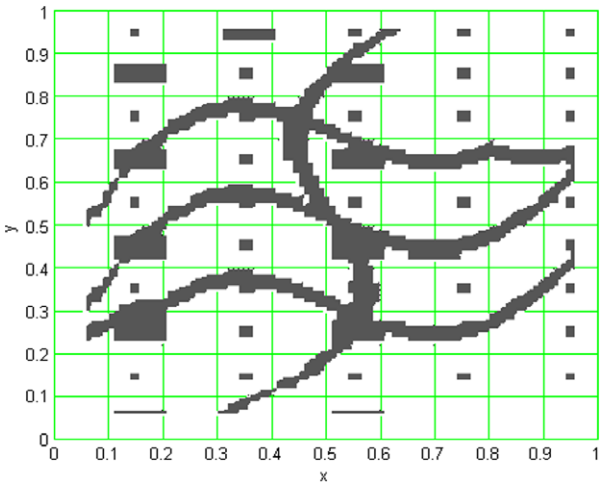


Fig. 1. Permeability field that model high conductivity channels within a homogeneous domain. The minimum (background) conductivity is taken to be $\kappa_{\min} = 1$, and the high conductivity (gray regions) with value of $\kappa_{\max} = \eta$ ($\eta = 10^6$).

Moreover, we define w_0 to be the solution of the problem

$$-\nabla \cdot \left(\kappa_q(x)\nabla w_0\right) = h(x) \quad \text{in } D, \tag{24}$$

to use it in the following examples as our initial guess. In the following, we show:

- In the first example, we compare the approximate solution of the reduced system obtained by applying the global–local approach against the solution of the original system with full dimension (N_f) and show the reduction we achieve in terms of the computational cost.
- In the second example, we show the variations of the error as we increase the number of local DEIM points, L_0^{local}, and global DEIM points, L_0^{global} for one selection of the parameter μ.
- In the third example, we show the effect of using several offline parameters to improve the reduced-order solutions. As such, we use two offline values of the parameter μ and solve an online problem for a different value of μ.
- In the fourth example, we use two offline values of μ and show the variations of the errors as we increase the number of local and global points.
- Random values of the parameter μ with a probability distribution are used in the fifth example. We demonstrate the applicability of our approach in this setup.

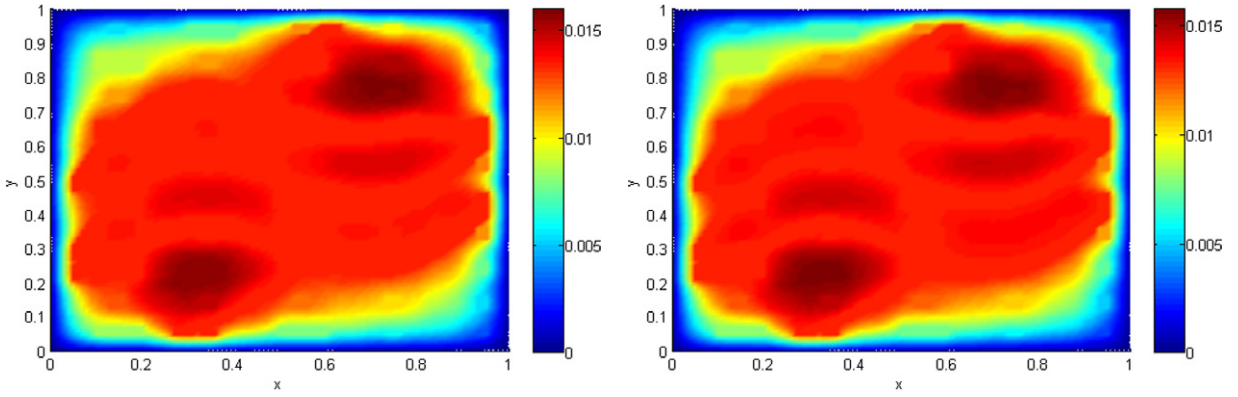
4.1. Single offline parameter

Example 4.1. We consider (1) along with the following offline and online parameters

$$\theta^{\text{off}} = \begin{cases} h^{\text{off}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{off}} = 10, \\ U_0^{\text{off}} = w_0, \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} h^{\text{on}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = 40, \\ U_0^{\text{on}} = w_0 * 0.5 \end{cases}$$

where the nonlinear function b_q is defined as $b_q(u, \mu) = e^{\mu u}$. Here, the source term does not need to be fixed for the method to work as we see below. We employ GMsFEM for the spatial discretization and the backward Euler method for time advancing as described in Section 3.1. Furthermore, we follow the steps given in Section 3.2 using three DEIM points ($L_0^{\text{local}} = 3$) per coarse region to approximate b_q in the offline stage. After generating the snapshots of the coarse-grid solutions using local DEIM, we compute the multiscale POD modes that are used in the online problem. We use $L_0^{\text{global}} = 5$ in the online stage to approximate b_q globally and then use the generated POD modes to



(a) Reference solution. (b) Approximate solution.

Fig. 2. Comparison between reference solution of the fine-scale problem with that obtained from the global–local multiscale approach.

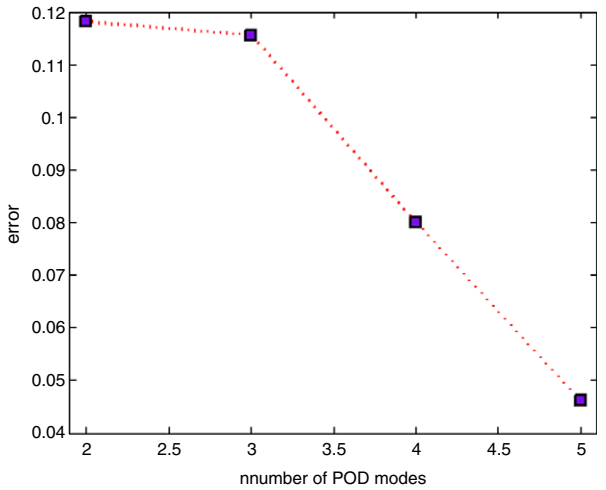


Fig. 3. Variations of the solution error with the number of POD modes.

approximate the coarse-scale solution. In Fig. 2, we compare the approximate solution obtained from the global–local nonlinear model reduction approach with the solution of the original system without using the DEIM technique to approximate the nonlinear function. A good approximation is observed in this figure, which demonstrates the capability of global–local nonlinear model reduction to reproduce accurately the fully resolved solution of a nonlinear PDE.

We have also considered a permeability field that is obtained by rotating the permeability field κ_q in Fig. 1 such that the three long channels are in the vertical direction. Our numerical results show similar accuracy and computational cost compared to the previous case (see Fig. 1). In general, we expect non-homogeneous boundary conditions to affect the numerical results.

The approximate solution shown in Fig. 2(b) is obtained using only two POD modes. As expected, increasing the number of POD modes used in the online stage yields a better approximation. That is, the error decreases as we increase the number of POD modes used as shown in Fig. 3. The error using two POD modes decreases slightly from 12% (at steady state) to 11.5% when using three POD modes. The decreasing trend is steeper when considering more POD modes. For instance, the use of 5 modes yields an error of 4.5%.

In order to illustrate the computational savings, we compute the time for solving the system of ordinary differential equations given in (7) with and without using the proposed method. We denote the time for solving the full system by T_{fine} and the time for solving the reduced system using global–local nonlinear model reduction by T_{GL}. Then, the

Table 1
Variation of the percentage of the simulation time corresponding to different number of local and global DEIM points. Here we use two POD modes.

L_0^{local}	L_0^{global}	R (%)
2	2	3.7832
2	3	3.3741
3	3	3.2093

Table 2
Variation of the percentage of the simulation time corresponding to different number of POD modes. Here we use $L_0^{\text{local}} = 2$ and $L_0^{\text{global}} = 3$.

POD modes	R (%)
2	3.3741
3	4.0387
4	4.9158
5	6.1414

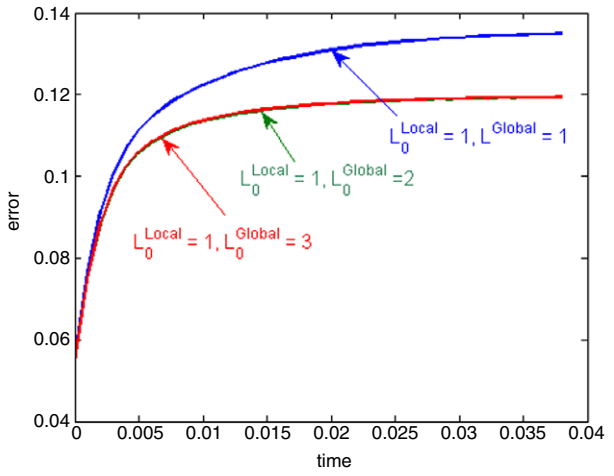
percentage of the simulation time is given by

$$R = \frac{T_{GL}}{T_{fine}} * 100. \tag{25}$$

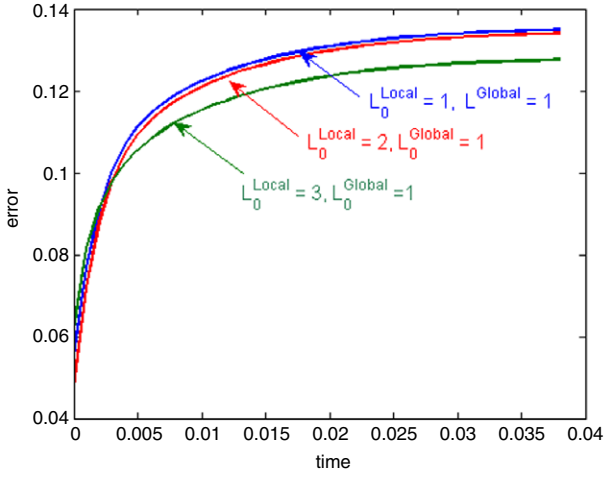
We compute R with respect to different number of DEIM points and POD modes and present the results in Tables 1 and 2, respectively. In Table 1, the first column shows the number of local DEIM points (L_0^{local}), the second column represents the number of global DEIM points (L_0^{global}), and the third column illustrates the percentage of the simulation time. Here two POD modes are used. As L_0^{local} and/or L_0^{global} increase, the percentage decreases accordingly. For example, R decreases from 3.7832% to 3.3741% by increasing L_0^{global} from two to three, and to 3.2093% by increasing both L_0^{local} and L_0^{global} from two to three. Decreasing R means that T_{GL}, time for solving the reduced system, decreases as we increase the number of DEIM points. Therefore, increasing the number of local and global DEIM points may speed up the simulation in addition to improving the accuracy as we see in the next example. In Table 2, the numbers of POD modes used for the global reduction are listed in the first column and the corresponding values of R are shown in the second column. In this case, we keep the number of local and global DEIM points constant and equal to two and three, respectively. Now, increasing the number of POD modes inversely affects the simulation speed-up. That is, increasing the number of POD modes increases the value of R which means T_{GL} is increasing and hence the speed-up of our simulation is decreasing. For example, R increases from 3.3741% when we use two POD modes to 4.0387% with three POD modes and keeps increasing as we increase the number of POD modes to be 6.1414% with five POD modes. Although, increasing the number of POD modes slows down the simulation, it improves the accuracy of the approximate solution (see Fig. 3). However, the following examples show the capability in terms of the accuracy of this method when using two POD modes for the global approximation.

Example 4.2. In this example, we use different numbers of local and global DEIM points, $L_0^{\text{local}} = \{1, 2, 3\}$ and $L_0^{\text{global}} = \{1, 2, 3\}$, to investigate how these numbers affect the error. As in Example 4.1, we consider $b_q(u, \mu) = e^{\mu u}$ and the following offline and online parameters:

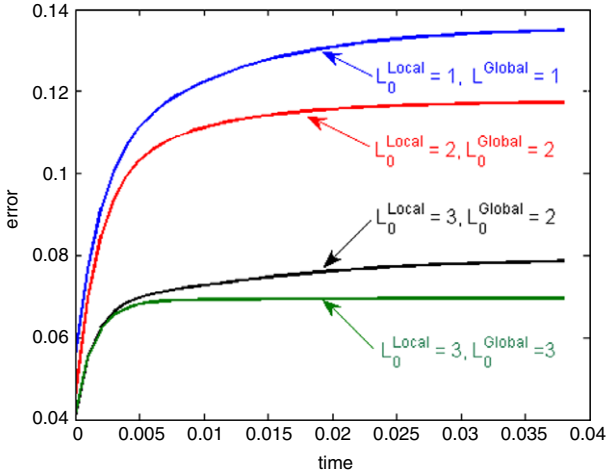
$$\theta^{\text{off}} = \begin{cases} h^{\text{off}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{off}} = 10, \\ U_0^{\text{off}} = w_0, \end{cases}$$



(a) Variations of global DEIM points.



(b) Variations of local DEIM points.



(c) Variations of both global and local DEIM points.

Fig. 4. Effect of the number of local and global DEIM points on the approximate solution accuracy.

$$\theta^{\text{on}} = \begin{cases} h^{\text{on}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = 40, \\ U_0^{\text{on}} = 0.5 w_0. \end{cases}$$

In Fig. 4(a), we plot the transient variations of the error while using different numbers of global DEIM points for a fixed number of local DEIM points equal to one. Increasing the number of global DEIM points from one to three results in a decrease in the error from 13% to 11% (at steady state). Further increases in the number of global DEIM points does not yield any improvement in the total error. This is due to the dominance of the local error. Fig. 4(b) shows the decreasing trend of the error as we increase the number of local DEIM points. In Fig. 4(c), we show the variations of the error with increasing the number of both local and global DEIM points. Increasing the number of DEIM points enables smaller error and then improves the solution accuracy. These examples show that the number of local and global DEIM points need to be chosen carefully to balance the local and global errors.

4.2. Multiple offline parameters

Example 4.3. In this example, we define the nonlinear function as $b_q(u, \mu) = e^{\mu(0.9+u)}$ and use $\mu_1^{\text{off}} = 2$ and $\mu_2^{\text{off}} = 5$, separately, in the offline problem to compute POD modes and DEIM points. We then combine these to use

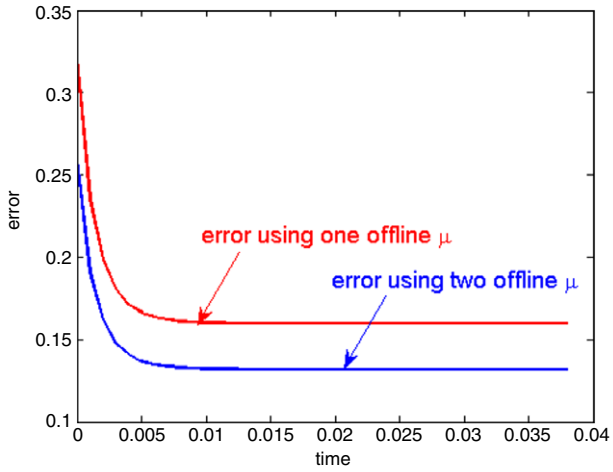


Fig. 5. Transient variations of the error (using different offline values of the parameter μ).

the total number of POD modes in the online problem with a different online value of μ ($\mu^{\text{on}} = 3$). In this example we keep the number of local and global DEIM points constant and equal to three (i.e., $L_0^{\text{local}} = L_0^{\text{global}} = 3$). Furthermore, we use different online initial conditions and source term. The following system parameters are considered.

$$\theta^{\text{off}} = \begin{cases} h^{\text{off}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu_1^{\text{off}} = 2, \\ \mu_2^{\text{off}} = 5, \\ U_0^{\text{off}} = w_0. \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} h^{\text{on}} = 1 + \sin(4\pi x_1)\sin(4\pi x_2), \\ \mu^{\text{on}} = 3, \\ U_0^{\text{on}} = 0. \end{cases}$$

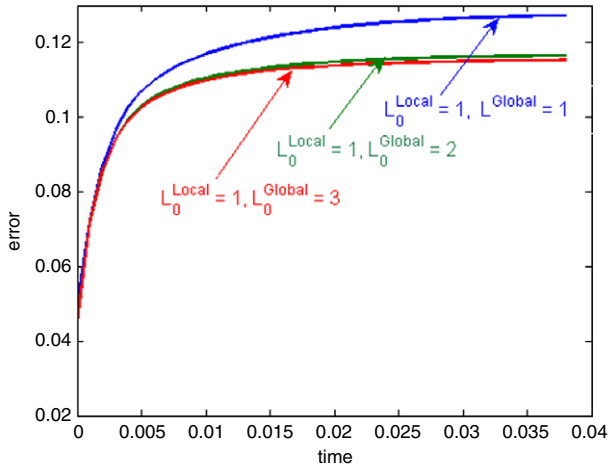
We show in Fig. 5 that the error decreases when combining two cases that correspond to different values of offline μ. For instance, the error when considering only one offline case is about 16% and it goes down to 13% when combining two cases with two different values of offline μ. Hence, using multiple parameter values in the offline stage improves the method's accuracy independently of the online parameters.

Example 4.4. Next, we consider the following parameters

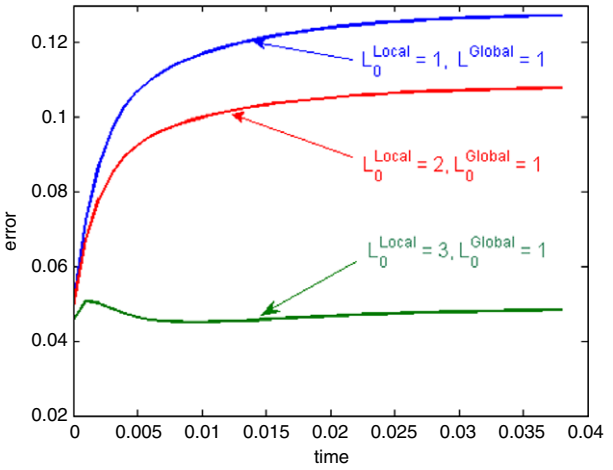
$$\theta^{\text{off}} = \begin{cases} h^{\text{off}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu_1^{\text{off}} = 10, \\ \mu_2^{\text{off}} = 40, \\ U_0^{\text{off}} = w_0, \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} h^{\text{on}} = 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = 24, \\ U_0^{\text{on}} = 0 \end{cases}$$

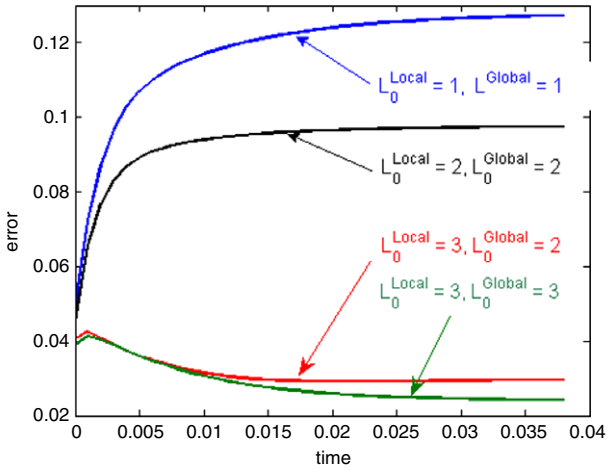
and the nonlinear function $b_q(u, \mu) = e^{\mu u}$. In this case, we use two offline values of μ while considering different numbers of local and global DEIM points. The effect of the number of local and global DEIM points on the error between the reference and approximate solutions when combining two cases that correspond to two different values of μ is shown in Fig. 6. Similar trends to those of Example 4.2 are observed. Increasing both local and global DEIM points improves the approximation to the solution. For instance, the error reduces from about 13% when using a local and a global DEIM point to 2% when using three local and global DEIM points. The error reduction in this case (when we use two offline μ) is bigger than the one we obtained when only using one offline μ value where the error decreased from 13% to 7% (see Fig. 4(c)). We conclude that using two offline μ values and increasing number



(a) Variations of global DEIM points.



(b) Variations of local DEIM points.



(c) Variations of both global and local DEIM points.

Fig. 6. Effect of the number of local and global DEIM points on the approximate solution accuracy (using two offline μ).

of local and global DEIM points yields a better approximation. Therefore, choosing the number of local and global DEIM points and the offline parameter values are the main factors to achieve high accuracy in the proposed method.

Example 4.5. In this example we consider the case with random values of the parameter μ that has a normal distribution with the mean 25 and variance 4. As in Example 4.3, we use different values of the offline parameter $\mu^{\text{off}} = \{10, 25, 39\}$, and compute the POD and DEIM modes. Further, we combine these modes to get the global POD and DEIM modes that we use in the online problem. In the online problem, we take uncorrelated random values of μ^{on} drawn from the above probability distribution. We rapidly compute the approximate solution and evaluate the relative error corresponding to each value of μ^{on}. Comparing the mean solutions of the fully-resolved model and the reduced model demonstrates the capability of the proposed method when random values of the parameter is employed in the nonlinear functional. Furthermore, we observe a good accuracy as shown from the error plotting in Fig. 7.

5. Conclusions

In this work, we present a global–local nonlinear model reduction approach to reduce the computational cost for solving high-contrast nonlinear parabolic PDEs. This is achieved through two main stages; offline and online. In the offline step, we use the generalized multiscale finite element method (GMsFEM) to represent the coarse-grid solutions

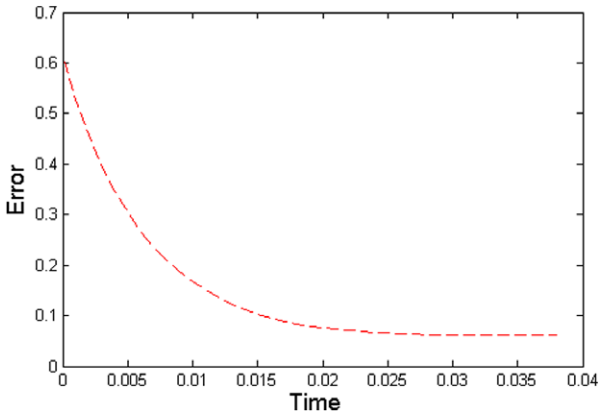


Fig. 7. Mean error of approximating the solution by using global–local multiscale approach with random values of the online parameter μ.

through applying the local discrete empirical interpolation method (DEIM) to approximate the nonlinear functions that arise in the residual and Jacobian. Using the snapshots of the coarse-grid solutions, we compute the proper orthogonal decomposition (POD) modes. In the online step, we project the governing equation on the space spanned by the POD modes and use the global DEIM to approximate the nonlinear functions. Although one can perform global model reduction independently of GMsFEM, the computations of the global modes can be very expensive. Combining both local and global mode reduction methods along with applying DEIM to inexpensively compute the nonlinear function can allow a substantial speed-up. We demonstrate the effectiveness of the proposed global–local nonlinear model reduction method on several examples of nonlinear multiscale PDEs that are solved using a fully-implicit time marching schemes. The results show the great potential of the proposed approach to reproduce the flow field with good accuracy while reducing significantly the size of the original problem. Increasing the number of the local and global modes to improve the accuracy of the approximate solution is examined. Furthermore, the robustness of proposed model reduction approach with respect to variations in initial conditions, permeability fields, nonlinear-function's parameters, and forcing terms is demonstrated.

Acknowledgments

YE's work is partially supported by the US Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-FG02-13ER26165, and the DoD Army ARO Project, and the grant FA9550-11-1-0341 from the Air Force Office of Scientific Research.

References

[1] I. Akhtar, A.H. Nayfeh, C.J. Ribbens, On the stability and extension of reduced-order Galerkin models in incompressible flows: a numerical study of vortex shedding, Theor. Comput. Fluid Dyn. 23 (3) (2009) 213–237.
[2] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Two-level discretizations of nonlinear closure models for proper orthogonal decomposition, J. Comput. Phys. 230 (1) (2011) 126–146.
[3] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, Comput. Methods Appl. Mech. Engrg. 237–240 (2012) 10–26.
[4] I. Akhtar, Z. Wang, J. Borggaard, T. Iliescu, A new closure strategy for proper orthogonal decomposition reduced-order models, J. Comput. Nonlinear Dyn. 7 (3) (2012) 034503.
[5] M. Ghommem, I. Akhtar, M.R. Hajj, A low–dimensional tool for predicting force decomposition coefficients for varying inflow conditions, Prog. Comput. Fluid Dyn. Int. J. 13 (6) (2013) 368–381.
[6] H.P. Bakewell, J.L. Lumley, Viscous sublayer and adjacent wall region in turbulent pipe flow, Phys. Fluids 10 (9) (1967) 1880–1889.
[7] L. Sirovich, Turbulence and the dynamics of coherent structures, Quart. Appl. Math. 45 (1987) 561–590.
[8] A.E. Deane, I.G. Kevrekidis, G.E. Karniadakis, S.A. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinder, Phys. Fluids A 3 (10) (1991) 2337–2354.
[9] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 53 (1993) 321–575.
[10] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, Cambridge, UK, 1996.

[11] A. Hay, J. Borggaard, I. Akhtar, D. Pelletier, Reduced-order models for parameter dependent geometries based on shape sensitivity analysis, J. Comput. Phys. 229 (2010) 1327–1352.
[12] A. Hay, I. Akhtar, J.T. Borggaard, On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions, Internat. J. Numer. Methods Fluids 68 (1) (2012) 122–134.
[13] T. Arbogast, Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems, SIAM J. Numer. Anal. 42 (2) (2004) 576–598 (electronic).
[14] C.-C. Chu, I.G. Graham, T.-Y. Hou, A new multiscale finite element method for high-contrast elliptic interface problems, Math. Comp. 79 (272) (2010) 1915–1955.
[15] Y. Efendiev, J. Galvis, X. Wu, Multiscale finite element methods for high-contrast problems using local spectral basis functions, J. Comput. Phys. 230 (2011) 937–955.
[16] Y. Efendiev, T. Hou, Multiscale Finite Element Methods: Theory and Applications, in: Surveys and Tutorials in the Applied Mathematical Sciences, vol. 4, Springer, New York, 2009.
[17] Y. Efendiev, T. Hou, V. Ginting, Multiscale finite element methods for nonlinear problems and their applications, Commun. Math. Sci. 2 (2004) 553–589.
[18] M. Ghommem, V. Calo, Y. Efendiev, Mode decomposition methods for flows in high-contrast porous media. Part II. Local-global approach, J. Comput. Phys. 253 226–238.
[19] J. Aarnes, On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, SIAM J. Multiscale Model. Simul. 2 (2004) 421–439.
[20] J. Aarnes, Y. Efendiev, Mixed multiscale finite element for stochastic porous media flows, SIAM J. Sci. Comput. 30 (5) (2008) 2319–2339.
[21] T. Arbogast, K. Boyd, Subgrid upscaling and mixed multiscale finite elements, SIAM J. Numer. Anal. 44 (3) (2006) 1150–1171 (electronic).
[22] O. Iliev, R. Lazarov, J. Willems, Variational multiscale finite element method for flows in highly porous media, Multiscale Model. Simul. 9 (4) (2011) 1350–1372.
[23] P. Jenny, S. Lee, H. Tchelepi, Multi-scale finite volume method for elliptic problems in subsurface flow simulation, J. Comput. Phys. 187 (2003) 47–67.
[24] T. Arbogast, G. Pencheva, M. Wheeler, I. Yotov, A multiscale Mortar mixed finite element method, Multiscale Model. Simul. 6 (1) (2007) 319–346.
[25] M. Wheeler, G. Xue, I. Yotov, A multiscale Mortar multipoint flux mixed finite element method, ESAIM Math. Model. Numer. Anal. 46 (4) (2012) 759–796.
[26] Y. Efendiev, J. Galvis, E. Gildin, Local–global multiscale model reduction for flows in highly heterogeneous media, J. Comput. Phys. 231 (2012) 8100–8113.
[27] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high contrast media, SIAM Multiscale Model. Simul. 8 (2010) 1461–1483.
[28] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high contrast media. Reduced dimensional coarse spaces, SIAM Multiscale Model. Simul. 8 (2010) 1621–1644.
[29] V.M. Calo, Y. Efendiev, J. Galvis, M. Ghommem, Multiscale empirical interpolation for solving nonlinear PDEs (using generalized multiscale finite element methods), J. Comput. Phys. 278 (2014) 204–220.
[30] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.
[31] Y. Efendiev, J. Galvis, T. Hou, Generalized multiscale finite element methods, J. Comput. Phys. 251 (2013) 116–135.