GLOBAL-LOCAL NONLINEAR MODEL REDUCTION FOR FLOWS IN HETEROGENEOUS POROUS MEDIA

A Dissertation

by

MANAL THAWAB A. ALOTIBI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Yalchin Efendiev
Co-Chair of Committee,	Eduardo Gildin
Committee Members,	Raytcho Lazarov
	Peter Howard
Head of Department,	Emil Straube

December 2016

Major Subject: Mathematics

ABSTRACT

Many problems in engineering and science are represented by nonlinear partial differential equations (PDEs) with high contrast parameters and multiple scales. Solving these equations involves expensive computational cost because the fine-grid needs to resolve smallest scales and high contrast. In such cases, reduced-order methods are often needed.

Reduced-order methods can be divided into local reduction methods and global reduction methods. Local reduced-order methods such as upscaling, Multiscale Finite Element Method (MsFEM) and Generalized Multiscale Finite Element Method (GMsFEM) divide the computational domain into coarse grids, where each grid contains small-scale heterogeneities and high contrast, and represents the computations for macroscopic simulations. In local model reduction, reduced-order models are constructed in each coarse region. Some known approaches, such as homogenization and numerical homogenization, are developed for problems with and without scale separation, respectively. Global reduced-order models, such as Proper Orthogonal Decomposition (POD), construct the reduced-order models via global finite element basis functions. These basis functions are constructed by solving many forward problems that can be expensive.

In this dissertation, we propose global-local model reduction methods. The idea of global-local model reduction methods is to approximate the global basis functions locally and adaptively. However, in the case of nonlinear systems, additional interpolation techniques are required, such as Discrete Empirical Interpolation Method (DEIM).

We propose a general global-local approach that uses the GMsFEM to construct

adaptive approximation for the global basis functions. The developments of these methods require adaptive offline and adaptive online reduced-order model strategies, which we pursue in this work.

We consider the applications to nonlinear flow problems, such as nonlinear Forchheimer flow. In this case, we construct multiscale basis functions for the velocity field following mixed GMsFEM. In addition, we present a local online adaptive method for the basis enrichment of the function space based on an error indicator depending on the local residual norm. Finally, we propose a global online adaptive method to add new global basis functions to the POD subspace. We use local error indicators and solve the global residual problem using the GMsFEM, with local online adaptation.

DEDICATION

This dissertation is dedicated to the memory of my father, Thawab Alotibi. He was behind this achievement with his support and encouragement. I miss him every day, but I believe that his soul is surrounding me. If he was still alive today, he would be proud and happy for making his dream into reality. I also dedicate my dissertation to my lovely daughter, Awrad.

ACKNOWLEDGMENTS

First and foremost, I cannot express enough thanks to my advisor, Prof. Yalchin Efendiev, for his continued support and encouragement. Without his effort, help, and valuable suggestions this research could not have been accomplished. I owe a deep sense of gratitude to Prof. Efendiev for improving my research skills and extending my relationship with many researchers working in the same field.

I would like to extend my deepest thanks to my co-advisor from the Petroleum Engineering Department, Prof. Eduardo Gildin, for his appreciated support. I feel sincere gratitude to Prof. Raytcho Lazarov, who put my foot on the first step at the very beginning of my study. I would like to extend my appreciation to Prof. Peter Howard, for his assistance, not only for me, but for all graduate students in our department.

I am grateful to Texas A&M University's Department of Mathematics, for helping their students to overcome the difficulties. I would also like to acknowledge the support from The Ministry of Education in Saudi Arabia and Saudi Arabian Cultural Mission to the U.S. I am particularly grateful for the funding provided by the government of Saudi Arabia to complete my study.

I thank profusely Prof. Victor Calo from King Abdullah University of Science and Technology (KAUST), Saudi Arabia. Prof. Calo with Prof. Efendiev offered a great opportunity to visit KAUST and participate in my first work with their collaborators who I would also like to thank: Prof. Juan Galvis and Dr. Mehdi Ghommem. Additionally, I would like to thank Prof. Eric Chung, for valuable suggestions and help. I wish to thank all my friends and classmates, especially Chak Lee.

Finally, to my loving and supportive husband, Faris Alotaibi: my deepest gratitude. I appreciate your patience and your management of our household activities throughout this experience. Your continued encouragement makes the completion of this work possible. I also will not forget to thank my mother, for her love and support. My warmest thanks must be to my father, who passed away before seeing this achievement.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a dissertation committee consisting of Professors Yachin Efendiev (advisor), Raytcho Lazarov, Peter Howard of the Department of Mathematics, and Eduardo Gildin (co-advisor) of the Department of Petroleum Engineering. All other work for the dissertation was completed independently by the student.

Graduate study was supported by a fellowship and funding from The Ministry of Education in Saudi Arabia and Saudi Arabian Cultural Mission to the U.S.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Nonlinear partial differential equations (PDEs), with complex heterogeneities, multiple scales and high contrast in media properties, represent an important class of problems with many relevant engineering and scientific applications in porous media. For example, in fractured media, fracture widths can be very small and fracture length can vary over a rich hierarchy of scales. The various sized fractures can connect and form a network with a complex geometry. Simulations of complex processes through these systems can be challenging because of multiple scales and high contrast. Similar examples appear when modeling other subsurface flows (e.g., through carbonates), where the permeability field has a high-contrast coefficient and heterogeneous distributions. Another application is the gas flow in shale formation. Shale is a rock with small organic and inorganic pores which add more complexities to the shale system. These rocks are fractured to produce natural gas. The resulting fractures can have complex geometries and simulations, these systems are prohibitively expensive.

In general, solving these types of problems requires resolving all scales and uncertainties. Using standard methods to resolve all the scales at once can be very expensive. Moreover, solving nonlinear flows, such as Forchheimer flow, has further difficulties due to the need of many iterations. 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. This complicates the simulation of a large number of degrees of freedom. As such, some model reduction techniques are required to reduce the degrees of freedom. This is a general objective of this dissertation. Figure 1.1 shows the general concept of reduced-ordered methods.

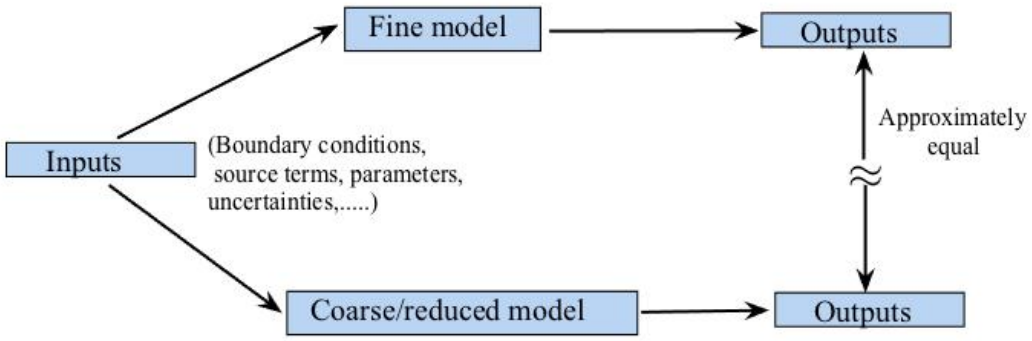


Figure 1.1: General concept of reduced-order modeling.

Typically, model reduction methods can be divided into two main categories: global reduced-order modeling techniques and local reduced-order modeling techniques. In this dissertation, we discuss and develop some global and local model reduction techniques for nonlinear flows in highly-heterogeneous media with high contrast.

Global model reduction methods. Global reduction techniques construct global finite element basis functions to solve the underlying PDEs. These basis functions are defined in the entire physical domain and use boundary conditions that are relevant to the physical problem. For example, in the applications, where the source terms are defined via injection and production wells, one uses various rates to solve the global problem. These rates are typically time-dependent. Therefore, in the offline stage, various time-dependent rates are defined to obtain snapshots, which are used for generating global basis functions. In other applications, one can use different physical boundary conditions or source terms to obtain these snapshots. From this set of snapshots, the global basis functions are extracted.

Several techniques, such as Proper Orthogonal Decomposition (POD), Dynamic

Mode Decomposition (DMD) and Balanced Truncation (BT), are used for global model reduction. The main purpose of these techniques is to reduce the dimension of the dynamical system by projecting the high-dimensional system into a lower-dimensional space using a set of orthonormal basis functions constructed from a sequence of snapshots [3, 51, 52, 4, 35]. In addition to order reduction, POD provides a powerful technique for extracting the most energetic modes from a linear or nonlinear dynamical process [12, 50, 25, 14, 40, 3, 51, 52, 4, 39, 38].

Local model reduction methods. The general concept of local model reduction is to reduce the dimension of the models within a certain acceptable accuracy instead of solving the full-resolved models. Typically, local model reduction techniques start with dividing the computational domain into some coarse grids, where each coarse grid is partitioned into connected fine grids (see Figure 1.2). Then, reduced-order models are constructed for each local coarse grid. Under this general concept, very successful approaches include homogenization and numerical homogenization have been proposed.

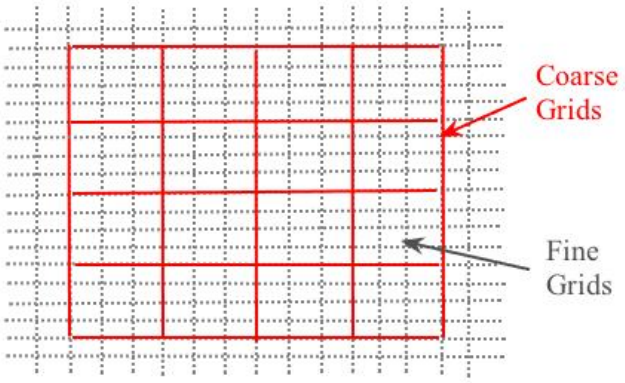


Figure 1.2: Coarse grids and fine grids.

Homogenization and numerical homogenization techniques [41, 19, 55, 26, 45, 34] are designed and used for problems with periodic heterogeneities or problems with scale separation. The scale separation assumption allows the computations to be localized. Homogenization and numerical homogenization techniques calculate effective properties by using some averaging methods. For example, in single-phase flow simulations, coarse-grid blocks or representative volumes are used to compute the effective properties of these blocks. The computation of effective properties is performed by solving local problems with some boundary conditions. Once the effective properties are computed, the global problem is solved with upscaled permeability fields. In nonlinear problems, one computes the effective nonlinear properties. For example, in Forchheimer flow [10], the effective properties consist of nonlinear functional relations between the fluxes and the pressure. These relations for each coarse block are computed by solving multiple local problems. In these homogenization and numerical homogenization approaches, one computes the effective properties by averaging the local solutions. In contrast, in multiscale methods, we use these local solutions to construct spaces for approximations of the solution space.

In multiscale methods, one constructs a coarse space that is spanned by a set of independently computed multiscale basis functions. This allows capturing the effects of small scales on a coarse grid. Then, the multiscale basis functions are coupled using global formulation in order to compute the reduced-order solution. Some multiscale methods have been introduced and used in various applications. Multiscale Finite Volume Method (MsFVM) (see, e.g., [44]) and Multiscale Finite Element Method (MsFEM) (e.g., [7, 20, 29, 30, 31, 36]) are particular examples of multiscale methods that are very popular. Both methods construct the multiscale basis functions by solving local problems with artificial boundary conditions. The MsFVM is commonly used in subsurface applications and it constructs the basis

functions on a dual coarse mesh to produce mass conservative solutions. Although the efficiency of the MsFEM has been proven for many applications, a systematic enrichment of the coarse space is required in some situations. More precisely, if a high level of accuracy is desired, adding additional basis functions to the construction of the coarse space is needed. The systematic enrichment of the coarse space is important for the convergence to the fine solution. For this purpose, the Generalized Multiscale Finite Element Method (GMsFEM) has been introduced by Efendiev et al. [28, 21] and used in many applications [27, 29, 32, 33].

As in many multiscale and model reduction techniques, the GMsFEM method divides the computation into two main stages: (1) the offline stage and (2) the online stage. In the offline stage, a small dimensional space is constructed via appropriate spectral decomposition. This space can be efficiently used in the online stage to construct multiscale basis functions. In the online stage, the multiscale basis functions are used to solve the forward model on a coarse-grid for many input parameters. Thus, this approach provides a substantial computational saving at the online stage. In this research, we apply the GMsFEM as an effective tool for our local model reduction. Further discussion of this method will be in Section 3.

We would like to remark that besides MsFEM, MsFVM and GMsFEM, some other numerical homogenization methods have been efficiently used, such as variational multiscale methods [42], heterogeneous multiscale methods [53, 48], mortar multiscale methods [9, 54], mixed multiscale finite element methods [1, 2, 8, 43], and mixed generalized multiscale finite element methods [22].

In a number of applications, the conservation of mass is important. For this reason, finite volume or mixed finite element discretization techniques are used to couple multiscale basis functions. In [22], the mixed GMsFEM was presented for solving a mixed framework of flow in heterogeneous media where the conservation of

mass is essential. The author considered the *linear* case and constructed the velocity field for the snapshots and offline spaces and used piecewise constant basis functions to approximate the pressure. In this dissertation, we extend this method to the *nonlinear* flow, considering nonlinear Forchheimer flow as an application.

Forchheimer flow is a particularly important example of physical processes that can be modeled by a nonlinear equation. Forchheimer flow appears in many applications related to fast flow in porous media. For example, fast flows can appear in many regions of the domains, such as near wells. The homogenization of flows in porous media typically considers low velocity flows that are governed by Stokes' equations. When the nonlinear effects due to fast flows become important, Navier-Stokes equations are solved at pore level. The homogenization of Navier-Stokes equations leads to a nonlinear Darcy flow. These issues are investigated in the literature [15, 37, 47]. One way to model the nonlinear effects at the pore level on the Darcy scale is via Forchheimer flow.

Heterogeneities in Forchheimer flow can occur due to variable permeability field [45]. For example, in recent years, using near well data, e.g., core data, engineers create increasingly complex and detailed geocellular models near the well. To reduce the computational complexity, some type of coarsening is needed. Recently, many approaches have been proposed for solving Forchheimer flow. For example, some coarsening and upscaling methods have been used to reduce the computational complexity for solving this equation. In [45], a special nonlinear upscaled Forchheimer form is used to simplify the calculation. In [34], a local-global upscaling technique is iteratively used. A new formulation for Forchheimer flow is used in [10]. The authors reduced the original system of equations for pressure and velocity to one nonlinear equation for pressure only. This equivalent form is obtained by using the monotone nonlinear permeability function of the gradient of pressure.

The global-local model reduction. Global model reduction methods derive reduced-order models through the construction of global finite element basis functions (global modes). These basis functions are expensive to compute. This is because the global techniques define the basis functions in the entire physical domain using fine-scale snapshots. Therefore, to reduce the computational cost of the global modes, one can approximate the global snapshots first and then compute the global modes for the resulting reduced-order model. This can be achieved by combining local and global techniques. This combination defines the concept of global-local model reduction. Moreover, one can use local model reduction to effectively update the global modes and design adaptive strategies, which we pursue in this dissertation.

Global-local model reduction techniques start with using some local model reduction such as the GMsFEM to construct appropriate number of basis functions in each coarse block. These basis functions are then used to solve the coarse-scale problem. Then, the global modes are constructed by applying a global reduction method such as POD/DMD on the coarse-scale solutions. With this approach, a significant reduction can be achieved while preserving the main flow feature due to the use of local modes that capture these features.

A combination of local and global model reduction schemes has been used for linear problem [36, 27]. A significant reduction in the computational complexity when solving linear parabolic PDE in [36] has been achieved by combining the concepts of the GMsFEM and POD and/or DMD. In [27], balanced truncation is used to perform global model reduction and is efficiently combined with the local model reduction tools introduced in [29].

Adaptive methods. Some offline and online adaptive methods have been proposed for model reduction. In the offline adaptive method (see [17, 23]), a better reduced solution space is obtained by adding some basis functions locally in the

coarse regions based on an a-posteriori error indicator which depends on the local residual norm. However, the resulting reduced system, that is constructed in the offline stage, is kept unchanged in the online stage. Therefore, the online solution still relies on the pre-computed offline information. Unlike the offline adaption, the online adaptive methods modify the reduced system during the online computations [17, 23, 5, 21, 24]. All these studies are considered for the linear elliptic flows.

In the case of dealing with reduced-order models of nonlinear systems, the aforementioned reduction and adaptive methods are limited by the full cost of the evaluation of the nonlinear coefficients. In this case, some interpolation methods can be used to avoid performing fine-grid computations. The Discrete Empirical Interpolation Method (DEIM) introduced in [18] is one of these interpolation methods that can be used for local and global approximations of the nonlinear functions.

Discrete empirical interpolation method. Basically, the discrete empirical interpolation method approximates a nonlinear function through interpolatory projection of a few selected global snapshots of the function. The idea is to use empirical snapshots and information of the nonlinear function in some selected components (in local or global regions) to represent such function over the entire domain. The selection of the empirical snapshots is based on a singular value decomposition (SVD) for the set of snapshots of the nonlinear function. The DEIM modes used for interpolation are the dominant eigenmodes of the SVD. More details of this method will be presented in the next section.

This dissertation. Our main contributions in this dissertation are: (1) We combine three methods: a local reduction method (GMsFEM), a global mode decomposition method (POD) and a discrete empirical interpolation technique to reduce the computational complexity associated with time dependent nonlinear flows in highly heterogeneous porous media. (2) Specifically, we use DEIM to approxi-

mate the nonlinear coefficients locally (at selected points in each coarse region) at the offline stage and globally (at selected points in the entire domain) at the online stage for which we refer to our method as a global-local nonlinear approach. (3) We extend the proposed method to the mixed discretization framework. (4) We apply the mixed-GMsFEM to solve the Forchheimer equation using local and global DEIM. (5) We propose adaptive strategies in local and global methods. In the following, we briefly introduce these contributions as they are organized in this dissertation.

In Section 3, we apply the proposed global-local reduction method to the nonlinear multiscale parabolic equations with nonlinear diffusion coefficients. We use the GMsFEM to introduce the coarse-scale solution for computing the global snapshots. The global snapshots are used by the POD method to construct global basis functions. We use this low dimensional global space constructed via local multiscale basis functions to solve the forward problem for different online parameter inputs. During these forward computations, the local basis functions are kept fixed and, thus, the computational cost for solving for global snapshots is inexpensive and performed by solving coarse-scale problems. Moreover, we employ DEIM to approximate the nonlinear coefficients locally and globally in order to circumvent the issue of the fine-scale computation cost of these coefficients. We study the effect of the number of the local and global DEIM modes on the accuracy of our approximation. Additionally, we apply this method using several offline parameter inputs. Then, the online space is generated from the combination of the POD modes computed for each parameter. Our numerical examples show that using several offline parameters improves the reduced-order solutions. We present extensive numerical studies.

In Section 4, we propose a mixed GMsFEM for nonlinear flow and apply it to solve nonlinear Forchheimer flow in highly heterogeneous porous media. In this extension, we generate local snapshots and local spectral problems for solving Forchheimer flow

in a mixed formulation. We consider the two term law form of Forchheimer equation in the case of slightly-compressible single-phase flows and write the resulting system in terms of a nonlinear flow equation for pressure with the nonlinearity depending on the pressure gradient.

The proposed approach constructs multiscale basis functions for the velocity field following the mixed GMsFEM as developed in [22] for the linear case. Then, we apply the POD method to the coarse-scale problem obtained by the mixed GMsFEM. As in Section 3, we combine the concepts of the mixed GMsFEM with POD and DEIM. We also study the convergence rate of the proposed reduced-order model analytically. Furthermore, we extend local online concepts and use them adaptively. The local online basis functions are added locally to improve the accuracy of the solution.

In Section 5, our objective is to design a global-local online approach. In many applications, the global snapshot spaces do not contain sufficient information for all time instants. Thus, at some instants, one needs to compute new snapshots and new global basis functions. In [13], residual-based POD modes were used to improve the POD subspace for Navier-Stokes equations. In [49], the POD modes for nonlinear dissipative systems are updated based on residual indicators. However, computing the global basis functions can be expensive and one needs inexpensive error indicators. In Section 5, we investigate these issues. We introduce a global online adaptive method that is used to add new global basis functions to the POD subspace. Toward this goal, we use a criterion to decide if adapting the POD subspace is required. We use local error indicators to decide. Using local error indicators instead of global error indicators reduces the computational time. At any time step, if the adaption is needed, the new POD basis function is computed by solving the global residual problem. We solve the global residual problem using the GMsFEM, with the local online adaptation. We emphasize that global online adaptivity is

performed by incorporating new data that becomes available in the online stage. We apply this method to nonlinear Forchheimer flow. Numerical results are presented, where we vary the source terms.

2. PRELIMINARIES*

2.1 Model problem

For the first model problem in Section 3, we will consider a time-dependent nonlinear flow governed by the following parabolic partial differential equation

$$\frac{\partial p}{\partial t} - \text{div}\left(\kappa(x; p, \mu)\nabla p\right) = f(x) \quad \text{in } D, \tag{2.1}$$

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

In Sections 4 and 5, we are interested in solving the following form for the Forchheimer equation

$$v + \beta(x)|v|v = -\frac{1}{\mu}\kappa(x)\nabla p, \tag{2.2}$$

where $v(x)$ and $p(x)$ represent the velocity field and the pressure distribution, respectively; x is the spatial variable in $\mathbb{R}^2$, $\kappa(x)$ is a given high-contrast heterogeneous permeability field, μ is the viscosity of the fluid, and β is the coefficient of inertial flow resistance. Here, $|.|$ denotes the Euclidean norm in $\mathbb{R}^2$, ($|v|^2 = \sum_{i=1}^{2} v_i^2$). If no confusion arises, we will use the same variable to denote the continuous and fine-grid variables.

*Parts of this section have been reprinted with permission from [6] Manal Alotaibi, Victor M Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Globallocal nonlinear model reduction for flows in heterogeneous porous media. Computer Methods in Applied Mechanics and Engineering, 292:pp. 122137, 2015.

In this section, we will consider the first model problem represented by Equation (2.1) to explain the general concepts of our proposed global-local reduction approach. A more detailed description for the model problem of Forchheimer flow (2.2) will be delegated to Section 4.

First, we outline the fine-grid solution technique for solving Equation (2.1). We partition the domain D into a set of finite elements (e.g., quadrilaterals or triangles) called coarse-grid blocks. We denote the coarse discretization by $\mathcal{T}^H$, where $H > 0$ is the coarse mesh size. In addition, we assume that $\mathcal{T}^h$ is a refinement of $\mathcal{T}^H$ by a connected union of fine-grid blocks. We use $\{y_i\}_{i=1}^{N_v}$ to denote the vertices of the coarse mesh, and define the neighborhood of a node y_i by

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H;\ y_i \in \overline{K_j}\}. \tag{2.3}$$

See Figure 2.1 for an illustration of a coarse neighborhood.

To solve (2.1) using the finite element method (FEM), we search for $p(x,t) \in V_h = \text{span}\{\phi_i\}_{i=1}^{N_f}$, where ϕ_i are the standard finite element basis functions defined on $\mathcal{T}^h$, and N_f denotes the number of fine nodes. The finite element discretization of (2.1) yields a system of ordinary differential equations given by

$$\mathrm{M}\dot{\mathrm{P}} + \mathrm{N}(\mathrm{P}, \mu) = \mathrm{F}, \tag{2.4}$$

where

$$\mathrm{P} = \begin{pmatrix} p_1(t) & p_2(t) & \cdots & p_{N_f}(t) \end{pmatrix}$$

is the vector collecting the pressure values at all nodes in the domain, $\mathrm{M} := [\mathrm{m}_{ij}] = \int_D \phi_i \phi_j$ is a mass matrix, $\mathrm{F} := [\mathrm{f}_i] = \int_D \phi_i f$. Using the offline basis functions, we

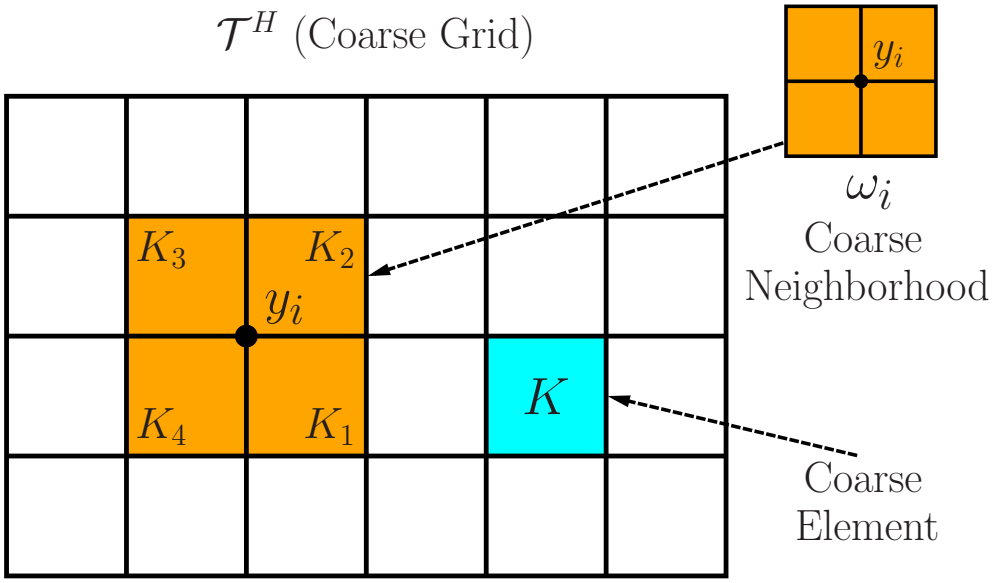


Figure 2.1: Illustration of the coarse grid ($\mathcal{T}^H$) and coarse neighborhood (ω_i).

can write (in a discrete form)

$$\kappa(x;p,\mu) = \sum_{q=1}^{Q} \kappa_q(x) b_q(p,\mu). \tag{2.5}$$

This results in

$$\mathrm{N}(\mathrm{P},\mu) \approx \sum_{q=1}^{Q} \mathrm{A}_q \Lambda_1^q(\mathrm{P},\mu)\mathrm{P},$$

where we have

$$\mathrm{A}_q := [\mathrm{a}_{ij}^q] = \int_D \kappa_q \nabla\phi_i \cdot \nabla\phi_j,$$
$$\Lambda_1^q(\mathrm{P},\mu) = \mathrm{diag}\begin{pmatrix} b_q(p_1,\mu) & b_q(p_2,\mu) & \cdots & b_q(p_{N_f},\mu) \end{pmatrix}.$$

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

$$\mathrm{P}^{n+1} + \Delta t\ \mathrm{M}^{-1}\mathrm{N}(\mathrm{P}^{n+1}, \mu) = \mathrm{P}^{n} + \Delta t\ \mathrm{M}^{-1}\mathrm{F}, \tag{2.6}$$

where Δt is the time-step size and the superscript n refers to the temporal level of the solution. We note that the square matrices are of size $N_f \times N_f$. Moreover, the nonlinear term $\mathrm{N}(\mathrm{P}, \mu)$ requires an iterative method to compute the nonlinear coefficients on the fine-grid. Therefore, solving Equation (2.6) involves an expensive computational cost. The main goal of this work is to derive a suitable reduced order model of size N_r such that $N_r \ll N_f$.

2.2 Generalized multiscale finite element method (local model reduction)

The main idea of the GMsFEM is to construct a small dimensional local solution space that can be used to generate an efficient and accurate approximation to the solution of a large system, which is the solution of (2.6) in our case. As in many multiscale and model reduction techniques, the GMsFEM divides the computation into offline stage and online stage (see [28]). Below we summarize the offline/online computational procedure in the following steps:

1. Offline computations:
   - Generation coarse grid.
   - Construction of snapshot space.
   - Construction of the offline space by performing dimension reduction in the space of local snapshots.
2. Online computations:

– Compute multiscale basis functions for each input parameter.

– For given forcing term and boundary conditions, solve a coarse-grid problem .

2.2.1 Offline stage

In the offline computation, we first construct the local snapshot space $V_{\text{snap}}^{\omega_i}$ for each coarse region ω_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. This space is used to construct the offline space in the following step via a spectral decomposition of the snapshot space. We denote each snapshot vector (listing the solution at each node in the domain) using a single index and create the following matrix

$$R_{\text{snap}} = \left[\psi_1^{\text{snap}}, \ldots, \psi_{M_{\text{snap}}}^{\text{snap}}\right],$$

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

The following step is to construct the offline space V_{off}. We want this space to be a subspace of the the snapshot space that can approximate with a sufficient accuracy any element of the original snapshot space. Toward this goal, we construct the offline space via an auxiliary spectral decomposition of the snapshot space. The spectral decomposition enables us to select the high-energy elements from the snapshot space by choosing the eigenvectors corresponding to the largest eigenvalues. These eigenvectors are then used to define the offline multiscale basis functions.

Moreover, in 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 parameter. Therefore, 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, where ν represents both p and μ in the following computations.

We consider the following eigenvalue problem in the space of snapshots,

$$A^{\text{off}}\Psi_k^{\text{off}} \;=\; \lambda_k^{\text{off}} S^{\text{off}}\Psi_k^{\text{off}}, \tag{2.7}$$

where

$$\begin{aligned} A^{\text{off}} &= [a_{mn}^{\text{off}}] = \int_{\omega_i} \overline{\kappa}(x,\nu)\nabla\psi_m^{\text{snap}}\cdot\nabla\psi_n^{\text{snap}} = R_{\text{snap}}^T\overline{A}R_{\text{snap}}, \\ S^{\text{off}} &= [s_{mn}^{\text{off}}] = \int_{\omega_i} \overline{\overline{\kappa}}(x,\nu)\psi_m^{\text{snap}}\psi_n^{\text{snap}} = R_{\text{snap}}^T\overline{S}R_{\text{snap}}. \end{aligned} \tag{2.8}$$

The weighted function $\tilde{\kappa}$ in the definition of S^{off} is defined by (see [28])

$$\tilde{\kappa} = \kappa\ H^2\sum_{i=1}^{N_v}|\nabla\hat{\chi}_i|^2.$$

Here, $\hat{\chi}_i$ is the standard multiscal partition of unity functions defined by

$$\begin{aligned} -\text{div}(\kappa(x;\nu)\nabla\hat{\chi}_i) = 0 \quad K\in\omega_i \\ \hat{\chi}_i = g_i \quad \text{on}\ \ \partial K, \end{aligned}$$

for all $K\in\omega_i$, where g_i is assumed to be linear. In Equations (2.8), the coefficients $\overline{\kappa}(x,\nu)$ and $\overline{\overline{\kappa}}(x,\nu)$ are parameter-averaged coefficients. The $\overline{A}$ and $\overline{S}$ denote the fine-scale matrices with parameter-averaged coefficients. The fine-scale stiffness matrix

A (similar for S) is constructed by integrating only on ω_i

$$A = [a_{mn}] = \int_{\omega_i} \kappa(x; p, \mu) \nabla \psi_m^{\text{snap}} \cdot \nabla \psi_n^{\text{snap}}. \tag{2.9}$$

Then, we choose the smallest M_{off} eigenvalues from Equation (2.7) and form the corresponding eigenvectors in the respective space of snapshots by setting $\psi_k^{\text{off}} = \sum_j \Psi_{kj}^{\text{off}} \psi_j^{\text{snap}}$ (for $k = 1, \ldots, M_{\text{off}}$), where Ψ_{kj}^{off} are the coordinates of the vector Ψ_k^{off}. We then create the offline matrix

$$R_{\text{off}} = \left[\psi_1^{\text{off}}, \ldots, \psi_{M_{\text{off}}}^{\text{off}}\right],$$

which maps the fine-grid vectors to the the vectors of coarse degrees of freedom. Using the transformation matrix $R_{\text{off}} \in \mathbb{R}^{N_f \times M_{\text{off}}}$, we can express Equation (2.6) as

$$\mathrm{P}_H^{n+1} + \Delta t (R_{\text{off}}^T \mathrm{M} R_{\text{off}})^{-1} R_{\text{off}}^T \mathrm{N}(R_{\text{off}}\, \mathrm{P}_H^{n+1}, \mu) = \mathrm{P}_H^n + \Delta t (R_{\text{off}}^T \mathrm{M} R_{\text{off}})^{-1} R_{\text{off}}^T \mathrm{H}, \tag{2.10}$$

where P_H denotes the coarse-scale solution (offline solution). The square matrices in (2.10) are of size $M_{\text{off}} \times M_{\text{off}}$, where $M_{\text{off}} \ll N_f$. Thus, Equation (2.10) is considered as a local model reduction of (2.6).

2.2.2 Online stage

In case of parameter dependent problem, we seek a subspace of the respective offline space such that it can approximate well any element of the offline space. At the online stage, the bilinear forms are *parameter-dependent*. The following eigenvalue

problems are posed in the reduced offline space:

$$A^{\text{on}}(\nu)\Psi_k^{\text{on}} \quad = \quad \lambda_k^{\text{on}} S^{\text{on}}(\nu)\Psi_k^{\text{on}}, \tag{2.11}$$

where

$$A^{\text{on}}(\nu) = [a^{\text{on}}(\nu)_{mn}] = \int_{\omega_i} \kappa(x,\nu)\nabla\psi_m^{\text{off}} \cdot \nabla\psi_n^{\text{off}} = R_{\text{off}}^T A(\nu) R_{\text{off}},$$

$$S^{\text{on}}(\nu) = [s_{mn}^{\text{on}}] = \int_{\omega_i} \widetilde{\kappa}(x,\nu)\psi_m^{\text{off}}\psi_n^{\text{off}} = R_{\text{off}}^T S(\nu) R_{\text{off}},$$

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 (2.11) and form the corresponding eigenvectors in the offline space by setting $\psi_k^{\text{on}} = \sum_j \Psi_{kj}^{\text{on}}\psi_j^{\text{off}}$ (for $k = 1, \ldots, M_{\text{on}}$), where Ψ_{kj}^{on} are the coordinates of the vector Ψ_k^{on}. Note that, if $\kappa(x,p)$ can be written as $\kappa(x,p) = k_0(x)\mathcal{N}(p)$, where $\mathcal{N}(p)$ is a nonlinear function of p, then one can use the parameter-independent case of the 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).

2.3 Proper orthogonal decomposition method (global model reduction)

In our proposed method, the main objective of the POD method is to construct a low dimensional solution space that can be used to solve the forward problem for any input parameter in the online stage. The first step of POD is to collect a sequence of n instantaneous coarse-solution p_H^i, where $p_H^i = p_H(t_i)$. Then, we define the space

$$\mathcal{V} = \text{span}\{p_H^1, p_H^2, \cdots, p_H^n\}.$$

We assume the time spacing between two consecutive snapshots in the above sequence is constant. Using this sequence of snapshots, we introduce the correlation matrix

$W = [w_{i,j}]_{n\times n} \in \mathbb{R}$ i.e.

$$W = \mathcal{V}^T\mathcal{V},$$

and then compute the POD modes $\{\varphi^{\text{POD}}\}$ by performing eigenvalues and eigenvectors of the correlation matrix W; that is

$$Wz_i = \sigma_i^2 z_i \quad \text{and} \qquad \varphi_i^{\text{POD}} = \frac{1}{\sigma_i}\mathcal{V}z_i.$$

We assume $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_d^2 > 0$, where d is the maximum number of nonzero eigenvalues. More details on using the POD modes for online space construction will be discussed in Sections 3 and 4.

2.4 Discrete emprical interpolation method (DEIM)

We give a quick review of the discrete empirical interpolation method (DEIM) as presented in [18]. First of all, the need of using this method to approximate the nonlinear function $\mathcal{N}(p)$ is coming from our attempt to solve (2.10) (the coarse-scale problem) instead of (2.6) (the fine-scale problem). In the offline stage, the nonlinear function $\mathcal{N}(p)$ needs to be evaluated with vectors $p = R_{\text{off}}p_H$. In the online stage, $\mathcal{N}(p)$ is evaluated with vectors of the form $p = R_{\text{off}}R_{\text{on}}p_{\text{on}}$. In both cases the vectors p are the downscaling of solutions obtained by a reduced order model. For ease of the notation, we use the same variable to denote fine-grid and the continuous solution. This leads us to look for an approximation of $\mathcal{N}(p)$ at a reduced cost. We use discrete empirical interpolation method (DEIM) for the local approximation of the nonlinear functions in the offline stage and for the global approximation in the online stage. 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 in some

locations (or components).

Let $\mathcal{N}(\tau) \in \mathbb{R}^n$ 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 ζ_i is a basis that represents the solution space and $l \ll n_s$. This leads us to look for an approximation of $\mathcal{N}(\tau)$ at a reduced cost. To perform a reduced order approximation of $\mathcal{N}(\tau)$, we first define a reduced dimensional space for $\mathcal{N}(\tau)$. That is, we would like to find m basis vectors (where m is much smaller than n), ψ_1^*,..., ψ_m^*, such that we can write

$$\mathcal{N}(\tau) \approx \Psi^* d(\tau), \tag{2.12}$$

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

The goal of DEIM is to find $d(\tau)$ using only a few rows of (2.12). In general, one can define $d(\tau)$'s using m rows of (2.12) and invert a reduced system to compute $d(\tau)$. This can be formalized using the matrix $\mathcal{P}$

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

where $e_{\wp_i} = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathbb{R}^n$ is the $\wp_i^{\text{th}}$ column of the identity matrix $\mathrm{I}_n \in \mathbb{R}^{n \times n}$ for $i = 1, \cdots, m$. Multiplying Equation (2.12) by $\mathcal{P}^T$ and assuming that the matrix $\mathcal{P}^T \Psi^*$ is nonsingular, we obtain

$$\mathcal{N}(\tau) \approx \tilde{\mathcal{N}}(\tau) = \Psi^* d(\tau) = \Psi^* (\mathcal{P}^T \Psi^*)^{-1} \mathcal{P}^T \mathcal{N}(\tau). \tag{2.13}$$

To summarize, approximating the nonlinear function $\mathcal{N}(\tau)$, as given by Equa-

tion (2.13), requires the following:

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

To determine the projection basis $\Psi^* = (\psi_1^*, \cdots, \psi_m^*)$, we collect function evaluations in an $n \times n_s$ matrix $\mathrm{B} = [\mathcal{N}(\tau_1), \cdots, \mathcal{N}(\tau_{n_s})]$ and employ POD to select the most energetic modes. This selection uses the eigenvalue decomposition of the square matrix $\mathrm{B}^T\mathrm{B}$ (left singular values) and form the important modes using the dominant eigenvalues. These modes are used as the projection basis in the approximation given by Equation (2.12). In Equation (2.13), the term $\Psi^*(\mathcal{P}^T\Psi^*)^{-1} \in \mathbb{R}^{n\times m}$ is computed once and stored. The $d(\tau)$ is computed using the values of the function $\mathcal{N}(\tau)$ at m points with the indices $\wp_1, \cdots, \wp_m$ identified using the DEIM algorithm in Table (2.1).

The computational saving is due to the resulting fewer evaluations of $\mathcal{N}(\tau)$. 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.

DEIM Algorithm [18]:

Input: The projection basis matrix $\Psi^* = (\psi^*_1, \cdots, \psi^*_m)$
obtained by applying POD on a sequence of n_s function evaluations.

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

1: Set $[|\rho|, \wp_1] = \max\{|\psi^*_1|\}$

2: Set $\Psi^* = [\psi^*_1]$, $\mathcal{P} = [e_{\wp_1}]$, and $\overrightarrow{\wp} = (\wp_1)$

3: for $k = 2, ..., m$ do

- Solve $(\mathcal{P}^T\Psi^*)w = \mathcal{P}^T\psi^*_k$ for some w.

- Compute $r = \psi^*_k - \Psi^* w$

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

- Set $\Psi^* = [\Psi^* \ \psi^*_k]$, $\mathcal{P} = [\mathcal{P} \ e_{\wp_k}]$, and $\overrightarrow{\wp} = \begin{pmatrix} \overrightarrow{\wp} \\ \wp_k \end{pmatrix}$

end for

Table 2.1: Algorithm of the multiscale discrete empirical interpolation method.

3. GLOBAL-LOCAL METHOD FOR DIFFUSION MODEL FOR NONLINEAR PARABOLIC EQUATION*

In this section, we combine discrete empirical interpolation techniques, global mode decomposition methods, and local multiscale methods such as the Generalized Multiscale Finite Element Method (GMsFEM). We use this combination 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 the coarse-grid solutions. Computing the GMsFEM solution involves calculating the residual and the Jacobian on the 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 enables a significant reduction in the flow problem size while accurately capturing the behavior of the fully-resolved solutions.

3.1 Model problem

We consider the model problem presented in Section 2.1 by the following equation

$$\frac{\partial p}{\partial t} - \text{div}\left(\kappa(x; p, \mu)\nabla p\right) = f(x) \quad \text{in } D, \tag{3.1}$$

subject to some boundary and initial conditions. We recall that D is a bounded computational domain with high contrast permeability κ, μ is an input parameter, $p = p(t, x)$ denotes the pressure and f is the forcing term.

*Parts of this section have been reprinted with permission from [6] Manal Alotaibi, Victor M Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Globallocal nonlinear model reduction for flows in heterogeneous porous media. Computer Methods in Applied Mechanics and Engineering, 292:pp. 122137, 2015.

3.2 Finite element discretization and Newton method

Following the fine-scale and coarse-scale discretization of the computational domain D as described in Section 2.1, we will get the following system of ordinary differential equations

$$\mathrm{M}\dot{\mathrm{P}} + \mathrm{N}(\mathrm{P}, \mu) = \mathrm{F}. \tag{3.2}$$

Recall that P is the vector of pressure values at all fine nodes and F is the right-hand-side vector obtained by discretization. And using the offline basis functions, we can write (in a discrete form)

$$\kappa(x; p, \mu) = \sum_{q=1}^{Q} \kappa_q(x) b_q(p, \mu). \tag{3.3}$$

This results in

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

where we have

$$\begin{aligned} \mathrm{A}_q := [\mathrm{a}_{ij}^q] &= \int_D \kappa_q \nabla\phi_i \cdot \nabla\phi_j, \\ \Lambda_1^q(\mathrm{P}, \mu) &= \mathrm{diag}\begin{pmatrix} b_q(p_1, \mu) & b_q(p_2, \mu) & \cdots & b_q(p_{N_f}, \mu) \end{pmatrix}, \end{aligned}$$

and ϕ_i are piecewise linear basis functions defined on a fine triangulation of D.

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

$$\mathrm{P}^{n+1} + \Delta t\ \mathrm{M}^{-1}\mathrm{N}(\mathrm{P}^{n+1}, \mu) = \mathrm{P}^n + \Delta t\ \mathrm{M}^{-1}\mathrm{F}, \tag{3.4}$$

where Δt is the time-step size and the superscript n refers to the temporal level of

the solution. The residual is defined as:

$$\mathrm{R}(\mathrm{P}^{n+1}) = \mathrm{P}^{n+1} - \mathrm{P}^{n} + \Delta t\ \mathrm{M}^{-1}\mathrm{N}(\mathrm{P}^{n+1}, \mu) - \Delta t\ \mathrm{M}^{-1}\mathrm{F}$$

with derivative (Jacobian)

$$\begin{aligned}\mathrm{J}(\mathrm{P}^{n+1}) = D_p\mathrm{R}(\mathrm{P}^{n+1}) &= I + \Delta t\ \mathrm{M}^{-1} D_p\mathrm{N}(\mathrm{P}^{n+1}) \\ &= I + \sum_{q=1}^{Q} \Delta t\ \mathrm{M}^{-1}\mathrm{A}_q\Lambda_1^q(\mathrm{P}^{n+1}) + \sum_{q=1}^{Q} \Delta t\ \mathrm{M}^{-1}\mathrm{A}_q\Lambda_2^q(\mathrm{P}^{n+1}),\end{aligned}$$

where

$$\Lambda_2^q(\mathrm{P}, \mu) = \mathrm{diag}\left(\begin{array}{cccc} p_1 \frac{\partial b_q(p_1,\mu)}{\partial p} & p_2 \frac{\partial b_q(p_2,\mu)}{\partial p} & \cdots & p_{N_f} \frac{\partial b_q(p_{N_f},\mu)}{\partial p} \end{array} \right),$$

and D_p is the multi-variate gradient operator defined as $\left[D_p\mathrm{R}(\mathrm{P})\right]_{ij} = \partial \mathrm{R}_i / \partial \mathrm{P}_j$.

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

$$\begin{aligned}\mathrm{J}(\mathrm{P}_{(k)}^{n+1})\Delta\mathrm{P}_{(k)}^{n+1} &= -\Big(\mathrm{P}_{(k)}^{n+1} - \mathrm{P}^{n} + \Delta t\ \mathrm{M}^{-1}\mathrm{N}(\mathrm{P}_{(k)}^{n+1}, \mu) - \Delta t\ \mathrm{M}^{-1}\mathrm{F}\Big), \\ \mathrm{P}_{(k+1)}^{n+1} &= \mathrm{P}_{(k)}^{n+1} + \Delta\mathrm{P}_{(k)}^{n+1},\end{aligned}$$

where the initial guess is $\mathrm{P}_{(0)}^{n+1} = \mathrm{P}^{n}$ and k is the iteration counter. The above iterations are repeatedly applied until $\parallel \Delta\mathrm{P}_{(k)}^{n+1} \parallel$ is less than a specific tolerance. Therefore, the fine-grid discretization involves many solves with a large matrices of size $N_f \times N_f$ that may become prohibitively expensive to handle numerically. In the following we introduce a global-local model reduction technique that accurately approximates the solution using fewer number of degrees of freedom.

3.3 Local multiscale model reduction

In our proposed technique, we consider the GMsFEM as the tool for local model reduction. Recall from Section 2.2 that the GMsFEM divides the computations into offline stage and online stage. In the offline stage we construct snapshot space and offline multiscale space. The snapshot space, essentially, is the space containing the extensive set of basis functions which are the solutions of local problems. The offline space is constructed via spectral decomposition of the local snapshot space. In parameter-dependent problems, the online multiscale space is then constructed for each input parameter in the online stage. We remark that in our simulations, we use $Q = 1$ in Equation (3.3) 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 multiscale space is the same as the offline multiscale space). We refer to Section 2.2 for more details.

Suppose the offline multiscale basis functions $\{\psi_i^{\text{off}}\}_{i=1}^{M_{\text{off}}}$ are obtained following the GMsFEM framework. We then create the offline matrix

$$R_{\text{off}} = \left[\psi_1^{\text{off}}, \ldots, \psi_{M_{\text{off}}}^{\text{off}}\right],$$

which maps the fine-grid vectors to the vectors of coarse degrees of freedom. Using the transformation matrix $R_{\text{off}} \in \mathbb{R}^{N_f \times M_{\text{off}}}$, we use the solution expansion $p = R_{\text{off}}\, p_H$ and employ the multiscale framework to obtain a set of M_{off} ordinary differential equations that constitute a reduced-order model; that is,

$$\dot{P_H} = -(R_{\text{off}}^T \mathrm{M} R_{\text{off}})^{-1} R_{\text{off}}^T \mathrm{N}(R_{\text{off}}\, P_H, \mu) + (R_{\text{off}}^T \mathrm{M} R_{\text{off}})^{-1} R_{\text{off}}^T \mathrm{F}, \tag{3.5}$$

where P_H denotes the vector of the coarse-scale solutions (offline solutions). Thus, the original problem with N_f degrees of freedom is reduced to a dynamical system with M_{off} dimensions where $M_{\text{off}} \ll N_f$. Again, we use the same notations for continuous and fine-grid discrete variables for the notation simplicity.

The nonlinear term $(R_{\text{off}}^T \mathrm{M} R_{\text{off}})^{-1} R_{\text{off}}^T \mathrm{N}(R_{\text{off}}\ \mathrm{P}_H, \mu)$ in the reduced-order model, given by Equation (3.5), 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 the multiscale DEIM as described in Section 2.4.

To solve the reduced system (3.5), we employ the backward Euler scheme; that is,

$$P_H^{n+1} + \Delta t\ \widetilde{\mathrm{M}}^{-1} \widetilde{\mathrm{N}}(P_H^{n+1}) = P_H^n + \Delta t\ \widetilde{\mathrm{M}}^{-1} \widetilde{\mathrm{F}}, \tag{3.6}$$

where

$$\widetilde{\mathrm{M}} = R_{\text{off}}^T \mathrm{M} R_{\text{off}}, \quad \widetilde{\mathrm{N}}(P_H) = R_{\text{off}}^T \mathrm{N}(R_{\text{off}}\ P_H, \mu), \quad \text{and} \quad \widetilde{\mathrm{F}} = R_{\text{off}}^T \mathrm{F}.$$

We let

$$\widetilde{\mathrm{R}}(P_H^{n+1}) = P_H^{n+1} - P_H^n + \Delta t\ \widetilde{\mathrm{M}}^{-1} \widetilde{\mathrm{N}}(P_H^{n+1}) - \Delta t\ \widetilde{\mathrm{M}}^{-1} \widetilde{\mathrm{F}}, \tag{3.7}$$

with derivative

$$\begin{aligned}\widetilde{\mathrm{J}}(P_H^{n+1}) = D_p\widetilde{\mathrm{R}}(P_H^{n+1}) &= I + \Delta t\ \widetilde{\mathrm{M}}^{-1} D_p\widetilde{\mathrm{N}}(P_H^{n+1}) \\ &= I + \sum_{q=1}^{Q} \Delta t\ \widetilde{\mathrm{M}}^{-1} R_{\text{off}}^T \mathrm{A}_q \Lambda_1^q(R_{\text{off}} P_H^{n+1}, \mu) R_{\text{off}} \\ &\quad + \sum_{q=1}^{Q} \Delta t\ \widetilde{\mathrm{M}}^{-1} R_{\text{off}}^T \mathrm{A}_q \Lambda_2^q(R_{\text{off}} P_H^{n+1}, \mu) R_{\text{off}}.\end{aligned}$$

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

$$\widetilde{\mathrm{J}}(P_{H_{(k)}}^{n+1})\Delta P_{H_{(k)}}^{n+1} = -\Big(P_{H_{(k)}}^{n+1} - P_H^n + \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{N}}(P_{H_{(k)}}^{n+1}) - \Delta t\ \widetilde{\mathrm{M}}^{-1}\widetilde{\mathrm{F}}\Big), \tag{3.8}$$

$$P_{H_{(k+1)}}^{n+1} = P_{H_{(k)}}^{n+1} + \Delta P_{H_{(k)}}^{n+1}, \tag{3.9}$$

where the initial guess is $P_{H_{(0)}}^{n+1} = P_H^n$. The above iterations are repeated until $\parallel \Delta P_{H_{(k)}}^{n+1} \parallel$ is less than a specific tolerance. Furthermore, we use the 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.

3.4 Global-local nonlinear model reduction approach

We denote the offline parameters by θ^{off} which include samples of the right-hand side $f(x)$ denoted by f_i^{off}, samples of μ denoted by μ_i^{off}, and samples of initial conditions denoted by $P_{0,i}^{\text{off}}$. Similarly, the online parameter set is denoted by θ^{on} and includes the online source term f^{on}, the online μ (μ^{on}), and the online initial conditions P_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 $\theta^{\text{off}} = \{\theta_i^{\text{off}}\} = \{f_i^{\text{off}}, \mu_i^{\text{off}}, P_{0,i}^{\text{off}}\}$.
- 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 Section 2.4, 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:

$$\mathbf{Z}^{N_t} = \{P_H^1, P_H^2, \cdots, P_H^{N_t}\}, \tag{3.10}$$

  where N_t is the number of snapshots and M_{off} is the size of the column vectors P_H^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 $\psi_i^{\text{on}} := \varphi_i^{\text{POD}}$; that is, we let

$$p_H(x,t) \approx \tilde{p}_H(x,t) = \sum_{i=1}^{M_{\text{on}}} \alpha_i(t)\psi_i^{\text{on}}(x), \tag{3.11}$$

  or in a matrix form

$$P_H^n \approx \tilde{P}_H^n = R_{\text{on}}\alpha^n, \tag{3.12}$$

  where $R_{\text{on}} = \left[\psi_1^{\text{on}}, \cdots, \psi_{M_{\text{on}}}^{\text{on}}\right]$.

- **Online Stage**

The online stage includes the following steps:

- Given online $\theta^{\text{on}} = \{f^{\text{on}}, \mu^{\text{on}}, P_0^{\text{on}}\}$.
- Use the solution expansion given by (3.11) and project the governing equation of the coarse-scale problem onto the space formed by the modes to obtain a set of M_{on} ordinary differential equations that constitute a reduced-order model; that is,

$$\dot{\alpha} = -(\underbrace{R_{\text{on}}^T R_{\text{off}}^T \text{M} R_{\text{off}} R_{\text{on}}}_{M_{\text{on}} \times M_{\text{on}}})^{-1} \underbrace{R_{\text{on}}^T}_{M_{\text{on}} \times M_{\text{off}}} \underbrace{R_{\text{off}}^T}_{M_{\text{off}} \times N_f} \underbrace{\text{N}(R_{\text{off}} R_{\text{on}} \alpha)}_{N_f \times 1} + (R_{\text{on}}^T R_{\text{off}}^T \text{M} R_{\text{off}} R_{\text{on}})^{-1} R_{\text{on}}^T R_{\text{off}}^T \text{F}. \tag{3.13}$$

- 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

$$\widehat{\text{J}}(\alpha_{(k)}^{n+1}) \Delta\alpha_{(k)}^{n+1} = -\Big(\alpha_{(k)}^{n+1} - \alpha^n + \Delta t\, \widehat{\text{M}}^{-1} \widehat{\text{N}}(\alpha_{(k)}^{n+1}) - \Delta t\, \widehat{\text{M}}^{-1} \widehat{\text{F}}\Big), \tag{3.14}$$

where

$$\widehat{\text{M}} = R_{\text{on}}^T R_{\text{off}}^T \text{M} R_{\text{off}} R_{\text{on}} = R_{\text{on}}^T \tilde{\text{M}} R_{\text{on}}, \qquad \widehat{\text{F}} = R_{\text{on}}^T R_{\text{off}}^T \text{F} = R_{\text{on}}^T \tilde{\text{F}},$$

$$\widehat{\text{N}} = R_{\text{on}}^T R_{\text{off}}^T \text{N} = R_{\text{on}}^T \tilde{\text{N}}.$$

Then

$$\alpha_{(k+1)}^{n+1} = \alpha_{(k)}^{n+1} - (\widehat{\text{J}}(\alpha_{(k)}^{n+1}))^{-1} \Big(\alpha_{(k)}^{n+1} - \alpha^n + \Delta t\, \widehat{\text{M}}^{-1} \widehat{\text{N}}(\alpha_{(k)}^{n+1}) - \Delta t\, \widehat{\text{M}}^{-1} \widehat{\text{F}}\Big).$$

Thus, the original problem with N_f degrees of freedom is reduced to a dynamical system with M_{on} dimensions where $M_{\text{on}} \ll M_{\text{off}} \ll N_f$.

– Use global DEIM to approximate the nonlinear functions that appear in the residual and Jacobian. To do so, we write the nonlinear function $\mathbf{N}(R_{\text{off}}R_{\text{on}}\alpha)$ in Equation (3.13) as

$$\mathrm{N}(R_{\text{off}}R_{\text{on}}\alpha) \approx \Psi^* d, \tag{3.15}$$

where $\Psi^* = [\psi_1^*, ..., \psi_{L_0^{\text{global}}}^*]$ 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 $\mathrm{N}(R_{\text{off}}P_H)$ computed offline and employ the POD technique to select the most energetic modes (see Section 2.4). The coefficient vector d is computed using the values of the function N at L_0^{global} global points.

– Use the solution expansion given by (3.11) in terms of POD modes to approximate the coarse-scale solution and then use the operator matrix R_{off} to downscale the approximate solution and evaluate the flow field on the fine-grid.

3.5 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 P_H is given by $R_{\text{off}} P_H$, which is a fine-grid vector.

In the following numerical examples, we consider (3.1) with specified boundary and initial conditions, where the permeability coefficient and the forcing term are given by

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

Here, κ_q represents the permeability field with high-conductivity channels as shown in Figure 3.1 and $b_q(p, \mu)$ is defined later for each example. We use the GMsFEM along with the Newton method to discretize (3.1). 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 (3.1), as defined in (2.9), we introduce the relative energy error as

$$\|E\|_{\mathrm{A}} = \sqrt{\frac{(\mathrm{P} - \widetilde{\mathrm{P}})^T \mathrm{A} (\mathrm{P} - \widetilde{\mathrm{P}})}{\mathrm{P}^T \mathrm{A} \mathrm{P}}}. \tag{3.16}$$

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

$$-\mathrm{div}\,(\kappa_q(x) \nabla w_0) = f(x) \quad \text{in} \quad D, \tag{3.17}$$

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

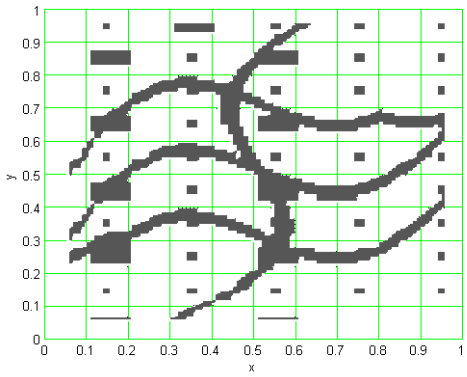


Figure 3.1: Permeability field that model high conductivity channels within a homogeneous domain. The minimum (background) conductivity is taken to be $\kappa_{min} = 1$, and the high conductivity (gray regions) with value of $\kappa_{max} = \eta$ ($\eta = 10^6$).

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.

3.5.1 Single offline parameter

Example 3.5.1. We consider (3.1) along with the following offline and online parameters

$$\theta^{\text{off}} = \begin{cases} f^{\text{off}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{off}} = & 10, \\ P_0^{\text{off}} = & w_0, \end{cases}$$

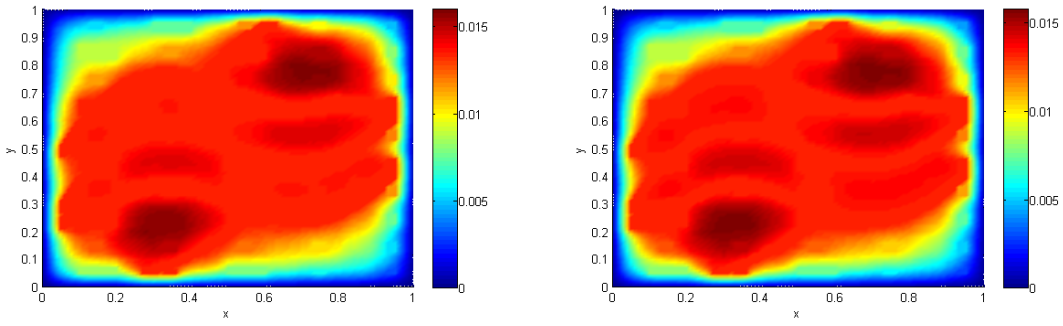
$$\theta^{\text{on}} = \begin{cases} f^{\text{on}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = & 40, \\ P_0^{\text{on}} = & w_0 * 0.5, \end{cases}$$

where the nonlinear function b_q is defined as $b_q(p, \mu) = e^{\mu p}$. Here, the source term does not need to be fixed for the method to work as we see below. We employ the GMsFEM for the spatial discretization and the backward Euler method for time advancing as described in Section 3.2. Furthermore, we follow the steps given in Section 3.4 using three DEIM points ($L_0^{local} = 3$) 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 $L_0^{global} = 5$ in the online stage to approximate b_q globally and then use the generated POD modes to approximate the coarse-scale solution. In Figure 3.2, 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 Figure 3.1 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 Figure 3.1). In general, we expect non-homogeneous boundary conditions to affect the numerical results.

The approximate solution shown in Figure 3.2(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 Figure 3.3. 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



(a) Reference Solution (b) Approximate Solution

Figure 3.2: Comparison between reference solution of the fine-scale problem with that obtained from the global-local multiscale approach.

savings, we compute the time for solving the system of ordinary differential equations

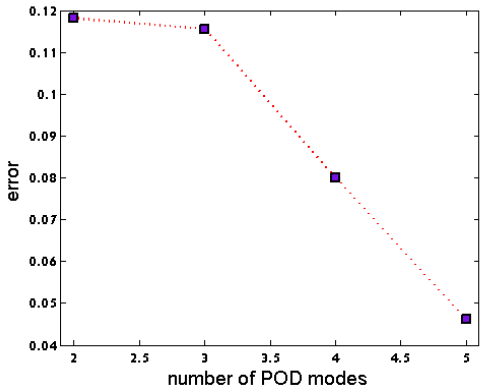


Figure 3.3: Variations of the solution error with the number of POD modes.

given in (3.2) 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 percentage of the simulation time is given by

$$PST = \frac{T_{GL}}{T_{fine}} * 100. \tag{3.18}$$

We compute PST with respect to different number of DEIM points and POD modes and present the results in Tables 3.1 and 3.2, respectively. In Table 3.1, 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, PST 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 PST 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 3.2, the numbers of POD modes used for the global reduction are listed in the first column and the corresponding values of PST 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 PST which means T_{GL} is increasing and hence the speed-up of our simulation is decreasing. For example, PST 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 Figure 3.3). However, the following examples show the capability in terms of the accuracy of this method when using two POD modes for the global reduction.

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

Table 3.1: Variation of the percentage of the simulation time corresponding to different number of local and global DEIM points. Here we use two POD modes.

Example 3.5.2. In this example, we use different numbers of local and global DEIM points, $L_0^{local} = \{1, 2, 3\}$ and $L_0^{global} = \{1, 2, 3\}$, to investigate how these numbers affect the error. As in Example 3.5.1, we consider $b_q(p, \mu) = e^{\mu p}$ and the following

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

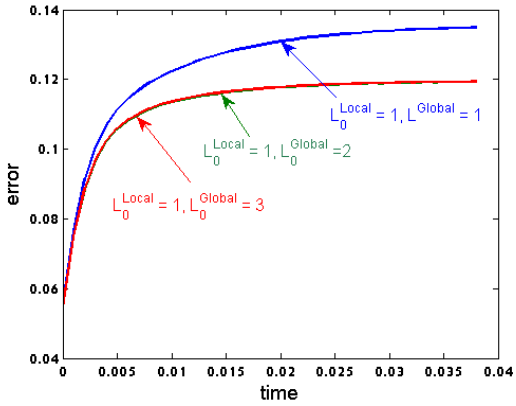
Table 3.2: Variation of the percentage of the simulation time corresponding to different number of POD modes. $L_0^{local} = 2$ and $L_0^{global} = 3$.

offline and online parameters:

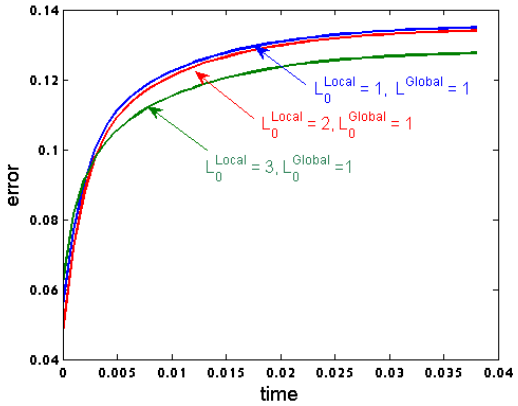
$$\theta^{\text{off}} = \begin{cases} f^{\text{off}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{off}} = & 10, \\ P_0^{\text{off}} = & w_0, \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} f^{\text{on}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = & 40, \\ P_0^{\text{on}} = & 0.5w_0. \end{cases}$$

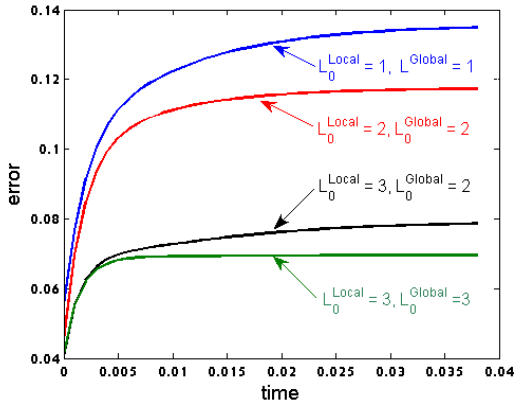
In Figure 3.4(a), we plot the transient variations of the error while using different numbers of global DEIM points for a fixed number of local DEIM points equal to one. Increasing the number of global DEIM points from one to three results in a decrease in the error from 13% to 11% (at steady state). 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. Figure 3.4(b) shows the decreasing trend of the error as we increase the number of local DEIM points. In Figure 3.4(c), we show the variations of the error with increasing the number of both local and



(a) Variations of global DEIM points



(b) Variations of local DEIM points



(c) Variations of both global and local DEIM points

Figure 3.4: Effect of the number of local and global DEIM points on the approximate solution accuracy.

global DEIM points. Increasing the number of DEIM points enables a 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.

3.5.2 Multiple offline parameters

Example 3.5.3. In this example, we define the nonlinear function as $b_q(p,\mu) = e^{\mu(0.9+p)}$ 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 modes to use 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., $L_0^{local} = L_0^{global} = 3$). Furthermore, we use different online initial conditions and source term. The following system parameters are considered.

$$\theta^{\text{off}} = \begin{cases} f^{\text{off}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu_1^{\text{off}} = & 2, \\ \mu_2^{\text{off}} = & 5, \\ P_0^{\text{off}} = & w_0. \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} f^{\text{on}} = & 1 + \sin(4\pi x_1)\sin(4\pi x_2), \\ \mu^{\text{on}} = & 3, \\ P_0^{\text{on}} = & 0. \end{cases}$$

We show in Figure 3.5 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.

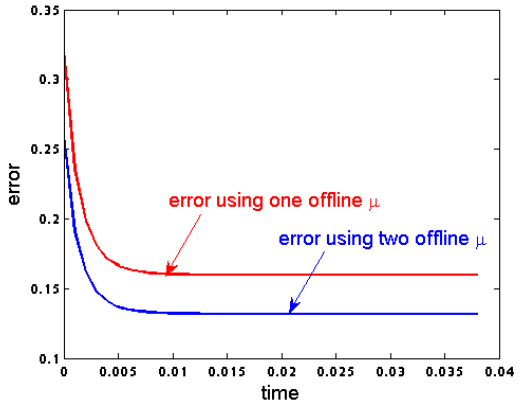


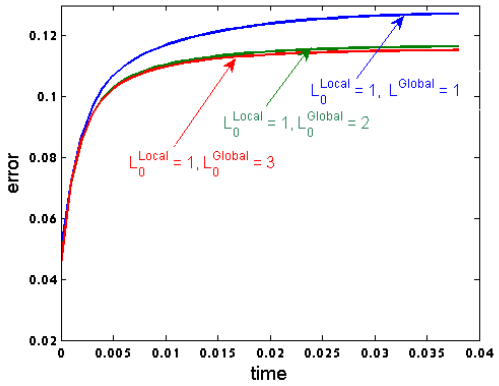
Figure 3.5: Transient variations of the error (using different offline values of the parameter μ).

Example 3.5.4. Next, we consider the following parameters

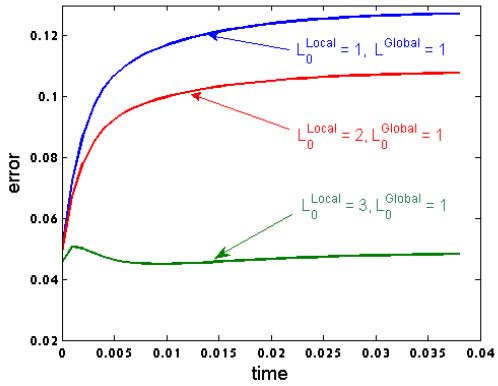
$$\theta^{\text{off}} = \begin{cases} f^{\text{off}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu_1^{\text{off}} = & 10, \\ \mu_2^{\text{off}} = & 40, \\ P_0^{\text{off}} = & w_0, \end{cases}$$

$$\theta^{\text{on}} = \begin{cases} f^{\text{on}} = & 1 + \sin(2\pi x_1)\sin(2\pi x_2), \\ \mu^{\text{on}} = & 24, \\ P_0^{\text{on}} = & 0, \end{cases}$$

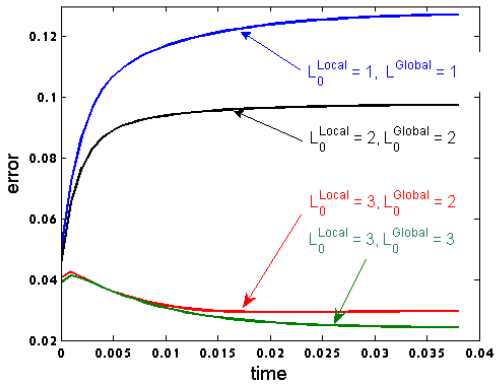
and the nonlinear function $b_q(p, \mu) = e^{\mu p}$. In this case, we use two offline values of μ while considering different numbers of local and global DEIM points. The effect of the number of local and global DEIM points on the error between the reference and



(a) Variations of global DEIM points



(b) Variations of local DEIM points



(c) Variations of both global and local DEIM points

Figure 3.6: Effect of the number of local and global DEIM points on the approximate solution accuracy (using two offline μ).

approximate solutions when combining two cases that correspond to two different values of μ is shown in Figure 3.6. Similar trends to those of Example 3.5.2 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 Figure 3.4(c)). We conclude that using two offline μ values and increasing number 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 3.5.5. 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 3.5.3, 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 Figure 3.7.

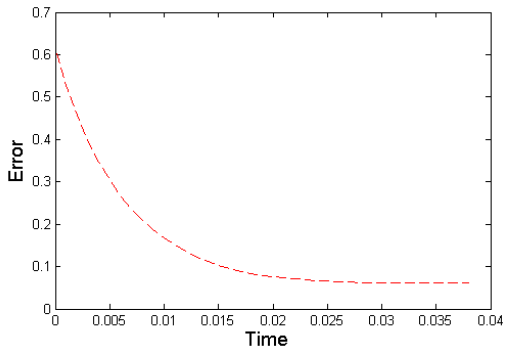


Figure 3.7: Mean error of approximating the solution by using global-local multiscale approach with random values of the online parameter μ.

4. GLOBAL-LOCAL MODEL REDUCTION FOR HETEROGENEOUS FORCHHEIMER FLOW

In this section, we propose a mixed Generalized Multiscale Finite Element Method (GMsFEM) for solving nonlinear Forchheimer flow in highly heterogeneous porous media. We consider the two term law form of the Forchheimer equation in the case of slightly-compressible single-phase flows. We write the resulting system in terms of a nonlinear flow equation for pressure when the nonlinearity depends on the pressure gradient. The proposed approach constructs multiscale basis functions for the velocity field following the mixed GMsFEM as developed in [22] for the linear case. To reduce the computational cost resulting from solving nonlinear system, we combine the mixed GMsFEM with Discrete Empirical Interpolation Method (DEIM) to compute the nonlinear coefficients in some selected degrees of freedom at each coarse domain. In addition, a global reduction method such as Proper Orthogonal Decomposition (POD) is used to construct the online space to be used to solve the reduced-order system for different inputs. We present numerical and theoretical results to show that in addition to speeding up the simulation we can achieve good accuracy with a few basis functions per coarse edge. Moreover, we present an online adaptive method for basis enrichment of the multiscale space based on an error indicator depending on the local residual norm. We use this enrichment method to add some online local multiscale basis functions at some fixed time steps. Our numerical experiments show that these additional multiscale basis functions will reduce the error if we start with a sufficient number of initial offline basis functions.

4.1 Problem statement

We apply our developed global-local reduction method in Section 3 to a nonlinear parabolic PDE in a mixed form. In particular, our interest is in solving the Forchheimer equation presented in Section 2 by Equation (2.2) as following

$$v + \beta(x)|v|v = -\frac{1}{\mu}\kappa(x)\nabla p. \tag{4.1}$$

Recall that $v(x)$ and $p(x)$ are the velocity field and the pressure distribution, respectively; $\kappa(x)$ is a given high-contrast heterogeneous permeability field, μ is the viscosity of the fluid, and β is the coefficient of inertial flow resistance. For simplicity from now on we assume the viscosity $\mu = 1$ and define

$$\mathcal{N}(v) := 1 + \beta(x)|v|, \tag{4.2}$$

then we write the nonlinear form of Darcy's law as

$$\kappa^{-1}\mathcal{N}(v)v + \nabla p = 0.$$

And the equation describing the conservation of mass is given by

$$\phi(x)\frac{\partial \rho}{\partial t} = -\operatorname{div}(\rho v) + f(x),$$

where ρ is the fluid density, ϕ is the rock porosity, and $f(x)$ is an external mass flow rate. By scaling the time variable we write continuity equation as following

$$\frac{\partial \rho}{\partial t} = -\operatorname{div}(\rho v) + f(x). \tag{4.3}$$

For slightly compressible fluid (such as the compressible liquid), the equation of state has the form, see [10],

$$\rho(p) = \rho_0 e^{\gamma p}, \tag{4.4}$$

where ρ_0 is the density at the reference pressure p_0 and γ is the inverse of the compressibility constant. Substituting (4.4) in Equation (4.3), yields

$$\frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} = -\rho \ \mathrm{div}(v) - \frac{d\rho}{dp} v \cdot \nabla p + f(x).$$

Then,

$$\frac{\partial p}{\partial t} = -\rho \frac{\partial p}{\partial \rho} \ \mathrm{div}(v) - v \cdot \nabla p + \frac{\partial p}{\partial \rho} f(x). \tag{4.5}$$

From (4.4), we have

$$\frac{\partial \rho}{\partial p} = \rho_0 \gamma e^{\gamma p} = \gamma \rho.$$

Hence,

$$\frac{\partial p}{\partial \rho} = \frac{1}{\gamma \rho}. \tag{4.6}$$

Substituting (4.6) into (4.5), gives

$$\frac{\partial p}{\partial t} = -\frac{1}{\gamma} \ \mathrm{div}(v) - v \cdot \nabla p + \frac{1}{\gamma \rho} f(x). \tag{4.7}$$

Since for most slightly compressible fluids in porous media γ is very small, we drop the second term of the RHS and write

$$\gamma \frac{\partial p}{\partial t} + \ \mathrm{div}(v) = \frac{1}{\rho} f.$$

Note that $\frac{1}{\rho} = \frac{1}{\rho_0 e^{\gamma p}} \approx \frac{1}{\rho_0}$. Since γ and ρ_0 are constants, we write

$$f = \frac{1}{\rho} f \qquad \text{and} \qquad \frac{\partial p}{\partial t} = \gamma \frac{\partial p}{\partial t}.$$

Then, the mixed formulation describing the fluid flow is:

$$\begin{aligned} \kappa^{-1} \mathcal{N}(v)\ v + \nabla p &= 0 && \text{in } D \times J. \\ \frac{\partial p}{\partial t} + \mathrm{div}(v) &= f(x) && \text{in } D \times J, \\ p(x,0) &= p_0 && \text{in } D, \\ v.\nu &= g && \text{on } \partial D \times J, \end{aligned} \tag{4.8}$$

where $D \subset \mathbb{R}^2$ is a bounded convex domain with boundary ∂D, $J = [0, T]$ is the computational time interval, and ν is the outward unit-normal vector on ∂D. We remark that Picard iteration method will be used as the nonlinear solver for the above system along with backward Euler implicit time-stepping scheme.

To describe the general formulation, we introduce the following notations and basic definitions to be used throughout the following sections.

- $\mathcal{T}^H$, as defined in Section 2, denotes the usual conforming partition of the computational domain D into finite elements (triangles, quadrilaterals, tetrahedrals, etc.), called coarse-grid blocks, where $H > 0$ is the coarse mesh size.

- $\mathcal{T}^h$ is a refinement of $\mathcal{T}^H$ by a connected union of of fine-grid blocks, which are

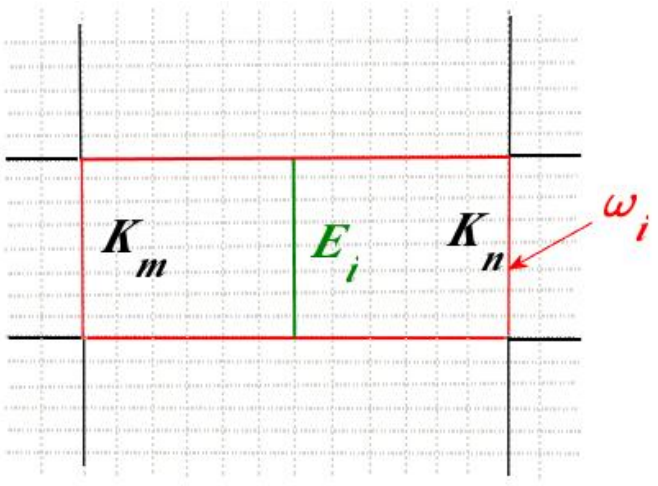


Figure 4.1: Coarse neighborhood $\omega_i = K_m \cup K_n$ corresponding to the coarse edge E_i.

conforming across coarse-grid edges.

- $\mathcal{E}^H := \bigcup_{i=1}^{N_e} \{E_i\}$ (where N_e is the number of coarse edges E_i) denotes the set of all edges of the coarse mesh $\mathcal{T}^H$, and $\mathcal{E}_0^H$ denotes the set of all interior coarse edges.

- We define the coarse neighborhood ω_i corresponding to the coarse edge E_i as the union of all coarse-grid blocks having the edge E_i, namely,

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H; \quad E_i \in \partial K_j\}. \tag{4.9}$$

  See Figure 4.1 for an illustration.

- Each coarse edge E_i can be written as a union of fine-grid edges, namely, $E_i = \bigcup_{j=1}^{J_i} e_j$, where J_i is the total number of fine-grid edges on E_i and e_j denotes a fine-grid edge.

- For a scalar function $q \in L^2(\Omega)$, where Ω is a given open set, the L^2 norm is

$$\|q\|_\Omega^2 = \langle q, q \rangle_\Omega = \int_\Omega q^2.$$

- For a vector field v, we define the weighted L^2 norm

$$\|v\|_{\kappa^{-1},\Omega}^2 = \langle v, v \rangle_{\kappa^{-1},\Omega} = \int_\Omega \kappa^{-1} v^2.$$

- For the time interval $J = [0, T]$ we define the norm

$$\|v\|_{(J;\kappa^{-1},\Omega)}^2 = \int_0^T \|v(., s)\|_{\kappa^{-1},\Omega}^2 ds.$$

- We define the Sobolev space

$$V = H(\mathrm{div}; \Omega; \kappa^{-1}) := \{v : v \in L^2(\Omega)^2, \ \ \mathrm{div}(v) \in L^2(\Omega)\},$$

and is equipped with the norm

$$\|v\|_V^2 = \|v\|_{H(\mathrm{div};\Omega;\kappa^{-1})}^2 = \|v\|_{\kappa^{-1},\Omega}^2 + \|\mathrm{div}(v)\|_{L^2(\Omega)}^2.$$

If $\kappa = 1$, we write

$$H(\mathrm{div}; \Omega) = H(\mathrm{div}; \Omega; \kappa^{-1}),$$

and

$$\|v\|_{H(\mathrm{div};\Omega)}^2 = \|v\|_{L^2(\Omega)}^2 + \|\mathrm{div}(v)\|_{L^2(\Omega)}^2.$$

4.2 Finite element discretization

Let $V_h \times Q_h$ be the standard lowest-order-Raviart-Thomas space for the semi-discrete approximation of (4.8) on the fine-grid $\mathcal{T}^h$. Then, the fine-grid solution $(v_h, p_h) \in V_h \times Q_h$ satisfies

$$\begin{aligned} \langle \mathcal{N}(v_h)\ v_h,\ w_h \rangle_{\kappa^{-1},D} - \langle \mathrm{div}(w_h),\ p_h \rangle_D = 0, \qquad & \forall w_h \in V_h^0, \\ \langle \frac{\partial p_h}{\partial t},\ q_h \rangle_D + \langle \mathrm{div}(v_h),\ q_h \rangle_D = \langle f, q_h \rangle_D, \qquad & \forall q_h \in Q_h, \end{aligned} \tag{4.10}$$

and $p_h(0) = p_{0,h}$, $v_h.\ \nu = g_h$ on $\partial D \times J$, where $p_{0,h}$ and g_h are approximations to p_0 and g in Q_h and V_h, respectively. We define $V_h^0 = V_h \cap \{v \in V_h : v.\ \nu = 0 \ \text{ on } \partial D\}$.

Suppose $V_h = \mathrm{span}\{\phi_i\}_{i=1}^{M_v}$ and $Q_h = \mathrm{span}\{\chi_i\}_{i=1}^{M_p}$. Then, we write

$$v_h = \sum_{i=1}^{M_v} \hat{V}_i \phi_i \quad \text{and} \quad p_h = \sum_{i=1}^{M_p} \hat{P}_i \chi_i,$$

Define $V = (\hat{V}_1, \cdots, \hat{V}_{M_v})^T$ and $P = (\hat{P}_1, \cdots, \hat{P}_{M_p})^T$, the fully discrete system of the above problem using backward Euler scheme with time step size Δt can be written in the matrix form as the following:

$$\begin{aligned} N_{\text{fine}}(V^{n+1})\ \ V^{n+1} - \ B_{\text{fine}}^T P^{n+1} = 0, \\ M_{\text{fine}} \frac{P^{n+1} - P^n}{\Delta t} + B_{\text{fine}} V^{n+1} = F, \end{aligned} \tag{4.11}$$

where the superscripts n indicate the time step. Here, $N_{\text{fine}}(V) := M_\kappa \Gamma(V)$, where

$$M_\kappa = [m_{\kappa_{i,j}}] = \int_D \kappa^{-1} \phi_i \phi_j, \qquad \Gamma(V) = \mathrm{diag}\left(\mathcal{N}(\hat{V}_1), \cdots, \mathcal{N}(\hat{V}_{M_v})\right).$$

The matrices B_{fine}, M_{fine}, and the vector F are respectively defined as:

$$B_{\text{fine}} = [b_{i,j}] = \int_D \text{div}(\phi_i)\chi_j, \quad M_{\text{fine}} = [m_{i,j}] = \int_D \chi_i\chi_j, \quad F = [f_j] = \int_D f(x)\chi_j.$$

The above scheme involves, at each time step, the following iteration:

$$N_{\text{fine}}(V_{(k)}^{n+1}) \;\; V_{(k+1)}^{n+1} - \;\; B_{\text{fine}}^T P_{(k+1)}^{n+1} = 0, \tag{4.12}$$

$$M_{\text{fine}} \frac{P_{(k+1)}^{n+1} - P_{(k+1)}^{n}}{\Delta t} + B_{\text{fine}} V_{(k+1)}^{n+1} = F, \tag{4.13}$$

where the initial guess is $V_{(0)}^{n+1} = V^n$, the subscripts k denote the respective Picard iteration level. From (4.12) we have:

$$V_{(k+1)}^{n+1} = \left(N_{\text{fine}}(V_{(k)}^{n+1}) \right)^{-1} \;\; B_{\text{fine}}^T P_{(k+1)}^{n+1}, \tag{4.14}$$

substitute (4.14) into (4.13),

$$M_{\text{fine}} \frac{P_{(k+1)}^{n+1} - P_{(k+1)}^{n}}{\Delta t} + B_{\text{fine}} \left(N_{\text{fine}}(V_{(k)}^{n+1}) \right)^{-1} \;\; B_{\text{fine}}^T P_{(k+1)}^{n+1} = F.$$

$$\underbrace{M_{\text{fine}} + \Delta t \left(B_{\text{fine}} \left(N_{\text{fine}}(V_{(k)}^{n+1}) \right)^{-1} \;\; B_{\text{fine}}^T \right)}_{A_{(k)}^{n+1}} P_{(k+1)}^{n+1} = \underbrace{\Delta t \; F + M_{\text{fine}} P_{(k+1)}^{n}}_{X_{(k+1)}^{n}}.$$

Thus, we have the matrix equation

$$P_{(k+1)}^{n+1} = \left(A_{(k)}^{n+1} \right)^{-1} \;\; X_{(k+1)}^{n}.$$

The solution of this equation is then used to compute $V_{(k+1)}^{n+1}$ in (4.14). The iterations

are repeatedly applied until $\|V_{(k+1)}^{n+1} - V_{(k)}^{n+1}\|$ is less than a specific tolerance. Therefore, solving the nonlinear system (4.8) in the fine-grid using the standard FEM is very expensive. Our aim in the following sections is to perform an efficient reduced-order model to approximate the solution in the coarse-grid. Moreover, when solving the reduced-order model, we will use DEIM to approximate the nonlinear function, $\mathcal{N}(v)$, at some selected components in the local and global domains. This will reduce the computational cost for evaluating the nonlinear coefficients.

We remark that the fine-grid solutions (v_h, p_h) are considered as the reference solutions in our convergence analysis in Section 4.4.

4.3 Global-local reduction method

In our presented method, we employ the mixed generalized multiscale finite element method described in [22] to represent the coarse-grid solutions with multiscale basis functions for the velocity field v. In our case, computing the coarse-grid solutions involves computing the nonlinear function defined in (4.2) on the fine-grid. To avoid the cost of this computation, we apply the discrete empirical interpolation method locally at each coarse region in the offline stage. Using the snapshots of the coarse-grid solutions of the velocity field, we compute the proper orthogonal decomposition (POD) modes and define the online space to be the linear span of these modes. We project the governing equations on the online space and apply the global DEIM to approximate the nonlinear coefficients globally (i.e. at selected points in the whole domain). This is a general overview of the global-local model reduction method for a mixed type problem. In the following, we give a detailed description for this method.

4.3.1 Mixed generalized multiscale finite element method

In this section, we introduce the mixed GMsFEM as presented in [22] for the linear case. In the mixed GMsFEM, we approximate the pressure p in the space of piecewise constant functions with respect to the coarse-grid $\mathcal{T}^H$, denoted by $Q_H \subset Q_h$. For the velocity field v, we define a set of multiscale basis functions for each coarse edge $E_i \in \mathcal{E}^H$. These basis functions are supported in the coarse neighborhood ω_i corresponding to the coarse edge E_i. Specifically, to obtain a basis function for a coarse edge E_i, we use the terminology introduced in [28], where the construction of the multisacle basis functions is done in the offline computation stage. We first construct a snapshot space V_{snap} by solving a local elliptic problem in the coarse neighborhood ω_i with a given normal velocity on E_i and zero normal velocity on the boundary $\partial\omega_i$. The solutions of local elliptic problems with all possible boundary conditions up to the fine-grid resolution form an extensive set of basis functions for the snapshot space V_{snap}. We will present a space reduction technique which provides a systematic way to select the dominant modes in the snapshot space. This technique is based on a carefully designed local spectral problem giving a rapidly decaying error. The selected dominant modes are the multiscale basis functions for the velocity field. Let $\{\Psi_j\}$ be the set of multiscale basis functions for the edge E_i. We define the multiscale space (the offline space) for the velocity field v as the linear span of all local basis functions which is denoted as

$$V_{\text{off}} = \bigoplus_{\mathcal{E}^H} \{\Psi_i\}.$$

We also define $V_{\text{off}}^0 = V_{\text{off}} \cap \{v \in V_{\text{off}} \,:\, v \cdot \nu = 0 \text{ on } \partial D\}$ as a subspace of V_{off} consisting of vector fields with zero normal component on ∂D; that is,

$$V_{\text{off}}^0 = \bigoplus_{\mathcal{E}_0^H} \{\Psi_i\}.$$

We then apply the proper orthogonal decomposition as a global model reduction to construct the online space $V_{\text{on}} \subset V_{\text{off}}$. Next, we discuss the constructions of V_{snap}, V_{off}, and V_{on}.

4.3.2 Snapshot space

To construct the basis functions of the snapshot space, we will find $(v_j^{(i)}, p_j^{(i)})$ by solving the following elliptic problem on the coarse neighborhood ω_i corresponding to the edge $E_i \in \mathcal{E}^H$

$$\begin{aligned} \kappa^{-1} v_j^{(i)} + \nabla p_j^{(i)} &= 0 \quad \text{in} \quad \omega_i, \\ \operatorname{div}(v_j^{(i)}) &= \alpha_j^{(i)} \quad \text{in} \quad \omega_i, \end{aligned} \tag{4.15}$$

subject to the boundary condition $v_j^{(i)} \cdot \nu_i = 0$ on $\partial\omega_i$, where ν_i denotes the outward unit-normal vector on $\partial\omega_i$. The above problem (4.15) will be solved separately on each coarse-grid element $K_l \subset \omega_i$ with extra boundary condition on E_i given by

$$v_j^{(i)} \cdot m_i = \delta_j^{(i)} \quad \text{on } E_j, \tag{4.16}$$

where m_i is a fixed unit-normal vector on E_i and $\delta_j^{(i)}$ is a piecewise constant function on E_i defined as

$$\delta_j^{(i)} = \begin{cases} 1, & \text{on } e_j, \\ 0, & \text{on other fine grid edges on } E_i, \end{cases} \qquad j = 1, 2, \cdots, J_i.$$

The function $\alpha_j^{(i)}$ in (4.15) is constant on each coarse-grid block and it satisfies $\int_{K_l} \alpha_j^{(i)} = \int_{E_i} \delta_j^{(i)}$ for all $K_l \subset \omega_i$. We remark that, the vector field $v_j^{(i)}$ can be extended to the rest of the domain D by defining $v_j^{(i)} = 0$ outside ω_i because $v_j^{(i)} \cdot \nu_i = 0$ on the boundary of ω_i.

The set of the solutions of (4.15) is the snapshot basis $\Psi_j^{i,\text{snap}} := v_j^{(i)}$. Using the snapshot basis, we define the snapshot space V_{snap} by

$$V_{\text{snap}} = \text{span}\{\Psi_j^{i,\text{snap}} : \quad 1 \leq j \leq J_i, \quad 1 \leq i \leq N_e\}.$$

To simplify notation, we will use the following single-index notation

$$V_{\text{snap}} = \text{span}\{\Psi_i^{\text{snap}} : \quad 1 \leq i \leq M_{\text{snap}}\},$$

where $M_{\text{snap}} = \sum_{i=1}^{N_e} J_i$ is the total number of snapshot fields.

4.3.3 Offline space

For each coarse neighborhood ω_i corresponding to the coarse edge E_i, we perform a local space reduction on the local snapshot space $V_{\text{snap}}^{(i)}$ through the use of some local spectral problems. The purpose of this is to determine the important local modes in the local snapshot space and to obtain a smaller space for approximating the solution. The local snapshot space $V_{\text{snap}}^{(i)}$ corresponding to the coarse edge E_i is

defined by

$$V_{\text{snap}}^{(i)} = \text{span}\{\Psi_j^{i,\text{snap}} : \quad 1 \leq j \leq J_i\}.$$

The local spectral problem is: find a real number $\lambda^{(i)} > 0$ and a function $v \in V_{\text{snap}}^{(i)}$ such that

$$a_i(v, w) = \lambda^{(i)} s_i(v, w), \qquad \forall w \in V_{\text{snap}}^{(i)}, \tag{4.17}$$

where

$$a_i(v, w) = \int_{E_i} \kappa^{-1}(v \cdot m_i)(w \cdot m_i), \quad s_i(v, w) = \int_{\omega_i} \kappa^{-1} v \cdot w + \int_{\omega_i} \text{div}(v)\,\text{div}(w). \tag{4.18}$$

Recall that m_i is a fixed unit-normal on the coarse edge E_i. However, one can use different spectral problem, for example see [22]. Assume that the eigenvalues of (4.17) are arranged in increasing order

$$\lambda_1^{(i)} < \lambda_2^{(i)} < \cdots < \lambda_{J_i}^{(i)}, \tag{4.19}$$

where $\lambda_k^{(i)}$ denotes the k-th eigenvalue for the coarse neighborhood ω_i. We then select the eigenfunctions, $Z_k^{(i)}$, corresponding to the first l_i eigenvalues to form the offline space. Thus, we define the offline multiscal basis functions as

$$\Psi_k^{i,\text{off}} = \sum_{j=1}^{J_i} Z_{kj}^{(i)} \Psi_j^{i,\text{snap}}, \qquad k = 1, 2, \cdots, l_i,$$

where $Z_{kj}^{(i)}$ is the j-th component of the vector $Z_k^{(i)}$. The global offline space is then

$$V_{\text{off}} = \text{span}\{\Psi_k^{i,\text{off}} : \quad 1 \leq k \leq l_i, \quad 1 \leq i \leq N_e\}.$$

To simplify the notations, we will use the following single-index notation

$$V_{\text{off}} = \text{span}\{\Psi_k^{\text{off}} : \quad 1 \leq k \leq M_{\text{off}}\},$$

where $M_{\text{off}} = \sum_{i=1}^{N_e} l_i$ is the total number of offline basis functions. Furthermore, we define V_{off}^0 such that all vectors in V_{off}^0 have zero normal component on the global domain boundary ∂D.

We define the transformation matrix, R_{off}, that maps from the offline space to the fine space as following

$$R_{\text{off}} = \left[\psi_1^{\text{off}}, \ldots, \psi_{M_{\text{off}}}^{\text{off}}\right],$$

where ψ_k^{off} is a vector containing the coefficients in the expansion of Ψ_k^{off} in the fine-grid basis functions.

Given the offline space, the mixed GMsFEM is to find $(v_H, p_H) \in V_{\text{off}} \times Q_H$ such that:

$$\begin{aligned} \langle \mathcal{N}(v_H)\, v_H,\ w_H \rangle_{\kappa^{-1},D} - \langle \text{div}(w_H),\ p_H \rangle_D = 0, \qquad & \forall w_H \in V_{\text{off}}^0, \\ \langle \frac{\partial p_H}{\partial t},\ q_H \rangle_D + \langle \text{div}(v_H),\ q_H \rangle_D = \langle f, q_H \rangle_D, \qquad & \forall q_H \in Q_H, \end{aligned} \tag{4.20}$$

where $p_H(0) = p_{0,h}$, $v_H.\ \nu = g_H$ on $\partial D \times J$, and for each coarse edge $E_i \in \partial D$, we have

$$\int_{E_i} (g_H - g)\Psi_j^{\text{off}} \cdot \nu = 0,$$

for all basis functions Ψ_j^{off} corresponding to the edge E_i. The GMsFEM system (4.20)

can be represented in the matrix form as follows.

$$\begin{aligned} R_{\text{off}}^T N_{\text{fine}}(R_{\text{off}} V_H) R_{\text{off}} V_H - R_{\text{off}}^T B_{\text{fine}}^T G_H P_H = 0, \\ G_H^T M_{\text{fine}} G_H^T \dot{P}_H + G_H^T B_{\text{fine}} R_{\text{off}} V_H = G_H^T F, \end{aligned} \tag{4.21}$$

where P_0 is given, G_H is the restriction operator from Q_H into Q_h, V_H and P_H are vectors of coefficients in the expansions of the solutions v_H and p_H in the spaces V_{off} and Q_H, respectively. Therefore, the original system (4.11) with M_v degrees of freedom for the velocity field is reduced to a system with M_{off} velocity dimension where $M_{\text{off}} \ll M_v$. The nonlinear term $R_{\text{off}}^T N_{\text{fine}}(R_{\text{off}} V_H) R_{\text{off}}$ in the reduced-order model, given by (4.21), has a computation complexity that depends on the dimension of the full system M_v. To reduce the computational requirements, we employ the discrete empirical interpolation method to approximate the nonlinear term locally at each coarse neighborhood. Let $\mathcal{N}_L$ denotes the approximation of $\mathcal{N}$ using local DEIM. Then, instead of solving the offline problem given by (4.20), we consider the following offline problem: Find $(v_H, p_H) \in V_{\text{off}} \times Q_H$ such that

$$\begin{aligned} \langle \mathcal{N}_L(v_H)\ v_H,\ w_H \rangle_{\kappa^{-1},D} - \langle \text{div}(w_H),\ p_H \rangle_D = 0, \qquad \forall w_H \in V_{\text{off}}^0, \\ \langle \frac{\partial p_H}{\partial t},\ q_H \rangle_D + \langle \text{div}(v_H),\ q_H \rangle_D = \langle f, q_H \rangle_D, \qquad \forall q_H \in Q_H, \end{aligned} \tag{4.22}$$

where $p_H(0) = p_{0,h}$, $v_H.\ \nu = g_H$ on $\partial D \times J$.

4.3.4 Online space

The online space is used to solve the problem for different parameters such as source terms, initial conditions, and boundary conditions. In our simulation, the online space is constructed via proper orthogonal decomposition method presented in Section 2.3. Next, we recall a brief description of this method with some details.

Computation of the POD modes [46]: Let us divide the interval $[0, T]$ into $(n-1)$ sub-intervals for given $n \in \mathbb{N}$,

$$0 = t_1 < t_2 < \cdots < t_n = T.$$

Let v_H^i for $i \in \{1, \cdots, n\}$ denotes the coarse-grid solution of problem (4.22) at time t_i, i.e., $v_H^i = v_H(t_i)$ and set

$$\mathcal{V} = \text{span}\{v_H^1, \cdots, v_H^n\}.$$

We refer to $\mathcal{V}$ as the ensemble consisting of the snapshots $\{v_H^i\}_{i=1}^n$. Let $\{\varphi_i\}_{i=1}^d$ denote an orthonormal basis of $\mathcal{V}$ with $d = \dim\mathcal{V}$. Then each element in the space $\mathcal{V}$ can be written as a linear combination of $\{\varphi_i\}_{i=1}^d$. In particular, the collected offline snapshots can be expressed as

$$v_H^i = \sum_{j=1}^{d} \langle v_H^i, \varphi_j \rangle_{\kappa^{-1},D} \ \ \varphi_j \qquad \text{for} \ \ i = 1, \cdots, n. \tag{4.23}$$

The POD method is to choose an orthonormal basis such that for every $l \in \{1, \cdots, d\}$ the mean square error between the elements v_H^i, $0 \le i \le n$, and the corresponding l-th partial sum of (4.23) is minimized on average:

$$\min_{\varphi_1, \cdots, \varphi_l} \frac{1}{(n-1)} \sum_{i=1}^{n} \| v_H^i - \sum_{j=1}^{l} \langle v_H^i, \varphi_j \rangle_{\kappa^{-1},D} \ \varphi_j \|_{\kappa^{-1},D}^2, \tag{4.24}$$

subject to

$$\langle \varphi_i, \varphi_j \rangle = \delta_{ij} \qquad \text{for} \ \ 1 \le i \le l, \ \ 1 \le j \le i. \tag{4.25}$$

A solution $\{\varphi_i\}_{i=1}^{l}$ to (4.24) and (4.25) is called the POD basis of rank l. We introduce the correlation matrix $W = (w_{i,j})_{n\times n} \in \mathbb{R}$ corresponding to the snapshots $\{v_H^i\}_{i=1}^{n}$ by

$$w_{i,j} = \frac{1}{(n-1)}\langle v_H^i, v_H^j\rangle_{\kappa^{-1},D}.$$

The matrix W is positive semidefinite and has rank d.

Proposition 1. Let $\mu_1^2 \geq \mu_2^2 \geq \cdots \geq \mu_d^2$ denote the positive eigenvalues of W and $u_1, u_2, \cdots, u_d$ the associated orthonormal eigenvectors. Then a POD basis of rank $l \leq d$ is given by

$$\varphi_i = \sum_{j=1}^{n}(u_i)_j v_H^j,$$

where $(u_i)_j$ denotes the jth component of the eigenvector u_i. Furthermore, the following error formula holds

$$\frac{1}{(n-1)}\sum_{i=1}^{n}\|v_H^i - \sum_{j=1}^{l}\langle v_H^i, \varphi_j\rangle_{\kappa^{-1},D}\ \varphi_j\|_{\kappa^{-1},D}^2 = \sum_{j=l+1}^{d}\mu_j^2.$$

Online problem: We first define the online space to be the subspace of V_{off} spanned by the POD basis functions with dimension $M_{\text{on}} = l$ and is denoted by V_{on},

$$V_{\text{on}} = \text{span}\{\Psi_k^{\text{on}} : \quad 1 \leq k \leq M_{\text{on}}\},$$

where $\Psi_k^{\text{on}} := \varphi_k, \quad 1 \leq k \leq M_{\text{on}}$. Furthermore, we define V_{on}^0 to be the restriction of V_{on} with all vectors have zero normal component on ∂D. Then, the online problem

is to find $(v_{\text{on}}, p_{on}) \in V_{\text{on}} \times Q_H$ such that:

$$\begin{aligned} \langle \mathcal{N}_G(v_{on})\ v_{\text{on}},\ w_{\text{on}} \rangle_{\kappa^{-1},D} - \langle \text{div}(w_{\text{on}}),\ p_{\text{on}} \rangle_D = 0, \qquad & \forall w_{\text{on}} \in V^0_{\text{on}}, \\ \langle \frac{\partial p_{\text{on}}}{\partial t},\ q_{\text{on}} \rangle_D + \langle \text{div}(v_{\text{on}}),\ q_{\text{on}} \rangle_D = \langle f, q_{\text{on}} \rangle_D, \qquad & \forall q_{\text{on}} \in Q_{\text{H}}, \end{aligned} \tag{4.26}$$

where $p_{\text{on}}(0) = p_{0,h}$, $v_{\text{on}}.\ \nu = g_{\text{on}}$ on $\partial D \times J$, and for each coarse edge $E_i \in \partial D$, we have

$$\int_{E_i} (g_{\text{on}} - g)\Psi^{\text{on}}_j \cdot \nu = 0,$$

for all basis functions Ψ^{on}_j corresponding to the edge E_i. $\mathcal{N}_G$ denotes the approximation of the nonlinear function (4.2) on the global domain D using global DEIM. Since $V_{\text{on}} \subseteq V_{\text{off}}$, each of Ψ^{on}_k is represented in terms of a vector ψ^{on}_k containing the coefficients in the expansion of Ψ^{on}_k in the offline basis functions. Thus we define the mapping from the online space to the offline space by the following matrix:

$$R_{\text{on}} = \left[\psi^{\text{on}}_1, \ldots, \psi^{\text{on}}_{M_{\text{on}}}\right].$$

Let $\mathcal{R} = R_{\text{off}} R_{\text{on}} \in \mathbb{R}^{M_v \times M_{\text{on}}}$, then $\mathcal{R}$ is the transformation matrix from online space to the fine space. For given P_0, the matrix form of the online system (4.26) is as follows:

$$\begin{aligned} \mathcal{R}^T N_{\text{fine}}(\mathcal{R}V_{\text{on}})\mathcal{R}V_{\text{on}} - \mathcal{R}^T B^T_{\text{fine}} G_H P_{\text{on}} = 0, \\ G^T_H M_{\text{fin}} G^T_H \dot{P}_{\text{on}} + G^T_H B_{\text{fine}} \mathcal{R} V_{\text{on}} = G^T_H F, \end{aligned} \tag{4.27}$$

where we use the global DEIM to avoid the expensive computational cost of the nonlinear term $\mathcal{R}^T N_{\text{fine}}(\mathcal{R}V_{\text{on}})\mathcal{R}$. We emphasize that the degrees of freedom for the velocity field is reduced to $M_{\text{on}} \ll M_{\text{off}} \ll M_v$.

4.4 Convergence analysis

In this section, we will estimate the L^2- norm error between the solution to Forchheimer equation (4.1) using the global-local reduction method and the fine-scale solution obtained by solving the fine-grid problem (4.10) in the lowest-order Raviart-Thomas FE space, $V_h \times Q_h$. First, we will estimate the error between the fine-scale solutions $(v_h, p_h) \in V_h \times Q_h$ and the offline solutions $(v_H, p_H) \in V_{\text{off}} \times Q_H$ through introducing the projection of the fine-grid velocity v_h to the snapshots space, V_{snap}. Next we will derive an estimate for the difference between the offline solutions and the online solutions, $(v_{\text{on}}, p_{\text{on}}) \in V_{\text{on}} \times Q_H$, obtained by the global reduction for the offline velocity field using POD method.

In the following analysis we will define a projection $\hat{v} \in V_{\text{snap}}$ as follows. Let K be a coarse-grid block and let $\bar{f} = \frac{1}{|K|}\int_K f$ be the average value of f over K. Then the restriction of $\hat{v}$ on K is obtained by solving the following problem:

$$\begin{aligned} \kappa^{-1}\mathcal{N}(\hat{v})\ \hat{v} + \nabla\hat{p} &= 0 && \text{in } K \times J, \\ \frac{\partial \hat{p}}{\partial t} + \operatorname{div}(\hat{v}) &= \bar{f}(x) && \text{in } K \times J, \end{aligned} \tag{4.28}$$

subject to the following conditions:

$$\hat{p}(0) = p_h(0) \quad \text{in } K, \qquad \hat{v}.\,\nu = v_h.\,\nu \quad \text{on } \partial K \times J.$$

We solve problem (4.28) on the fine-grid. Then we have $\hat{v} \in V_h$ and by construction we also have $\hat{v} \in V_{\text{snap}}$. Furthermore, our results in the following analysis are obtained based on the following assumptions and approximation properties:

A 1: Assume $f \in L^2$.

A 2: There exists constants α_0, α_1 such that

$$0 < \alpha_0 \leq \mathcal{N}(v) \leq \alpha_1.$$

Similarly, there exists positive constants α_0^L, α_0^G, α_1^L, and α_1^G such that

$$0 < \alpha_0^L \leq \mathcal{N}_L(v) \leq \alpha_1^L, \qquad\qquad 0 < \alpha_0^G \leq \mathcal{N}_G(v) \leq \alpha_1^G.$$

A 3: The nonlinear function, $\mathcal{N}(v)$, defined by Equation (4.2) is Lipschitz continuous with respect to the $\|.\|_{\kappa^{-1},D}$ norm, i.e. there exists a real constant $C_L > 0$ such that:

$$\|\mathcal{N}(v_1) - \mathcal{N}(v_2)\|^2_{\kappa^{-1},D} \leq C_L \|v_1 - v_2\|^2_{\kappa^{-1},D} \qquad \forall v_1, v_2 \in V_h.$$

A 4: $\mathcal{N}(v)$ is strictly monotone function (see [11]). More precisely, there is a positive constant $C_m > 0$ such that:

$$(\mathcal{N}(v_1)v_1 - \mathcal{N}(v_2)v_2).(v_1 - v_2) \geq C_m |v_1 - v_2|^2 \qquad \forall\ v_1, v_2 \in V_h.$$

From now on, we will use the notation $a \preceq b$ whenever there is a uniform constant $C > 0$ such that $a \leq Cb$. Thus, we write the inequalities in **A3** and **A4**, respectively, as following:

$$\|\mathcal{N}(v_1) - \mathcal{N}(v_2)\|^2_{\kappa^{-1},D} \preceq \|v_1 - v_2\|^2_{\kappa^{-1},D} \qquad \forall v_1, v_2 \in V_h.$$

$$\langle \mathcal{N}(v_1)v_1 - \mathcal{N}(v_2)v_2, v_1 - v_2 \rangle_{\kappa^{-1},D} \succeq \|v_1 - v_2\|_{\kappa^{-1},D} \qquad \forall\ v_1, v_2 \in V_h.$$

A 5: Let $\hat{v}$ be the projection of v_h to the snapshot space obtained by solving (4.28). We assume $\|\hat{v}\|_\infty \leq \tilde{C}$, for some positive constant $\tilde{C}$.

A 6: Let $P_l : V_{\text{off}} \longrightarrow V_{\text{on}}$ be a projection defined as following:

$$P_l u_H = \sum_{i=1}^{l} \langle u_H(t), \varphi_i \rangle_{\kappa^{-1},D} \ \varphi_i \qquad \forall u_H \in V_{\text{off}}, \tag{4.29}$$

where $\{\varphi_1, \cdots, \varphi_l\}$ are the POD basis functions span the online space as discussed in Section 4.3.4. Recall that from Proposition.1 we have the following estimate:

$$\frac{1}{(n-1)} \sum_{i=1}^{n} \|v_H^i - \sum_{j=1}^{l} \langle v_H^i, \varphi_j \rangle_{\kappa^{-1},D} \ \varphi_j\|_{\kappa^{-1},D}^2 = \sum_{j=l+1}^{d} \mu_j^2.$$

The sum on the LHS of the above estimate is the trapezoidal approximation for the integral

$$\int_0^T \|v_H(s) - \sum_{j=1}^{l} \langle v_H(s), \varphi_j \rangle_{\kappa^{-1},D} \ \varphi_j\|_{\kappa^{-1},D}^2 \ ds.$$

Therefore, we can write:

$$\int_0^T \|v_H(s) - P_l v_H(s)\|_{\kappa^{-1},D} \ ds = \sum_{j=l+1}^{d} \mu_j^2. \tag{4.30}$$

To estimate the online error $\|v_{\text{on}} - v_h\|_{\kappa^{-1},D}$, we assume $\forall v_H \in V_{\text{off}}, \exists \ \tilde{C}_l > 0$ such that $\|P_l v_H\|_\infty \leq \tilde{C}_l$.

A 7: The error of approximating a nonlinear function using discrete empirical interpolation method (DEIM) has been discussed in [16]. In our analysis we will use the following notations to denote the local and the global DEIM errors,

respectively,

$$E^{\text{DEIM}}_{\text{local}}(v_H) := ||\mathcal{N}(v_H) - \mathcal{N}_L(v_H)||^2_{\kappa^{-1},D}. \tag{4.31}$$

$$E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) := ||\mathcal{N}(v_{\text{on}}) - \mathcal{N}_G(v_{\text{on}})||^2_{\kappa^{-1},D}. \tag{4.32}$$

Lemma 4.4.1. Let $(v_h, p_h) \in V_h \times Q_h$ be the solution of (4.10), and $(\hat{v}, \hat{p})$ be the solution obtained by solving the variational problem of (4.28). Under the assumption **A1-A4** the following estimate holds true for any $t \in [0, T]$,

$$\int_0^t \|v_h(s) - \hat{v}(s)\|^2_{\kappa^{-1},D} ds + \|p_h(t) - \hat{p}(t)\|^2_D \preceq C_{\max} \sum_{i=1}^{N_e} \int_0^t \|f(s) - \bar{f}(s)\|^2_{K_i} ds, \tag{4.33}$$

where $C_{\max} = \max_{K \in \mathcal{T}^H} \left(\kappa^{-1}_{\min,K}\right)$, $\kappa_{\min,K}$ is the minimum of κ over K.

Proof. Subtracting the variational problem of (4.28) from (4.10), we have:

$$\begin{aligned} \langle \mathcal{N}(v_h)\ v_h - \mathcal{N}(\hat{v})\ \hat{v},\ w_h \rangle_{\kappa^{-1},K} - \langle \text{div}(w_h),\ (p_h - \hat{p}) \rangle_K = 0, \quad & \forall w_h \in V^0_h(K), \\ \langle (p_h - \hat{p})_t,\ q_h \rangle_K + \langle \text{div}(v_h - \hat{v}),\ q_h \rangle_K = \langle (f - \bar{f}), q_h \rangle_K, \quad & \forall q_h \in Q_h(K). \end{aligned} \tag{4.34}$$

Taking $w_h = v_h - \hat{v}$ and $q_h = p_h - \hat{p}$ in (4.34) and adding the resulting equations, we get:

$$\langle \mathcal{N}(v_h)\ v_h - \mathcal{N}(\hat{v})\ \hat{v},\ (v_h - \hat{v}) \rangle_{\kappa^{-1},K} + \frac{1}{2}\frac{d}{dt}\|p_h - \hat{p}\|^2_{L^2(K)} = \langle (f - \bar{f}), (p_h - \hat{p}) \rangle_K.$$

Using monotone property in **A4**, we have:

$$\|v_h - \hat{v}\|^2_{\kappa^{-1},K} + \frac{1}{2}\frac{d}{dt}\|p_h - \hat{p}\|^2_{L^2(K)} \preceq \|f - \bar{f}\|_{L^2(K)} \ \ \|p_h - \hat{p}\|_{L^2(K)}, \tag{4.35}$$

which implies

$$\|v_h - \hat{v}\|^2_{\kappa^{-1},K} + \frac{1}{2}\frac{d}{dt}\|p_h - \hat{p}\|^2_{L^2(K)} \preceq \frac{C_1}{2}\|f - \bar{f}\|^2_{L^2(K)} + \frac{1}{2C_1}\ \|p_h - \hat{p}\|^2_{L^2(K)},$$

where C_1 is a positive constant to be determined later. Recall that the Raviart-Thomas element satisfies the following inf-sup condition

$$\|q_h\|_{L^2(K)} \preceq \sup_{w_h \in V_h(K)} \frac{\int_K \mathrm{div}(w_h) q_h}{\|w_h\|_{H(\mathrm{div};K)}}, \quad \forall\ q_h \in Q_h(K). \tag{4.36}$$

Using (4.36) and first equation of (4.34) gives:

$$\begin{aligned}
\|p_h - \hat{p}\|_{L^2(K)} &\preceq \sup_{w_h \in V_h(K)} \frac{\langle \mathcal{N}(v_h)\ (v_h - \hat{v}), w_h\rangle_{\kappa^{-1},K}}{\|w_h\|_{H(\mathrm{div};K)}} \\
&\leq \sup_{w_h \in V_h(K)} \frac{\left(\int_K (\kappa^{-1}\mathcal{N}(v_h)\ (v_h - \hat{v}))^2\right)^{\frac{1}{2}} \left(\int_K |w_h|^2\right)^{\frac{1}{2}}}{\|w_h\|_{H(\mathrm{div};K)}} \\
&\leq \alpha_1\ \kappa_{\min,K}^{-\frac{1}{2}} \|v_h - \hat{v}\|_{\kappa^{-1},K}.
\end{aligned}$$

Thus we have:

$$\|p_h - \hat{p}\|^2_{L^2(K)} \preceq\ \kappa^{-1}_{\min,K}\|v_h - \hat{v}\|^2_{\kappa^{-1},K}.$$

Let C_2 be the hidden constant in the above inequality, we define the constant C_1 in

(4.35) to be $C_1 := C_2\ \kappa^{-1}_{\min,K}$, then (4.35) becomes

$$\|v_h - \hat{v}\|^2_{\kappa^{-1},K} + \frac{d}{dt}\|p_h - \hat{p}\|^2_K \leq C_2\ \kappa^{-1}_{\min,K}\|f - \bar{f}\|^2_K.$$

Finally, integrate over $(0,t)$ and sum over all elements K_i to get the desired estimate in (4.33).

□

Next, we will estimate the error $\|v_h - v_H\|_{\kappa^{-1},D}$ in Theorem 4.4.3. To prove this theorem, we will make use of the inf-sup condition given by Theorem 4.4.2, which is proved in [22].

Theorem 4.4.2. *Let N_0 be the number of interior coarse edges. For each interior coarse edge E_i, assume that there exists a basis function $\Psi^{i,\text{off}}_r \in V^0_{\text{off}}$, $1 \leq r \leq l_i$, such that $\int_{E_i} \Psi^{i,\text{off}}_r \cdot m_i \neq 0$. Then, for all $p \in Q_H$, we have*

$$\|p\|_{L^2(D)} \preceq C_{\text{infsup}} \sup_{w \in V^0_{\text{off}}} \frac{\int_D \text{div}(w)p}{\|w\|_V}, \tag{4.37}$$

where $C_{\text{infsup}} = (\max_{1\leq i\leq N_0} \min_r \int_{w_i} \kappa^{-1}\Psi^{i,\text{off}}_r.\Psi^{i,\text{off}}_r + 1)^2$ and the minimum is taken over all indices r with the property $\int_{E_i} \Psi^{i,\text{off}}_r.\ m_i \neq 0$.

Proof. see [22]. □

Theorem 4.4.3. *Let $(v_h, p_h) \in V_h \times Q_h$ be the solution of (4.10), $(v_H, p_H) \in V_{\text{off}} \times Q_H$ be the solution for solving (4.22). Then, under the assumptions* **A1-A6**, *the following*

error estimate holds true,

$$\int_0^t \|v_h(s)-v_H(s)\|^2_{\kappa^{-1},D}+\|p_h(t)-p_H(t)\|^2_D \preceq \max_{K\in\mathcal{T}^H}\left(\kappa^{-1}_{min,K}\right)\sum_{i=1}^{N_e}\int_0^t \|f(s)-\bar{f}(s)\|^2_{K_i} + \Lambda^{-1}\sum_{i=1}^{N_e}\int_0^t a_i\left(\hat{v}(s),\hat{v}(s)\right)ds + tE^{DEIM}_{local}(v_H), \tag{4.38}$$

for all $t\in[0,T]$*, where* $E^{DEIM}_{local}(v_H)$ *is the local DEIM error defined in* (4.31).

Proof. We will split the proof to the following steps:

Step 1: Using the fact that $V^0_{\text{off}}\subseteq V^0_h$ and $Q_H\subset Q_h$, we can take $w_h=w_H\in V^0_{\text{off}}$ and $q_h=q_H\in Q_H$ in (4.22) and subtract the resulting system from (4.10) to obtain:

$$\begin{aligned}\langle\mathcal{N}(v_h)v_h-\mathcal{N}_L(v_H)v_H,\ w_H\rangle_{\kappa^{-1},D}-\langle\text{div}(w_H),\ (p_h-p_H)\rangle_D=0, \quad \forall w_H\in V^0_{\text{off}},\\ \langle(p_h-p_H)_t,\ q_H\rangle_D+\langle\text{div}(v_h-v_H),\ q_H\rangle_D=0, \quad \forall q_H\in Q_H,\end{aligned} \tag{4.39}$$

Since q_H is a constant function over each coarse grid K, then

$$\langle(f-\bar{f}),q_H\rangle_K=\int_K(f-\bar{f})\ q_H\ \ dx=q_H\left[\int_K f\ \ dx-|K|\bar{f}\right]=0.$$

Therefore, if we let $q_h=q_H$ in the second equation of (4.34), we will get:

$$\langle(\hat{p}-p_h)_t,\ q_H\rangle_K+\langle\text{div}(\hat{v}-v_h),\ q_H\rangle_K=0 \quad \forall q_H\in Q_H.$$

Sum over all K and add the resulting equation to the system (4.39), we obtain the following system:

$$\langle\mathcal{N}(v_h)v_h-\mathcal{N}_L(v_H)v_H,w_H\rangle_{\kappa^{-1},D}-\langle\text{div}(w_H),(p_h-p_H)\rangle_D=0\ \forall w_H\in V^0_{\text{off}}, \tag{4.40}$$

$$\langle (\hat{p} - p_H)_t,\ q_H \rangle_D + \langle \mathrm{div}(\hat{v} - v_H),\ q_H \rangle_D = 0\ \forall q_H \in Q_H. \tag{4.41}$$

Note that

$$\langle \mathrm{div}(w_H),\ (p_h - p_H) \rangle_D = \langle \mathrm{div}(w_H),\ (p_h - \hat{p}) \rangle_D + \langle \mathrm{div}(w_H),\ (\hat{p} - p_H) \rangle_D.$$

Using the first equation of (4.34) we have:

$$\langle \mathrm{div}(w_H), (p_h - p_H) \rangle_D = \langle \mathcal{N}(v_h)v_h - \mathcal{N}(\hat{v})\hat{v}, w_h \rangle_{\kappa^{-1},D} + \langle \mathrm{div}(w_H), (\hat{p} - p_H) \rangle_D. \tag{4.42}$$

Substituting (4.42) into (4.40) gives:

$$\begin{aligned} \langle \mathcal{N}(v_h)v_h - \mathcal{N}_L(v_H)v_H, w_H \rangle_{\kappa^{-1},D} - \langle \mathrm{div}(w_H), (p_h - p_H) \rangle_D = \\ \langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, w_H \rangle_{\kappa^{-1},D} - \langle \mathrm{div}(w_H), (\hat{p} - p_H) \rangle_D. \end{aligned}$$

Thus, we rewrite the system (4.40) - (4.41) as following

$$\begin{aligned} \langle \mathcal{N}(\hat{v})\ \hat{v} - \mathcal{N}_L(v_H)v_H,\ w_H \rangle_{\kappa^{-1},D} - \langle \mathrm{div}(w_H),\ (\hat{p} - p_H) \rangle_D = 0, \quad \forall w_H \in V_{\mathrm{off}}^0, \\ \langle (\hat{p} - p_H)_t,\ q_H \rangle_D + \langle \mathrm{div}(\hat{v} - v_H),\ q_H \rangle_D = 0, \quad \forall q_H \in Q_H. \end{aligned} \tag{4.43}$$

Recall that $\hat{v} \in V_{\mathrm{snap}}$. We can therefore write $\hat{v}$ as

$$\hat{v} = \sum_{i=1}^{N_e} \sum_{k=1}^{J_i} \hat{v}_{ij}\ \Psi_k^{i,\mathrm{off}}. \tag{4.44}$$

Let us define $\hat{v}_H \in V_{\text{off}}$ by

$$\hat{v}_H = \sum_{i=1}^{N_e} \sum_{k=1}^{l_i} \hat{v}_{ij} \ \Psi_k^{i,\text{off}}, \tag{4.45}$$

where we recall that $l_i \leq J_i$ is the number of eigenfunctions selected for the coarse neighborhood ω_i and $\Psi_k^{i,\text{off}}$ are the eigenfunctions of the local spectral problem (4.17). Notice that $\hat{v}_H \in V_{\text{off}}^0$. Then, we can take $w_H = \hat{v}_H - v_H$ and $q_H = \hat{p} - p_H$ in (4.43), and add the resulting equations, we obtain:

$$\begin{aligned}\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, \hat{v} - v_H\rangle_{\kappa^{-1},D} + \langle (\hat{p} - p_H)_t, \hat{p} - p_H\rangle_D = \langle \text{div}(\hat{v}_H - \hat{v}), \hat{p} - p_H\rangle_D \\ + \langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D}.\end{aligned}$$

Then we can write,

$$\begin{aligned}&\underbrace{\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}(v_H)v_H, \hat{v} - v_H\rangle_{\kappa^{-1},D}}_{I_1} + \langle (\hat{p} - p_H)_t, \hat{p} - p_H\rangle_D = \\ &\qquad \underbrace{\langle \text{div}(\hat{v}_H - \hat{v}), \hat{p} - p_H\rangle_D}_{I_2} + \\ &\qquad \underbrace{\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D}}_{I_3} + \\ &\qquad \underbrace{\langle \mathcal{N}_L(v_H)v_H - \mathcal{N}(v_H)v_H, \hat{v} - v_H\rangle_{\kappa^{-1},D}}_{I_4}.\end{aligned} \tag{4.46}$$

Step 2: By the assumption **A4**, we have

$$I_1 = \langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}(v_H)v_H, \ \hat{v} - v_H\rangle_{\kappa^{-1},D} \succeq \|\hat{v} - v_H\|^2_{\kappa^{-1},D}.$$

Therefore, (4.46) can be written as

$$\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + \frac{1}{2}\frac{d}{dt}\|\hat{p} - p_H\|^2_{L^2(D)} \preceq I_2 + I_3 + I_4. \tag{4.47}$$

Step 3: Using Cauchy-Schwarz inequality, we have

$$I_2 = \langle \mathrm{div}(\hat{v}_H - \hat{v}),\ \hat{p} - p_H\rangle_D \leq \|\mathrm{div}(\hat{v}_H - \hat{v})\|_{L^2(D)}\ \|\hat{p} - p_H\|_{L^2(D)}. \tag{4.48}$$

By the definition of the spectral problem (4.18), we write:

$$\begin{aligned}\|\mathrm{div}(\hat{v}_H - \hat{v})\|^2_{L^2(D)} = \int_D \left(\mathrm{div}(\hat{v}_H - \hat{v})\right)^2 &\preceq \sum_{i=1}^{N_e}\int_{w_i}\left(\mathrm{div}(\hat{v}_H - \hat{v})\right)^2 \\ &\preceq \sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}).\end{aligned} \tag{4.49}$$

By the inf-sup condition (4.37) and the error equation (4.43), we have:

$$\begin{aligned}\|\hat{p} - p_H\|_{L^2(D)} &\preceq C_{\mathrm{infsup}} \sup_{w\in V^0_{\mathrm{off}}} \frac{\int_D \mathrm{div}(w)(\hat{p} - p_H)}{\|w\|_V} \\ &= C_{\mathrm{infsup}} \sup_{w\in V^0_{\mathrm{off}}} \frac{\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, w\rangle_{\kappa^{-1},D}}{\|w\|_V} \\ &= C_{\mathrm{infsup}} \sup_{w\in V^0_{\mathrm{off}}} \{\frac{\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)\hat{v}, w\rangle_{\kappa^{-1},D}}{\|w\|_V} + \\ &\qquad \frac{\langle \mathcal{N}_L(v_H)(\hat{v} - v_H), w\rangle_{\kappa^{-1},D}}{\|w\|_V}\}.\end{aligned}$$

By **A5**, there is a positive constant $\tilde{C}$ such that $\|\hat{v}\|_\infty \leq \tilde{C}$. Then,

$$\begin{aligned}\|\hat{p} - p_H\|_{L^2(D)} \preceq C_{\mathrm{infsup}} \sup_{w\in V^0_{\mathrm{off}}} \{&\frac{\tilde{C}\|\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H)\|_{\kappa^{-1},D}\|w\|_{\kappa^{-1},D}}{\|w\|_V} + \\ &\frac{\alpha_1^L\|\hat{v} - v_H\|_{\kappa^{-1},D}\|w\|_{\kappa^{-1},D}}{\|w\|_V}\}.\end{aligned}$$

Thus,

$$\|\hat{p} - p_H\|_{L^2(D)} \preceq C_{\text{infsup}}\{\|\mathcal{N}(\hat{v}) - \mathcal{N}(v_H)\|_{\kappa^{-1},D} + \|\mathcal{N}(v_H) - \mathcal{N}_L(v_H)\|_{\kappa^{-1},D} + \|\hat{v} - v_H\|_{\kappa^{-1},D}\}.$$

Using **A3** and (4.31), we write

$$\|\hat{p} - p_H\|_{L^2(D)} \preceq C_{\text{infsup}}\{\|\hat{v} - v_H\|_{\kappa^{-1},D} + \sqrt{E^{\text{DEIM}}_{\text{local}}(v_H)}\}. \tag{4.50}$$

Substitute (4.49) and (4.50) into (4.48), we get :

$$\langle \text{div}(\hat{v}_H - \hat{v}), \hat{p} - p_H\rangle_D \preceq C_{\text{infsup}} \left[\sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v})\right]^{1/2} \|\hat{v} - v_H\|_{\kappa^{-1},D} + C_{\text{infsup}} \left[\sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}\right]^{1/2} \sqrt{E^{\text{DEIM}}_{\text{local}}(v_H)}.$$

Using Young's inequality,

$$I_2 = \langle \text{div}(\hat{v}_H - \hat{v}),\ \hat{p} - p_H\rangle_D \preceq \{\frac{1}{2C_3}\sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}) + \frac{C_3}{2}\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + \frac{1}{2}\sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}) + \frac{1}{2}E^{\text{DEIM}}_{\text{local}}(v_H)\},$$

where $C_3 > 0$ will be determined later.

Step 4: For I_3 we can write:

$$\langle \mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} = \langle (\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H))\hat{v}, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} + \langle \mathcal{N}_L(v_H)(\hat{v} - v_H), \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D}.$$

Note that

$$\langle(\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H))\hat{v}, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} \leq \tilde{C}\|\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H)\|_{\kappa^{-1},D} \ \|\hat{v} - \hat{v}_H\|_{\kappa^{-1},D} \tag{4.51}$$

and

$$\langle\mathcal{N}_L(v_H)(\hat{v} - v_H), \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} \leq \alpha_L\|\hat{v} - v_H\|_{\kappa^{-1},D} \ \|\hat{v} - \hat{v}_H\|_{\kappa^{-1},D}. \tag{4.52}$$

Using Young's inequality in (4.51):

$$\begin{aligned}\langle(\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H))\hat{v}, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} &\preceq \frac{C_4}{2}\|\mathcal{N}(\hat{v}) - \mathcal{N}_L(v_H)\|^2_{\kappa^{-1},D} + \frac{1}{2C_4}\|\hat{v} - \hat{v}_H\|^2_{\kappa^{-1},D} \\ &\preceq \frac{C_4}{2}\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + \frac{C_4}{2}E^{\text{DEIM}}_{\text{local}}(v_H) \\ &\quad + \frac{1}{2C_4}\|\hat{v} - \hat{v}_H\|^2_{\kappa^{-1},D}.\end{aligned}$$

Similarly, for (4.52),

$$\langle\mathcal{N}_L(v_H)(\hat{v} - v_H), \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} \preceq \frac{C_4}{2}\|\hat{v} - v_H\|^2_{\kappa^{-1},D} \ + \frac{1}{2C_4}\|\hat{v} - \hat{v}_H\|^2_{\kappa^{-1},D}.$$

Therefore:

$$\begin{aligned}I_3 = \langle\mathcal{N}(\hat{v})\hat{v} - \mathcal{N}_L(v_H)v_H, \hat{v} - \hat{v}_H\rangle_{\kappa^{-1},D} \preceq C_4\|\hat{v} - v_H\|^2_{\kappa^{-1},D} &+ \frac{C_4}{2}E^{\text{DEIM}}_{\text{local}}(v_H) \\ &+ \frac{1}{C_4}\|\hat{v} - \hat{v}_H\|^2_{\kappa^{-1},D}.\end{aligned}$$

Step 5: We can split I_4 as following:

$$\begin{aligned} I_4 = \langle \mathcal{N}_L(v_H)v_H - \mathcal{N}(v_H)v_H, \hat{v} - v_H \rangle_{\kappa^{-1},D} \leq \langle \mathcal{N}_L(v_H)(v_H - \hat{v}), \hat{v} - v_H \rangle_{\kappa^{-1},D} \\ + \langle (\mathcal{N}_L(v_H) - \mathcal{N}(v_H))\hat{v}, \hat{v} - v_H \rangle_{\kappa^{-1},D} \\ + \langle \mathcal{N}(v_H)(\hat{v} - v_H), \hat{v} - v_H \rangle_{\kappa^{-1},D}. \end{aligned}$$

Using **A2**, we have

$$\begin{aligned} \langle \mathcal{N}_L(v_H)(v_H - \hat{v}), \hat{v} - v_H \rangle_{\kappa^{-1},D} &\leq \alpha_1^L \ \langle v_H - \hat{v}, \hat{v} - v_H \rangle_{\kappa^{-1},D} \\ &\leq \max\{\alpha_1, \alpha_1^L\} \langle v_H - \hat{v}, \hat{v} - v_H \rangle_{\kappa^{-1},D} \end{aligned} \tag{4.53}$$

and

$$\begin{aligned} \langle \mathcal{N}(v_H)(\hat{v} - v_H), \hat{v} - v_H \rangle_{\kappa^{-1},D} &\leq \alpha_1 \ \langle \hat{v} - v_H, \hat{v} - v_H \rangle_{\kappa^{-1},D} \\ &\leq \max\{\alpha_1, \alpha_1^L\} \langle \hat{v} - v_H, \hat{v} - v_H \rangle_{\kappa^{-1},D}. \end{aligned} \tag{4.54}$$

Consequently,

$$I_4 = \langle \mathcal{N}_L(v_H)v_H - \mathcal{N}(v_H)v_H, \hat{v} - v_H \rangle_{\kappa^{-1},D} \leq \langle (\mathcal{N}_L(v_H) - \mathcal{N}(v_H))\hat{v}, \hat{v} - v_H \rangle_{\kappa^{-1},D}.$$

Using Cauchy-Schwarz inequality:

$$I_4 \leq \tilde{C} \|\mathcal{N}_L(v_H) - \mathcal{N}(v_H)\|_{\kappa^{-1},D} \|\hat{v} - v_H\|_{\kappa^{-1},D}.$$

By Young's inequality,

$$I_4 \preceq \frac{1}{2C_5} E_{\text{local}}^{\text{DEIM}}(v_H) + \frac{C_5}{2} \|\hat{v} - v_H\|_{\kappa^{-1},D}^2.$$

Step 6: From steps 3, 4 and 5, we have:,

$$I_2 + I_3 + I_4 \preceq \left(\frac{1}{2C_3+2}\right)\sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}) + \left(\frac{C_3+C_5+2C_4}{2}\right)\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + \left(\frac{1+C_4}{2} + \frac{1}{2C_5}\right) E^{\text{DEIM}}_{\text{local}}(v_H) + \frac{1}{C_4}\|\hat{v} - \hat{v}_H\|^2_{\kappa^{-1},D}. \tag{4.55}$$

We notice that

$$\|\hat{v}_H - \hat{v}\|^2_{\kappa^{-1},D} \;\leq\; \sum_{i=1}^{N_e}\|\hat{v}_H - \hat{v}\|^2_{\kappa^{-1},\omega_i} \leq \sum_{i=1}^{N_e} s_i\left(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}\right).$$

Let C_6 be the hidden constant in (4.55), we then choose C_3, C_4 and C_5 such that $C_6\ \frac{C_3+C_5+2C_4}{2} = \frac{1}{2}$. Then,

$$I_2 + I_3 + I_4 \preceq \sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}) + \frac{1}{2}\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + E^{\text{DEIM}}_{\text{local}}(v_H).$$

Step 7: Substituting the above inequality into (4.47), we get:

$$\frac{1}{2}\|\hat{v} - v_H\|^2_{\kappa^{-1},D} + \frac{1}{2}\frac{d}{dt}\|\hat{p} - p_H\|^2_{L^2(D)} \preceq \sum_{i=1}^{N_e} s_i(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}) + E^{\text{DEIM}}_{\text{local}}(v_H). \tag{4.56}$$

By (4.44) and (4.45) and the fact that $\Psi^{i,\text{off}}_k$ are eigenfunctions of (4.17), we have:

$$s_i\left(\hat{v}_H - \hat{v}, \hat{v}_H - \hat{v}\right) = \sum_{k=l_i+1}^{J_i}\left(\lambda^{(i)}_k\right)^{-1}(\hat{v}_{ik})^2 a_i\left(\Psi^{i,\text{off}}_k,\ \Psi^{i,\text{off}}_k\right).$$

Assume eigenvalues of (4.17) are ordered as $\lambda^{(i)}_1 < \lambda^{(i)}_2 < ... < \lambda^{(i)}_{j_i}$, then by orthogo-

nality of $\Psi_k^{i,\text{off}}$, we obtain:

$$s_i\left(\hat{v}_H-\hat{v},\hat{v}_H-\hat{v}\right)\leq\left(\lambda_{l_i+1}^{(i)}\right)^{-1}a_i\left(\hat{v}_H-\hat{v},\hat{v}_H-\hat{v}\right)\leq\left(\lambda_{l_i+1}^{(i)}\right)^{-1}a_i\left(\hat{v},\hat{v}\right).$$

Taking $\Lambda=\min_{1\leq i\leq N_e}\lambda_{l_i+1}^{(i)}$, we obtain:

$$\sum_{i=1}^{N_e}s_i(\hat{v}_H-\hat{v},\hat{v}_H-\hat{v})\ \preceq\ \sum_{i=1}^{N_e}\Lambda^{-1}a_i\left(\hat{v},\hat{v}\right).$$

Substitute in (4.56), we have:

$$\|\hat{v}_h-v_H\|_{\kappa^{-1},D}^2+\frac{d}{dt}\|\hat{p}-p_H\|_D^2\preceq\sum_{i=1}^{N_e}\Lambda^{-1}a_i\left(\hat{v},\hat{v}\right)+E_{\text{local}}^{\text{DEIM}}(v_H).\tag{4.57}$$

Step 8: Finally, by the triangle inequality, we have

$$\begin{aligned}\|v_h-v_H\|_{\kappa^{-1},D}^2+\frac{d}{dt}\|p_h-p_H\|_D^2\leq\|v_h-\hat{v}\|_{\kappa^{-1},D}^2+\frac{d}{dt}\|p_h-\hat{p}\|_D^2+\|\hat{v}_h-v_H\|_{\kappa^{-1},D}^2\\+\frac{d}{dt}\|\hat{p}-p_H\|_D^2.\end{aligned}$$

Integrating with respect to time and using Lemma 4.4.1 and inequality (4.57), we obtain the desired estimate (4.38). □

Next, we will estimate the online error $\|v_h-v_{\text{on}}\|_{\kappa^{-1},D}$ through the use of the projection $P_l:V_{\text{off}}\longrightarrow V_{\text{on}}$ defined in (4.29).

Theorem 4.4.4. *Let* $(v_h,p_h)\in V_h\times Q_h$ *be the solution of* (4.10) *and* $(v_{\text{on}},p_{\text{on}})\in$

$V_{\text{on}} \times Q_H$ *be the solution for solving* (4.26)*. Using the assumptions* **A1-A6***, we have*

$$\int_0^t \|v_h - v_{on}\|^2_{\kappa^{-1},D} + \|p_h(t) - p_{on}(t)\|^2_D \preceq \max_{K\in\ \mathcal{T}^H} \left(\kappa^{-1}_{min,K}\right) \sum_{i=1}^{N_e} \int_0^t \|f(s) - \bar{f}(s)\|^2_{K_i} ds$$
$$+ \Lambda^{-1} \sum_{i=1}^{N_e} \int_0^t a_i\left(\hat{v}(s), \hat{v}(s)\right) ds + tE^{DEIM}_{local}(v_H)$$
$$+ \sum_{j=1}^{d} \mu_j^2 + tE^{DEIM}_{global}(v_{on}), \tag{4.58}$$

for all $t \in [0, T]$*, where* $E^{DEIM}_{local}(v_H)$ *is the local DEIM error given by* (4.31) *and* $E^{DEIM}_{global}(v_{on})$ *is the global DEIM error given by* (4.32) *.*

Proof. **Step 1:** Consider the following problem in $V_{\text{on}} \times Q_H$ space:

$$\begin{aligned} \langle \mathcal{N}(v_H)\ P_l v_H,\ w_{\text{on}}\rangle_{\kappa^{-1},D} - \langle \text{div}(w_{\text{on}}),\ p_l\rangle_D = 0, \qquad & \forall w_{\text{on}} \in V^0_{\text{on}}, \\ \langle \frac{\partial p_l}{\partial t},\ q_{\text{on}}\rangle_D + \langle \text{div}(P_l v_H),\ q_{\text{on}}\rangle_D = \langle f, q_{\text{on}}\rangle_D, \qquad & \forall q_{\text{on}} \in Q_H. \end{aligned} \tag{4.59}$$

Since $V_{\text{on}} \subset V_{\text{off}}$, we can replace w_H in (4.20) by w_{on} and subtract (4.59) to get:

$$\begin{aligned} \langle \mathcal{N}(v_H)\ (v_H - P_l v_H),\ w_{\text{on}}\rangle_{\kappa^{-1},D} - \langle \text{div}(w_{\text{on}}),\ p_H - p_l\rangle_D = 0, \qquad & \forall w_{\text{on}} \in V^0_{\text{on}}, \\ \langle (p_H - p_l)_t,\ q_{\text{on}}\rangle_D + \langle \text{div}(v_H - P_l v_H),\ q_{\text{on}}\rangle_D = 0, \qquad & \forall q_{\text{on}} \in Q_H. \end{aligned}$$

By restricting the proof of Theorem 4.4.2 to the online space, we can obtain the following inf-sup condition. For all $p \in Q_H$, we have:

$$\|p\|_{L^2(D)} \preceq C^{\text{on}}_{\text{infsup}} \sup_{w\in V^0_{\text{on}}} \frac{\int_D \text{div}(w) p}{\|w\|_V}, \tag{4.60}$$

where $C^{\text{on}}_{\text{infsup}} = (\max_{1\le i\le N_0} \min_r \int_{w_i} \kappa^{-1} \Psi^{i,\text{on}}_r . \Psi^{i,\text{on}}_r + 1)^2$ and the minimum is taken

over all indices r with the property $\int_{E_i} \Psi_r^{i,\mathrm{on}}.m_i \neq 0$. Using this condition we have:

$$\|p_H - p_l\|_{L^2(D)} \preceq C_{\mathrm{infsup}}^{\mathrm{on}} \sup_{w \in V_{\mathrm{on}}^0} \frac{\int_D \mathrm{div}(w)(p_H - p_l)}{\|w\|_V}$$

$$\preceq C_{\mathrm{infsup}}^{\mathrm{on}} \sup_{w \in V_{\mathrm{on}}^0} \frac{\langle \mathcal{N}(v_H)\ (v_H - P_l v_H),\ w \rangle_{\kappa^{-1},D}}{\|w\|_V}.$$

Hence,

$$\|p_H - p_l\|_{L^2(D)} \preceq \|v_H - P_l v_H\|_{\kappa^{-1},D}. \tag{4.61}$$

Step 2: Subtracting (4.26) from (4.59) gives us:

$$\begin{aligned} \langle \mathcal{N}(v_H)\ P_l v_H - \mathcal{N}_G(v_{\mathrm{on}}) v_{\mathrm{on}},\ w_{\mathrm{on}} \rangle_{\kappa^{-1},D} - \langle \mathrm{div}(w_{\mathrm{on}}),\ p_l - p_{\mathrm{on}} \rangle_D = 0, \qquad & \forall w_{\mathrm{on}} \in V_{\mathrm{on}}^0, \\ \langle (p_l - p_{\mathrm{on}})_t,\ q_{\mathrm{on}} \rangle_D + \langle \mathrm{div}(P_l v_H - v_{\mathrm{on}}),\ q_{\mathrm{on}} \rangle_D = 0, \qquad & \forall q_{\mathrm{on}} \in Q_H. \end{aligned}$$

Take $w_{\mathrm{on}} = P_l v_H - v_{\mathrm{on}} \in V_{\mathrm{on}}$ and $q_{\mathrm{on}} = p_l - p_{\mathrm{on}} \in Q_H$ and add the resulting equations to obtain:

$$\langle \mathcal{N}(v_H)\ P_l v_H - \mathcal{N}_G(v_{\mathrm{on}}) v_{\mathrm{on}},\ P_l v_H - v_{\mathrm{on}} \rangle_{\kappa^{-1},D} + \langle (p_l - p_{\mathrm{on}})_t,\ p_l - p_{\mathrm{on}} \rangle_D = 0,$$

which implies:

$$\alpha_0 \|P_l v_H - v_{\mathrm{on}}\|_{\kappa^{-1},D}^2 + \frac{1}{2}\frac{d}{dt}\|p_l - p_{\mathrm{on}}\|_{L^2}^2 \leqslant \langle (\mathcal{N}_G(v_{\mathrm{on}}) - \mathcal{N}(v_H)) v_{\mathrm{on}}, P_l v_H - v_{\mathrm{on}} \rangle_{\kappa^{-1},D}. \tag{4.62}$$

Note that:

$$\begin{aligned}\langle(\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H))v_{\text{on}}, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D} &= \langle\mathcal{N}_G(v_{\text{on}})(v_{\text{on}}-P_l v_H), P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}\\ &+\langle(\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H))P_l v_H, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}\\ &+\langle\mathcal{N}(v_H)(P_l v_H - v_{\text{on}}), P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}.\end{aligned}$$

By the assumption **A2**, we have

$$\begin{aligned}\langle\mathcal{N}_G(v_{\text{on}})(v_{\text{on}}-P_l v_H), P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D} &\leq \alpha_1^G\langle v_{\text{on}}-P_l v_H, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}\\ &\leq \max\{\alpha_1^G,\alpha_1\}\langle v_{\text{on}}-P_l v_H, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}.\end{aligned} \tag{4.63}$$

Similarly,

$$\begin{aligned}\langle\mathcal{N}(v_H)(P_l v_H - v_{\text{on}}), (P_l v_H - v_{\text{on}})\rangle_{\kappa^{-1},D} &\leq \alpha_1\langle P_l v_H - v_{\text{on}}, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}\\ &\leq \max\{\alpha_1^G,\alpha_1\}\langle P_l v_H - v_{\text{on}}, P_l v_H - v_{\text{on}}\rangle_{\kappa^{-1},D}.\end{aligned} \tag{4.64}$$

By the assumption **A6** and Young's inequality, we have

$$\begin{aligned}\langle(\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H))P_l v_H, (P_l v_H - v_{\text{on}})\rangle_{\kappa^{-1},D} \leq \tilde{C}_l\{&\frac{1}{2C_7}\|\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H)\|^2_{\kappa^{-1},D}\\ &+\frac{C_7}{2}\|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D}\}.\end{aligned} \tag{4.65}$$

From (4.63), (4.64) and (4.65), we get:

$$\begin{aligned}\langle(\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H))v_{\text{on}}, (P_l v_H - v_{\text{on}})\rangle_{\kappa^{-1},D} \preceq &\frac{1}{2C_7}\|\mathcal{N}_G(v_{\text{on}})-\mathcal{N}(v_H)\|^2_{\kappa^{-1},D}\\ &+\frac{C_7}{2}\|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D},\end{aligned} \tag{4.66}$$

where the hidden constant in the above inequality is $\max\{\alpha_1^G,\alpha_1,\tilde{C}_l\}$. We can rewrite

(4.66) as follows:

$$\begin{aligned}\langle(\mathcal{N}_G(v_{\text{on}}) - \mathcal{N}(v_H)\ )\ v_{\text{on}},\ (P_l v_H - v_{\text{on}})\rangle_{\kappa^{-1},D} &\preceq \frac{1}{2C_7}\|\mathcal{N}_G(v_{\text{on}}) - \mathcal{N}(v_{\text{on}})\|^2_{\kappa^{-1},D} \\ &+ \frac{1}{2C_7}\|\mathcal{N}(v_{\text{on}}) - \mathcal{N}(v_H)\|^2_{\kappa^{-1},D} \\ &+ \frac{C_7}{2}\ \|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D} \\ \\ &\preceq \frac{1}{2C_7}E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) \\ &+ \frac{C_L}{2C_7}\|v_{\text{on}} - v_H\|^2_{\kappa^{-1},D} \\ &+ \frac{C_7}{2}\ \|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D}.\end{aligned}$$

Using the triangle inequality, $\|v_{\text{on}} - v_H\|^2_{\kappa^{-1},D} \leqslant \|P_l v_H - v_H\|^2_{\kappa^{-1},D} + \|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D}$. Therefore,

$$\begin{aligned}\langle(\mathcal{N}_G(v_{\text{on}}) - \mathcal{N}(v_H))v_{\text{on}}, (P_l v_H - v_{\text{on}})\rangle_{\kappa^{-1},D} &\preceq \frac{1}{2C_7}\left[E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) + \|P_l v_H - v_H\|^2_{\kappa^{-1},D}\right] \\ &+ \left(\frac{1}{2C_7} + \frac{C_7}{2}\right)\|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D}.\end{aligned} \tag{4.67}$$

Let $C_8 > 0$ be the hidden constant in the above inequality, then we choose C_7 such that $C_8\left(\frac{1}{2C_7} + \frac{C_7}{2}\right) < \alpha_0$ and substitute (4.67) in (4.62), we have

$$\|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D} + \frac{d}{dt}\|p_l - p_{\text{on}}\|^2_{L^2(D)} \preceq E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) + \|P_l v_H - v_H\|^2_{\kappa^{-1},D}.$$

Step 3: By the triangle inequality, we have:

$$\begin{aligned}\|v_H - v_{\text{on}}\|^2_{\kappa^{-1},D} + \frac{d}{dt}\|p_H - p_{\text{on}}\|^2_{L^2(D)} &\leqslant \|v_H - P_l v_H\|^2_{\kappa^{-1},D} + \|P_l v_H - v_{\text{on}}\|^2_{\kappa^{-1},D} \\ &\quad + \frac{d}{dt}\|p_l - p_{\text{on}}\|^2_{L^2(D)} + \frac{d}{dt}\|p_H - p_l\|^2_{L^2(D)} \\ &\preceq E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) + \|P_l v_H - v_H\|^2_{\kappa^{-1},D} \\ &\quad + \frac{d}{dt}\|p_H - p_l\|^2_{L^2(D)}.\end{aligned}$$

Integrating in time, we obtain:

$$\begin{aligned}\int_0^t \|v_H(s) - v_{\text{on}}(s)\|^2_{\kappa^{-1},D} ds + \|p_H(t) - p_{\text{on}}(t)\|^2_{L^2(D)} &\preceq t E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}) \\ &\quad + \int_0^t \|P_l v_H - v_H\|^2_{\kappa^{-1},D} \\ &\quad + \|p_H(t) - p_l(t)\|^2_{L^2(D)}.\end{aligned} \tag{4.68}$$

By (4.61), we have

$$\|p_H(t) - p_l(t)\|^2_{L^2(D)} \preceq \|v_H(t) - P_l v_H(t)\|^2_{\kappa^{-1},D},$$

for $t \in [0, T]$. Therefore, from Proposition 1, we have:

$$\|p_H(t) - p_l(t)\|^2_{L^2(D)} \preceq \|v_H(t) - P_l v_H(t)\|^2_{\kappa^{-1},D} \leqslant \sum_{i=1}^{n} \|v_H^i - P_l v_H^i\|^2_{\kappa^{-1},D} \preceq \sum_{j=1}^{d} \mu_j^2.$$

Also, using (4.30), we have: $\int_0^t \|P_l v_H - v_H\|^2_{\kappa^{-1},D} \leqslant \sum_{j=1}^{d} \mu_j^2$. Hence, (4.68) becomes

$$\int_0^t \|v_H(s) - v_{\text{on}}(s)\|^2_{\kappa^{-1},D} ds + \|p_H(t) - p_{\text{on}}(t)\|^2_{L^2(D)} \preceq \sum_{j=1}^{d} \mu_j^2 + t E^{\text{DEIM}}_{\text{global}}(v_{\text{on}}).$$

Using the triangle inequality and Theorem 4.4.3 we obtain the desired result in (4.58). □

4.5 Local online adaptive method

The local online adaptive method is an enrichment algorithm used to add new local multiscale basis functions, at some coarse regions, to the local offline multiscale basis functions. More precisely, we first start with the offline multiscale space, V_{off}, as our initial online space and use an enrichment algorithm to adaptively add some new basis functions based on the residual of the previous solution and special minimum energy snapshots. The advantage of adding these online basis functions is to accelerate the convergence based on the initial number of offline basis functions as we will show in our numerical examples. In the earlier works [17, 23], such method was used for linear elliptic problems. In our case, which is time-dependent, we do this enrichment at some fixed time steps and update the online basis functions in every 10-th time step to save the computational time.

We define V_Ω for a given region $\Omega \subseteq D$, to be the space of functions in V_{snap} which are supported in Ω, i.e. $V_\Omega = \oplus_{\omega_i \subseteq \Omega} V_{\text{snap}}^{(i)}$. Let $\hat{V}_\Omega$ denote the divergent free subspace of V_Ω. We denote the local online multiscal space by V_{ms}. Furthermore, we use the index $m \geq 0$ to represent the enrichment level and V_{ms}^m is the corresponding online multiscale space obtained by applying the proposed enrichment algorithm. The multiscale solutions obtained by solving the variational problem in $V_{ms}^m \times Q_H$ is then denoted by (v_{ms}^m, p_{ms}^m). At $m = 0$, we define $V_{ms}^0 = V_{\text{off}}$ and we have $(v_{ms}^0, p_{ms}^0) = (v_H, p_H)$.

At the fixed n-th time step, define the nonlinear functional R_Ω^n on V_Ω by

$$R_\Omega^n(u) = \int_\Omega \kappa^{-1} \mathcal{N}(v_{ms}^{n,m}) \;\; v_{ms}^{n,m}.u - \int_\Omega \text{div}(u) \;\; p_{ms}^{n,m}, \qquad \forall u \in V_\Omega,$$

and define the norm

$$\|R_{\Omega}^{n}\|_{V_{\Omega}^{*}} = \sup_{u \in V_{\Omega}} \frac{R_{\Omega}^{n}(u)}{\|u\|_{H(\mathrm{div};\Omega,\kappa^{-1})}}.$$

To simplify the notation, from now on, we will not use the time step index representation, n, as we know all the following computations are conducted in a specific fixed time step. The online adaptive method is summarized in the following steps:

Step 1: Find the multiscale solutions $v_{ms}^{m} \in V_{ms}^{m}$ and $p_{ms}^{m} \in Q_H$.

Step 2: Select non-overlapping regions $\Omega_1, ..., \Omega_J \subseteq D$, where each Ω_j is a union of some coarse grid neighborhood.

Step 3: Find the online basis functions. For each Ω_j, find $\hat{\phi}_j \in \hat{V}_{\Omega_j}$ such that

$$R_{\Omega_j}(u) = \int_{\Omega_j} \kappa^{-1}\mathcal{N}(v_{ms}^{m-1})\ \hat{\phi}_j.\ u \quad \forall u \in \hat{V}_{\Omega_j}.$$

Step 4: Update the velocity space by setting $V_{ms}^{m+1} = V_{ms}^{m} \oplus \mathrm{span}\{\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_J\}$ and go back to Step 1. Repeat these steps until the global error indicator is small or we have a certain number of basis functions.

Remark 4.5.1. In our calculations, we compute the error to be the difference between the online solution v_{ms}^{m} and the snapshot solution $v_{\mathrm{snap}} \in V_{\mathrm{snap}}(\Omega_j)$. If this error is very small, i.e., v_{ms}^{m} is close to v_{snap}, the norm of $\hat{\phi}_j$ will be small. In this case, we normalize it before computing the matrix in the finite element method to avoid having a singular matrix in our computation.

The convergence rate of the online adaptive method has been studied and proved in [17]. The result states that the error $\|v_{\mathrm{snap}} - v_{ms}^{m+1}\|_{\kappa^{-1},D}$ depends on the value of the first eigenvalue of the offline spectral problem that the corresponding eigenvector

was not included in the construction of the offline space. In other words, recall that we ordered the eigenvalue of the spectral problem (4.17) as in (4.19), the convergence rate of the online adaptive method can be small if $\Lambda_{\min}^{(I)} = \min_{i \in I} \lambda_{l_i+1}^{(i)}$ is large, where l_i is the initial number of the offline basis functions determined at the beginning of the method. That is we should choose l_i so that $\lambda_{l_i+1}^{(i)}$ is significantly large. We demonstrate this result in the next section.

4.6 Numerical results

In this section, we will present some numerical examples using global-local adaptive method for mixed framework of nonlinear Forchheimer flow given in (4.1) . We will use different numbers of local and global DEIM modes and different numbers of POD modes to perform the online space. We use these examples to illustrate the effect of these numbers on the accuracy and complexity of our computation. We also present examples to test the efficiency of using the online adaptive method to reduce the error. We will use different numbers of initial offline basis functions to show that the convergence rate depends on the number of the initial offline basis functions.

4.6.1 Global-local reduction method

In our simulations, the computational domain is $D = (0, 1)^2$. The number of coarse grids in each direction is $N = 10$ and the number of fine grids in each direction is $n = 100$. We also use time step size $\Delta t = 5$ in the time interval $t \in [0, 100]$. We are considering the permeability field shown in Figure 4.2 with contrast 10^3, i.e., $\max \kappa(x) / \min \kappa(x) = 10^3$.

4.6.1.1 Comparing the reference solution and the approximate solution

We use the fine solution as our reference solution. The approximate solution is the solution obtained by our proposed global-local method. In this example,

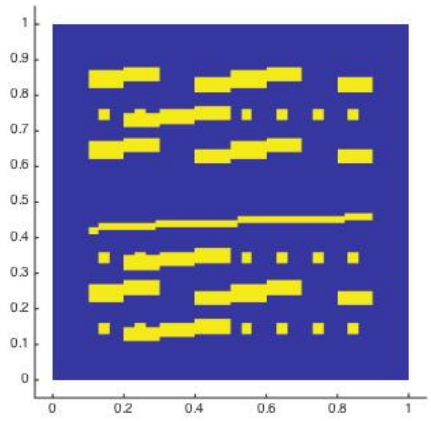
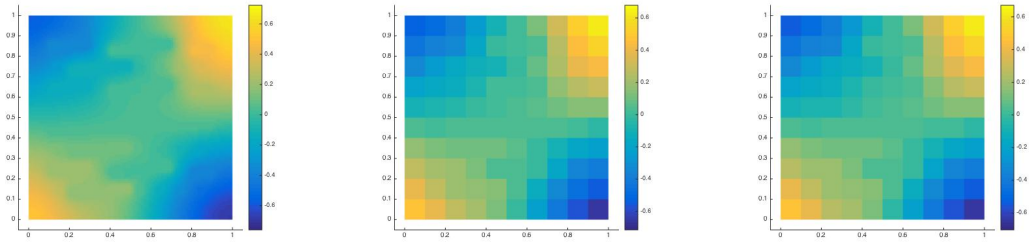


Figure 4.2: Permeability field (κ).

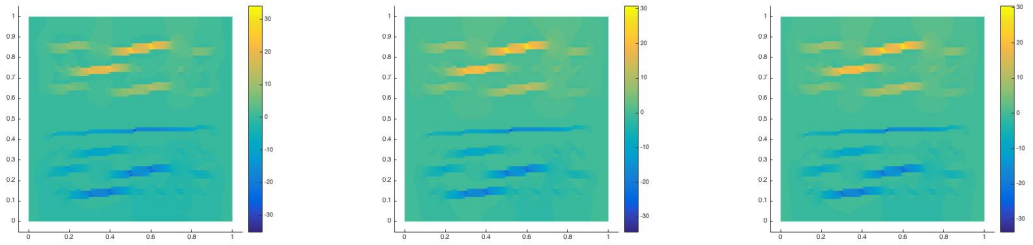
we obtained our approximate solution using two DEIM points to approximate the nonlinear function $\mathcal{N}(v)$ locally at each coarse region in the offline stage. In the online stage, we, also, used two DEIM points to approximate $\mathcal{N}(v)$ in the global domain. Furthermore, to perform the online space we used four POD modes. Therefore, the dimensions of the fine-grid space, offline space and online space are: $M_v = 20200$, $M_{\text{off}} = 540$ and $M_{\text{on}} = 4$, respectively. In Figure 4.3, we compare the pressure solution of the fine-grid problem with the pressure solution obtained from the offline problem (4.22) and the online problem (4.26). The corresponding velocity solutions in x-direction and y-direction are shown in Figures (4.4) and (4.5), respectively. From this comparison, we observe a good approximation, which shows the capability of the global-local reduction technique to present the fully resolved solution of the nonlinear Forchheimer equation.

In addition to this comparison, we show the accuracy of the proposed method by computing the relative L^2 error for the online pressure and online velocity using the



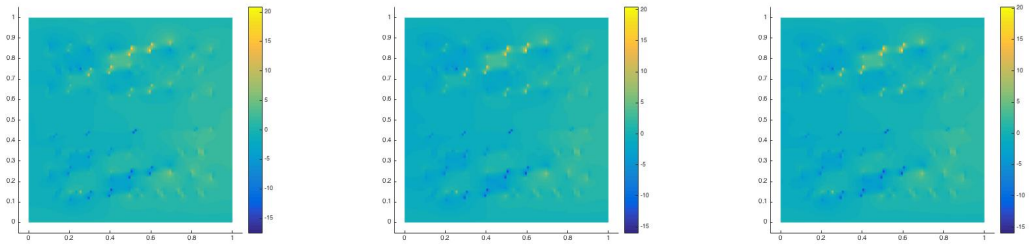
(a) Fine Pressure (b) Offline Pressure (c) Online Pressure

Figure 4.3: Comparison of pressure solutions.



(a) Fine Velocity (b) Offline Velocity (c) Online Velocity

Figure 4.4: Comparison of velocity solutions (x-direction).



(a) Fine Velocity (b) Offline Velocity (c) Online Velocity

Figure 4.5: Comparison of velocity solutions (y-direction).

following error equations:

$$RE_{\text{on}}(P) = \frac{\|p_h - p_{\text{on}}\|_{L^2(D)}}{\|p_h\|_{L^2(D)}}, \qquad RE_{\text{on}}(V) = \frac{\|v_h - v_{\text{on}}\|_{\kappa^{-1},D}}{\|v_h\|_{\kappa^{-1},D}}. \tag{4.69}$$

In Figure 4.6, we plot the error of the online pressure and velocity solutions computed using (4.69). The velocity error (on the top) is less than 7% and the pressure error (on the bottom) is less than 27%. We, also, observe that the errors for the velocity and the pressure is getting stable after almost 10 time steps.

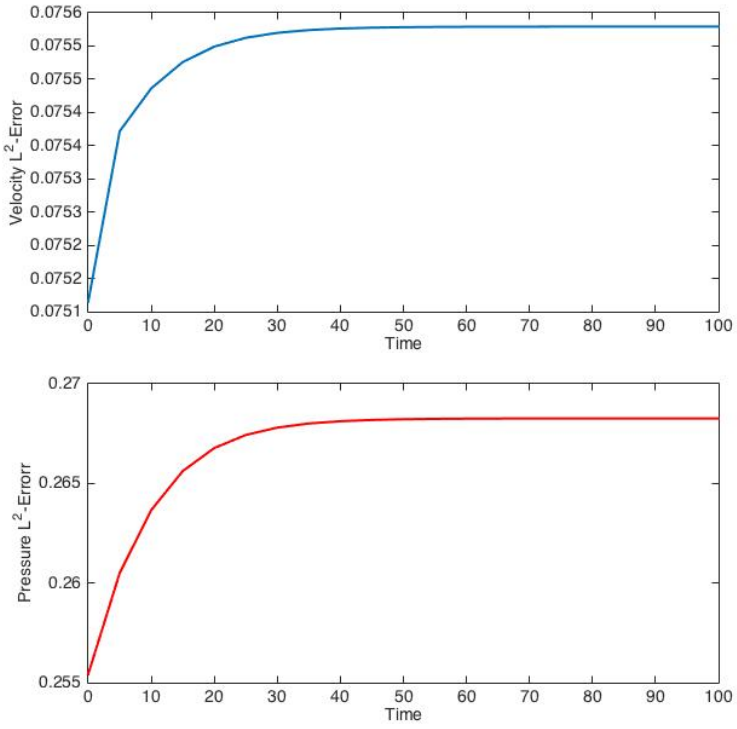


Figure 4.6: Top: the relative L^2 error for the online velocity field. Bottom: the relative L^2 error for the online pressure. Here, we use 2 local DEIM points, 2 global DEIM points and 4 POD modes.

Beside the good accuracy, we observe a significant reduction in the computational time as our principle goal for this proposed method. We first compute the time for solving the original system, T_0, and the time for the solving the reduced systems in the offline and online stages T_{off} and T_{on}, respectively. Then, we compute the percentage of the simulation time using:

$$PST = \frac{T_0}{T_i} \times 100,$$

where T_i denotes either T_{off} or T_{on}. In Table (4.1), we list these numbers to show the efficiency of the global-local approach in terms of saving the computational time.

Full system (fine-scale problem (4.10))	$T_0 = 702.6800$
Offline reduced system (offline problem (4.22))	$T_{\text{off}} = 2.0096$ (PST $= 0.2860\%$)
Online reduced system (online problem (4.26))	$T_{\text{on}} = 0.0822$ (PST$= 0.0117\%$)

Table 4.1: Time record.

4.6.1.2 Using different number of POD modes

The approximate solutions in the first example are obtained by using 4 POD modes. However, increasing the numbers of POD modes used in the online stage yields a better approximation. This is shown in Figure 4.7 where we plot the relative L^2 error for the velocity corresponding two different numbers of the POD modes.

4.6.1.3 Using different number of the local and global DEIM points

The effect of the number of the local and global modes used to approximate the nonlinear function $\mathcal{N}(v)$ in the offline and online problems is shown in Figure

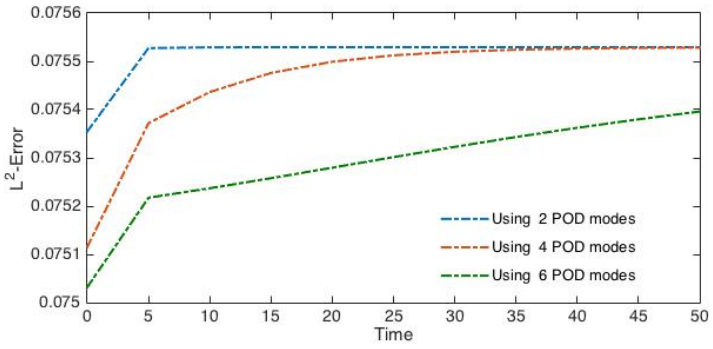
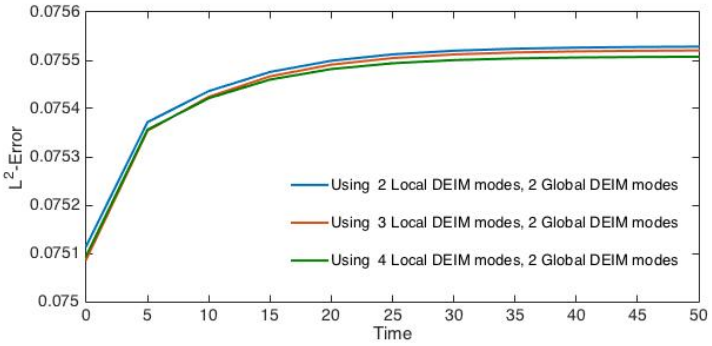


Figure 4.7: Variations of the velocity error with the number of POD modes.

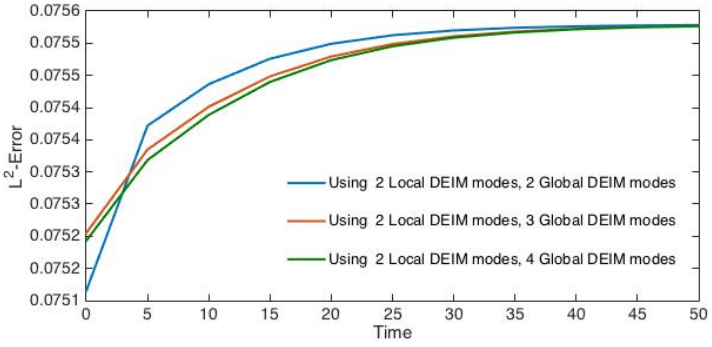
4.8. In (a), we show the velocity L^2 error with using 2, 3 and 4 DEIM modes to approximate $\mathcal{N}(v)$ locally while using fixed number for the global DEIM modes equal to two. In (b), we fix the number of the local DEIM and use different numbers for the global DEIM points. From these figures we see that the error is decreasing by increasing either local or global DEIM points used to approximate the nonlinear function locally or globally. However, further increasing for the global DEIM points does not affect the error as shown in (b). This is related to the dominance of the local error. In (C), we plot the velocity error with different numbers of both local and global DEIM points to show that increasing these numbers improves the solution accuracy. Therefore, to balance the local and global error one needs to carefully choose these numbers.

4.6.2 Local online adaptive method

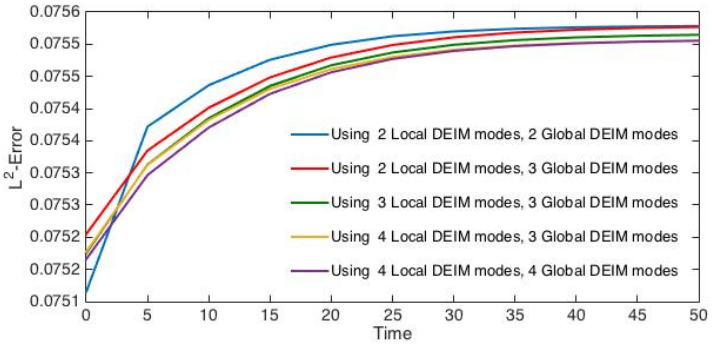
We apply the local online adaptive method to solve the system of equations given by (4.8). This example shows the performance of using local online adaptive method to get better approximation of our solution and demonstrates the effect of choosing different initial number of offline basis functions. We consider the domain



(a) Variations of local DEIM points.



(b) Variations of global DEIM points.



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

Figure 4.8: Variations of the velocity error with the number of local and global DEIM points. Here, we used 4 POD modes in online problem.

$[0,1] \times [0,1]$ with homogeneous boundary conditions, i.e., $g = 0$. We divide the domain into 15×15 coarse grid and divide each coarse grid into 40×40 fine grid. The source term f is chosen to be 1 on top left coarse grid block, -1 on bottom right coarse grid block and zero elsewhere. We use the permeability filed in Figure 4.2 with the contrast value of 10^2. For the time variable, we use time step size $\Delta t = 5$ in the time interval $t \in [0, 100]$. In each enrichment level, the regions $\Omega_1, ..., \Omega_J$ are chosen to be disjoint coarse-grid neighborhoods. We also emphasize that the online basis functions are added at the first time step and are updated in every 10-th time step. In Figure 4.9, we plot the snapshot velocity solution as our reference solution on the top and the online velocity solution in the bottom. We can see from this figure that we get a good approximation with using two initial number of basis functions.

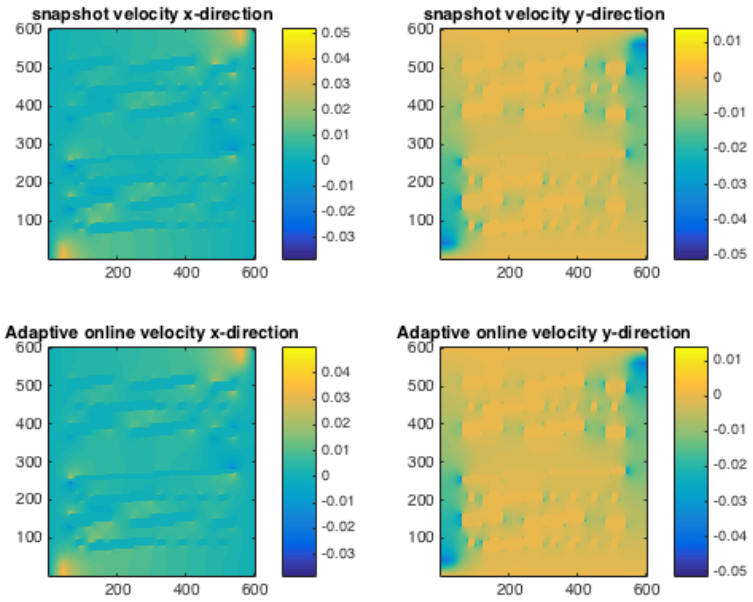


Figure 4.9: Comparing the snapshot velocity solution with online solution using two initial basis functions.

To show the effect of using different numbers of initial basis functions, we solve the equations using 1, 2, 3, and 4 basis functions and compute the error e given by:

$$e = \frac{\|v_{\text{snap}} - v_{ms}\|_{\kappa^{-1},D}}{\|v_{\text{snap}}\|_{\kappa^{-1},D}}. \tag{4.70}$$

In Figure 4.10, we plot the snapshot error e against the number of the basis function used initially. We observe that the error decay faster with larger number for the initial basis functions. This is also observed from Table 4.2, which shows the value of e. The first column of Table 4.2 represents the number of basis functions used for each coarse neighborhood and the total degrees of freedom (DOF), which are the numbers in parentheses. The other columns represent the snapshots errors when using 1, 2, 3, and 4 basis functions. This observation can be explained by the value of Λ_{min}, which depends on the number of the initial basis functions used offline. Choosing larger number for the initial basis functions gives larger value for Λ_{min}. For example, with 1, 2, 3, and 4 basis functions, the value of Λ_{min} are $0.0776, 1.9511, 3.6204$ and 5.2765, respectively. In online adaptive method the rate of convergence is bounded above by $\Lambda_{\text{min}}^{-1}$. Therefore, increasing the number of the initial basis functions yields to the increase of Λ_{min}, and hence, the decrease of the error.

number of basis (DOF)	e (1 basis) $\Lambda_{\text{min}} = 0.0776$	e (2 basis) $\Lambda_{\text{min}} = 1.9511$	e (3 basis) $\Lambda_{\text{min}} = 3.6204$	e (4 basis) $\Lambda_{\text{min}} = 5.2765$
1(420)	0.1772	/	/	/
2(840)	0.0563	0.0474	/	/
3(1260)	0.0413	0.0086	0.0345	/
4(1680)	0.0213	0.0068	0.0074	0.0263
5(2100)	0.0096	0.0062	0.0064	0.0067

Table 4.2: Snapshot error of online adaptive method with 1, 2, 3 and 4 initial bases.

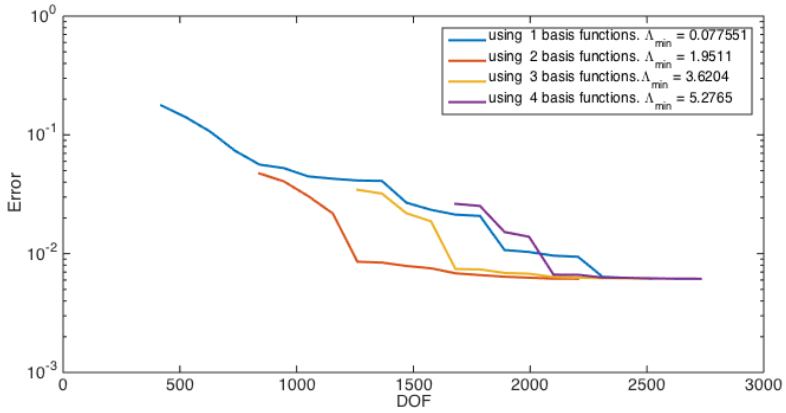


Figure 4.10: Snapshot error with different number of initial basis functions.

5. GLOBAL-LOCAL ONLINE ADAPTIVE REDUCTION METHOD FOR HETEROGENEOUS FORCHHEIMER FLOW

In this section, we introduce a global online adaptive method that is used to add new global basis functions to the POD subspace based on an inexpensive error indicators. Since the global error indicators are expensive to compute, we use local error indicators as our criterion to reduce the computational time. At any time step, if the adaption is needed, the new POD basis function is computed by solving the global residual problem. We solve the global residual problem using the GMsFEM, with the local online adaptation as presented in Section 4.5. For that we refer to the proposed method as the global-local online adaptive method. We emphasize that the global online adaptivity is performed by incorporating new data that become available in the online stage. This feature plays an important role to improve the accuracy of the approximate solution as we will see in the numerical experiment. We will consider the nonlinear Forchheimer flow (see Section 4.1) as our model problem.

To introduce the proposed method, we first recall the fine-scale problem under our consideration.

5.1 Fine-scale model

Recall the fine-scale problem for Forchheimer flow, presented in Section 4.2, is given by:

$$N_{\text{fine}}(V^n)\ \ V^n - \ B_{\text{fine}}^T P^n = 0, \tag{5.1}$$

$$M_{\text{fine}}\frac{P^n - P^{n-1}}{\Delta t} + B_{\text{fine}} V^n = F, \tag{5.2}$$

where V and P are the velocity and pressure vectors, respectively. F is the vector represents the source term. The matrices $N_{\text{fine}}, B_{\text{fine}}$ and M_{fine} are defined in Section 4.3. Let $\tilde{F} = \Delta t F + M_{\text{fine}} P^{n-1}$. From Equation (5.2), we have

$$P^n = M_{\text{fine}}^{-1}(\tilde{F} - \Delta t B_{\text{fine}} V^n),$$

and substitute the above equation in Equation (5.1) to get:

$$N_{\text{fine}}(V^n)\ V^n - B_{\text{fine}}^T(M_{\text{fine}}^{-1}\tilde{F}) + \Delta t B_{\text{fine}}^T(M_{\text{fine}}^{-1}B_{\text{fine}})V^n = 0.$$

Then,

$$\left[N_{\text{fine}}(V^n) + \Delta t B_{\text{fine}}^T(M_{\text{fine}}^{-1}B_{\text{fine}})\right] V^n = B_{\text{fine}}^T(M_{\text{fine}}^{-1}\tilde{F}). \tag{5.3}$$

Define $\mathcal{A}(V^n) = \left[N_{\text{fine}}(V^n) + \Delta t B_{\text{fine}}^T(M_{\text{fine}}^{-1}B_{\text{fine}})\right]$ and $\mathcal{H} = B_{\text{fine}}^T(M_{\text{fine}}^{-1}\tilde{F})$, we can re-write Equation (5.3) in a simple form as:

$$\mathcal{A}(V^n)V^n = \mathcal{H}. \tag{5.4}$$

This equation is expensive to solve. To avoid the complexity of solving Equation (5.4), we use a global reduction method, precisely, POD method, to solve the problem in a reduced-order dimension which will be discussed in the following section.

5.2 Global offline space

The global offline space is defined to be the reduced space of global basis functions (POD modes) construed offline. The construction of the global offline space starts with constructing the snapshot space by solving some local problems. In our method, we use Problem (4.15) to construct the snapshot space, V_{snap}. Using the whole

snapshot space, we solve the Forchheimer flow equation to obtain the global snapshot solutions $v_{\text{snap}} \in V_{\text{snap}}$. These snapshot solutions are then used to construct the global POD basis functions. More precisely, we collect N_t snapshot solutions and define

$$\mathcal{V} = \text{spsan}\{v_{\text{snap}}^1, \cdots, v_{\text{snap}}^{N_t}\}.$$

Then, we apply the POD method introduced in Section 2.3 to the correlation matrix ($W = \mathcal{V}^T\mathcal{V}$) to obtain the global offline POD basis functions. We denote the global POD modes by ψ_i^G and define the POD projection matrix, Ψ^G, to be the matrix whose columns are ψ_i^G. In the following sections, new global basis functions will be constructed during online process. To distinguish the global basis functions computed offline from those computed online, we will use ψ_{off}^G for offline global basis and ψ_{on}^G for online global basis. Similarly, Ψ_{off}^G is the offline POD projection matrix and Ψ_{on}^G is the online POD projection matrix which is obtained by updating Ψ_{off}^G with new online basis functions, ψ_{on}^G, this will be discussed in details in Section 5.3.

Once the offline global space is constructed, we can solve the original large-scale dynamical system (5.1)-(5.2) in a lower dimension. For example, in Section 5.4, the offline solution is obtained using only 2 offline global basis functions, i.e., the dimension of the offline reduced system is 2. The POD reduced-order system for Equation (5.4) is:

$$(\Psi_{\text{off}}^G)^T \ \mathcal{A}(\Psi_{\text{off}}^G \ V_r^n) \ \Psi_{\text{off}}^G \ V_r^n = (\Psi_{\text{off}}^G)^T \ \mathcal{H}, \tag{5.5}$$

where the subscript r in the above equation indicates the representation of the vector in the reduced dimension. In other words, V_r is a vector of reduced-order offline solutions $v_r \in \text{span}\{\psi_{\text{off}}^G\}$. Therefore, the accuracy of the reduced-order solution, v_r,

depends on the offline information and how these information represents the problem with initial input parameters.

In many applications, as time advancing, the global offline space (POD subspace) may not be sufficient to represent the full resolved solution at some time instants. To explain this, we call Case 1 of Example 5.4.1 (see Section 5.4). In this case, the reduced system is construed in the offline stage and kept unchanged in the online stage. In other words, no global online basis is added to the reduced solution space. Therefore, the reduced-order solution v_r, relies only on the pre-computed information from the offline stage. In the online stage, we solve the forward reduced system with different source term at some time instants. Since this change was not incorporated at the offline stage, the pre-computed offline information will not be sufficient to approximate the online solution. For example, Figure 5.1 shows that the error is large (10%) and has two jumps at the time instants when the source term has been changed in the online stage. We refer to Section 5.4 for more details.

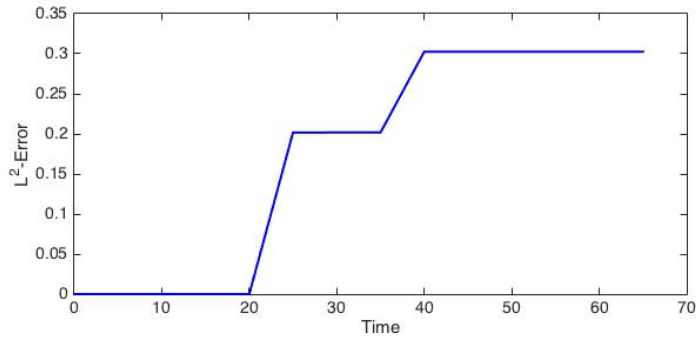


Figure 5.1: L^2- Error of the reduced-order solution in Case 1, $v_r \in \{\psi_{\text{off}}^G\}$.

We conclude that in this case and similar situations, the constructed global offline space is not sufficient to approximate the solution within desired accuracy. For this

reason, one needs to add some new global basis functions to the global POD subspace in the online process to improve the accuracy of the approximate solution. In order to add new global basis and adapt the POD subspace, we must know two main things: (1) at which time step to adapt and (2) how to adapt the POD subspace. The following sections are specified to address these issues.

5.3 Global online space

Global online space consists of global offline basis functions, $\{\psi^G_{\text{off}}\}$, and some new global online basis functions, $\{\psi^G_{\text{on}}\}$, that are computed whenever updating the global POD subspace is required based on an error indicator. To decide in which time instant updating the global POD subspace is needed, one needs to compute the norm of the following residual,

$$\text{Res} = \mathcal{H} - \mathcal{A}(\Psi^G_{\text{off}}\, V^n_r)\Psi^G_{\text{off}}\, V^n_r. \tag{5.6}$$

Then, if the residual norm is larger than a specific error tolerance at time step k, updating the global basis functions is required and the new global online basis function to be added in this time step is then given by solving

$$\mathcal{A}(\psi^{G,k}_{\text{on}})\psi^{G,k}_{\text{on}} = \text{Res}. \tag{5.7}$$

Notice that Equation (5.7) is a fine-grid problem. Therefore, computing the global residual and the new global online basis can be very expensive. To avoid the fine-scale computational cost, we solve Equation (5.7) adaptively using local model reduction and local online adaptive method introduced in Section 4.5. This will be discussed in Section 5.3.1.

Moreover, instead of using the global error indicator based on the global residual

given by Equation (5.6), we use local error indicator based on the norm of the local residual which is cheaper to compute. Precisely, we define the local residual R_i to be the restriction of the residual given by Equation (5.6) in the coarse blocks $\omega_i,\ \forall\ i = 1 \cdots, N_e$ (recall that N_e denotes the total number of the coarse edges E_i in the coarse mesh $\mathcal{T}^H$). Then, we compute their corresponding energy norm. If the number of coarse blocks with large error (i.e. the residual norm is greater than a certain tolerance) is greater than a specific number, then the POD subspace cannot give a good approximation and the adaption is needed. Next, we discuss how to adapt the POD subspace using the proposed global-local adaptive reduction method.

5.3.1 Global-local online adaptive method

To reduce the computational cost for solving Equation (5.7), we employ a local reduced order model to approximate the global solution by a coarse-scale solution which is cheaper to compute. Toward this goal, we apply the mixed GMsFEM introduced in Section 4.3.1 to construct the local multiscale basis functions. These basis functions are computed by performing a local spectral decomposition in the snapshot space, V_{snap}, and choosing the eigenvectors corresponding to the smallest eigenvalues. We denote the local multiscale basis functions by ψ_i^L and let

$$\Psi_{\text{off}}^L = \text{span}\{\psi_1^L \cdots \psi_{M_{\text{off}}}^L\},$$

to be the local offline multiscale space. We refer to Section 4.3.1 for more details. As in global basis functions, the local multiscale space constructed in the offline stage will be updated with new local online multiscale basis functions using the local online adaptive method discussed in Section 4.5. For that, from now on, we use ψ_{off}^L and ψ_{on}^L for offline and online local multiscale basis functions, respectively.

Suppose at a fixed time step k, the global space is spanned by N_{k-1} number of initial POD modes, that is

$$\Psi_k^G = \{\psi_1^G, \cdots, \psi_{N_{k-1}}^G\}.$$

Assume at this time step, Ψ_k^G was not sufficient to approximate the velocity solution for Forchhiemer flow based on the above described local error indicator. Then, Ψ_k^G needs to be updated by adding a new online global basis $\psi_{\text{on}}^{G,k}$. This basis function is obtained by solving Equation (5.7). Since this equation is nonlinear, we may linearize it to simplify the computation. That is instead of (5.7), we consider the following equation

$$\mathcal{A}(\psi_{N_{k-1}}^G)\psi_{\text{on}}^{G,k} = \text{Res}. \tag{5.8}$$

Using online local multiscale space Ψ_{on}^L, we will compute the coarse-scale solution of Equation (5.8) and then the global solution, $\psi_{\text{on}}^{G,k}$, will be obtained by projection. We emphasize that we will start with the offline local multiscale space Ψ_{off}^L as the initial online local space. To clarify this, we describe the whole process in the following steps.

Step 1: Solve for the multiscale solution of Equation (5.8)

We solve for $\hat{\psi} \in \Psi_{\text{on}}^L$ the following reduced-order system:

$$\hat{\mathcal{A}}\ \hat{\psi} = \hat{\text{R}}, \tag{5.9}$$

where $\hat{\mathcal{A}} = (\Psi_{\text{on}}^L)^T\ \mathcal{A}(\Psi_{\text{on}}^L\ \psi_{N_{k-1}}^G)\ (\Psi_{\text{on}}^L)$ and $\hat{\text{R}} = (\Psi_{\text{on}}^L)^T$ Res. The global online basis is then given by $\psi_{\text{on}}^G = \Psi_{\text{on}}^L\ \hat{\psi}$.

Step 2: Compute the local residual of the approximation in Step 1:

The multiscale residual is given by:

$$\mathrm{R}_{\mathrm{ms}} = \mathcal{A}(\psi^G_{\mathrm{on}})\ \psi^G_{\mathrm{on}} - \mathrm{Res}. \tag{5.10}$$

For each coarse region ω_i, we define the local residual $\mathrm{R}^i_{\mathrm{ms}}$ to be the restriction of R_{ms} in ω_i.

Step 3: Update the local online multiscale space $\Psi^L_{\mathbf{on}}$:

To update Ψ^L_{on}, we will use the index $m \geq 0$ to represent the enrichment level of the local multiscale space. We denote the local multiscale space at the enrichment level m by Ψ^L_m and let $\Psi^L_0 = \Psi^L_{\mathrm{off}}$. Moreover, we denote the restriction of Ψ^L_m on ω_i by $\Psi^{L,i}_m$.

Suppose at the enrichment level m, $\Psi^L_m = \{\psi^L_1, \cdots, \psi^L_r\}$. For each coarse region ω_i, we solve for $\hat{\phi}^i \in \Psi^{L,i}_m$ such that

$$\mathrm{R}^i_{\mathrm{ms}} = \mathcal{A}(\psi^L_{\mathrm{r}})|_{\omega_i}\ \hat{\phi}^i. \tag{5.11}$$

Then, we define $\psi^L_{\mathrm{on}} = \oplus_{i=1}^{N_e} \hat{\phi}^i$. We update the online local space by setting $\Psi^L_{m+1} = \{\Psi^L_m, \psi^L_{\mathrm{on}}\}$ and let $\Psi^L_{\mathrm{on}} = \Psi^L_{m+1}$. Before going to the next step, we repeat from Step 1 to Step 3 until the multiscale residual in (5.10) is less than a specific tolerance or we have certain number of local basis functions.

Step 4: Update the global POD space:

Finally, we update the global online POD subspace at the time step k to be

$$\Psi^G_k = \{\psi^G_1, \cdots, \psi^G_{N_{k-1}}, \psi^G_{\mathrm{on}}\}.$$

In this way, the number of POD modes increased by one at the time step requires

the adaption.

5.4 Numerical results

In this section, we present numerical results to show the performance of the global-local online adaptive method for approximating Forchheimer flow problem (see Section 4.1). We will consider two examples. The first example is to demonstrate the efficiency of adapting the global snapshot space using our adaptive method to improve the accuracy of the approximate solution of the reduced system. In this example, we compare the accuracy of the approximate solution in three cases. (1) The solution space is the global offline space, i.e., no global online basis is involved. (2) The solution space is the global online space, which consists of the offline global basis and some online global basis which are obtained using global residual and global error indicator. (3) The solution space is the global online space where, in this case, the global online basis are computed cheaply using our proposed adaptive method. In the second example, we will show the effect of the initial number of the offline global basis functions on the number of the adaption needed throughout the computational time interval to achieve the desired accuracy. In this example, we will show that the number of the time instants, where the adaption is needed, is increased as we decrease the number of the global offline basis functions. This is because the sufficiency of the global offline space in approximating the reduced solution is decreasing using fewer number of global basis functions. As a result, more adaption is needed.

Example 5.4.1. In this example, the computational domain is $[0,1] \times [0,1]$. For applying the mixed GMsFEM, we divide the domain into 15×15 coarse grids, where each coarse grid contains 40×40 fine grids. We solve the problem using homogeneous boundary conditions and consider the permeability field κ_1 as depicted in Figure 5.2

with the contrast value of 10^2. For the time variable, we use the time step size $\Delta t = 5$ in the time interval $[0, 100]$. The source term f, in the offline stage, is chosen to be 1 on the top left coarse-grid block, -1 on the bottom right coarse-grid block and zero elsewhere. In the online stage, we vary the source term at some time instants.

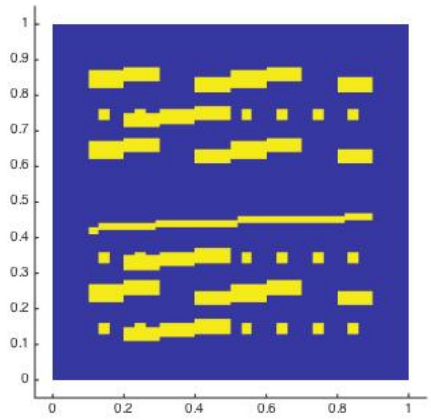


Figure 5.2: Permeability field κ_1.

Moreover, in the offline stage, we use the initial number of multiscale basis functions to be 3 for each coarse region (coarse-grid neighborhood) to construct the offline local space. Using 3 basis functions per coarse edge gives 1260 total number of degrees of freedom. This number is increased during enrichment process in the online stage using local online adaptive method. For the POD subspace, we initially start with 2 POD modes to define the offline global space. Using global-local online adaptive method as described in Section 5.3.1, the number of POD modes is increased by one whenever the adaption is required based on the error indicator.

To show the effect of adapting the POD subspace at some instance using global-

local online adaptive method, we compare the L^2 relative error for the velocity solution of reduced-order model, v_r, with and without the adaption. We use the snapshot solution, v_{snap}, as our reference solution and compute the error using the following equation:

$$e_{\text{online}} = \frac{\|v_{\text{snap}} - v_r\|_{\kappa^{-1},D}}{\|v_{\text{snap}}\|_{\kappa^{-1},D}}. \tag{5.12}$$

We consider 3 cases to compute the approximate online solution and the corresponding relative error.

- **Case 1:** In this case, the reduced-order solution, v_r, is obtained by solving the global reduced model generated by global POD basis functions computed offline, i.e., by ψ^G_{off} only without adding any online global basis functions. The error in this case is plotted in Figure 5.1. We see that the error in this case has jumps at $t \in \{25, 40\}$. These jumps are due to having different source term f at these time instants, which change the flow field drastically.

- **Case 2:** For this case, we use global error indicator to monitor the accuracy of the approximate solution. The global error indicator depends on the global residual given by (5.6). A new global online basis function, ψ^G_{on}, is then added to the offline POD basis functions whenever the adaption is need. That is the online solution belongs to $\{\psi^G_{\text{off}}, \psi^G_{\text{on}}\}$. The error of this case is shown in Figure 5.3. The POD subspace is updated by adding one global online basis function at time instants $\{25, 40\}$, where the error had jumps in the first case.

- **Case 3:** We use local error indicator as discussed in Section 5.3 and use the global-local adaptive method described in Section 5.3.1 for updating the POD subspace. In this case the global online basis functions are computed using

local multiscal basis functions. At $t = 25$, the number of local basis functions increased from 1260 to 2520 and to 3150 at $t = 40$. This increasing is resulting from adding some local online multiscale basis functions at some coarse blocks, where the error is greater than a specific error tolerance. Once the multiscale basis functions is updated it then used to compute a new global online basis function to be added to the POD space generated offline. Figure 5.4 represents the error in this case.

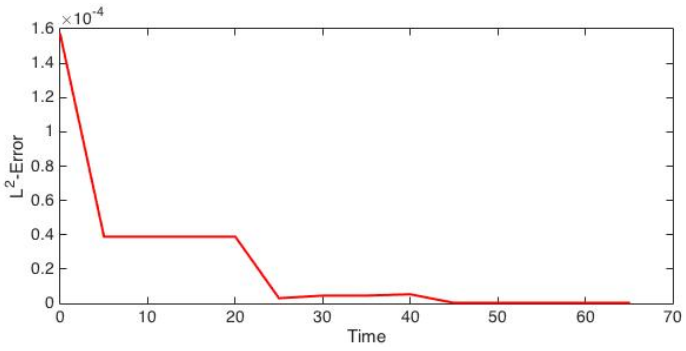


Figure 5.3: L^2- Error of the reduced-order solution in Case 2, $v_r \in \{\psi^G_{\text{off}}, \psi^G_{\text{on}}\}$. ψ^G_{on} is computed using global error indicator and global residual.

In Table 5.1, we list the error computed in the above three cases at the time instants $\{25, 40\}$ when the error in case 1 has jumps and the POD subspace is updated in case 2 and 3. The first column represents the time instants. The other columns are the error corresponding to the three cases. We observe that the error drops from 10^{-1} in the first case to 10^{-6} in case 2 and 3 at $t = 25$. At $t = 40$ the error in case 2 drops to 10^{-7} and in case 3 the error is 10^{-6}. Although Case 2 gives a good accuracy, it is expensive since we need to solve Equation (5.6) for new global online basis function in the fine-grid dimension. To conclude, among the three cases

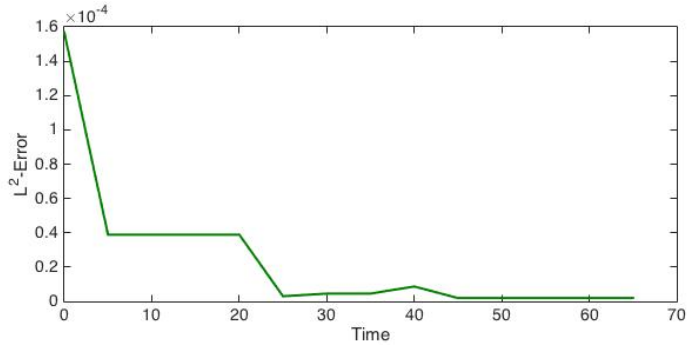


Figure 5.4: L^2- Error of the reduced-order solution in Case 3, $v_r \in \{\psi^G_{\text{off}}, \psi^G_{\text{on}}\}$. ψ^G_{on} is computed using local error indicator and local residual.

the global-local online adaptive method used in Case 3 provides an inexpensive way to improve the accuracy of the approximate velocity solution.

t	**Case** 1	**Case** 2	**Case** 3
25	0.2017	$3.0134e-6$	$3.0153e-6$
40	0.3026	$3.0903e-7$	$1.9406e-6$

Table 5.1: L^2 relative error for the online approximate solution at the time instants $t = 25$ and $t = 40$ in Cases 1, 2 and 3.

Example 5.4.2. Our objective in this example is to show the effect of the initial number of offline global basis functions on the number of time steps when updating the online solution space is needed. In our simulation, the computational domain is, also, $[0, 1] \times [0, 1]$. We divide this domain into 8×8 coarse grids and divide each coarse grid into 32×32 fine grids. The source term and boundary conditions are the same as in Example 5.4.1. The permeability field, κ_2, considered for this example is shown in Figure 5.5. In this example, we construct the global offline space using $1, 2$ and 3 POD modes. Therefore, the dimension of the initial global online space is

1, 2 and 3, respectively. Then, we apply the global-local online adaptive method to update the global online space, in other words, to add more global online basis.

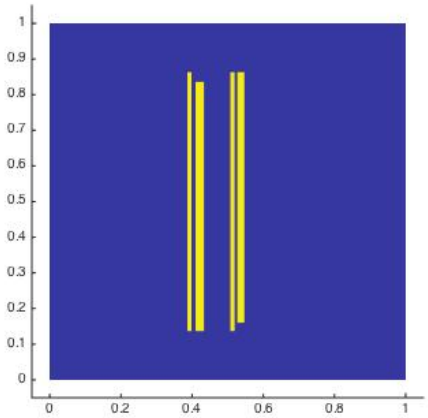


Figure 5.5: Permeability field κ_2.

We observe that using only one POD mode to construct the offline global space yields to more adaption steps during the time interval. For example, in this case, the online global space is updated by adding one online global basis at each time step. This is due to the initial poor choice of snapshots used. In Figure 5.6, we plot the L^2- error of the reduced order solution in case of using one offline global basis. This figure shows that even with the adaptive enrichment of the online space, the accuracy is not satisfied comparing with the other cases (see Figures 5.7 and 5.8). In contrast, using two or three POD modes for constructing the offline global space reduces the number of adaption steps to two times. In addition, the accuracy of the approximate solution drops from 10^{-1}, in case of using one offline global basis, to 10^{-5}. We, also, observe that the number of the adaption steps is still two even with using further

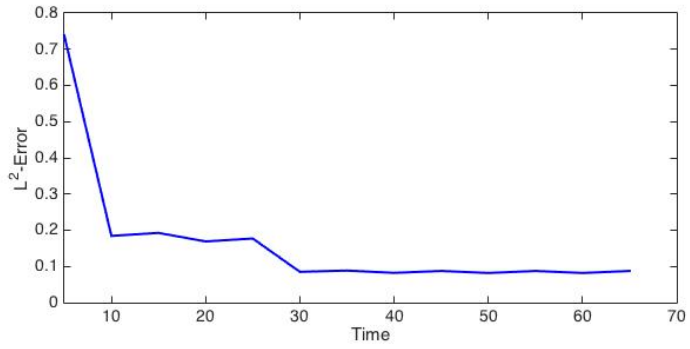


Figure 5.6: L^2- Error of the reduced order solution using one POD mode for the initial online global space.

number of global offline basis. However, the accuracy of the approximate solution when using three POD modes is better than in case of using two POD modes. This shows that the number of the offline global basis functions need to be chosen carefully to efficiently employ the global-local online adaptive method.

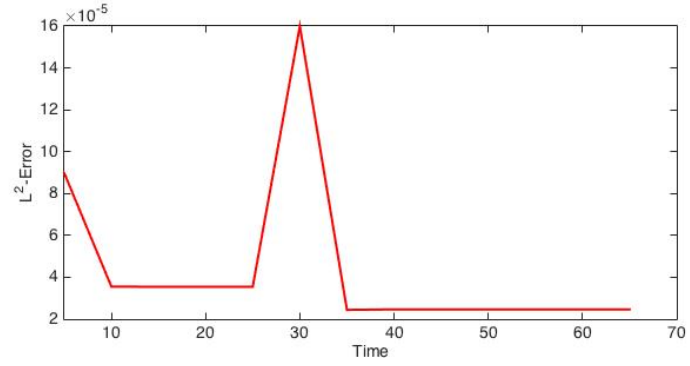


Figure 5.7: L^2- Error of the reduced order solution using two POD modes for the initial online global space.

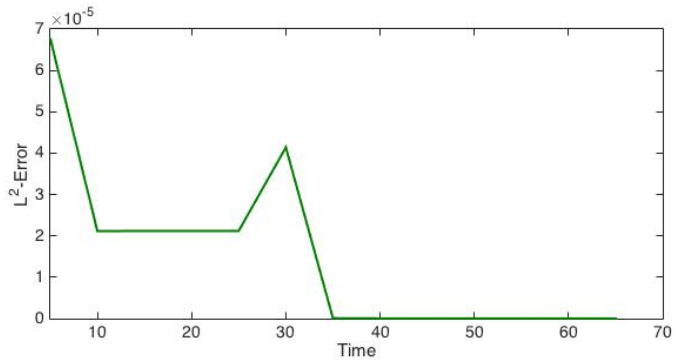


Figure 5.8: L^2- Error of the reduced order solution using three POD modes for the initial online global space.

6. CONCLUSION

In this dissertation, we present a global-local nonlinear model reduction approach to reduce the computational cost for solving high-contrast nonlinear parabolic PDEs in Section 3 and for solving nonlinear Forchheimer flow in Section 4. The reduction is achieved through two main stages: offline and online.

In Section 3, the generalized multiscale finite element method (GMsFEM) is used in the offline step to represent the coarse-scale solutions 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-scale 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 the GMsFEM, the computations of the global modes can be very expensive. Combining both local and global model 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 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 the proposed model reduction approach with respect to variations in initial conditions, permeability fields, nonlinear function's parameters, and forcing

terms is demonstrated.

In Section 4, we employ the mixed GMsFEM in the offline stage to represent coarse-scale velocity solution. As in Section 3, we approximate the nonlinear function using local DEIM when solving the offline reduced model. The main contribution in this stage is to construct the snapshot space and the offline space for the velocity field. Then, a multiscale proper orthogonal decomposition technique is used as a global reduction method to find the best subspace of offline (multiscale) space. We then use DEIM in the global domain when solving the online problem to circumvent the issue of the fine-grid computations associated with the projected nonlinear terms. Convergence analysis and some numerical experiments are presented to show the performance of our method. Additionally, we present an enrichment algorithm to adaptively improve the offline space by adding some local online multiscale basis functions at some selected time steps. The numerical tests show the effect of this enrichment method and the initial dimension of the offline space.

In the last section, we propose a global-local online adaptive method that is used to adapt the global POD solution space during the online stage. The purpose of this adaption is to improve the accuracy of the online approximate solution in some applications where the global POD basis functions do not contain sufficient information for all time instants. In this method, we add new global online basis functions to the global POD modes computed offline. We employ local techniques to reduce the computational cost for evaluating the residual and for computing the online global basis functions. We apply this method to Forchhiemer flow and present numerical examples to compare the accuracy of the proposed method vs the accuracy of using the static POD space without updating.

REFERENCES

[1] Jorg E Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Modeling & Simulation*, 2(3):pp. 421–439, 2004.

[2] Jorg E Aarnes and Yalchin Efendiev. Mixed multiscale finite element methods for stochastic porous media flows. *SIAM Journal on Scientific Computing*, 30(5):pp. 2319–2339, 2008.

[3] Imran Akhtar, Ali H Nayfeh, and Calvin J Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theoretical and Computational Fluid Dynamics*, 23(3):pp. 213–237, 2009.

[4] Imran Akhtar, Zhu Wang, Jeff Borggaard, and Traian Iliescu. A new closure strategy for proper orthogonal decomposition reduced-order models. *Journal of Computational and Nonlinear Dynamics*, 7(3):eid 034503, 2012.

[5] Ibrahim Y Akkutlu, Yalchin Efendiev, and Maria Vasilyeva. Multiscale model reduction for shale gas transport in fractured media. *Computational Geosciences*, 20(5):pp. 953–973, 2016.

[6] Manal Alotaibi, Victor M Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Global–local nonlinear model reduction for flows in heterogeneous porous media. *Computer Methods in Applied Mechanics and Engineering*, 292:pp. 122–137, 2015.

[7] Todd Arbogast. Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems. *SIAM Journal on Numerical Analysis*, 42(2):pp. 576–598, 2004.

[8] Todd Arbogast and Kirsten J Boyd. Subgrid upscaling and mixed multiscale finite elements. *SIAM Journal on Numerical Analysis*, 44(3):pp. 1150–1171, 2006.

[9] Todd Arbogast, Gergina Pencheva, Mary F Wheeler, and Ivan Yotov. A multiscale mortar mixed finite element method. *Multiscale Modeling & Simulation*, 6(1):pp. 319–346, 2007.

[10] Eugenio Aulisa, Lidia Bloshanskaya, Yalchin Efendiev, and Akif Ibragimov. Upscaling of Forchheimer flows. *Advances in Water Resources*, 70:pp. 77–88, 2014.

[11] Eugenio Aulisa, Lidia Bloshanskaya, Luan Hoang, and Akif Ibragimov. Analysis of generalized Forchheimer flows of compressible fluids in porous media. *Journal of Mathematical Physics*, 50(10):eid 103102, 2009.

[12] Henry P Bakewell Jr and John L Lumley. Viscous sublayer and adjacent wall region in turbulent pipe flow. *Physics of Fluids (1958-1988)*, 10(9):pp. 1880–1889, 1967.

[13] Michel Bergmann, C-H Bruneau, and Angelo Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):pp. 516–538, 2009.

[14] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):pp. 539–575, 1993.

[15] Alain Bourgeat, Eduard Marušić-Paloka, and Andro Mikelić. Weak nonlinear corrections for Darcys law. *Mathematical Models and Methods in Applied Sciences*, 6(08):pp. 1143–1155, 1996.

[16] Victor M Calo, Yalchin Efendiev, Juan Galvis, and Mehdi Ghommem. Multiscale empirical interpolation for solving nonlinear PDEs. *Journal of Computational Physics*, 278:pp. 204–220, 2014.

[17] Ho Y Chan, Eric T Chung, and Yalchin Efendiev. Adaptive mixed GMsFEM for flows in heterogeneous media. *arXiv preprint arXiv:1507.01659*, 2015.

[18] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):pp. 2737–2764, 2010.

[19] Gregori A Chechkin, Andre L Piatnitskiĭ, and Alekseĭ S Shamev. *Homogenization: methods and applications*, volume 234 of Translations of Mathematical Monographs. American Mathematical Society, 2007.

[20] C-C Chu, Ivan Graham, and T-Y Hou. A new multiscale finite element method for high-contrast elliptic interface problems. *Mathematics of Computation*, 79(272):pp. 1915–1955, 2010.

[21] Eric Chung, Yalchin Efendiev, and Thomas Y Hou. Adaptive multiscale model reduction with generalized multiscale finite element methods. *Journal of Computational Physics*, 320:pp. 69–95, 2016.

[22] Eric T Chung, Yalchin Efendiev, and Chak Shing Lee. Mixed generalized multiscale finite element methods and applications. *Multiscale Modeling & Simulation*, 13(1):pp. 338–366, 2015.

[23] Eric T Chung, Yalchin Efendiev, and Wing T Leung. Residual-driven online generalized multiscale finite element methods. *Journal of Computational Physics*, 302:pp. 176–190, 2015.

[24] Eric T Chung, Wing T Leung, and Maria Vasilyeva. Mixed GMsFEM for second order elliptic problem in perforated domains. *Journal of Computational and Applied Mathematics*, 304:pp. 84–99, 2016.

[25] AE Deane, IG Kevrekidis, G Em Karniadakis, and SA Orszag. Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 3(10):pp. 2337–2354, 1991.

[26] LJ Durfolsky. Numerical calculation of equivalent grid block permeability tensors of heterogeneous porous media. *Water Resources Research*, 27(5):pp. 699–708, 1991.

[27] Yalchin Efendiev, Juan Galvis, and Eduardo Gildin. Local–global multiscale model reduction for flows in high-contrast heterogeneous media. *Journal of Computational Physics*, 231(24):pp. 8100–8113, 2012.

[28] Yalchin Efendiev, Juan Galvis, Guanglian Li, and Michael Presho. Generalized multiscale finite element methods. Nonlinear elliptic equations. *Communications in Computational Physics*, 15(03):pp. 733–755, 2014.

[29] Yalchin Efendiev, Juan Galvis, and Xiao-Hui Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *Journal of Computational Physics*, 230(4):pp. 937–955, 2011.

[30] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4 of Surveys and Tutorials in the Applied Mathematical Sciences. Springer Science & Business Media, New York, 2009.

[31] Yalchin Efendiev, Thomas Y Hou, Victor Ginting, et al. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):pp. 553–589, 2004.

[32] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling & Simulation*, 8(4):pp. 1461–1483, 2010.

[33] Juan Galvis and Yalchin Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Modeling & Simulation*, 8(5):pp. 1621–1644, 2010.

[34] Cristiano Rodrigues Garibotti and M Peszyńska. Upscaling non-Darcy flow. *Transport in Porous Media*, 80(3):pp. 401–430, 2009.

[35] Mehdi Ghommem, Imran Akhtar, and Muhammad R Hajj. A low-dimensional tool for predicting force decomposition coefficients for varying inflow conditions. *Progress in Computational Fluid Dynamics, an International Journal*, 13(6):pp. 368–381, 2013.

[36] Mehdi Ghommem, Michael Presho, Victor M Calo, and Yalchin Efendiev. Mode decomposition methods for flows in high-contrast porous media. Global–local approach. *Journal of Computational Physics*, 253:pp. 226–238, 2013.

[37] Tiziana Giorgi. Derivation of the Forchheimer law via matched asymptotic expansions. *Transport in Porous Media*, 29(2):pp. 191–206, 1997.

[38] Alexander Hay, Imran Akhtar, and Jeff T Borggaard. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 68(1):pp. 122–134, 2012.

[39] Alexander Hay, Jeff Borggaard, Imran Akhtar, and Dominique Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of Computational Physics*, 229(4):pp. 1327–1352, 2010.

[40] Philip Holmes, John L Lumley, and Gal Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 1998.

[41] Ulrich Hornung. *Homogenization and porous media*, volume 6 of Interdisciplinary Applied Mathematics. Springer-Verlag, New York, 1997.

[42] Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1):pp. 3–24, 1998.

[43] O Iliev, R Lazarov, and Joerg Willems. Variational multiscale finite element method for flows in highly porous media. *Multiscale Modeling & Simulation*, 9(4):pp. 1350–1372, 2011.

[44] P Jenny, SH Lee, and HA Tchelepi. Multiscale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187(1):pp. 47–67, 2003.

[45] Mohammad Karimi-Fard, Louis J Durlofsky, et al. Detailed near-well Darcy-Forchheimer flow modeling and upscaling on unstructured 3d grids. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009.

[46] Zhendong Luo, Jing Chen, IM Navon, and Xiaozhong Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 47(1):pp. 1–19, 2008.

[47] Eduard Marušić-Paloka and Andro Mikelić. The derivation of a nonlinear filtration law including the inertia effects via homogenization. *Nonlinear Analysis: Theory, Methods & Applications*, 42(1):pp. 97–137, 2000.

[48] Pingbing Ming, Pingwen Zhang, et al. Analysis of the heterogeneous multiscale method for elliptic homogenization problems. *Journal of the American Mathematical Society*, 18(1):pp. 121–156, 2005.

[49] M-L Rapún, F Terragni, and JM Vega. Adaptive POD-based low-dimensional modeling supported by residual estimates. *International Journal for Numerical Methods in Engineering*, 104(9):pp. 844–868, 2015.

[50] Lawrence Sirovich. Turbulence and the dynamics of coherent structures part i: coherent structures. *Quarterly of Applied Mathematics*, 45(3):pp. 561–571, 1987.

[51] Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *Journal of Computational Physics*, 230(1):pp.126–146, 2011.

[52] Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:pp. 10–26, 2012.

[53] E Weinan, Bjorn Engquist, et al. The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1):pp. 87–132, 2003.

[54] Mary Fanett Wheeler, Guangri Xue, and Ivan Yotov. A multiscale mortar multipoint flux mixed finite element method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(4):pp. 759–796, 2012.

[55] Xiao-Hui Wu, Y Efendiev, and Thomas Y Hou. Analysis of upscaling absolute permeability. *Discrete and Continuous Dynamical Systems Series B*, 2(2):pp. 185–204, 2002.