LOCAL-GLOBAL MODEL REDUCTION TECHNIQUES USING BALANCED TRUNCATION

A Dissertation

by

ANASTASIYA N. PROTASOV

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
	Alexei Poltoratski
Head of Department,	Emil Straube

May 2016

Major Subject: Mathematics

ABSTRACT

Many applications such as porous media and material science possess multiscale nature of media properties and high number of variables. As a result, forward problems on todays computer architectures require high cost of computations. The dissertation is devoted to creating novel model reduction techniques in order to carry out these prohibitively expensive computations.

Model reduction techniques can be divided into two categories, local reduced-order modeling techniques and global reduced-order modeling techniques. Local reduced-order modeling techniques include many multiscale and homogenization type methods. Multiscale techniques provide significant computational savings since the same multiscale basis functions are used for all forward simulations. While homogenization methods allow reducing cost of computations by solving cell problems with varying order of accuracy. As for global model reduction technique, we consider the earlier approach called Balanced Truncation (BT), where the system is written in terms of a mapping from input to output. In the dissertation local model reduction techniques and the global model reduction technique are combined.

Local-global model reduction techniques are designed for different problem settings. First, we examine the flow in porous media with separable scales. We employ hierarchical approaches for solving local problems. Then the obtained coarse-grid models are coupled with BT approach.

Next problem formulation describes the flow in porous media without scale separation. Two cases of media properties are considered: general heterogenous media with a parameter and a time-varying heterogeneous media, where the media properties depend on time. For these type of problems we use appropriate form of BT and

combine it with the offline - online GMsFEM procedure.

Finally, we consider a coupled flow and transport problem, where the transport equation is convection dominated. Both flow and transport equations are discretized on a coarse grid using GMsFEM. We bring together the mixed coarse-grid discretization of convection-diffusion equation and BT approaches to obtain an accurate local-global model reduction.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Prof. Yalchin Efendiev for the guidance, enormous help and valuable suggestions. Without his encouragement, experience and special topic classes on "Multiscale methods on partial differential equations" this research can't be performed.

I am grateful to Texas A&M University, Department of Mathematics for awarding me the Teacher Assistant position. I appreciate National Science Foundation for providing such great fellowships as IGERT Fellowship New Mathematical Tools for Next-generation Materials and EAPSI fellowship to spend summer 2012 in Singapore. I am highly indebted to a National Priorities Research Program grant NPRP 701482-1-278 from QNRF, Army and DOE for providing financial support of my research. I am benefited of being a member of Exxon Mobil project "Development of Novel Nonlinear Model Reduction Techniques for Production Optimization and Parameter Estimation in Heterogeneous Porous Media Flow".

I am beholden to the tremendous support of my co-advisor from Petroleum Engineering Department Prof. Eduardo Gildin. I appreciate the help of petroleum engineering students, especially Reza Ghasemi, Sardar Afra and Yuhe Wang.

I would like to extend my appreciation to Prof. Juan Galvis and Postdoctoral scholar Michael Presho for colossal support with numerical examples and openness for discussions. I appreciate enormous support of Prof. Donald Brown with discussing homogenization techniques. I would like to thank Wing Tat Leung for sharing numerical results for coupled flow and transport system, and immediate openness to all my questions.

Additionally, I would like to thank the visitor Dr. Mehdi Ghommem for valuable suggestions. I would like to thank Xia Bing Xing for sharing numerical results and for the support during Nanyang Technological University visit.

I would like to thank all my family for their love and support.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION*

Many relevant engineering and physical problems in porous media flow involve model equations that are inherently multiscale in nature. In particular, the underlying properties of the reservoir, such as porosity and permeability, often exhibit heterogeneous and high-contrast behavior (see , e.g. [16, 22, 29, 50, 56]). For example, flow conduits (such as fractures or channels) represent regions of media whose permeability may be many orders of magnitude larger than surrounding regions. On the other hand, flow barriers (such as deformation bands or compressed shale) are modeled by a permeability that is many orders of magnitude smaller than surrounding regions. Furthermore, the description of a porous medium typically involves some level of uncertainty that must be incorporated into the model. In turn, solving such problems on a fully-resolved scale may become a prohibitively expensive task, especially when numerous computations must be performed for uncertainty quantification, model optimization, and/or sensitivity analysis. As a result, the need for more computationally efficient and suitably accurate solution techniques for single-phase flow models as well as for coupled flow and transport models is of particular interest in this work.

In the dissertation, the first problem we consider is devoted to a single-phase flow model with scale separation (Section 3). For problems with scale separation, one can use homogenization techniques to derive macroscopic equations. Homogenization techniques are limited to problems with scale separation and are more effective compared to multiscale methods, which are designed for problems without

*Parts of this section have been reprinted from E. Gildin, M. Ghasemi, A. Protasov, Y. Efendiev, Nonlinear Complexity Reduction for Fast Simulation of Flow in Heterogeneous Porous Media, paper SPE 163618 presented at SPE Annual Technical Conference and Exhibition, Woodlands, Texas, USA, 18-20 February, 2013.

scale separation. In this section, we combine Balanced Truncation (BT) and the homogenization. The global reduced-order model is formulated for the homogenized equation and we study the convergence of the reduced-order homogenized model and the fine-scale detailed model analytically.

Over the past two decades, coarse-grid multiscale methods have been used as suitable alternatives for solving fully-resolved computational models (see, e.g. [1, 6, 27, 41, 42, 43]). This class of methods involves the construction of a coarse solution space in which solutions are sought within the span of a set of multiscale basis functions. For this work, we follow the framework of the Multiscale Finite Element Method (MsFEM), where the basis functions are independently pre-computed through solving a set of localized problems that resemble the global operator [41]. The multiscale basis functions are then consolidated in order to form a coarse space in which a specified global formulation is used to construct a reduced-order solution. Due to the localized fine-grid computations, relevant behavior of the global system is inherently embedded into the multiscale basis functions. As such, these types of multiscale spaces are equipped with a direct method of projecting the coarse-grid solutions to the fine grid. Furthermore, as the basis functions are independent, MsFEM (and related approaches) offers a more efficient solution technique in a parallel setting. While standard multiscale methods have been effective for a variety of applications [26, 27, 28, 30], recent literature has offered a generalized framework for the enrichment of coarse solution spaces. In particular, the Generalized Multiscale Finite Element Method (GMsFEM) is a robust solution technique where the standard coarse-grid solution spaces from MsFEM may be systematically enriched to converge to the fine grid solution [7, 24, 25]. The additional basis functions are constructed through solutions of localized eigenvalue problems that are used in order to further capture the underlying behavior of the system. Due to this type of

construction, GMsFEM offers an approach where the number of basis functions and resulting coarse space dimension may be carefully chosen to adhere to a desired level of accuracy. However, since the localized eigenvalue problems scale with respect to the fine mesh, in this work we wish to avoid direct eigenvector calculations when sources of uncertainty are incorporated into the problem formulation.

When considering parameter-dependent problems that may require repeated calculations, the coarse space enrichment may be split into two stages for an added boost in efficiency. More specifically, in the Section 4, we implement an offline-online procedure in which the basis function computations and coarse space construction are executed (see also [9, 12, 49, 51]). The main goal of doing so is to allow for the efficient construction of an online space for each fixed parameter value; in turn, avoiding the need to directly recompute the eigenvectors for separate realizations. The first stage of the computation involves forming a larger-dimensional (as compared to the online space) parameter-independent offline space. The offline space accounts for a suitable range of parameter values that may be used in the online stage, and the construction constitutes a one-time preprocessing step. The offline space is created by initially producing a set of "snapshot" functions in which a number of localized problems are solved on each coarse subdomain for a specified number of parameter values. The offline space is then obtained by solving a set of localized eigenvalue problems that use averaged parameter quantities within the space of snapshots. The use of averaged quantities at this point of the offline stage is what makes this computation parameter-independent, and suitable as a pre-processing step. At the online stage, we finally solve analogous reduced-order eigenvalue problems within the offline space for a given parameter value in order to construct the desired online coarse space. In order to perform global model reduction we use BT approach on a coarse-grid problem obtained by GMsFEM.

In Section 5 we develop a method for a single-phase flow equation in time-varying heterogeneous media. The above approach was adapted to a time-varying case by using a special form of BT developed in [53]. The main advantage of this method is the fact that it doesn't require finding expensive solutions of Lyapunov equations. We offer a novel technique based on described GMsFEM approach coupled with the time-varying BT. Numerical results validate the computational efficiency of the suggested global-local model reduction method.

In Section 6, we extend above concepts and consider coupled flow and transport processes. Our previous studies in Section 3, 4, and 5 consider elliptic type equations, while in Section 6, we also consider convection-dominated transport equation, which is coupled to the elliptic flow equations. The global model reduction approach is proposed based on BT and using local model reduction techniques. For local model reduction technique, we use a mixed GMsFEM for both flow and transport equations. The mixed finite element approach gives the mass conservation, which is very important for subsurface applications. We present numerical results, which show that with only a few degrees of freedom one can achieve accurate approximations.

2. PRELIMINARIES

2.1 First model problem

For our first model problem, we concentrate on a single-phase parameter-dependent flow equation in heterogeneous media. We assume that $k(x;\mu)$ is a high contrast coefficient (i.e., the ratio $k_{\max}(x)/k_{\min}(x)$ is large) and an input u (controllable source term) varies on a coarse grid and μ is used to represent the parameter dependence. Next, we introduce a mathematical formulation of the problem. We search for a solution $p(x,t)$ of the system

$$\frac{\partial p}{\partial t} = \operatorname{div}\big(\kappa(x;\mu)\nabla p\big) + f(u) \quad \text{in } D, \tag{2.1a}$$

$$p = g \quad \text{on } \partial D, \tag{2.1b}$$

where the input (or control) $u \in \mathbb{R}^m$ is given, D is a domain in $\mathbb{R}^2$, the right hand side $f(u)$ is square integrable and depends linearly on u, g denotes the boundary condition. The above model is solved along with a specified initial condition $p(x,0) = p_0(x)$. We also suppose that an output (measurable quantity) is given by an operator acting on the state (solution) as $q = Cp = (C_1(p), \ldots, C_n(p))$. Modeling subsurface reservoir flow can be considered as an example, where an input is described by well rates and an output is the pressure at the wells. In this case, C can be considered as a selection matrix of grid cells that have been allocated to wells.

In order to solve Equation (2.1), in Section 4 and Section 5, we will use the GMsFEM, which is based on continuous Galerkin framework. We partition the domain D into a set of finite elements (e.g., quadrilaterals or triangles) into a coarse grid which we denote by $\mathcal{T}^H$. We use $\{y_i\}_{i=1}^{N_v}$ to denote the vertices of the coarse

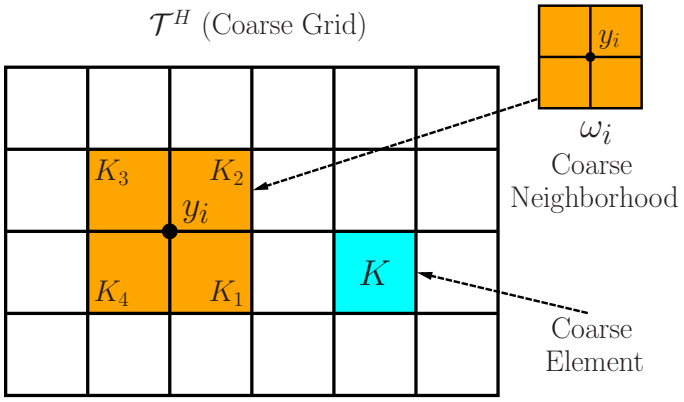


Figure 2.1: Illustration of the coarse grid, coarse elements, and a coarse neighborhood

mesh, and define the neighborhood of a node y_i by

$$\omega_i = \bigcup \{K_j \in \mathcal{T}^H;\ y_i \in \overline{K_j}\}. \tag{2.2}$$

See Figure 2.1 for an illustration of a coarse neighborhood. In addition, we assume that there is a refinement of the coarse mesh and we denote this fine grid by $\mathcal{T}^h$.

We denote the space of linear finite element functions subordinated to $\mathcal{T}^h$ by V^h and we search for $p(x,t) \in V^h$ such that

$$\int_D \frac{\partial p}{\partial t} v = -\int_D \kappa(x;\mu) \nabla p \, \nabla v + \int_D f(u)\, v \ \text{ for all } v \in V^h. \tag{2.3}$$

The above equation yields the discrete form

$$M p_t = -A(\mu) p + Bu, \tag{2.4}$$

where $M := [m_{ij}] = \displaystyle\int_D \phi_i \phi_j$ is a mass matrix, $A(\mu) := [a(\mu)_{ij}] = \displaystyle\int_D \kappa(x;\mu) \nabla \phi_i \, \nabla \phi_j$ is a stiffness matrix, $Bu := [(bu)_i] = \displaystyle\int_D f(u)\, \phi_i$, and ϕ_i is used to denote standard

bilinear basis functions that span V^h. Given matrices M, $A(\mu)$, and B we solve (2.4) using a backward Euler, implicit time marching scheme

$$p^{k+1} = \big(M + \Delta t A(\mu)\big)^{-1} M p^k + \big(M + \Delta t A(\mu)\big)^{-1} \Delta t B u, \tag{2.5}$$

where p^k denotes the discrete pressure solution at the k time level. Using N_f to denote the number of fine grid points, we note that the square matrices are of size $N_f \times N_f$. A 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 (a local approach)

Below we briefly discuss the Generalized Multiscale Finite Element Method (GMsFEM) that we use in Sections 4-6. A more detailed description will be delegated to Section 4.2. As GMsFEM hinges on a set of localized spectral problems that are used to construct a suitable coarse scale solution space, we refer to GMsFEM as a local model reduction technique. The main purpose of GMsFEM is to derive a suitable reduced order model of the original system. In the context of GMsFEM, this is done through the systematic construction of a coarse solution space that is formed through a series of independent, local computations [25]. In particular, standard multiscale spaces may be systematically enriched through a careful choice of localized eigenvalue problems that capture the underlying structure of the system. Formally speaking, GMsFEM yields a transformation matrix $R \in \mathbb{R}^{N_f \times N_c}$ such that we may express Equation (2.4) as

$$R^T M R (p_c)_t = -R^T A(\mu) R p_c + R^T B u \tag{2.6}$$

or

$$(p_c)_t = -A^c(\mu)p_c + B^c u, \tag{2.7}$$

where $A^c(\mu) = (R^T M R)^{-1} R^T A(\mu) R$, $B^c = (R^T M R)^{-1} R^T B$, and p_c denotes the coarse scale solution. Similarly, an approximation to the measurable output may be written as $q^c = CRp_c = C^c p_c$. The square matrices in (2.6) are of size $N_c \times N_c$, where $N_c \ll N_f$ denotes the size of the respective coarse solution space. Thus, Equation (2.7) (or Equation (2.6)) serves as a local model reduction of Equation (2.4).

2.3 Homogenization (a local approach)

Next, we introduce a mathematical formulation of the problem with separable scales. For a given input $u \in R^m$, we seek of state $p(t, \cdot) \in H^1(D)$ such that

$$\frac{dp}{dt} = div(\kappa(x, \frac{x}{\epsilon})\nabla p) + f(u). \tag{2.8}$$

We assume D is a domain in R^N (R^2 or R^3) with a small period $\epsilon > 0$ compared to the size of the whole domain D. The coefficient $\kappa(x, y)$ is assumed to be periodic with respect to y and the observed quantity $q = \mathcal{C}p$, where $\mathcal{C}(p) = (\mathcal{C}_1(p), \ldots, \mathcal{C}_n(p))$. Moreover, we assume the following boundary conditions: $p = g \ \ on \ \ \partial D, g \in H^1(\partial D)$ and an initial condition $p(x, y, 0) = 0$.

Next, we briefly discuss the idea of homogenization. We introduce a microscopic variable or fast periodic variable $y = \frac{x}{\epsilon}$ and a macroscopic variable or slow periodic variable x.

We assume the solution can be expanded as the following sum

$$p^\epsilon(x,t) = \sum_{i=0}^{+\infty} \epsilon^i p_i = p_0(x, \frac{x}{\epsilon}, t) + \epsilon p_1(x, \frac{x}{\epsilon}, t) + \epsilon^2 p_2(x, \frac{x}{\epsilon}, t) + \epsilon^3 p_3(x, \frac{x}{\epsilon}, t) + ..., \tag{2.9}$$

where the functions $p_i(x, y, t), i = 1, 2, ...$ are periodic in y with period 1. It is shown in Section 3.1 that the homogenized equation can be written as

$$\begin{aligned} \frac{dp_o}{dt} - div_x \kappa^* \nabla_x p_0 = f(u) \ \ in \ \ D, \\ p_0 = g_0 \ \ on \ \ \partial D, \\ p_0(x, 0) = 0, \end{aligned} \tag{2.10}$$

where the homogenized coefficient is defined as $\kappa^*_{ij} = \int_Y \big[\kappa(x, y)(e_i + \nabla_y N_i)(e_j + \nabla_y N_j)\big](x, y)dy$ and $N(x, y)$ is a solution of a cell problem defined by:

$$\begin{aligned} -div_y \kappa(x, y)(e_i + \nabla_y N_i(x, y)) = 0 \ \ in \ \ Y, \\ N(x, y) \ \ is \ \ periodic \ \ in \ \ Y. \end{aligned} \tag{2.11}$$

As usual, $\{e_i\}_{i=1}^N$ denotes standard basis in R^N.

Moreover, it was shown in Section 3.1 that the expansion can be written as $p^\epsilon(x, t) = p_0(x, \frac{x}{\epsilon}, t) + \epsilon p_1(x, \frac{x}{\epsilon}, t) + \epsilon^2 p_2(x, \frac{x}{\epsilon}, t) + \epsilon^3 p_3(x, \frac{x}{\epsilon}, t) + ... = p_0(x, t) + \epsilon N(x, y) \cdot \nabla p_0 + \theta$, where $||\theta||_{H^1(D)} = O(\epsilon^{1/2})$.

2.4 Balanced Truncation (a global approach)

Below we describe a general idea of model reduction technique called Balanced Truncation (BT) followed from the theory summarized in [38]. BT can be successfully applied in order to approximate the input-output behavior of linear systems. We consider the following time-invariant system

$$\dot{p}(t) = \tilde{A}p(t) + \tilde{B}u(t) \ \ \text{for} \ \ t \in (0, \infty), \tag{2.12a}$$

$$p(0) = p_0, \tag{2.12b}$$

$$q = \tilde{C}p(t), \tag{2.12c}$$

where p_0 is an initial condition and q is an output.

Now we introduce Laplace transform for a time-varying vector p(t) by

$$L[p](s) = \int_0^\infty e^{-st} p(t)dt \quad \text{for } s \in R.$$

Similarly, we can write $L[u](s) = \int_0^\infty e^{-st}u(t)dt$ and $L[q](s) = \int_0^\infty e^{-st}q(t)dt$.

Notice that $L[\dot{p}](s) = \int_0^\infty e^{-st}\dot{p}(t)dt = -\int_0^\infty (-s)e^{-st}p(t)dt + (e^{-st}p(t)) \mid_{t=0}^{t=\infty}$.

Next, we can write the Laplace transform of the system (2.12) as $sL[p](s) - p(0) = \tilde{A}L[p](s) + \tilde{B}L[u](s)$ or $L[p](s) = (sI - \tilde{A})^{-1}p(0) + (sI - \tilde{A})^{-1}\tilde{B}L[u](s)$.

Finally, we can find the output as $L[q] = \tilde{C}L[p](s) = \tilde{C}(sI - \tilde{A})^{-1}p(0) + \tilde{C}(sI - \tilde{A})^{-1}\tilde{B}L[u](s)$.

In a case of $p(0) = 0$ we get $L[q] = G(s)L[u](s)$, for as called transfer matrix of the system $G(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B}$. Then, we call a state-space modal $(\tilde{A}, \tilde{B}, \tilde{C}, 0)$ satisfying

$$G(s) = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & 0 \end{array}\right]. \tag{2.13}$$

a realization of G(s).

Next, we define the observability Gramian as $W_o(t) = \int_0^t e^{s\tilde{A}^T}\tilde{C}^T\tilde{C}e^{s\tilde{A}}ds$ and the controllability Gramian as $W_c(t) = \int_0^t e^{s\tilde{A}}\tilde{B}\tilde{B}^T e^{s\tilde{A}^T}ds$. It can be shown that the controllability and observability Gramians satisfy the following Lyapunov equations respectively,

$$\tilde{A}\, W_c + W_c\, \tilde{A}^T + \tilde{B}\tilde{B}^T = 0 \quad \text{and} \quad \tilde{A}^T W_o + W_o\, \tilde{A} + \tilde{C}^T\tilde{C} = 0. \tag{2.14}$$

Now we compute a basis such that transforms the controllability and observability Gramians to be equal and diagonal. In this case, the quantities $\sigma_i = \sqrt{\lambda_i\,(W_c W_o)}$,

$i = 1, \cdots, n$ are called the Hankel Singular Values. The reduced system G_r can be obtained by selecting the dominant eigenvalues.

In [38] it has been shown that the reduced system $G_r(s)$ is balanced and asymptotically stable, moreover the following error estimate takes place

$$\| G(s) - G_r(s) \|_{\mathcal{H}_\infty} \preceq (\sigma_{r+1} + \cdots + \sigma_n) . \tag{2.15}$$

2.5 Second model problem

We consider a coupled flow and transport system in the following form

$$\frac{\partial c}{\partial t} = G\Delta c - v \cdot \nabla c + f \quad \text{in } D \times (0, T), \tag{2.16a}$$

$$v = -\kappa \nabla p \quad \text{in } D, \tag{2.16b}$$

$$\text{div} v = 0 \quad \text{in } D, \tag{2.16c}$$

where D is a given domain in $\mathbb{R}^2$, G is a diffusivity of the medium, $T > 0$ is a fixed time moment, κ is a conductivity, f is a source term, c is a concentration, and p is a pressure. The boundary conditions are $c(x, t) = 0$ on $\partial D \times [0, T]$, $p(x) = 0$ on ∂D. The initial condition is $c(x, 0) = 0$ in D.

In order to achieve mass conservation, we use mixed finite element methods. We introduce a new auxiliary variable w (flux or flux-concentration), so the system (2.16)

can be written in a mixed formulation

$$\frac{\partial c}{\partial t} = -\nabla \cdot w + f \quad \text{in } D \times (0,T), \tag{2.17a}$$

$$w = -G\nabla c + vc \quad \text{in } D \times (0,T), \tag{2.17b}$$

$$v = -\kappa \nabla p \quad \text{in } D, \tag{2.17c}$$

$$\text{div} v = 0 \quad \text{in } D. \tag{2.17d}$$

We assume that the basis functions for the concentration c and the pressure p are piecewise constant functions on a coarse grid. Following [16], we will construct multiscale basis functions for the flux w and the velocity v.

Below we write a discretized form of the problem (2.17). We assume that the domain D is partitioned in a usual way into a union of rectangles which we denote by $\mathcal{T}^H$. The fine mesh we denote as $\mathcal{T}^h$. We define by $\mathcal{E}^H = \{E_i : 1 \leq i \leq N_e\}$ the set of all edges/faces of elements in $\mathcal{T}^H$, where N_e is the number of coarse edges/faces. We introduce the coarse neighborhood of the edge/face E_i as

$$\omega_i = \{K \in \mathcal{T}^H : \partial K \cap E_i \neq \emptyset\}. \tag{2.18}$$

Figure 2.2 for an illustration of a coarse neighborhood was given at [16], where the coarse mesh is shown by black lines and the fine mesh is shown by grey lines.

We denote the coarse trial space (will be introduced in Section 6) for the pressure p and the concentration c by

$$Q_H = \{\psi : \psi|_{K_i} \in \mathcal{P}_0(K_i) \text{ for all } K_i \in \mathcal{T}^H\}. \tag{2.19}$$

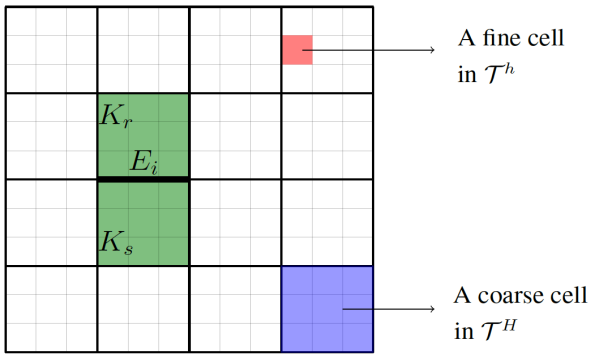


Figure 2.2: Illustration of a course neighborhood ω_i (in green) corresponding to the coarse edge E_i (see [16])

We introduce the multiscale solution space for the flux w and the velocity v as

$$V_H^w = \bigcup_{E_i} \Phi_i^w \quad \text{and} \quad V_H^v = \bigcup_{E_i} \Phi_i^v. \tag{2.20}$$

We discretize the time interval $[0, T]$ uniformly by the points $t^n = n\delta t, \quad n = 0, 1, \cdots, N$, where δt is the time step and $N = T/\delta t$.

Next GMsFEM approximation of (2.17) can be written as following (see [16]). For all $n \geq 1$, find $c_H^n \in Q_H$, $w_H^n \in V_H^w$, $p_H \in Q_H$, $v_H \in V_H^v$, such that

$$\begin{aligned}
\int_D \frac{c_H^n - c_H^{n-1}}{\delta t}\tilde{c} + \int_D (\nabla \cdot w_H^{n-\theta})\tilde{c} &= \int_D f^{n-\theta}\tilde{c} && \text{for } \forall \tilde{c} \in Q_H, \\
\int_D \frac{1}{G} w_H^{n-\theta} \cdot \tilde{w} &= \int_D (\nabla \cdot \tilde{w} + \frac{v_H}{G} \cdot \tilde{w}) c_H^{n-\theta} && \text{for } \forall \tilde{q} \in W_H^w, \\
\int_D \frac{1}{\kappa} v_H \cdot \tilde{v} &= \int_D (\nabla \cdot \tilde{v}) p_H && \text{for } \forall \tilde{v} \in V_H^v, \\
\int_D (\nabla \cdot v_H)\tilde{p} &= 0 && \text{for } \forall \tilde{p} \in Q_H,
\end{aligned} \tag{2.21}$$

where the boundary condition $p_H = g$. For $\theta = 0, 1/2$ and 1, we get the Backward Euler, Crank-Nicolson and Forward Euler Methods, respectively. The brief construction of V_H^q, V_H^v, and W_H^q will be discussed in the section 6 as well as the novel approach based on BT model reduction.

3. COARSE-GRID MODEL REDUCTION WITH BALANCED TRUNCATION FOR FLOWS IN POROUS MEDIA WITH SEPARABLE SCALES

In this section, we combine efficient homogenization methods on a coarse grid with global model reduction techniques. In [22], this is done using multiscale finite element methods; however, the cost of computing local solutions can be expensive due to small-scales and high-contrast in the coefficients. We employ hierarchical approaches for solving local problems. In particular, we propose solving local problems with different resolutions and then consider combining these multi-resolution solutions to achieve an accurate approximation of the effective properties. These coarse-grid models are coupled with BT.

3.1 Homogenization

Below we remind the theory of a periodic homogenization (see [3]). The main goal of the homogenization is to move from a microscopic description of a problem to an averaged/macroscopic formulation. As mentioned before we introduce a microscopic variable or fast periodic variable $y = \frac{x}{\epsilon}$ and a macroscopic variable or slow periodic variable x.

We assume the solution can be expanded as the following sum

$$p^{\epsilon}(x,t) = \sum_{i=0}^{+\infty} \epsilon^i p_i = p_0(x, \frac{x}{\epsilon}, t) + \epsilon p_1(x, \frac{x}{\epsilon}, t) + \epsilon^2 p_2(x, \frac{x}{\epsilon}, t) + \epsilon^3 p_3(x, \frac{x}{\epsilon}, t) + ..., \quad (3.1)$$

where the functions $p_i(x,y,t), i = 0, 1, ...$ are periodic in y with period 1. The following fact is the obvious implementation of chain rule and will be used for ensuing calculations.

Remark. *We assume* $F(x,y,t) \in H^1(D)$, *then* $\frac{d}{dx}F(x, \frac{x}{\epsilon}, t) = \frac{\partial}{dx}F(x,y,t) +$

$\frac{1}{\epsilon}\frac{\partial}{dy}F(x,y,t)$, *where* y *is evaluated at point* $y=\frac{x}{\epsilon}$.

Without loss of generality, we will limit ourselves to a one-dimensional problem. For further calculations we denote the operator $F^\epsilon := \frac{dp_\epsilon}{dt} - div(\kappa(x,\frac{x}{\epsilon})\nabla p_\epsilon) = f(u)$. Using Equation (3.1) and some obvious regrouping, the following expression is obtained:

$$\begin{aligned} F^\epsilon = -\epsilon^{-2}\big[div_y\kappa(x,y)\nabla_y p_0\big](x,\frac{x}{\epsilon},t) - \epsilon^{-1}\big[div_y\kappa(x,y)(\nabla_x p_0+\nabla_y p_1)+ \\ +div_x\kappa(x,y)\nabla_y p_0\big](x,\frac{x}{\epsilon},t) + +\epsilon^0\big(\big[-div_x\kappa(x,y)(\nabla_x p_0+\nabla_y p_1)- \\ -div_y\kappa(x,y)(\nabla_x p_1+\nabla_y p_2)\big](x,\frac{x}{\epsilon},t) + \frac{dp_o}{dt}(x,\frac{x}{\epsilon},t)\big)+ \\ +\sum_{i=1}^{+\infty}\epsilon^i\big[\frac{dp_i}{dt} - div_x\kappa(x,y)(\nabla_x p_i+\nabla_y p_{i+1}) - div_y\kappa(x,y)(\nabla_x u_{i+1}+\nabla_y p_{i+2})\big](x,\frac{x}{\epsilon},t). \end{aligned} \tag{3.2}$$

In other words, we are looking for a solution of the equation $F^\epsilon := \sum_{i=-2}^{+\infty} a_i\epsilon^i = f(u)$, where coefficients $a_i (i=-2,-1,0,1,2,...)$ of the power series F^ϵ are defined from Equation (3.2). Clearly, $a_i = 0$ for $i \neq 0$ and $a_0 = f(u)$. We consider two-scale problem, so only first three terms are sufficient to use. We introduce corresponding three equations:

$$a_{-2} = \big[div_y\kappa(x,y)\nabla_y p_0\big](x,y,t) = 0, \tag{3.3}$$

$$a_{-1} = \big[div_y\kappa(x,y)(\nabla_x p_0+\nabla_y p_1) + div_x\kappa(x,y)\nabla_y p_0\big](x,y,t) = 0, \tag{3.4}$$

$$\begin{aligned} a_0 = \big[-div_x\kappa(x,y)(\nabla_x p_0+\nabla_y p_1) - div_y\kappa(x,y)(\nabla_x p_1+\nabla_y p_2)\big](x,y,t)+ \\ +\frac{dp_o}{dt}(x,y,t) = f(u). \end{aligned} \tag{3.5}$$

Since $p_0(x,y,t)$ is periodic in y, then Equation (3.3) implies $p_0(x,y,t)$ actually doesn't depend on y and we can rewrite $p_0(x,y,t) \equiv p_0(x,t)$.

Moreover, taking into account $\nabla_y p_0(x,t) = 0$, Equation (3.4) can be transformed as following:

$$\big[div_y\kappa(x,y)(\nabla_x p_0 + \nabla_y p_1) + div_x\kappa(x,y)\nabla_y p_0\big](x,y,t) =$$

$$= \big[div_y\kappa(x,y)(\nabla_x p_0 + \nabla_y p_1)\big](x,y,t) = 0$$

or

$$-div_y\kappa(x,y)\nabla_y p_1(x,y,t) = div_y\kappa(x,y)\nabla_x p_0(x,t).$$

Therefore, the last equation gives an expression for $p_1(x,y,t)$ in terms of $p_0(x,t)$. More precisely, $p_1(x,y,t) = \sum_{i=1}^{N} \frac{\partial p_0}{\partial x_i} N_i(x,y) = N(x,y)\nabla p_0$ up to some constant function $c(x)$, where $N(x,y)$ is a solution for a cell problem defined by:

$$\begin{aligned} -div_y\kappa(x,y)(e_i + \nabla_y N_i(x,y)) = 0 \quad in \quad Y, \\ N(x,y) \quad is \quad periodic \quad in \quad Y. \end{aligned} \tag{3.6}$$

As usual, $\{e_i\}_{i=1}^{N}$ denotes standard basis in R^N.

Equation (3.5) represents an equation for the unknown p_2 in a unit periodic cell $Y = [0,1] \times [0,1]$ and it can be rewritten as following:

$$-div_y\kappa(x,y)\nabla_y p_2(x,y,t) =$$

$$= -\frac{dp_o}{dt}(x,t) + \big[div_x\kappa(x,y)(\nabla_x p_0 + \nabla_y p_1) + div_y\kappa(x,y)\nabla_x p_1\big](x,y,t) + f(u). \tag{3.7}$$

Notice that $p_2(x,y)$ has periodic boundary condition, so

$$\int_Y div_y\kappa(x,y)\nabla_y p_2(x,y,t)dy = \int_{\partial Y} \big(\kappa(x,y)\nabla_y p_2(x,y,t)\big) \cdot n dS = 0.$$

Hence after integrating both sides of Equation (3.7) over Y, the following expression

takes place:

$$\int_Y \big[div_x\kappa(x,y)(\nabla_x p_0+\nabla_y p_1)+div_y\kappa(x,y)\nabla_x p_1\big](x,y,t)dy=\int_Y(\frac{dp_o}{dt}(x,t)-f(u))dy. \tag{3.8}$$

Notice that a term

$$\int_Y \big[div_y\kappa(x,y)\nabla_x p_1\big](x,y,t)dy=\int_Y \big[div_y\kappa(x,y)\nabla_x\big(N(x,y)\nabla p_0(x,t)\big)\big](x,y,t)dy=0$$

according to the periodicity of $N(x,y)$ and the following fact.

Remark. *We assume $F(x,y)$ is a periodic function with respect to $y\in Y=[0,1]$, then $\int_Y \frac{d}{dy}F(x,y)dy=F(x,1)-F(x,0)=0$.*

Equation (3.8) can be written as

$$\frac{dp_o}{dt}(x,t)-div_x\int_Y\big[\kappa(x,y)(\nabla_x p_0+\nabla_y p_1)\big](x,y,t)dy=f(u)$$

and the homogenized equation can be written as

$$\begin{aligned}\frac{dp_o}{dt}(x,t)-div_x\kappa^*\nabla_x p_0(x,t)=f(u)\ \ in\ \ D,\\ p_0=g_0\ \ on\ \ \partial D,\\ p_0(x,0)=0,\end{aligned} \tag{3.9}$$

where the homogenized coefficient is defined as $\kappa^*_{ij}=\int_Y\big[\kappa(x,y)(e_i+\nabla_y N_i)(e_j+\nabla_y N_j)\big](x,y)dy$.

We show below that the expansion can be written as $p^\epsilon(x,t)=p_0(x,t)+\epsilon p_1(x,\frac{x}{\epsilon},t)+\epsilon^2 p_2(x,\frac{x}{\epsilon},t)+\epsilon^3 p_3(x,\frac{x}{\epsilon},t)+...=p_0(x,t)+\epsilon N(x,y)\cdot\nabla p_0+\theta(x,y,t)$, where $||\theta||_{H^1(D)}=O(\epsilon^{1/2})$.

For simplifications, we assume that the right hand side $f(u)\equiv 0$. Note that θ

satisfies the following system:

$$\begin{aligned} \frac{d\theta}{dt} - div_x \kappa^* \nabla_x \theta = 0, \\ \theta = -\epsilon N(x, \frac{x}{\epsilon}). \end{aligned} \tag{3.10}$$

Since the minimum energy is achieved on solution, we have $\int_D |\nabla\theta| \leq \int_D |\nabla\theta_{arb}|$ for any arbitrary θ_{arb} with boundary conditions $\theta_{arb} = -\epsilon N(x, \frac{x}{\epsilon})$ on ∂D. We consider a function $\tau \in C^\infty(D)$ such that $\tau = 1$ on ∂D and $\tau = 0$ on D^0. For simplicity, we assume $D = [0,1] \times [0,1]$, $\tau = 1$ in a boundary strip with length δ and $\tau = 0$ otherwise. Then it can be easily shown $\nabla\tau \sim \frac{1}{\delta}$. The following computations take place:

$$\int_D |\nabla\theta_{arb}|^2 \leq c \int_D \epsilon^2 |\nabla N(x, \frac{x}{\epsilon})|^2 \tau^2 + c \int_D \epsilon^2 |N(x, \frac{x}{\epsilon})|^2 |\nabla\tau|^2 \leq C\delta H + c\delta H \epsilon^2 \frac{1}{\delta^2} =$$

$= CH(\delta + \frac{\epsilon^2}{\delta}) = CH\epsilon$, where $\int_Y |N(x, \frac{x}{\epsilon})|^2 dy = \delta H$.

Therefore, $||\theta||_{H^1(D)} = O(\epsilon^{1/2})$ and it serves to take into account boundary conditions. In order to keep simple explanation, we ignore θ. Hence the pressure is equal to

$$p^\epsilon(x,t) = p_0(x,t) + \epsilon p_1(x, \frac{x}{\epsilon}, t), \tag{3.11}$$

where $p_1(x,y,t)$ is a function periodic in y (see [3]). Therefore, our problem becomes

$$\frac{dp^\epsilon}{dt} = div(\kappa(x, \frac{x}{\epsilon}) \nabla p^\epsilon) + f(u). \tag{3.12}$$

We notice that the computational cost for finding such solution is cheap due to the following fact that the solution can be found for a single RVE and is independent of ϵ.

We assume $\zeta^\epsilon \in H_0^1(D) \times H_{per}^1(Y)$ is a test function, then the weak form of the

equation (3.12) can be written as:

$$\int_D (\frac{dp^\epsilon}{dt}, \zeta^\epsilon)dx = \int_D \kappa(x, \frac{x}{\epsilon})\nabla p^\epsilon \nabla \zeta^\epsilon dx + \int_D (f, \zeta^\epsilon)dx. \tag{3.13}$$

Note that $\frac{dp^\epsilon}{dt} = \frac{dp_0}{dt} + \epsilon \frac{dp_1(x, \frac{x}{\epsilon}, t), t}{dt} \overset{\epsilon \to 0}{\to} \frac{dp_0}{dt}$ and the two-scale convergence holds: $\nabla p^\epsilon \overset{\epsilon \to 0}{\to} \nabla_x p_0(x,t) + \nabla_y p_1(x,y,t)$.

Therefore, the limiting equation for $p(x)$ can be rewritten as following:

$$\int_D \frac{dp_0}{dt}\zeta_0 dx = \int_D \int_Y \kappa(x,y)(\nabla_x p_0 + \nabla_y p_1)(\nabla_x \zeta_0 + \nabla_y \zeta_1)dydx + \int_D f(u)\zeta_0 dx. \tag{3.14}$$

3.2 Full tensor product FEM

3.2.1 Reduced BT approach

Full tensor product FEM is considered in this section for solving the homogenized system. For simplicity, we omit dependence of t in the derivations below. Matrix formulation is described below. We denote basis functions in $H_0^1(D)$ as $\{\Phi_i(x)\}(i = 1, ..., N)$, basis functions in $L^2(D)$ as $\{\Psi_k(x)\}(k = 1, ..., M)$ and basis functions in $H^1_{per}(Y)$ as $\{\tilde{\Psi}_l(y)\}(l = 1, ..., \tilde{M})$.

We assume that $p_0(x) = \sum_{i=1}^N c_i \Phi_i(x)$, $p_1(x,y) = \sum_{k=1}^M \sum_{l=1}^{\tilde{M}} d_{kl}\Psi_k(x)\tilde{\Psi}_l(y)$.

Remark. *Notice that the relationship between coefficients d_{kl} and e_{kl} can be found from the following equation:*

$$p_1(x,y) = \sum_{k=1}^M \sum_{l=1}^{\tilde{M}} d_{kl}\Psi_k(x)\tilde{\Psi}_l(y) = \sum_{i=1}^N \sum_{k=1}^M \sum_{l=1}^{\tilde{M}} e_{kl}\Psi_k(x)\tilde{\Psi}_l(y)c_i\Phi_i(x). \tag{3.15}$$

We assume $\zeta_0 = \Phi_j(x)$ for some $j = 1, ..., N$ and $\zeta_1 = \Psi_r(x)\tilde{\Psi}_s(y)$ for some $r = 1, ..., M, s = 1, ..., \tilde{M}$.

Therefore, Equation (3.14) can be rewritten as following:

$$\int_D \sum_{i=1}^{N} \frac{dc_i(t)}{dt}\Phi_i(x)\Phi_j(x)dx = \int_D \int_Y \kappa(x,y)(\sum_{i=1}^{N} c_i \nabla_x \Phi_i(x))\nabla_x \Phi_j(x)dydx+$$

$$+\int_D \int_Y \kappa(x,y)(\sum_{i=1}^{N} c_i \nabla_x \Phi_i(x))\Psi_r(x)\nabla_y \tilde{\Psi}_s(y)dydx+ \tag{3.16}$$

$$+\int_D \int_Y \kappa(x,y)(\sum_{k=1}^{M}\sum_{l=1}^{\tilde{M}} d_{kl}\Psi_k(x)\nabla_y \tilde{\Psi}_l(y))\nabla_x \Phi_j(x)dydx+$$

$$+\int_D \int_Y \kappa(x,y)(\sum_{k=1}^{M}\sum_{l=1}^{\tilde{M}} d_{kl}\Psi_k(x)\nabla_y \tilde{\Psi}_l(y))\Psi_r(x)\nabla_y \tilde{\Psi}_s(y)dydx+$$

$$+\int_D f(u)\Phi_j(x)dx.$$

We reorganize terms of Equation (3.16):

$$\sum_{i=1}^{N}(\int_D \Phi_i(x)\Phi_j(x)dx)\frac{dc_i(t)}{dt} = \sum_{i=1}^{N}(\int_D \int_Y \kappa(x,y)\nabla_x \Phi_i(x)\nabla_x \Phi_j(x)dydx)c_i+$$

$$+\sum_{i=1}^{N}(\int_D \int_Y \kappa(x,y)\nabla_x \Phi_i(x)\Psi_r(x)\nabla_y \tilde{\Psi}_s(y)dydx)c_i+$$

$$+\sum_{k=1}^{M}\sum_{l=1}^{\tilde{M}}(\int_D \int_Y \kappa(x,y)\Psi_k(x)\nabla_y \tilde{\Psi}_l(y)\nabla_x \Phi_j(x)dydx)d_{kl}+$$

$$+\sum_{k=1}^{M}\sum_{l=1}^{\tilde{M}}(\int_D \int_Y \kappa(x,y)\Psi_k(x)\nabla_y \tilde{\Psi}_l(y)\Psi_r(x)\nabla_y \tilde{\Psi}_s(y)dydx)d_{kl}+$$

$$+\int_D f(u)\Phi_j(x)dx.$$

Using matrix notation, the equation can be written in the following form:

$$\sum_{i=1}^{N} Q_{ij}\frac{dc_i(t)}{dt} = \sum_{i=1}^{N} A_{ji}c_i + \sum_{i=1}^{N}\sum_{r=1}^{M}(B_r^*)_{ji}c_i + \sum_{k=1}^{M}\sum_{l=1}^{\tilde{M}}(B_k)_{jl}d_{kl} + \sum_{k,r=1}^{M}\sum_{l=1}^{\tilde{M}}(S_{rk})_{jl}d_{kl} + F_j, \tag{3.17}$$

where

$A_{ij} = \int_D \int_Y \kappa(x,y)\nabla_x\Phi_i(x)\nabla_x\Phi_j(x)dydx$ for $i = 1, ..., N, j = 1, ..., N$;

$(B_l)_{ij} = \int_D \int_Y \kappa(x,y)\Psi_l(x)\nabla_y\tilde{\Psi}_j(y)\nabla_x\Phi_i(x)dydx$ for $l = 1, .., M, i = 1, ..., N, j = 1, ..., \tilde{M}$;

$F_j = \int_D f(u)\Phi_j(x)dx$ for $j = 1, ..., N$;

$(S_{rj})_{li} = \int_D \int_Y \kappa(x,y)\Psi_r(x)\nabla_y\tilde{\Psi}_j(y)\Psi_i(x)\nabla_y\tilde{\Psi}_l(y)dydx$ for $r = 1, .., M, j = 1, ..., \tilde{M}, l = 1, ..., \tilde{M}, i = 1, ..., M$;

$Q_{ij} = \int_D \Phi_i(x)\Phi_j(x)dx$ for $i = 1, ..., N, j = 1, ..., N$ (assume $Q_{ij} = 0$ for all $i = N+1, ..., N+M\tilde{M}, j = N+1, ..., N+M\tilde{M}$).

Therefore, the matrix form of Equation (3.17):

$$Q\begin{bmatrix} \frac{dc_1}{dt} \\ ... \\ \frac{dc_N}{dt} \\ \frac{d(d_{11})}{dt} \\ ... \\ \frac{d(d_{1\tilde{M}})}{dt} \\ ... \\ \frac{d(d_{M\tilde{M}})}{dt} \end{bmatrix} = \begin{bmatrix} (A)_{N\times N}(B_1)_{N\times\tilde{M}}...(B_M)_{N\times\tilde{M}} \\ (B_1^*)_{\tilde{M}\times N}(S_{11})_{M\times\tilde{M}}...(S_{1M})_{M\times\tilde{M}} \\ ..... \\ (B_M^*)_{\tilde{M}\times N}(S_{\tilde{M}1})_{M\times\tilde{M}}...(S_{\tilde{M}M})_{M\times\tilde{M}} \end{bmatrix} \begin{bmatrix} c_1 \\ ... \\ c_N \\ d_{11} \\ ... \\ d_{1\tilde{M}} \\ ... \\ d_{M\tilde{M}} \end{bmatrix} + \begin{bmatrix} F_1 \\ ... \\ F_N \\ 0 \\ ... \\ 0 \\ ... \\ 0 \end{bmatrix} =$$

$$= \begin{bmatrix} (A)_{N\times N} \\ (B_1^*)_{\tilde{M}\times N} \\ ..... \\ (B_M^*)_{\tilde{M}\times N} \end{bmatrix} \begin{bmatrix} c_1 \\ ... \\ c_N \end{bmatrix} + \begin{bmatrix} (B_1)_{N\times\tilde{M}}...(B_M)_{N\times\tilde{M}} \\ (S_{11})_{M\times\tilde{M}}...(S_{1M})_{M\times\tilde{M}} \\ ..... \\ (S_{\tilde{M}1})_{M\times\tilde{M}}...(S_{\tilde{M}M})_{M\times\tilde{M}} \end{bmatrix} \begin{bmatrix} d_{11} \\ ... \\ d_{1\tilde{M}} \\ ... \\ d_{M\tilde{M}} \end{bmatrix} + \begin{bmatrix} F_1 \\ ... \\ F_N \\ 0 \\ ... \\ 0 \\ ... \\ 0 \end{bmatrix} =$$

$$= \begin{bmatrix} (A)_{N\times N} \\ (B^*)_{M\tilde{M}\times N} \end{bmatrix} \begin{bmatrix} (c)_{1\times N} \end{bmatrix} + \begin{bmatrix} (B)_{N\times M\tilde{M}} \\ (S)_{M\tilde{M}\times M\tilde{M}} \end{bmatrix} \begin{bmatrix} (d)_{1\times M\tilde{M}} \end{bmatrix} + \begin{bmatrix} (F)_{1\times N} \\ (0)_{1\times M\tilde{M}} \end{bmatrix}, \qquad (3.18)$$

where $B = [B_1, ...B_M]$, $c = [c_1, ..., c_N]'$, $d = [d_{11}, ..., d_{1\tilde{M}}, ..., d_{M\tilde{M}}]'$,
$F = [F_1, ..., F_N]'$ and $S = \begin{bmatrix} (S_{11})_{M\times\tilde{M}}...(S_{1M})_{M\times\tilde{M}} \\ ..... \\ (S_{\tilde{M}1})_{M\times\tilde{M}}...(S_{\tilde{M}M})_{M\times\tilde{M}} \end{bmatrix}$.
From the last Equation (3.18)

$$B^*c + Sd = 0,$$

and the expression of d in terms of c can be found as $d = -S^{(-1)}B^*c$.

The size of matrix S is comparable to the size of the whole system. The sparse tensor product finite element method can be used in order to reduce the dimension of S and be able to compute $S^{(-1)}$ as well as a hierarchical approach.

We denote the 'non-zero' part of the matrix Q as $\tilde{Q}$, i.e., $\tilde{Q}$ is $N \times N$ matrix, where coefficients $\tilde{Q}_{ij} = \int_D \Phi_i(x)\Phi_j(x)dx$ for $i = 1, ..., N, j = 1, ..., N$. Then we can rewrite the first equation of the system (3.18) as

$$\tilde{Q}\frac{dc}{dt} = Ac + Bd + F = Ac - BS^{(-1)}B^*c + F = (A - BS^{(-1)}B^*)c + F.$$

Finally, we get the system:

$$(\tilde{Q})_{N\times N}\frac{dc}{dt} = (A - BS^{(-1)}B^*)c + F, \tag{3.19}$$

$$q = Cc.$$

For simplicity, we assume that $F = Ku$. The BT Approach will be applied in approximating the input-output behavior of linear systems and a-priori error bounds can be easily computed. In order to apply the BT approach, we need to take Laplace transform of dynamic system (3.19). We recall that the Laplace transform for some function $g(t)$ is defined as

$$L[g](s) = \int_0^{+\infty} e^{-st}g(t)dt, s \in R. \tag{3.20}$$

Moreover,

$$L[\dot{c}](s) = \int_0^{+\infty} e^{-st}\dot{c}(t)dt = -\int_0^{+\infty}(-s)e^{-st}c(t)dt + (e^{-st}c(t))\Big|_{s=0}^{s=+\infty} = sL[c](s) - c_0.$$

Therefore, the Laplace transform of dynamic system (3.19) yields

$$s\tilde{Q}L[c](s) - c(0) = (A - BS^{(-1)}B^*)L[c](s) + KL[u](s),$$

which gives

$$L[c](s) = (s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}x(0) + (s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}KL[u](s)$$

and the output

$$L[q](s) = C(s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}x(0) + C(s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}KL[u](s).$$

We assume $c(0) = 0$, then

$$L[q](s) = C(s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}KL[u](s) = G(s)L[u](s)$$

for transfer matrix

$$G(s) = C(s\tilde{Q} - A + BS^{(-1)}B^*)^{(-1)}K.$$

The balanced realization can be written as

$$G(s) = \begin{bmatrix} \tilde{Q}^{(-1)}(A - BS^{(-1)}B^*) & \tilde{Q}^{(-1)}K \\ C & 0 \end{bmatrix}. \tag{3.21}$$

By BT approach, there exists matrices A_{11}, B_1, C_1 such that

$$G(s) = \begin{bmatrix} A_{11} & B_1 \\ C_1 & 0 \end{bmatrix} \tag{3.22}$$

is also a realization of G(s).

We note that the computational cost for S^{-1} can be substantially high, there-

fore, it is useful to apply several sparse techniques in advance. Such techniques are described in the next section.

3.2.2 Lowest order basis

We assume that $\{\Phi_i(x)\}(i = 1, ..., N), \{\tilde{\Psi}_l(y)\}(l = 1, ..., \tilde{M})$ are piecewise-linear functions and $\{\Psi_k(x)\}(k = 1, ..., M)$ are piecewise constant functions. We divide D into intervals $I_1, I_2, ...I_S$ and denote $\langle\kappa(x, y)\rangle_{x,I_r} = \int_D \kappa(x, y)\Psi_r(x)\Psi_i(x)dx$. Also, for simplicity we assume

$$D = [0, 1] = \bigcup_{i=1}^{N-1} [x_i, x_{i+1}], Y = [0, 2\pi] = \bigcup_{j=1}^{\tilde{M}-1} [y_j, y_{j+1}]. \tag{3.23}$$

Then the matrices of Equation (3.17) can be written as following:

$$(S_{rj})_{li} = \int_D \int_Y \kappa(x, y)\Psi_r(x)\nabla_y\tilde{\Psi}_j(y)\Psi_i(x)\nabla_y\tilde{\Psi}_l(y)dydx =$$

$$= \int_Y \langle\kappa(x, y)\rangle_{x,I_r}\nabla_y\tilde{\Psi}_j(y)\nabla_y\tilde{\Psi}_l(y)\delta_{ri}dy = \begin{cases} \int_{y_{l-1}}^{y_{l+1}} \langle\kappa(x, y)\rangle_{x,I_r}\delta_{ri}\frac{2}{H_y^2}dy, & j = l, \\ -\int_{y_l}^{y_{l+1}} \langle\kappa(x, y)\rangle_{x,I_r}\delta_{ri}\frac{1}{H_y^2}dy, & j = l + 1, \\ -\int_{y_{l-1}}^{y_l} \langle\kappa(x, y)\rangle_{x,I_r}\delta_{ri}\frac{1}{H_y^2}dy, & j = l - 1, \end{cases} \tag{3.24}$$

for

$$r = 1, .., M, j = 1, ..., \tilde{M}, l = 1, ..., \tilde{M}, i = 1, ..., M, H_y = \frac{2\pi}{\tilde{M}}.$$

$$(B_r)_{ij} = \int_D \int_Y \kappa(x, y)\Psi_r(x)\nabla_y\tilde{\Psi}_j(y)\nabla_x\Phi_i(x)dydx =$$

$$= \int_Y \langle\kappa(x, y)\rangle_{x,I_r}\nabla_y\tilde{\Psi}_j(y)\nabla_x\Phi_i(x)dy =$$

$$= \begin{cases} \int_{y_{i-1}}^{y_{i+1}} \langle \kappa(x,y) \rangle_{x,I_r} \delta_{ri} \frac{1}{H_x} \frac{1}{H_y} dy & \text{for } j, i-odd; j, i-even, \\ -\int_{y_{i-1}}^{y_i} \langle \kappa(x,y) \rangle_{x,I_r} \delta_{ri} \frac{1}{H_x} \frac{1}{H_y} dy & \text{for } j-odd, i-even, \end{cases}$$

for

$$r = 1, .., M, i = 1, ..., N, j = 1, ..., \tilde{M}.$$

Now we will show that a cell problem (3.6) can be derived from the equation Equation (3.17).

We rewrite a cell problem (3.6) as following

$$\int_Y \nabla_y N_j(x,y) \kappa(x,y) \nabla_y \tilde{\Psi}_j(y) \nabla_y \tilde{\Psi}_l(y) dy = -\int_Y \kappa(x,y) \nabla_y \tilde{\Psi}_l(y) dy.$$

After averaging on the interval I_r we get the following equation:

$$\int_Y \nabla_y N_j(x,y) \langle \kappa(x,y) \rangle_{x,I_r} \nabla_y \tilde{\Psi}_j(y) \nabla_y \tilde{\Psi}_l(y) dy = -\int_Y \langle \kappa(x,y) \rangle_{x,I_r} \nabla_y \tilde{\Psi}_l(y) dy.$$

Therefore, the equation

$$B^* c + Sd = 0$$

can be written as

$$-(\int_Y \langle \kappa(x,y) \rangle_{x,I_r} \nabla_y \tilde{\Psi}_l(y) dy) \Psi_l(x) c_l - \int_Y \langle \kappa(x,y) \rangle_{x,I_r} \nabla_y \tilde{\Psi}_l(y) dy = 0.$$

3.2.3 Discussion on computational savings

As we saw in a previous section full tensor product FEM involves calculations solutions of cell problems for each macroscopic point $x \in D$. Depending on a structure of the domain D such calculations can be very expensive or even impossible to

make. Also note that the computational complexity of BT approach is $O(N^3)$ for N being a size of a problem. Therefore, we have to carefully chose the size of an input-output system (3.19).

In order to reduce computational cost for BT method, two techniques are suggested: a hierarchical approach and a sparse tensor product approach. The first method avoids enormously massive computations by using a grid hierarchy and a precomputed information for nearby cell equations. The second method uses a special form of sparse tensor product FEM and it achieves convergence rates comparable to the full FEM.

3.3 Hierarchical approach

Below we offer a basic computational algorithm that is followed by method justification. We briefly discuss the main concept of a hierarchical approach. It uses a sequence of finite element spaces where we compute cell equations with different order of accuracy. Firstly, for chosen macro-grids, we calculate solutions for cell problems with high accuracy. Secondly, for a finer grid we calculate respective solutions with less accuracy and correct them using high accurate nearby solutions.

3.3.1 Algorithm

We assume $D = \bigcup_{\vec{k}} \Xi_{\vec{k}}$, where $\Xi_{\vec{k}} = \Xi + \vec{k}$ for a center of grid-block $\vec{k}$ and a Ξ - rescaled unit periodic cell. We can associate $\vec{k}$ with labeling denoted by k. Henceforward, by $N(x, y)$ we mean the analytic solution for (3.6). We fix $k \in \{1, ..., G\}, G \in Z_+$ and the corresponding macro-grid Ξ_k. By $N^k(x, y)$ we denote the solution for the following problem

$$-div_y((\int_{\Xi_k} \kappa(x, y)dx)(I+\nabla N^k(x, y))) = 0, \int_Y N^k(x, y)dy = 0, N^k(x, y)\, is\, y-periodic. \tag{3.25}$$

Now we define variational forms as following:

$$A_y(\Xi_k)(N^k, \Phi) = \int_{\Xi_k} \int_Y \kappa(x,y) \nabla_y N(x,y) \nabla_y \phi(y) dy dx, \tag{3.26}$$

$$(f^k(y), \Phi)_{L_2} = -\int_{\Xi_k} \int_Y \kappa(x,y) \nabla_y \phi(y) dy dx. \tag{3.27}$$

Therefore, the weak variation form of Equation (3.25) can be written as

$$A_y(\Xi_k)(N^k, \Phi) = (f^k(y), \Phi)_{L_2}. \tag{3.28}$$

The problem (3.28) can be solved by standard Galerkin FEM.

We suppose that the basis functions $\{\Psi_k(x)\}_{k=1}^G$ correspond to the grid-blocks $\{\Xi_k\}_{k=1}^G$, so we may write

$$N^k(x,y) \approx \sum_{l=1}^{\tilde{M}} e_{kl} \Psi_k(x) \tilde{\Psi}_l(y). \tag{3.29}$$

The above approximation is exact in the numerical or discrete sense if the basis $\{\Psi_k(x)\}$ are piecewise constants. We assume $\phi(x,y) = \tilde{\Psi}_s(y)$ and after plugging the above expression into (3.25), we obtain

$$\int_{\Xi_k} \int_Y \kappa(x,y) \left(I + \sum_{l=1}^{\tilde{M}} e_{kl} \Psi_k(x) \nabla_y \tilde{\Psi}_l(y) \right) \nabla_y \tilde{\Psi}_s(y) dy dx = 0. \tag{3.30}$$

Solving the above system for each Ξ_k, we may construct

$$N(x,y) \approx \sum_{k=1}^{M} \sum_{l=1}^{\tilde{M}} e_{kl} \Psi_k(x) \tilde{\Psi}_l(y).$$

We denote such finite element approximation of solution by

$$\bar{N}(x,y) = \sum_{k=1}^{M} \sum_{l=1}^{\tilde{M}} e_{kl} \Psi_k(x) \tilde{\Psi}_l(y). \tag{3.31}$$

Below we suggested a hierarchical approach for solving problem (3.28) and we denote the corresponding solution by $\hat{N}(x,y)$.

For clearness, we describe a hierarchical algorithm for 2-level case. For multilevel case it can be easily done by induction.

Step 1 : Build FEM Spaces. A fixed macro-grid Ξ_k is considered. We wish to find an approximation $\bar{N}(\Xi_k, \cdot)$ satisfying (3.28) by using Galerkin FEM. We denote a local coarse space by V^c and local fine space by V^f, $V^c \subset V^f$. We denote the error in V^f by h and the coarsening FE factor by r.

Step 2 : Build Macro-Grid. First, we build a nested macro-grid for D denoted by

$$T_0 \subset T_1 \subset D.$$

Suppose we have an initial grid T_0 with distance between neighboring nodes at most H and T_1 can be obtained by refinement with grid spacing Hr^{-1}.

We then define the hierarchy of macro-grids $\{S_0, S_1\}$ as $S_0 = T_0$, $S_1 = T_1 \backslash S_0$. We refer to the coarsest grid S_0 as the anchor points. We require that the hierarchy of macro-grids be dense. That is, we require that for macro-block $x \in \Xi_k$, there exists a nearby grid-block $\Xi_{k'}$ such that

$$\text{dist}\,(k,k') < O(H\kappa^{-1}). \tag{3.32}$$

Step 3 : Calculating the Correction Term.

We will rewrite the equation (3.28) in a matrix form. In order to do that, for

fixed grid block Ξ_k we denote

$$(A_k)_{sl} = \int_{\Xi_k} \int_Y \kappa(x,y)\Psi_k(x)\nabla_y \tilde{\Psi}_l(y)\nabla_y \tilde{\Psi}_s(y)dydx, \tag{3.33}$$

and

$$(b_k)_s = -\int_{\Xi_k} \int_Y \kappa(x,y)\nabla_y \tilde{\Psi}_s(y)dydx \text{ and } e_{kl} = (e_k)_l. \tag{3.34}$$

Therefore, (3.28) in a matrix form can be written as

$$\sum_{l=1}^{\tilde{M}} (A_k)_{sl} (e_k)_l = (b_k)_s \text{ or } A_k e_k = b_k. \tag{3.35}$$

We outline an idea from multilevel algorithm. Suppose we have solved e_k in grid-block Ξ_k and suppose we wish to solve the equation in a nearby grid-block $\Xi_{k'}$. Then we write the correction term as $w_k = e_{k'} - e_k$, and so the correction term satisfies

$$A_{k'} w_{k'} = b_{k'} - A_{k'} e_k. \tag{3.36}$$

The idea is to eventually solve $w_{k'}$ in a coarser space than we used to solve e_k. Then, we let the approximation

$$e_{k'} = w_{k'} + e_k. \tag{3.37}$$

3.3.2 Assumptions

In order to guarantee the existence and uniqueness for the solution, we make a few assumptions on $\kappa(x,y)$ and $N(x,y)$.

Assumption 3.3.1 *$\kappa(x,y)$ is*

a) a smooth function of x and y, i.e. $\kappa(.,y) \in C^l(D)$ and $\kappa(x,.) \in C^k(Y)$ for some positive integers k and l,

b) Lipschitz in x, i.e. there exists a positive constant C so that $|\kappa(x_1,y) - \kappa(x_2,y)| \leq C \cdot dist(x_1,x_2)$,

c) bounded above and below, i.e. there are positive constants α and β so that for all $x \in D, y \in Y$ the following inequality $\alpha \cdot ||x|| \cdot ||y|| \leq \kappa(x,y) \leq \beta \cdot ||x|| \cdot ||y||$ takes place.

Assumption 3.3.2 *$N(x,y)$ is*

a) a smooth function of x, i.e. $N(.,y) \in C^{l_1}(D)$ for some $l_1 \in R$.

b) Lipschitz in x, i.e. there exists a positive constant C_1 so that $|N(x_1,y) - N(x_2,y)| \leq C_1 \cdot dist(x_1,x_2)$.

Assumption 3.3.3 *The solution $N(x,y)$ of the problem is in $C^1(\Xi_k \times Y)$. For the finite element approximation $\bar{N}(x,y) = \sum_{k=1}^{M} \sum_{l=1}^{\tilde{M}} e_{kl} \Psi_k(x) \tilde{\Psi}_l(y)$ there exists a positive constant C such that*

$$||N(x,y)) - \bar{N}(x,y)||_{H^1_{(\Xi_k)}} \leq Ch||N||_{H^2_{(\Xi_k)}}. \tag{3.38}$$

Notice that from Assumption 3.3.1 we can easily derive the Lipschitz condition on $A_y(\Xi_k)(N^k, \Phi)$:

Proposition 3.3.1 *There exists a constant γ so that*

$$|(A_y(\Xi_k)(N^k,\Phi) - A_y(\Xi_{k'})(N^k,\Phi))\,(v,w)\,| \leq \gamma |k - k'|_{l_2} \|v\|_{\mathcal{V}} \|w\|_{\mathcal{W}} \tag{3.39}$$

for all k, k' satisfying (3.32).

Remark. *A hierarchical solution may not be periodic on cells, but it can be fixed by applying oversampling techniques [27].*

3.3.3 Main theorem

We apply the same rectangular hierarchy of macro-grids as in [13]. Our goal is to prove that suggested hierarchical approach gives the same order of accuracy as the full solve.

We use the FE solution and trial spaces $(V_i, W_i)_{i=1}^{L}$ described in step 1 of algorithm. We assume that $V_0 = W_0 = \emptyset$ and we denote macro-grids by $\{S_0, S_1 ..., S_L\}$. A macrogrid $\Xi_{\vec{k}}$ for some fixed k is considered. We apply Galerkin method to find a solution $N_{\vec{k}} \in V_i$ with high accuracy for the following problem:

$$A_y(\Xi_{\vec{k}})(N_{\vec{k}}, \Phi) = (f_{\vec{k}}, \Phi) \qquad \forall \Phi \in W_i. \tag{3.40}$$

Assume we found a solution $N_{\vec{k}}$ and we consider a macrogrid $\Xi_{\vec{k'}}$. Then by finding the correction term from $A_y(\Xi_{\vec{k'}})(\omega^c_{\vec{k'}}, \Phi^c) = R((f^f_{\vec{k'}}, \Phi^f) - A_y(\Xi_{\vec{k'}})(N^f_{\vec{k}}, \Phi^f))$, we can find an approximation of N as $N^f_{\vec{k'}} = R'\omega^c_{\vec{k'}} + N^f_{\vec{k}}$, where R - the fine-to-coarse operator.

Our goal is to prove the following estimates.

Theorem 1. *There is a positive constant C independent of an error in the finest FE space h such that*

$$||N - \hat{N}_{\vec{k}}||_{H^1(Y)} \leq C \cdot h \cdot log(h^{-1}), \tag{3.41}$$

where $N(x, y)$ is an analytical solution of the cell Equation (3.6) *and $\hat{N}_{\vec{k}} = \bar{N} + \bar{\omega^c}$ is an approximated solution obtained by hierarchical algorithm Step1-Step3.*

Theorem 2. *There is a positive constant $\tilde{C}$ independent of an error in the finest*

FE space h such that

$$||p - \hat{p}||_{H^1} \leq \tilde{C} \cdot h \cdot log(h^{-1}), \tag{3.42}$$

where p is an analytical pressure solution of Equation (2.8) *and $\hat{p}$ is an approximated pressure solution obtained by hierarchical approximation of $\hat{N}$.*

Theorem 3. *There is a positive constant $\overline{C}$ independent of an error in the finest FE space h such that*

$$||q - \hat{q}||_{H^1} \leq \overline{C} \cdot h \cdot log(h^{-1}), \tag{3.43}$$

where q is an analytical output and $\hat{q}$ is an approximated output obtained by hierarchical approximation of $\hat{N}$.

First, let's prove the following regularity relation. We remind the definition of $\vec{k}$. We assume that $D = \bigcup_{\vec{k}} \Xi_{\vec{k}}$, where $\Xi_{\vec{k}} = \Xi + \vec{k}$ for a center of grid-block $\vec{k}$ and Ξ - a rescaled unit periodic cell. We can associate $\vec{k}$ with labeling denoted by k.

Lemma 1. For centers of grid-blocks $\vec{k}, \vec{k}'$ the following error estimate holds

$$||N_{\vec{k}} - N_{\vec{k}'}||_{H^1(Y)} \leq C|\vec{k} - \vec{k}'|_{l_2} \tag{3.44}$$

for some constant C.

Proof of Lemma 1. We fix centers of grid-blocks $\vec{k}, \vec{k}'$. Therefore, equations (3.25) for $N_{\vec{k}}$, $N_{\vec{k}'}$ can be written as following:

$$\int_Y \left(\int_{\Xi_{\vec{k}}} \kappa(x,y)dx \right) \nabla_y N_{\vec{k}} \nabla_y \phi(y) dy = -\int_Y \left(\int_{\Xi_{\vec{k}}} \kappa(x,y)dx \right) \nabla_y \phi(y) dy, \tag{3.45}$$

$$\int_Y \left(\int_{\Xi_{\vec{k}'}} \kappa(x,y)dx \right) \nabla_y N_{\vec{k}'} \nabla_y \phi(y) dy = - \int_Y \left(\int_{\Xi_{\vec{k}'}} \kappa(x,y)dx \right) \nabla_y \phi(y) dy. \quad (3.46)$$

Since we assume that the domain D is periodic, then each macro-grid differ by translation, i.e. we can write $\Xi_{\vec{k}} = \Xi + \vec{k}$ and $\Xi_{\vec{k}'} = \Xi + \vec{k}'$ for unit cell Ξ. We assume $x_{\vec{k}} = x_0 + \vec{k}$ and $x_{\vec{k}'} = x_0 + \vec{k}'$.

We make change of variables in both equations (3.45), (3.46) and subtract one from another:

$$\begin{aligned} &\int_Y \left(\int_\Xi (\kappa(x+\vec{k},y)dx \right) \nabla_y N^{\vec{k}} \nabla_y \phi(y) dy - \int_Y \left(\int_\Xi (\kappa(x+\vec{k}',y)dx \right) \nabla_y N^{\vec{k}'} \nabla_y \phi(y) dy \\ &= - \int_Y \left(\int_\Xi (\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y))dx \right) \nabla_y \phi(y) dy. \end{aligned} \quad (3.47)$$

We denote the right hand side $- \int_Y \left(\int_\Xi (\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y))dx \right) \nabla_y \phi(y) dy = I_1$. The equation (3.47) can be rewritten as

$$\begin{aligned} &\int_Y \left(\int_\Xi (\kappa(x+\vec{k},y)dx \right) \left(\nabla_y N^{\vec{k}} - \nabla_y N^{\vec{k}'} \right) \nabla_y \phi(y) dy = \\ &I_1 - \int_Y \left(\int_\Xi (\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y))dx \right) \nabla_y N^{\vec{k}'} \nabla_y \phi(y) dy. \end{aligned} \quad (3.48)$$

We denote $I_2 = - \int_Y \left(\int_\Xi (\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y))dx \right) \nabla_y N^{\vec{k}'} \nabla_y \phi(y) dy$. Then I_1 and I_2 can be easily estimated through Lipschitz inequality.

First, from Assumption 1b we have

$$\begin{aligned} &|I_1| \leq \int_Y \int_\Xi |\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y)| \cdot ||\nabla_y \phi(y)||_{L_2(\Xi \times Y)} dxdy \leq \\ &\int_\Xi \int_Y C|\vec{k} - \vec{k}'| \cdot ||\nabla_y \phi(y)||_{L_2(\Xi \times Y)} dydx. \end{aligned} \quad (3.49)$$

Secondly, we have:

$$|I_2| \leq \int_{\Xi}\int_{Y} |\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y)| \cdot ||\nabla_y N^{\vec{k}}||_{L^2(\Xi\times Y)} \cdot ||\nabla_y \phi(y)||_{L^2(\Xi\times Y)} dydx \leq$$
$$\int_{\Xi}\int_{Y} C \cdot |\vec{k}-\vec{k}'| \cdot ||\nabla_y N^{\vec{k}}||_{L^2(\Xi\times Y)} \cdot ||\nabla_y \phi(y)||_{L^2(\Xi\times Y)} dydx. \tag{3.50}$$

Since the gradient of Lipschitz function is bounded by Lipschitz constant, we have

$$\int_{\Xi}\int_{Y} C \cdot |\vec{k}-\vec{k}'| \cdot ||\nabla_y N^{\vec{k}}||_{L^2(\Xi\times Y)} \cdot ||\nabla_y \phi(y)||_{L^2(\Xi\times Y)} dydx \leq \tag{3.51}$$

$$\int_{\Xi}\int_{Y} C \cdot |\vec{k}-\vec{k}'| \cdot C1 \cdot ||\nabla_y \phi(y)||_{L^2(\Xi\times Y)} dydx.$$

We take $\phi = N^{\vec{k}} - N^{\vec{k}'}$, then the left hand side of the equation (3.48) can be written as

$$|\int_{\Xi}\int_{Y} (\kappa(x+\vec{k},y)\nabla_y N^{\vec{k}} - \kappa(x+\vec{k}',y)\nabla_y N^{\vec{k}})\nabla_y (N^{\vec{k}} - N^{\vec{k}'}) dydx| \leq$$

$$\int_{\Xi}\int_{Y} |\kappa(x+\vec{k},y) - \kappa(x+\vec{k}',y)| \cdot ||\nabla_y (N^{\vec{k}} - N^{\vec{k}'})||^2 dydx.$$

The second term:

$$|\int_{\Xi}\int_{Y} (\kappa(x+\vec{k}',y)\nabla_y N^{\vec{k}} - \kappa(x+\vec{k}',y)\nabla_y N^{\vec{k}'})\nabla_y \phi(y) dydx| \leq$$

$$\int_{\Xi}\int_{Y} |(\kappa(x+\vec{k}',y)\nabla_y N^{\vec{k}} - \kappa(x+\vec{k}',y)\nabla_y N^{\vec{k}'})\nabla_y \phi(y)| dydx \leq$$

$$\int_{\Xi}\int_{Y} |(\kappa(x+\vec{k}',y)| \cdot |\nabla_y N^{\vec{k}} - \nabla_y N^{\vec{k}'}| \cdot |\nabla_y \phi(y)| dydx.$$

Since the gradient of Lipschitz function on bounded interval is bounded, we have

$$\int_{\Xi}\int_{Y} |(\kappa(x+\vec{k}',y)| \cdot |\nabla_y N^{\vec{k}} - \nabla_y N^{\vec{k}'}| \cdot |\nabla_y \phi(y)| dydx \leq$$

$$\int_{\Xi}\int_{Y} \tilde{C} \cdot |\nabla_y N^{\vec{k}} - \nabla_y N^{\vec{k}'}| \cdot C^* dydx.$$

Therefore, $||N_{\vec{k}} - N_{\vec{k}'}||_{H^1(Y)} \leq C|\vec{k} - \vec{k}'|_{l_2}$. □

Proof of theorem 1.

Assume that $\bar{N}_k^f \in V^f$. We choose the nearby grid-block Ξ_k by assumption (3.32). We will define the hierarchical solution by $\hat{N}_{k'}^f = \bar{\omega}^{corr} + \bar{N}_k^f$ for smooth correction term $\bar{\omega}^{corr} = N_{k'} - N_k$.

From the variational formulation we get

$$A_y(\Xi_k)(\omega,\Phi) = A_y(\Xi_k)(N_k,\Phi) - A_y(\Xi_k)(N_{k'},\Phi) = (f_k,\Phi) - A_y(\Xi_k)(N_{k'},\Phi). \quad (3.52)$$

Since from (3.13) we have $A_y(\Xi_{k'})(N_{k'},\Phi) = (f_{k'},\Phi)$, then the equation (3.52) can be rewritten as following:

$$(f_k - f_{k'},\Phi) - (A_y(\Xi_k) - A_y(\Xi_{k'}))(N_k',\Phi) = A_y(\Xi_k)(\omega,\Phi) = A_y(\Xi_k)(\bar{\omega}^{corr},\Phi). \quad (3.53)$$

From Cea's lemma, Lipschitz inequality the following error estimate holds for fixed micro-grid Ξ_k:

$$||\omega - \bar{\omega}^{corr}|| \leq Crh \cdot r^{-1} = Ch. \quad (3.54)$$

Therefore, from the definition of correction term and the equation $A_y(\Xi_{k'})(\bar{\omega}^c,\Phi) = (f_{k'},\Phi)$,

$$A_y(\Xi_k)(\bar{\omega}^c, \Phi) = (f_k - f_{k'}, \Phi) - (A_y(\Xi_k) - A_y(\Xi_{k'}))(N'_k, \Phi). \tag{3.55}$$

From Assumptions, Equation (3.53), Equation (3.55) and following the proof of Theorem 3.1 at [13]:

$$\|\bar{\bar{\omega}}_{corr}(x,\cdot) - \bar{\omega}_{corr}(x,\cdot)\|_{\mathcal{V}} \leq$$

$$\leq 1/\beta||((f_k - f_{k'})(\omega - \bar{\omega}_{corr})|| + 1/\beta||A_y(\Xi_k) - A_y(\Xi_{k'}))(N'_k, (\omega - \bar{\omega}_{corr}))|| \leq log(\frac{1}{h}).$$

Finally, we proved that $||N - \hat{N}_{\vec{k}}||_{H^1(Y)} \leq C \cdot h \cdot log(\frac{1}{h})$. □

Proof of theorem 2.

Notice that $p_1(x,y) = N(x,y)\nabla p_0(x)$, therefore it is enough to prove the estimate just for $p_0(x)$.

Using the definition of the homogenized coefficient κ^*, multiplying by a test function Φ_j and integrating over the domain D, the Equation (3.9) can be written as

$$-\int_D \frac{dp_0}{dt}\Phi_j = \int_D \int_Y \kappa(x,y)\nabla_x p_0 \nabla_x \Phi_j dy dx + \sum_{k=1}^{M} \int_D \int_Y \kappa(x,y)\nabla_y p_1 \nabla_x \Phi_j(y) dy dx + \\ + \int_D f(u)\Phi_j(x) dx \tag{3.56}$$

and

$$-\int_D \frac{d\hat{p}_0}{dt}\Phi_j = \int_D \int_Y \hat{\kappa}(x,y)\nabla_x \hat{p}_0 \nabla_x \Phi_j dy dx + \sum_{k=1}^{M} \int_D \int_Y \hat{\kappa}(x,y)\nabla_y \hat{p}_1 \nabla_x \Phi_j(y) dy dx + \\ + \int_D f(u)\Phi_j(x) dx, \tag{3.57}$$

where $j = 1, ..., N$.

Subtracting equations (Equation (3.56) - Equation (3.57)) and taking into account that p_0 just depends on x, we have:

$$\begin{aligned}
-\int_D \frac{d(p_0-\hat{p}_0)}{dt}\Phi_j = &\int_D\int_Y \kappa(x,y)\nabla_x(p_0-\hat{p}_0)\nabla_x\Phi_j dydx+\\
&\int_D\int_Y \kappa(x,y)\nabla_y N\nabla_x(p_0-\hat{p}_0)\nabla_x\Phi_j dydx+\\
&\int_D\int_Y (\kappa(x,y)-\hat{\kappa}(x,y))\nabla_x\hat{p}_0\nabla_x\Phi_j dydx+\\
&\sum_{k=1}^{M}\int_D\int_Y (\kappa(x,y)-\hat{\kappa}(x,y))\nabla_y\hat{p}_1\nabla_x\Phi_j(y)dydx\\
&\int_D\int_Y \kappa(x,y)\nabla_x p_0\nabla_y(N-\hat{N})\nabla_x\Phi_j dydx.
\end{aligned} \tag{3.58}$$

Now take $\Phi_j = (p_0 - \hat{p}_0)$ and integrating over the time, the Equation (3.58) can be rewritten as

$$\begin{aligned}
\frac{1}{2}\int_0^T\int_D \frac{d(p_0-\hat{p}_0)^2}{dt}dxdt + \int_0^T\int_D\int_Y \kappa(x,y)(I+\nabla_y N)(\nabla_x(p_0-\hat{p}_0))^2 dydxdt = \\
\int_0^T\int_D\int_Y \kappa(x,y)\nabla_y(\hat{N}-N)\nabla_x p_0\nabla_x(p_0-\hat{p}_0)dydxdt+\\
\sum_{k=1}^{M}\int_0^T\int_D\int_Y \kappa(x,y)\nabla_y(\hat{N}-N)\nabla_y\hat{p}_1\nabla_x(p_0-\hat{p}_0)dydxdt+\\
\int_0^T\int_D\int_Y \kappa(x,y)\nabla_y(\hat{N}-N)\nabla_x\hat{p}_0\nabla_x(p_0-\hat{p}_0)dydxdt.
\end{aligned} \tag{3.59}$$

We assume $< \kappa(x,y)(I+\nabla_y N) >\geq \delta$ for some $\delta > 0$, then the following estimate can be obtained

$$min(\frac{1}{2},\delta)\cdot\int_0^T\int_D((p_0-\hat{p}_0)^2+(\nabla_x(p_0-\hat{p}_0))^2)dxdt \leq C'\cdot\|N-\hat{N}\|\cdot\|p_0-\hat{p}_0\|_{H^1}.$$

Finally, $\|p_0-\hat{p}_0\|_{H^1} \leq C''\cdot\|N-\hat{N}\|_{H^1} \leq \tilde{C}\cdot h\cdot log(h^{-1})$.

□

Proof of theorem 3.

According Theorem 2 we have that $\|p_0 - \hat{p}_0\|_{H^1} \leq \tilde{C} \cdot h \cdot log(h^{-1})$.

Clearly, $\|q_0 - \hat{q}_0\|_{H^1} = \|C\|_{L^\infty} \cdot \|p_0 - \hat{p}_0\|_{H^1} \leq \overline{C} \cdot h \cdot log(h^{-1})$ for some constant $\overline{C}$. □

3.4 Sparse approximation

Below we offer a general construction of sparse tensor product FEM and apply it for given input-output system (3.19). It was showed in [40] that the convergence rate for proposed approach is fundamentally the same as for full FEM.

To construct the sparse tensor product finite element spaces in $\Omega \times Y$ we define hierarchical sequence of finite element spaces: $V^0 \subset V^1 \subset V^2 \subset ...$ - subspaces of $H^1(\Omega)$, $V_\sharp^0 \subset V_\sharp^1 \subset V_\sharp^2 \subset ...$ - subspaces of $H_\sharp^1(Y)$. Notice that $p_1 \in L^2(\Omega, H^1(Y)) \approx L^2(\Omega) \bigotimes H^1(Y)$, so we assume the finite element basis for p_1 is $\underline{V}^L = V^L \bigotimes V_\sharp^L$. Define $W^l = V^l - V^{l-1}$ such as $V^l = W^l \bigoplus V^{l-1}, l = 1, .., L$.

Then the full tensor product space can be written as

$$\underline{V}^L = \bigoplus_{0 \leq l_0, l_1 \leq L} W^{l_0} \bigotimes W_\sharp^{l_1},$$

while the sparse tensor product space can be written as

$$\underline{\hat{V}}^L = \bigoplus_{0 \leq l_0 + l_1 \leq L} W^{l_0} \bigotimes W_\sharp^{l_1}.$$

We say that $dim(\underline{V}^L) = K$.

Notice that the dimension of $\underline{V}^L$ is $O(2^{2dL})$ and the dimension of $\underline{\hat{V}}^L$ is much less, $O(2^{dL}L)$.

Now we can define the basis of spaces as: $W^{l_0} = span(\Psi_{l_0 1}, \Psi_{l_0 2}, ..., \Psi_{l_0 j_{l_0}})$, $W^{l_1} = span(\tilde{\Psi}_{l_1 1}, \tilde{\Psi}_{l_1 2}, ..., \tilde{\Psi}_{l_1 j_{l_1}})$ for some fixed $l_0, l_1 \in R$. Then the following rep-

resentations hold $p_1(x,y) = \sum_{0\le l_0+l_1\le L}\sum_{k=1}^{j_{l_0}}\sum_{m=1}^{j_{l_1}} d_{kl}^{l_0 l_1}\Psi_{l_0 k}(x)\tilde{\Psi}_{l_1 m}(y)$ and $p_0(x) = \sum_{i=1}^{N} c_i\Phi_i(x)$.

Now we can follow the same procedure as for full tensor space.

We assume $\zeta_0 = \Phi_j(x)$ for some $j = 1, ..., N$ and $\zeta_1 = \Psi_{l_0 r}(x)\tilde{\Psi}_{l_1 s}(y)$ for some $r = 1, ..., j_{l_0}$ and $s = 1, ..., j_{l_1}$.

Therefore, the equation (3.14) can be rewritten as following:

$$\int_D \sum_{i=1}^{N}\frac{dc_i(t)}{dt}\Phi_i(x)\Phi_j(x)dx = \int_D\int_Y \kappa(x,y)(\sum_{i=1}^{N} c_i\nabla_x\Phi_i(x))\nabla_x\Phi_j(x)dydx+$$

$$+\int_D\int_Y \kappa(x,y)(\sum_{i=1}^{N} c_i\nabla_x\Phi_i(x))\Psi_{l_0 r}(x)\nabla_y\tilde{\Psi}_{l_1 s}(y)dydx+ \tag{3.60}$$

$$+\int_D\int_Y \kappa(x,y)(\sum_{0\le l_0+l_1\le L}\sum_{k=1}^{j_{l_0}}\sum_{m=1}^{j_{l_1}} d_{km}^{l_0 l_1}\Psi_{l_0 k}(x)\nabla\tilde{\Psi}_{l_1 m}(y))\nabla_x\Phi_j(x)dydx+$$

$$+\int_D\int_Y \kappa(x,y)(\sum_{0\le l_0+l_1\le L}\sum_{k=1}^{j_{l_0}}\sum_{m=1}^{j_{l_1}} d_{km}^{l_0 l_1}\Psi_{l_0 k}(x)\nabla\tilde{\Psi}_{l_1 m}(y))\Psi_{l_0 r}(x)\nabla_y\tilde{\Psi}_{l_1 s}(y)dydx+$$

$$+\int_D f(u)\Phi_j(x)dx.$$

We reorganize terms of equation (3.60):

$$\sum_{i=1}^{N}(\int_D \Phi_i(x)\Phi_j(x)dx)\frac{dc_i(t)}{dt} = \sum_{i=1}^{N}(\int_D\int_Y \kappa(x,y)\nabla_x\Phi_i(x)\nabla_x\Phi_j(x)dydx)c_i+$$

$$+\sum_{i=1}^{N}(\int_D\int_Y \kappa(x,y)\nabla_x\Phi_i(x)\Psi_{l_0 r}(x)\nabla_y\tilde{\Psi}_{l_1 s}(y)dydx)c_i+$$

$$+\sum_{0\le l_0+l_1\le L}\sum_{k=1}^{j_{l_0}}\sum_{m=1}^{j_{l_1}}(\int_D\int_Y \kappa(x,y)\Psi_{l_0k}(x)\nabla\tilde{\Psi}_{l_1m}(y)\nabla_x\Phi_j(x)dydx)d_{km}^{l_0l_1}+$$

$$+\sum_{0\le l_0+l_1\le L}\sum_{k=1}^{j_{l_0}}\sum_{m=1}^{j_{l_1}}(\int_D\int_Y \kappa(x,y)\Psi_{l_0k}(x)\nabla\tilde{\Psi}_{l_1m}(y)\Psi_{l_0r}(x)\nabla_y\tilde{\Psi}_{l_1s}(y)dydx)d_{km}^{l_0l_1}+$$

$$+\int_D f(u)\Phi_j(x)dx.$$

Using matrix notation, the equation can be written in the following form:

$$Q\begin{bmatrix}\frac{dc_1}{dt}\\ ...\\ \frac{dc_N}{dt}\\ 0\end{bmatrix}=\begin{bmatrix}(A)_{N\times N}(\tilde{B})_{N\times K}\\ (\tilde{B}^*)_{K\times N}(\tilde{S})_{K\times K}\end{bmatrix}\begin{bmatrix}c_1\\ ...\\ c_N\\ d_1\\ ...\\ d_K\end{bmatrix}+\begin{bmatrix}F_1\\ ...\\ F_N\\ 0\\ ...\\ 0\\ ...\\ 0\end{bmatrix}=$$

$$=\begin{bmatrix}(A)_{N\times N}\\ (\tilde{B}^*)_{K\times K}\end{bmatrix}\begin{bmatrix}(c)_{1\times N}\end{bmatrix}+\begin{bmatrix}(\tilde{B})_{N\times K}\\ (\tilde{S})_{K\times K}\end{bmatrix}\begin{bmatrix}(d)_{1\times K}\end{bmatrix}+\begin{bmatrix}(F)_{1\times N}\\ (0)_{1\times K}\end{bmatrix}, \tag{3.61}$$

where $A_{ij}=\int_D\int_Y \kappa(x,y)\nabla_x\Phi_i(x)\nabla_x\Phi_j(x)dydx$ for $i=1,...,N, j=1,...,N$;

$(\tilde{B})_{i(kj)}=\int_D\int_Y \kappa(x,y)\Psi_k(x)\nabla\tilde{\Psi}_j(y)\nabla_x\Phi_i(x)dydx$ for $(kj)=1,...,K, i=1,...,N$;

$F_j=\int_D f(u)\Phi_i(x)dx$ for $j=1,...,N$;

$(\tilde{S})_{(rj)(il)}=\int_D\int_Y \kappa(x,y)\Psi_r(x)\nabla\tilde{\Psi}_j(y)\Psi_i(x)\nabla_y\tilde{\Psi}_l(y)dydx$ for $(rj)=1,...,K, (il)=1,...,K$;

$Q_{ij}=\int_D \Phi_i(x)\Phi_j(x)dx$ for $i=1,...,N, j=1,...,N$ (assume $Q_{ij}=0$ for all

$i = N+1, ..., N+K, j = N+1, ..., N+K$).

From the last equation of (3.61)

$$\tilde{B}^* c + \tilde{S} d = 0,$$

and the expression of d in terms of c can be found: $d = -\tilde{S}^{(-1)} \tilde{B}^* c$.

The size of matrix $\tilde{S}$ is comparable to the size of whole system. The sparse tensor product finite element method can be used in order to reduce the dimension of $\tilde{S}$ and be able to compute $\tilde{S}^{(-1)}$.

We denote the 'non-zero' part of the matrix Q as $\tilde{Q}$, i.e., $\tilde{Q}$ is $N \times N$ matrix, where coefficients $\tilde{Q}_{ij} = \int_D \Phi_i(x)\Phi_j(x)dx$ for $i = 1, ..., N, j = 1, ..., N$. Then (3.61) can be rewritten as:

$$\tilde{Q}\frac{dc}{dt} = Ac + \tilde{B}d + F = Ac - \tilde{B}\tilde{S}^{(-1)}\tilde{B}^* c + F = (A - \tilde{B}\tilde{S}^{(-1)}\tilde{B}^*)c + F.$$

Notice that the output:

$$q(p^\epsilon) = \int_D l \cdot \nabla p^\epsilon dx \approx \int_D l \cdot \nabla_x p_0 dx + O(\epsilon^{\frac{1}{2}}),$$

so denoting by C the operator such that $Cc = \int_D l(x) \cdot \nabla_x p_0 dx$, the output can be rewritten as $q = [C \;\; 0] \begin{bmatrix} c \\ d \end{bmatrix} = Cc + 0 = Cc$.

Finally, we get the system:

$$(\tilde{Q})_{N\times N}\frac{dc}{dt} = (A - \tilde{B}\tilde{S}^{(-1)}\tilde{B}^*)c + F, \tag{3.62}$$

$$q = Cc.$$

3.5 Conclusions

This section is devoted to an input-output formulation for a parabolic problem, where input is a defined, controllable quantity and output is a measurable quantity. Proposed approach reduces computational cost of obtaining a solution for the flow in heterogeneous porous media with separable scales.

As a first step of the process, in order to discretize the problem two types of FEM were considered standard (full tensor product) and sparse tensor product FEM. The main advantage of a sparse tensor product approach over standard is the fact of getting a system of much smaller size, while convergence rates are comparable to full tensor product FEM. As a second step, a combination of both global and local model reduction techniques was applied in order to carry out computations on a discretized system. The following approach was developed for both sparse and full tensor product FEM. First, two-scale homogenization was applied to the system. Hierarchical approach allows solving local problems with different resolutions and then combines these multi-resolution solutions to achieve an accurate approximation. Secondly, the system is transformed in order to apply a global model reduction technique BT Method. The latter enables to get a good approximation of a solution by solving a system of much smaller size than the original one.

4. COARSE-GRID MODEL REDUCTION WITH BALANCED TRUNCATION FOR FLOWS IN GENERAL HETEROGENEOUS POROUS MEDIA WITH A PARAMETER *

4.1 Preliminaries

In this section we concentrate on a case where the permeability doesn't have a scale separation. The following section was published at [50]. We consider a problem in which we are interested in the effects that a small dimensional set of forcing inputs has on a small dimensional set of measurable outputs. In the context of flow modeling, the forcing inputs may represent an injection-production well configuration, and the measurable outputs may represent fixed or averaged pressure values at specified regions of the porous medium. We emphasize that in a basic framework, each evaluation of the input-output mapping requires the solution(s) of the model equation such that the measurable outputs are available. As such, cost effective alternatives to using fully resolved approximation techniques are desirable. Due to the localized construction of the coarse spaces mentioned above, GMsFEM may be viewed as a local model reduction technique for the approximation of the model equation (or the subsequent output quantities). More specifically, GMsFEM yields a reduced-order model in which the mapping is approximated by solving smaller systems of equations to obtain the output values.

We propose a local-global model reduction technique in which GMsFEM is combined with balanced truncation (BT) (see, e.g., [22, 33]). BT is a global model reduction technique that offers rigorous apriori error bounds and solution stability, and

*Reprinted from Journal of Computational and Applied Mathematics, Vol. 271, M. Presho, A. Protasov, E. Gildin, "Local-global model reduction of parameter-dependent, single-phase flow models via balanced truncation", Pages 163-179, Copyright 2014, with permission from Elsevier.

has been shown to be an effective model reduction tool for a variety of applications. For example, BT was recently applied to single-phase flow equations in [22], and we refer the interested reader to [33] for a more extensive discussion on using BT as an effective global model reduction technique. The main idea is recast the partial the differential equations into the systems framework, in which an input-to-output mapping, that is, transfer function, is constructed based on the internal state-variables. Based on how much we can steer a state variable from an initial condition to a final condition given a finite set of inputs, or how much we can infer the state of these variables based on some measurements (output) of the system, we can determine the most important state variables in this input-output mapping. In order to apply the method, we solve a set of coupled Lyapunov equations that use matrices from a given input-output dynamical system [38, 57]. The solutions of the Lyapunov equations are called the observability and controllability Gramians, and their combination is used to construct a reduced order system through using truncated eigenvectors of the specified matrices. The associated eigenvalues (so-called Hankel singular values) that appear in the apriori bounds have been shown to decay rapidly (see [2, 31]), and therefore the measurable outputs of the system can be accurately approximated by solving a substantially reduced order model. An important consideration of BT is that the solutions of the Lyapunov equations can be very expensive to compute [11, 14, 20]. Thus, BT may not be a suitable global model reduction technique if a fully-resolved system of equations yields Lyapunov equations that are prohibitively expensive to solve. As such, applying BT to a coarse-scale set of equations obtained through GMsFEM is shown to be a tractable approach for solving the model equation considered in this work.

4.2 Offline-online Balanced Truncation approach

An application of BT approach to parameter-dependent problems can be computationally insufficient since we have to solve series of Lyapunov equations. In particular, solving these equations directly on the fine scale for varying permeability samples, boundary conditions, and forcing terms will quickly become prohibitively expensive. BT of linear time-varying system was studied in [54], where error bounds were derived. However, the computational cost of a problem linearly depends on time steps which further necessitate increasing the efficiency of the method. In turn, in this work we propose the use of a multiscale, offline-online BT approach in order to obtain inexpensive solutions. The method is based on GMsFEM that allows us to construct an independent set of online basis functions in which a coarse solution may be sought. As the resulting global system will then be posed on a coarse grid, we avoid the need to implement BT directly on the fine grid.

The method consists of two stages: offline and online. At the offline stage, a space of "snapshots" is first computed. Then, the fixed offline space is constructed through a spectral reduction of the original snapshot space. We emphasize that the offline space is constructed such that it is independent of the input space (e.g., the input parameters that define the permeability field), and is used during the online stage in order to efficiently construct a set of multiscale basis functions (and online solution space). In essence, the online step yields a small dimensional subspace of the offline space for each fixed μ. As such, at the online stage coarse solutions may be quickly computed for specified input parameters. We remark that this type of construction is a main contribution of this work, and serves to further decrease the computational cost associated with the local model reduction. In particular, the online space construction will no longer depend on eigenvalue problems that

scale with respect to the size of the local fine mesh. Rather, the online eigenvalue computations are performed within a significantly reduced space that depends on the pre-processed offline eigenfunctions. A related local-global model reduction approach was offered in [22], where the authors considered a parameter-independent model that did not necessitate the need for a more efficient offline-online approach.

4.2.1 Generalized Multiscale Finite Element Method

4.2.1.1 Offline computation

At the offline stage, we first construct a snapshot space $V_{\text{snap}}^{\omega_i}$ subordinated to each coarse neighborhood ω_i in the domain (refer back to Figure 2.1). Construction of the snapshot space involves solving a set of local problems for a suitable range of input parameters, and in order construct the space we solve the following eigenvalue problems on each coarse neighborhood ω_i:

$$A(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} = \lambda_{l,j}^{\omega_i,\text{snap}} S(\mu_j)\psi_{l,j}^{\omega_i,\text{snap}} \quad \text{in } \omega_i, \tag{4.1}$$

where μ_j $(j = 1, \ldots, J)$ is a specified set of fixed parameter values. Neumann boundary conditions are used for solving the eigenvalue problems, however mention that other global formulations may warrant the use of modified boundary conditions [23]. The matrices in Equation (4.1) are defined as

$$A(\mu_j) = [a(\mu_j)_{mn}] = \int_{\omega_i} \kappa(x;\mu_j)\nabla\phi_n \cdot \nabla\phi_m, \quad S(\mu_j) = [s(\mu_j)_{mn}] = \int_{\omega_i} \widetilde{\kappa}(x;\mu_j)\phi_n\phi_m, \tag{4.2}$$

where ϕ_n denotes the standard bilinear, fine-scale basis functions and $\widetilde{k}$ will be defined in the next section (cf. (4.8)). We point out that Equation (4.1) is the discretized

form of the continuous equation

$$-\text{div}(\kappa(x,\mu_j)\nabla\psi_{l,j}^{\omega_i,\text{snap}}) = \lambda_{l,j}^{\omega_i,\text{snap}}\psi_{l,j}^{\omega_i,\text{snap}} \quad \text{in } \omega_i.$$

For notational brevity we now omit the superscript ω_i for the eigenvalue problems, yet it is assumed throughout this section that the offline and online space computations are localized to respective coarse neighborhoods. After solving Equation (4.1), we keep the first L_i eigenfunctions that correspond to the dominant eigenvalues (asymptotically vanishing in this case) in order to form the snapshot space

$$V_{\text{snap}} = \text{span}\{\psi_{l,j}^{\text{snap}} : \quad 1 \leq j \leq J \text{ and } 1 \leq l \leq L_i\},$$

for each coarse neighborhood ω_i. We then reorder the snapshot functions using a single index to create the matrix

$$R_{\text{snap}} = \left[\psi_1^{\text{snap}}, \ldots, \psi_{M_{\text{snap}}}^{\text{snap}}\right], \tag{4.3}$$

where M_{snap} denotes the total number of functions that are kept in the snapshot matrix construction.

At this stage, we perform a dimension reduction of the space of snapshots using an auxiliary spectral decomposition in order to construct the offline space $V_{\text{off}}^{\omega_i}$. More precisely, we seek a subspace of the snapshot space such that it can approximate any element of the snapshot space in the appropriate sense defined via auxiliary bilinear forms. The main objective is to use the offline space to efficiently (and accurately) construct a set of multiscale basis functions for each μ value at the online stage. And we reiterate that the offline bilinear forms are chosen to be *parameter-independent*, such that there is no need to reconstruct the offline space for each μ value. The

analysis in [25] motivates the following eigenvalue problem in the space of snapshots:

$$A^{\text{off}}\Psi_k^{\text{off}} = \lambda_k^{\text{off}} S^{\text{off}}\Psi_k^{\text{off}}, \tag{4.4}$$

where

$$A^{\text{off}} = [a_{mn}^{\text{off}}] = \int_{\omega_i} \overline{\kappa}(x;\mu)\nabla\psi_m^{\text{snap}} \cdot \nabla\psi_n^{\text{snap}} = R_{\text{snap}}^T \overline{A} R_{\text{snap}}$$

and

$$S^{\text{off}} = [s_{mn}^{\text{off}}] = \int_{\omega_i} \overline{\overline{\kappa}}(x;\mu)\psi_m^{\text{snap}}\psi_n^{\text{snap}} = R_{\text{snap}}^T \overline{S} R_{\text{snap}},$$

where $\overline{\kappa}(x,\mu)$, and $\overline{\overline{\kappa}}(x,\mu)$ are domain-based averaged coefficients. Here, $\overline{A}$ and $\overline{S}$ denote similar fine scale matrices as defined in Equation (4.1), except that averaged coefficients are used in the construction. To finalize the offline space we then choose the smallest M_{off} eigenvalues from Equation (4.4) and form the corresponding eigenvectors in the 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

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

to be used in the online space construction.

4.2.1.2 Online computation

With the offline space available, we next construct the associated online coarse space $V_{\text{on}}^{\omega_i}(\mu)$ on each coarse subdomain for a specified μ realization. In principle, we want the online space to be a small dimensional subspace of the offline space for computational efficiency. More specifically, we seek a subspace of the offline space such that it can approximate any element of the offline space in an appropriate sense. We reiterate that at the online stage, the reduced-order bilinear forms are chosen to

be *parameter-dependent*. Similar analysis (see [25]) motivates the following eigenvalue problem to be solved in the offline space:

$$A^{\text{on}}(\mu)\Psi_k^{\text{on}} = \lambda_k^{\text{on}} S^{\text{on}}(\mu)\Psi_k^{\text{on}}, \tag{4.5}$$

where

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

and

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

and $\kappa(x;\mu)$ and $\widetilde{\kappa}(x;\mu)$ are now depend on the specified μ value. Finally, to generate the online space we choose the smallest M_{on} eigenvalues from Equation (4.5) 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}. We emphasize that the size matrices in (4.5) now solely depends on the dimension of the offline space. As such, the diminished computational cost of creating the online space is in contrast to the more expensive alternative of directly using localized fine-grid computations in the space construction.

4.2.1.3 Global formulation using online space

In order to incorporate our online basis functions into a reduced-order global formulation of Equation (2.1) we start with an initial coarse space $V^{\text{init}}(\mu) = \text{span}\{\chi_i\}_{i=1}^{N_v}$, where N_v denotes the number of coarse nodes. We use χ_i to denoted the standard multiscale partition of unity functions which are defined by

$$-\text{div}\big(\kappa(x;\mu)\,\nabla\chi_i\big) = 0 \quad K \in \omega_i, \tag{4.6}$$

$$\chi_i = g_i \quad \text{on } \partial K, \tag{4.7}$$

for all $K \in \omega_i$, where g_i is a bilinear boundary condition. Using the initial partition of unity we define the summed, pointwise energy $\widetilde{\kappa}$ (cf. Equation (4.2)) as

$$\widetilde{\kappa} = \kappa \sum_{i=1}^{N_v} H^2 |\nabla\chi_i|^2, \tag{4.8}$$

where H denotes the coarse mesh size. In order to construct the global, coarse-grid solution space we then multiply the partition of unity functions by the online eigenfunctions from the space $V_{\text{on}}^{\omega_i}(\mu)$ to form the basis functions

$$\psi_{i,k} = \chi_i \psi_k^{\omega_i,\text{on}} \quad \text{for } 1 \leq i \leq N_v \text{ and } 1 \leq k \leq M_{\text{on}}^{\omega_i}, \tag{4.9}$$

where we recall that $M_{\text{on}}^{\omega_i}$ denotes the number of online basis functions to keep for each ω_i. We note that the basis construction in Equation (4.9) yields a basis set to be used within a continuous Galerkin global formulation, however, continuity is not a requirement for a discontinuous Galerkin formulation (see [23]). The online, spectral multiscale space is then defined as

$$V_{\text{on}}(\mu) = \text{span}\{\psi_{i,k} : 1 \leq i \leq N_v \text{ and } 1 \leq k \leq M_{\text{on}}^{\omega_i}\}, \tag{4.10}$$

and using a single index notation, we write $V_{\text{on}}(\mu) = \text{span}\{\psi_i\}_{i=1}^{N_c}$, where N_c denotes the total number of basis functions that are used in the coarse scale formulation. Recalling Equation (2.6) we now introduce the final form for the operator matrix R

to be used in the global system construction. In particular, we set $R = [\psi_1, \ldots, \psi_{N_c}]$, where ψ_i represents the vector of nodal values of each basis function defined on the fine grid. To solve Equation (2.1) we seek $p_c(x,t;\mu) = \sum_i b_i(t)\psi_i(x;\mu) \in V_{\text{on}}$ such that

$$\int_D \frac{\partial p_c}{\partial t} v = -\int_D \kappa(x;\mu)\nabla p_c \, \nabla v + \int_D f(u)\, v \;\text{ for all } v \in V_{\text{on}}. \tag{4.11}$$

The above equation yields the discrete form

$$M^*(p_c)_t = -A^*(\mu)p_c + B^*u, \tag{4.12}$$

$M^* := [m_{IJ}] = \displaystyle\int_D \psi_i\psi_j$ is a coarse mass matrix, $A^*(\mu) := [a_{IJ}] = \displaystyle\int_D \kappa(x;\mu)\nabla\psi_i \, \nabla\psi_j$ is a coarse stiffness matrix, $B^* := [b_I] = \displaystyle\int_D f(u)\,\psi_i$, and ψ_i denotes the coarse basis functions that span V_{on}. As previously mentioned in Subsection 2.2, the coarse matrices may be written using the operator matrix R as follows: $M^* = R^T M R$, $A^*(\mu) = R^T A(\mu) R$, and $B^* = R^T B$, where M, $A(\mu)$, and R are the fine scale matrices from Equation (2.4). The operator matrix may analogously be used to project coarse scale solutions back to the fine grid. We note that expressing the linear system as

$$\begin{aligned} (p_c)_t &= -A^c(\mu)p_c + B^c u, \\ q^c &= C^c p_c, \end{aligned} \tag{4.13}$$

where $A^c(\mu) = (M^*)^{-1}A^*(\mu)$, $B^c = (M^*)^{-1}B^*$, and $C^c = CR$ is the appropriate form for applying the BT approach that is described in the next subsection.

4.2.2 Balanced Truncation

The process of global model reduction may be generally stated as follows: for a given model G (such as the system in Equation (4.13)), we wish find a lower order system G_r such that G and G_r are close in some sense [57]. In our case, to reduce the size of the coarse system

$$\begin{aligned}(p_c)_t &= -A^c(\mu)p_c + B^c u,\\ q^c &= C^c p_c,\end{aligned}$$

we will apply BT.

In order to reduce the complexity of the system, we can find state space variables that can be truncated without compromising the input-output behavior of G. This can be accomplished by finding the states that are very difficult to steer (weakly controllable) and simultaneously states that cannot be inferred from the measurements or outputs (weakly observable). To this end, we form a joint measure of observability and controllability through solving the Lyapunov equations

$$\begin{aligned}W_{\text{ob}}A^c(\mu) + \big(A^c(\mu)\big)^T W_{\text{ob}} + (C^c)^T C^c = 0,\\ A^c(\mu)W_{\text{co}} + W_{\text{co}}\big(A^c(\mu)\big)^T + B^c(B^c)^T = 0,\end{aligned} \tag{4.14}$$

where W_{ob} is the so-called observability Gramian, and W_{co} is the controllability Gramian [38]. The solution of these two Lyapunov equations play an important role in model reduction and in particular in the balanced truncation method. Several algorithms have been devised to efficiently compute their solution. In the simplest form, they depend on dense matrix computations such as in the the Bartels-Stewart [10] method as modified by Hammarling [35] and are in general of order $\mathcal{O}(n^3)$. Many other algorithms have been developed to reduced its computational complex-

ity, but they rely on iterative methods and low rank approximations of the solution [34, 48, 52], which in many cases may not give an appropriate solution. Once the Gramians are available, they are expressed in the form of their Cholesky decompositions: $W_{\text{ob}} = L_{\text{ob}}^T L_{\text{ob}}$ and $W_{\text{co}} = L_{\text{co}} L_{\text{co}}^T$. We then find the first N_r ordered eigenvalues and normalized eigenvectors of $L_{\text{co}}^T W_{\text{ob}} L_{\text{co}}$, that is, we solve

$$L_{\text{co}}^T W_{\text{ob}} L_{\text{co}}\, \xi_i = \rho_i \xi_i \quad \text{for} \quad i = 1, \ldots, N_r. \tag{4.15}$$

The eigenvalues and (row) eigenvectors of $L_{\text{ob}}^T W_{\text{co}} L_{\text{ob}}$ are given by $\{\rho_i\}$ and $\{\zeta_i\} = \{\sigma_i^{-1} \xi_i^T L_{\text{co}}^T L_{\text{ob}}^T\}$ $(i = 1, \ldots, N_r)$, where $\sigma_i = \rho_i^{1/2}$. With the respective eigenvalues and eigenvectors in place we form the matrices U and V such that

$$V = L_{\text{co}} \left[\xi_1 \sigma_1^{-1/2} \cdots \xi_{N_r} \sigma_{N_r}^{-1/2}\right] \quad \text{and} \quad U = \begin{bmatrix} \sigma_1^{-1/2} \zeta_1 \\ \vdots \\ \sigma_{N_r}^{-1/2} \zeta_{N_r} \end{bmatrix} L_{\text{ob}},$$

with $UV = I_{N_r}$. The reduced order model of Equation (4.13) is finally formed by creating the matrices $A^r(\mu) = UA^c(\mu)V$, $B^r = UB^c$, and $C^r = C^c V$, where the value of r depends on the decay of these eigenvalues. This will be shown in the numerical experiments in the next section. In particular, we arrive at the reduced dimension model (e.g., $A^r(\mu)$ has size $N_r \times N_r$) where

$$\begin{aligned} (p_r)_t &= -A^r(\mu) p_r + B^r u, \\ q^r &= C^r p_r, \end{aligned} \tag{4.16}$$

and the output q^r closely approximates q^c.

A notable property of BT is that the apriori error estimates simply depend on the eigenvalues of Equation (4.15) (see [31]). In particular, the error estimate de-

pends upon the magnitude of the summation of all truncated eigenvalues. Another advantage of BT is the fact that the stability of the resulting system is preserved [4]. However, a possible limitation of the method is that the Lyapunov equations in Equation (4.14) require costly, dense computations. As a result, applying BT directly to a large dimensional input-output system may become prohibitively expensive. In this section, the combined GMsFEM-BT approach avoids the need for large dimensional computations while still maintaining a suitable level of accuracy. More specifically, since the method is applied directly to a online coarse-scale system (which has already been reduced using local GMsFEM), the computational cost can be significantly reduced.

4.3 Numerical results

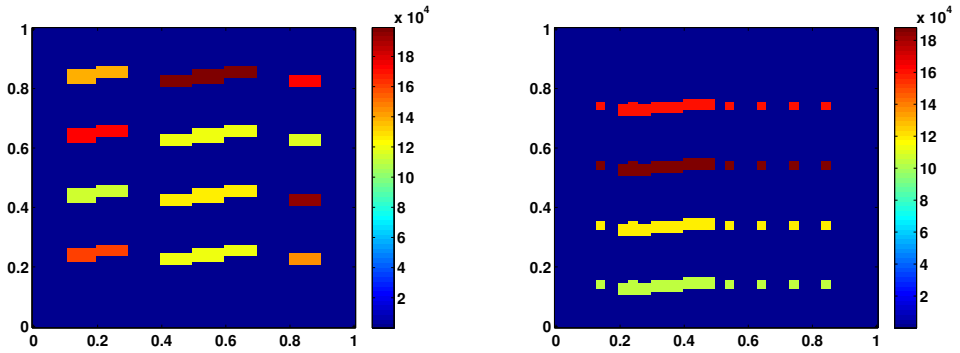
In this section we offer a variety of numerical results to validate the performance of the proposed offline-online balanced truncation approach. All solutions are computed on the two-dimensional unit domain $D = [0,1]\times[0,1]$, and we assume a homogeneous boundary condition $p = 0$. For the initial condition we use $p_0(x) = 0$. We let $f(u)$ be a piecewise constant forcing parameter on a coarse grid, such that the j^{th} column of B (recall Equation (2.4)) is defined as

$$B_{(:,j)} = \begin{cases} 1 & \text{if } x \in K_j, \\ 0 & \text{otherwise,} \end{cases}$$

where we use K_j to denote the j^{th} coarse element of the domain. In essence, the matrix B is a mapping of a coarse-dimensional parameter u to the fine grid. For all examples in this section, u is taken to be a multi-dimensional random parameter subordinated to the coarse grid such that each component is uniformly distributed on $[0,1]$. For each set of examples, a new input sample is drawn and used for the

respective comparisons. Additionally, we assume that the output selection matrix is defined by $C = B^T$. As a result, the output quantities of interest are taken as upscaled pressure solutions localized to the coarse elements. In order to solve the fine scale problem in Equation (2.1), we discretize D into a 100×100 fine element mesh.

For the first set of examples, we assume that the permeability field from Equation (2.1) takes the form $\kappa(x,\mu) = \mu_1\kappa_1(x) + \mu_2\kappa_2(x)$, where the pair (μ_1, μ_2) is assumed to be drawn from $[0,1]^2$. See Figure 4.1 for an illustration of the permeability structure. In addition, the resulting field exhibits a randomized high-contrast structure, where the contrast is taken to be $\kappa_{\max}/\kappa_{\min} \approx 10^5$. This type of permeability structure serves to model a field containing enhanced flow conduits such as fractures or channels. For all two-dimensional examples we use fixed values of $\mu_1 = 0.4$ and $\mu_2 = 0.7$ to create the composite field. In order to generate the snapshot space for the offline-online procedure (cf. (4.3)) we use three equally spaced points in each dimension and keep $L_i = 12$ eigenfunctions so that $M_{\text{snap}} = 3^2\times12 = 108$. For this set of examples we initially solve the using the fully-resolved system of equations from (2.4), and compute the corresponding output quantities. This computation is typically the most expensive, and is used as the benchmark for testing the accuracy of proposed method. In addition, we also solve the GMsFEM system in (4.13) and obtain the corresponding output quantities. Finally, we apply BT to the coarse scale equations from (4.13) to significantly reduce the dimension of the resulting system. Throughout this section, we are interested in comparing the outputs resulting from the fully resolved system and the reduced dimension system. In particular, we consider the errors of the output quantities of interest as the respective solutions advance in time. For the comparisons, we denote the fine-scale solutions as $p_f^1, p_f^2, \ldots, p_f^M$, the GMsFEM solutions as $p_c^1, p_c^2, \ldots, p_c^M$, and the reduced solutions obtained by BT



(a) $\kappa_1(x)$ (b) $\kappa_2(x)$

Figure 4.1: Permeability coefficients for a two-dimensional parameter-dependent problem

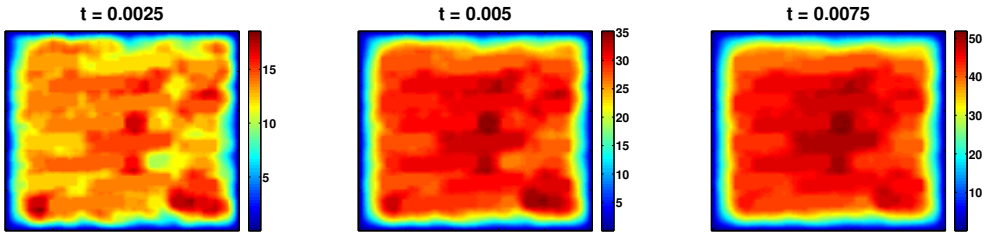
as $p_r^1, p_r^2, \ldots, p_r^M$. Here, we use M to denote the total number of time steps that are used in the simulations. Figure 4.2 represents the quality of GMsFEM for an arbitrary example, and serves as a motivation for online-offline balanced truncation approach.

We then compute the respective output quantities $q^{f,i} = C^f p_f^i$, $q^{c,i} = C^c p_c^i$, and $q^{r,i} = C^r p_r^i$ (for $i = 1, \ldots, M$), where the triangle inequality

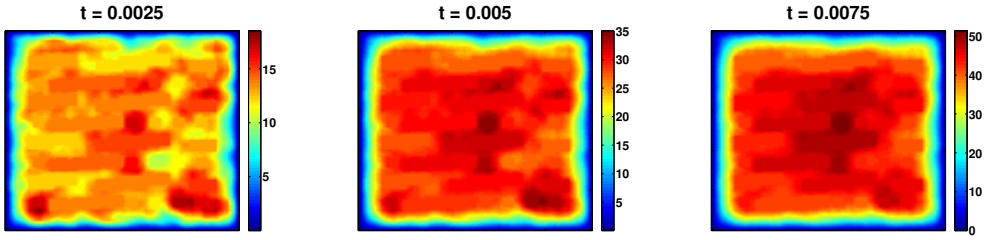
$$\|q^{f,i} - q^{r,i}\|_2 \leq \|q^{f,i} - q^{c,i}\|_2 + \|q^{c,i} - q^{r,i}\|_2 \tag{4.17}$$

serves as a guideline for choosing an appropriate BT dimension. In particular, we wish to choose the dimension such that the errors $\|q^{f,i} - q^{c,i}\|_2$ and $\|q^{c,i} - q^{r,i}\|_2$ are comparable. In doing so, we ensure that the error between the fully-resolved outputs and GMsFEM-BT reduced model is not dominated by a single term. Throughout this section we use a discrete, relative l^2-norm scaled by the coarse output norm.

For the first set of examples, a time step of $\Delta t = 5\times10^{-4}$ is used, and we run the simulations for 20 total time steps such that a steady state is observed. For



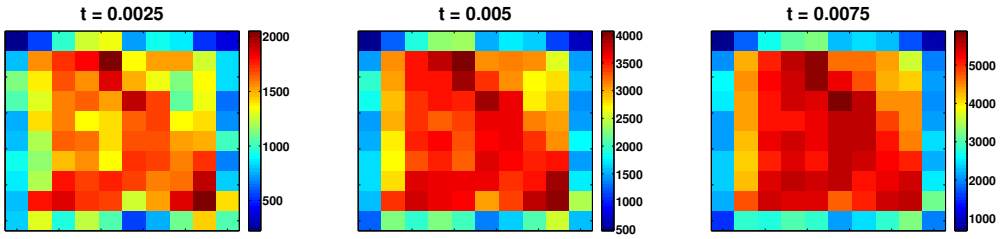
(a) Fine ($N_f = 10201$)



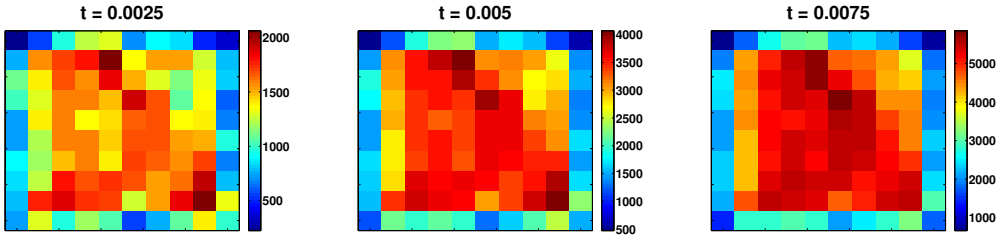
(b) GMsFEM ($N_c = 1524$)

Figure 4.2: Time variant pressure fields for fine-scale and coarse-scale systems

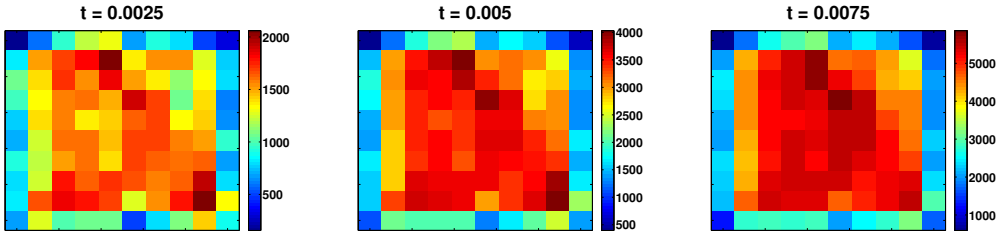
the following examples, we consider 10×10 and 20×20 coarse mesh configurations that yield a variety of online coarse space dimensions. In particular, for the 10×10 coarse mesh we consider online spaces of dimension $N_c = 364, 564$, and 779, and for the 20×20 coarse mesh we consider online spaces of dimension $N_c = 802, 1163$, and 1524. The time variant output for selected fine-scale, coarse-scale, and reduced-order systems is shown in Figures 4.3 and 4.4. Figure 4.3 corresponds to an online space of dimension $N_c = 564$ and Figure 4.4 corresponds to an online space of dimension $N_c = 1524$. In both figures, we plot the respective outputs for three time levels such that the fine output is in the first row, the coarse output is in the middle row, and the reduced output is in the third row. In Figure 4.3 we see from the illustrations that the reduced outputs are nearly indistinguishable from the fine outputs for the case when we use a reduced dimension of $N_r = 70$. Similarly, the outputs in Figure 4.4 are nearly indistinguishable for a reduced dimension of $N_r = 300$. For further comparison, we



(a) Fine ($N_f = 10201$)



(b) GMsFEM ($N_c = 564$)

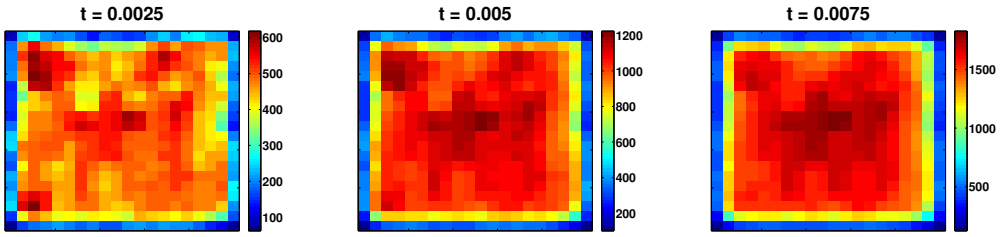


(c) BT ($N_r = 70$)

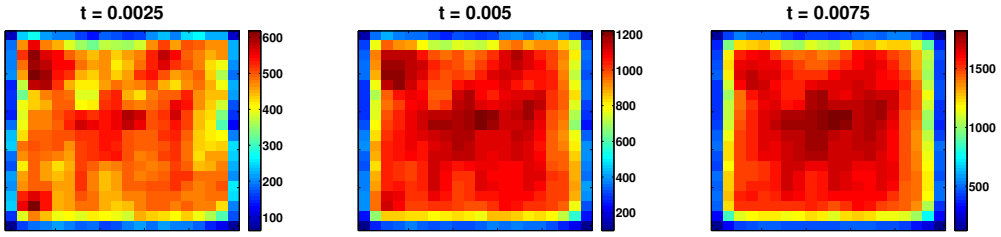
Figure 4.3: Time variant output comparisons for a 10×10 coarse mesh configuration

also offer sets of plots that illustrate the similarities between the fine, coarse, and reduced dimension outputs for either coarse mesh configuration at the steady state time in Figures 4.5 and 4.6. These initial illustrations serve as a motivation for more rigorous error comparisons below. In addition, a more detailed discussion on a suitable choice of reduced dimensions will also be offered.

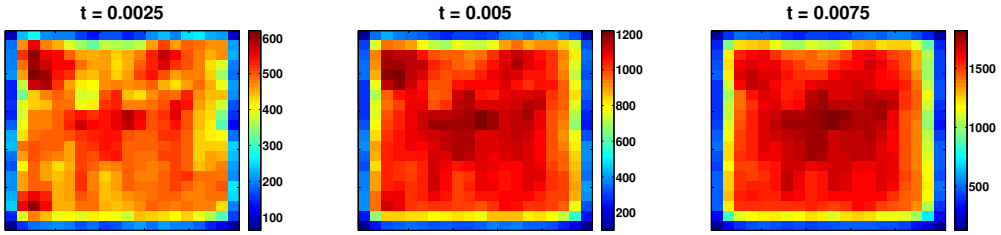
Table 4.1 compares the errors between the outputs of the respective steady state solutions using a variety of online coarse spaces and reduced model dimensions cor-



(a) Fine ($N_f = 10201$)



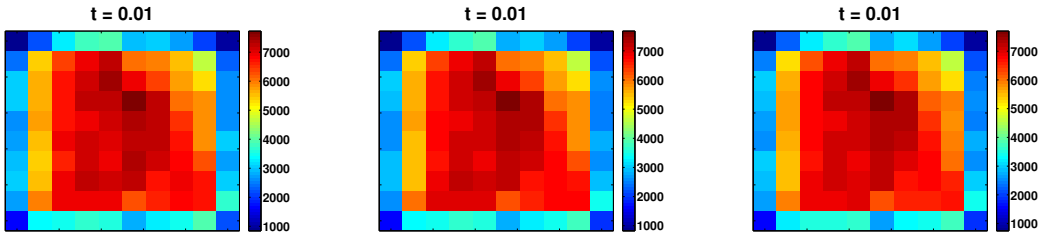
(b) GMsFEM ($N_c = 1524$)



(c) BT ($N_r = 300$)

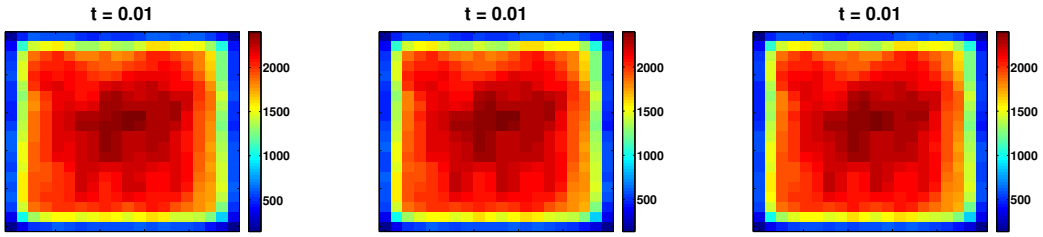
Figure 4.4: Time variant output comparisons for a 20×20 coarse mesh configuration

responding to a 10×10 coarse mesh. In particular, we test the error quantities from Equation (4.17) using online coarse space dimensions of $N_c = 364, 564$, and 779, and BT reduced dimensions of $N_r = 20, 30, 40, 50$; $N_r = 30, 50, 70, 90$ and $N_r = 100, 200, 300, 400$; respectively. We emphasize that the coarse spaces and reduced dimensions are much smaller than the fine-scale system of size $N_f = 10201$. We see from the first two columns of Table 4.1 that the error between the fine-scale and GMsFEM outputs decreases as the dimension of the coarse space increases. In



(a) Fine ($N_f = 10201$) (b) GMsFEM ($N_c = 564$) (c) BT ($N_r = 70$)

Figure 4.5: Steady state output comparisons for a 10×10 coarse mesh configuration



(a) Fine ($N_f = 10201$) (b) GMsFEM ($N_c = 1524$) (c) BT ($N_r = 300$)

Figure 4.6: Steady state output comparisons for a 20×20 coarse mesh configuration

addition, we see that it is sufficient to reduce the size of the BT system to $N_r = 50-70$ for the case of $N_c = 364$, $N_r = 70-90$ for the case of $N_c = 564$, and to $N_r = 110-150$ for the case of $N_c = 779$ in order to obtain a comparable level of reduced dimension accuracy. An analogous set of results corresponding to a 20×20 coarse mesh is offered in Table 4.2. The table suggests that it is sufficient to reduce the size of the BT system to $N_r = 40 - 50$ for the case of $N_c = 802$, $N_r = 150 - 200$ for the case of $N_c = 1163$, and to $N_r = 300 - 400$ for the case of $N_c = 1524$ in order to obtain a comparable level of reduced dimension accuracy. We note that the error estimates associated with the BT reduction (see [31]) loosely govern an appropriate size of the reduced system through a consideration of the Hankel singular values (i.e., the eigenvalues from Equation (4.15)). In particular, we generally choose a system

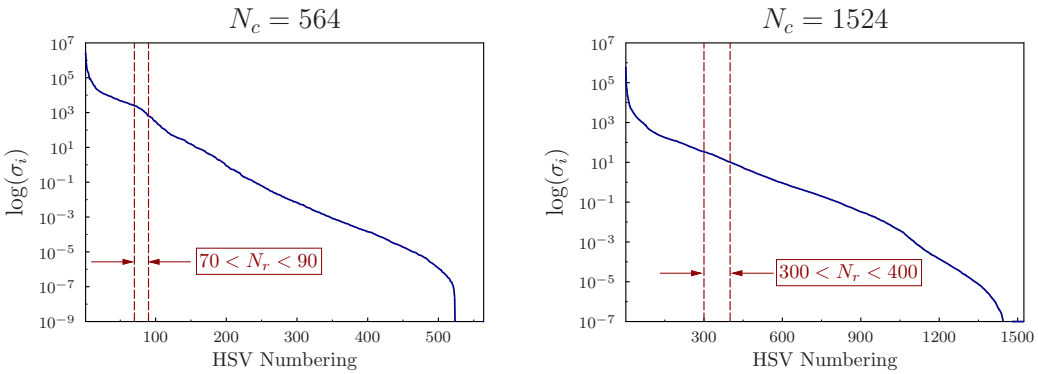


Figure 4.7: Hankel singular value decline as a guideline for choice of BT system size

whose size is consistent with a sufficient decline in the eigenvalue order. For example, we see from Figure 4.7 that the reduced dimensions of $N_r = 70 - 90$ ($N_c = 564$ case) and $N_r = 300 - 400$ ($N_c = 1524$ case) roughly correspond to $3 - 4$ orders of reduction in the respective eigenvalues. We again note that these reduced systems are in stark contrast to the size of the fully-resolved system ($N_f = 10201$), while offering a suitable level of accuracy.

For a broader range of comparisons, we also offer a set of plots in Figure 4.8 that show how the relative errors advance in time for 10×10 and 20×20 coarse mesh examples. For the 10×10 case, we consider $N_c = 564$, and for the 20×20 case we consider $N_c = 1524$. However, we mention that these results are representative for all cases. From Figure 4.8 we see that an increase in the size of reduced-dimension yields a decrease in the time-dependent errors. And as expected from Table 4.1, we see that a suitable range of reduced-dimensions essentially "encapsulates" the fine vs. coarse errors advancing in time, further validating an appropriate choice of BT dimensions.

For a final set of comparisons, we offer a set of computational times corresponding

dim(V_{on}) [N_c]	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size [N_r]
364	1.45	4.61	10
364	1.45	2.37	30
364	1.45	1.62	50
364	1.45	1.13	70
564	1.22	3.63	30
564	1.22	1.94	50
564	1.22	1.08	70
564	1.22	0.95	90
779	0.44	2.34	40
779	0.44	1.27	70
779	0.44	0.38	110
779	0.44	0.19	150

Table 4.1: Measurable output errors for a variety of reduced model dimensions at steady state: 10×10 coarse mesh

dim(V_{on}) [N_c]	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size [N_r]
802	1.55	3.03	20
802	1.55	2.14	30
802	1.55	1.37	40
802	1.55	1.33	50
1163	0.51	1.60	50
1163	0.51	0.87	100
1163	0.51	0.52	150
1163	0.51	0.33	200
1524	0.36	0.93	100
1524	0.36	0.43	200
1524	0.36	0.26	300
1524	0.36	0.01	400

Table 4.2: Measurable output errors for a variety of reduced model dimensions at steady state: 20×20 coarse mesh

to the fine, coarse, and reduced dimension solution procedures. All computations were performed serially on a dual-core desktop workstation where each core uses two Intel Core i3 3.20 GHz processors, each with 4GB of RAM. We emphasize that these timing comparisons do not account for the parallelizable nature of the GMsFEM online space construction, however still offer some promising behavior. In particular, a parallel implementation of the method would likely render the online GMsFEM computations negligible. The results in Table 4.3 are devoted to a 10×10 coarse mesh and Table 4.4 contains a set of 20×20 coarse mesh results. The top row shows the required time to calculate a solution for the given fine-scale system of the size $N_f = 10201$ (100×100 fine element mesh). Next, the actual time of finding a solution by the online-offline balanced truncation approach for the same coarse space dimensions is provided ($N_c = 364, 564, 779$ for Table 4.3, and $N_c = 802, 1163, 1524$ for Table 4.4). The final times are obtained by adding three components: a time for the construction of the GMsFEM online space, a time for solving the Lyapunov equations, and a time for finding a solution for the reduced GMsFEM-BT system. These numerical results show a significant reduction in the computational cost for the smallest coarse dimension of $N_c = 364$. In particular, a computational time of 78.3 seconds for the fine-scale solve can be serially reduced to a time of 8.1 seconds for this reduced model. We reiterate that even for this online space dimension, we obtain steady state output errors that are less than 1.5% (cf. Table 4.1). Additionally, the larger-dimensional choices resulting from a 10×10 coarse mesh are still noticeably smaller than the fine-scale counterpart while offering declining errors. In contrast, the 20×20 results in Table 4.4 show an increase in computational expense mainly due to an increase in the Lyapunov system dimension (cf. (4.14)). Thus, we conclude that a more tractable approach in this setting is to accept slightly larger errors resulting from reduced systems with more reasonable timing.

System	Size	Time (*sec*)
Fine	10201	**78.3**
GMsFEM (Online)	364	7.2
BT	50	0.9
GMsFEM-BT	50	**8.1**
GMsFEM (Online)	564	12.2
BT	70	3.9
GMsFEM-BT	70	**16.1**
GMsFEM (Online)	779	14.4
BT	110	19.1
GMsFEM-BT	110	**33.5**

Table 4.3: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: 10×10 coarse mesh

System	Size	Time (*sec*)
Fine	10201	**78.3**
GMsFEM (Online)	802	21.4
BT	40	13.2
GMsFEM-BT	40	**34.6**
GMsFEM (Online)	1163	41.3
BT	150	45
GMsFEM-BT	150	**86.4**
GMsFEM (Online)	1524	65.2
BT	300	115.5
GMsFEM-BT	300	**180.7**

Table 4.4: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: 20×20 coarse mesh

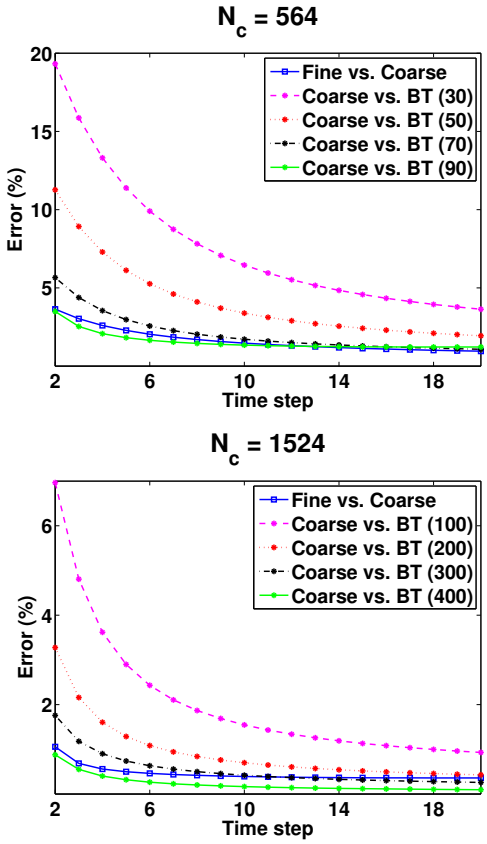


Figure 4.8: Comparison of relative errors for fine, coarse and reduced models

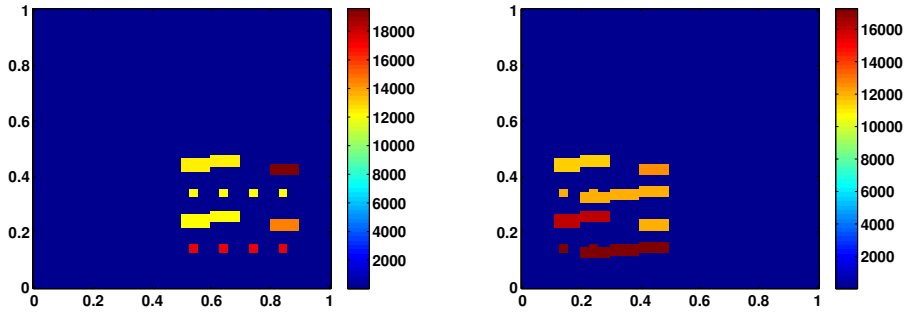
For a final set of examples, we consider a four-dimensional permeability field defined by $\kappa(x,\mu) = \mu_1\kappa_1(x) + \mu_2\kappa_2(x) + \mu_3\kappa_3(x) + \mu_4\kappa_4(x)$, where the values $(\mu_1,\mu_2,\mu_3,\mu_4)$ are assumed to be drawn from $[0,1]^4$. We introduce a slight abuse of notation by using κ_1 and κ_2 to denote different fields than the two-dimensional case before. See Figure 4.9 for an illustration of the individual fields for the current four-dimensional case. For all subsequent examples we use fixed values of $\mu_1 = 0.4$, $\mu_2 = 0.7$, $\mu_3 = 0.3$, and $\mu_4 = 0.8$. In order to generate the snapshot space in this case (cf. (4.3)) we use three equally spaced points in each dimension and keep $L_i = 12$ eigenfunctions so that $M_{\text{snap}} = 3^4 \times 12 = 972$.

In compliance with the two-dimensional problem, we offer an analogous set of

$\dim(V_{\text{on}})$ $[N_c]$	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size $[N_r]$
364	1.35	4.15	20
364	1.35	2.00	50
364	1.35	1.42	80
364	1.35	0.21	110
607	0.34	1.30	60
607	0.34	0.81	90
607	0.34	0.33	120
607	0.34	0.15	150
850	0.14	0.82	100
850	0.14	0.27	150
850	0.14	0.13	200
850	0.14	0.05	250

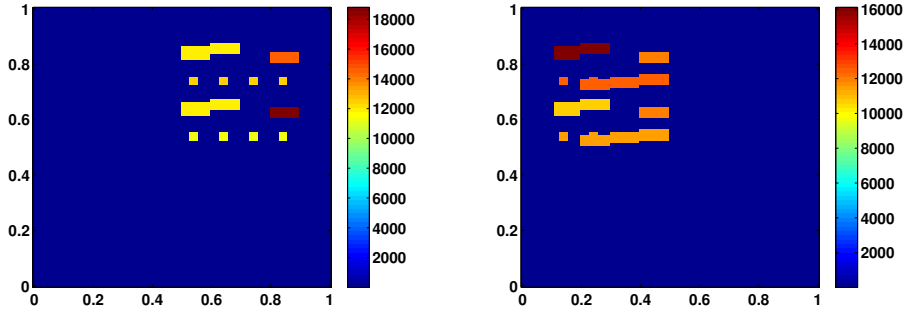
Table 4.5: Measurable output errors for a variety of reduced model dimensions at steady state for a four-dimensional problem

examples. The time step for these examples is chosen to be $\Delta t = 1\times10^{-4}$ and we run the simulations for a total number of 40 time steps. For this set of results we limit ourselves to a 10×10 coarse mesh configuration with online space dimensions of $N_c = 364, 607$, and 850. We offer a set of output illustrations corresponding to the fine, coarse, and reduced models in Figures 4.10 and 4.11. As before, we see that the respective outputs at a variety of fixed times are nearly indistinguishable. Furthermore, we see from Table 4.5 and Figure 4.12 that we are able to determine a suitable reduced dimension based on the corresponding output errors. And as expected, Table 4.6 shows that this range of online space and reduced dimensions offers serial simulation times that are smaller than the fine-scale solve. As before, the $N_c = 364$ case yields the best timing while still producing errors that are less than 1.5%.



(a) $\kappa_1(x)$

(b) $\kappa_2(x)$



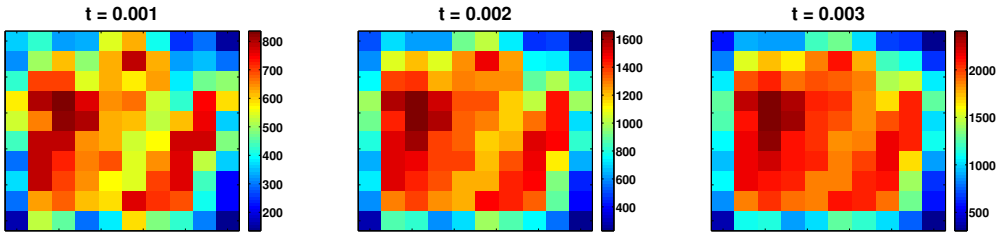
(c) $\kappa_3(x)$

(d) $\kappa_4(x)$

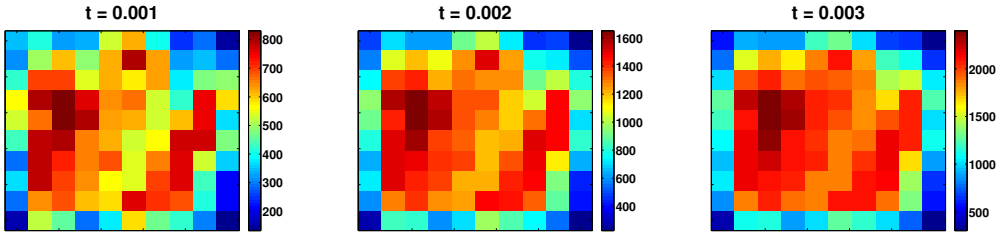
Figure 4.9: Permeability coefficients for a four-dimensional parameter-dependent problem

4.4 Conclusions

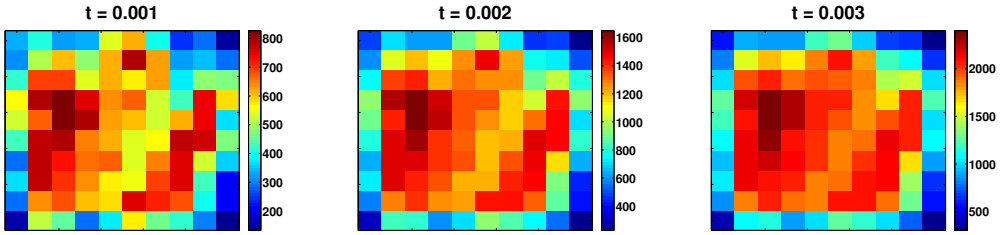
In this section we propose a local-global model reduction technique for the accurate calculation of output states of a parameter-dependent, single-phase flow model. We treat the problem using a combined approach in which GMsFEM is used as an effective local model reduction tool, and BT is used as an effective global model reduction tool. GMsFEM hinges on the independent construction of a set of multiscale basis functions that are used to form a lower-dimensional coarse solution space. The localized multiscale basis function computations are cast in the framework of an offline-online procedure in which a respective set of eigenvalue problems are used



(a) Fine ($N_f = 10201$)



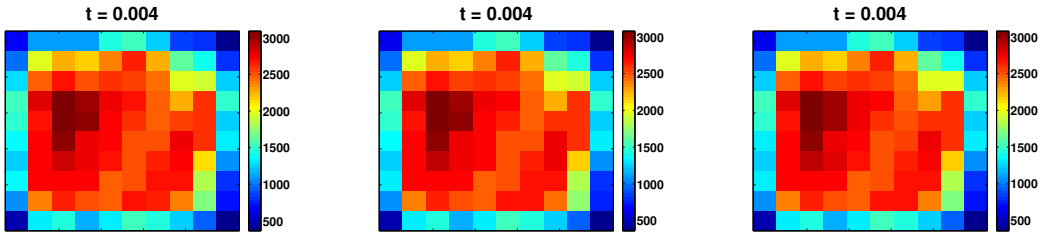
(b) GMsFEM ($N_c = 607$)



(c) BT ($N_r = 120$)

Figure 4.10: Time variant output comparisons for a four-dimensional problem

to capture the underlying behavior of the system. Since the computation of the snapshots and offline space accounts for a one-time preprocessing step, the online coarse space may be cheaply constructed for a fixed input state. Given the locally reduced coarse system, we then apply balanced truncation (BT) to further reduce the size of the system while maintaining a suitable level of output state accuracy. BT is a global model reduction technique in which the input-output mapping is approximated through the spectral construction of a reduced-order model, and re-



(a) Fine ($N_f = 10201$) (b) GMsFEM ($N_c = 607$) (c) BT ($N_r = 120$)

Figure 4.11: Steady state output comparisons for a four-dimensional problem

System	Size	Time (*sec*)
Fine	10201	**78.3**
GMsFEM (Online)	364	8.3
BT	80	1.3
GMsFEM-BT	80	**9.6**
GMsFEM (Online)	607	16.9
BT	120	6.6
GMsFEM-BT	120	**23.5**
GMsFEM (Online)	850	21.3
BT	200	22.4
GMsFEM-BT	200	**43.7**

Table 4.6: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms for a four-dimensional problem

quires the solution of a set of Lyapunov equations. The efficiency of the proposed method depends on the size of the online coarse space, due to the fact that the Lyapunov equations represent a main source of computational expense. However, the GMsFEM-BT approach is shown to be quite flexible with respect to the online space and reduced dimensions, and may be readily modified in order to ensure that the resulting output errors are comparable. In turn, the associated numerical examples suggest that the proposed method is a suitable approach for closely approximating the output state of the parameter-dependent, single-phase model equation.

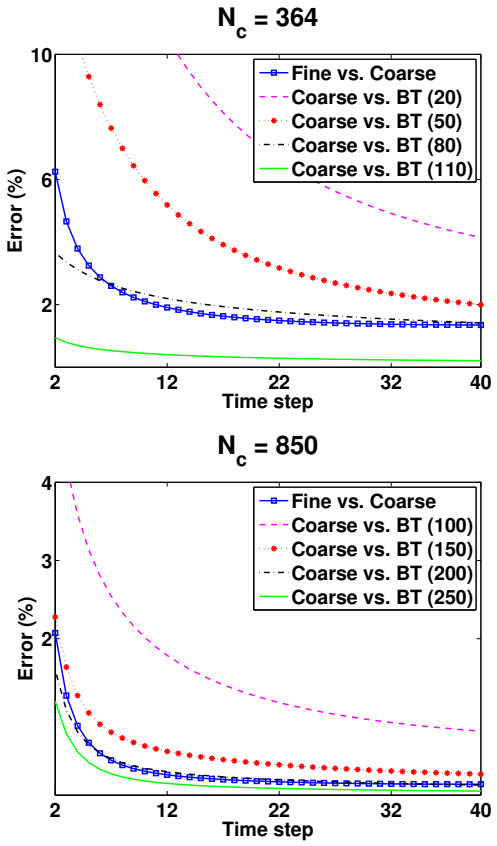


Figure 4.12: Comparison of relative errors for fine, coarse and reduced models for a four-dimensional problem

5. COARSE-GRID MODEL REDUCTION WITH BALANCED TRUNCATION FOR FLOWS IN TIME-VARYING HETEROGENEOUS POROUS MEDIA

5.1 Local-global model reduction for a time-varying system

In this section, we propose a local-global model reduction technique that combines GMsFEM method with BT for a time-varying system. As in Section 4, we use the GMsFEM as a local model reduction technique. But as a global model reduction technique, a special computationally efficient form of a time-varying BT is used. We offer a brief outline of the general procedure below.

We describe a time-variant model as following. Find a solution $p(x,t)$ of the system

$$\frac{\partial p}{\partial t} = \operatorname{div}\big(\kappa(x;t)\nabla p\big) + f(u;t) \quad \text{in } D, \tag{5.1a}$$

$$p(t) = g(t) \quad \text{on } \partial D, \tag{5.1b}$$

where the input (or control) $u \in \mathbb{R}^m$ is given, D is a domain in $\mathbb{R}^2$, the right hand side $f(u;t)$ is square integrable and depends linearly on u, g denotes the boundary condition. The above model is solved along with a specified initial condition $p(x,0) = p_0(x)$. We also suppose that an output is given as $q(t) = C(t)p(t)$.

For completeness, we rewrite the Equations (4.13) on a coarse grid (in a discrete setup) for a time-variant case. We denote the transformation matrix which was obtained by GMsFEM as $R \in \mathbb{R}^{N_f \times N_c}$ and the coarse-scale equation can be written as

$$R^T M(t) R (p_c)_t = -R^T A(t) R p_c + R^T B(t) u \tag{5.2}$$

or

$$(p_c)_t = -A^c(t)p_c + B^c(t)u, \tag{5.3}$$

where $A^c(t) = (R^T M(t)R)^{-1}R^T A(t)R$, $B^c(t) = (R^T M(t)R)^{-1}R^T B(t)$, and p_c denotes the coarse scale solution. Similarly, an approximation of the measurable output may be written as $q^c(t) = C(t)Rp_c(t) = C^c(t)p_c(t)$.

The next step is to apply BT for the time-varying system (5.3). We use the special computationally efficient form of a time-variant BT that was introduced in [53, 54]. The detailed algorithm is formulated in the next Section. In order to find a suitable approximation for the input-output map, we consider the transfer function $G(t)$. The approximation of $G(t)$ can be accomplished by finding projection matrices, namely $U(t)$ and $V(t)$, such that Equation (5.3) can be projected onto a much smaller subspace. To this end, the sequence of reduced-order model $G_r(t)$ can be computed by setting

$$A^r(t) = U(t)A^c(t)V(t); \quad B^r(t) = U(t)B^c; \quad C^r(t) = C^c V(t). \tag{5.4}$$

We remind the fact that the number of truncated states can be chosen according to error estimates that are expressed through Hankel singular values (HSV's).

Finally, a globally reduced system can be written as following

$$(p_r)_t = -A^r(t)p_r + B^r(t)u, \tag{5.5a}$$

$$q^r = C^r(t)p_r, \tag{5.5b}$$

such that $A^r(t)$ has size $N_r \times N_r$, where $N_r \ll N_c \ll N_f$. The performance of the proposed method will be assessed by comparing the measurable output quantities

$q(t)$, $q^c(t)$, and $q^r(t)$ resulting from Equations (5.1), (5.3), and (5.5), respectively.

As an approximation for the suggested reduced time-variant system (5.5), the discrete-time formulation is given below

$$p_r(t_{k+1}) = \left(I + \Delta t A^r(t_k)\right)^{-1} p_r(t_k) + \left(I + \Delta t A^r(t_k)\right)^{-1} \Delta t B^r(t_k) u, \tag{5.6a}$$

$$q^r(t_{k+1}) = C^r(t_k) p_r(t_{k+1}). \tag{5.6b}$$

5.1.1 Implementation of balanced truncation approach for time-varying systems

Sandberg developed the algorithm for a time-variant BT in [53] and the following theorem was proved.

Theorem. *In a case when G is stable and has initial realization at rest, there exists a reduced-order model $\hat{G} : \|G - \hat{G}\|_2 < \gamma$ with realization of order $\hat{n}(k) \leq n(k)$ if and only if there exist bounded semi-definite solutions $P(t)$ - controllability Gramian, $Q(t)$ - reachability Gramian to*

$$A^c(k) P(k) (A^c(k))^T - P(k+1) + B^c(k) (B^c)^T(k) < 0, \tag{5.7}$$

$$(A(k)^c)^T Q(k+1) A^c(k) - Q(k) + (C^c(k))^T C^c(k) < 0, \tag{5.8}$$

and $\lambda_{min}(P^c(k) \cdot Q^c(k)) = \gamma$ with multiplicity $\hat{n}(k) - n(k)$ for all k. Here $P(k)$ and $Q(k)$ are discrete time instants of $P(t)$ and $Q(t)$.

Optimal Gramians can be calculated as following:

$$A^c(k) P(k) (A^c(k))^T + B^c(k) (B^c(k))^T = P(k+1), \tag{5.9a}$$

$$P(0) = 0, \tag{5.9b}$$

$$(A^c(k))^T Q(k+1)A^c(k) + (C^c(k))^T C^c(k) = Q(k), \tag{5.10a}$$

$$Q(N+1) = 0. \tag{5.10b}$$

Time-varying BT algorithm. To find an output q given coarse matrices A^c, B^c, C^c and M^c:

INPUT coarse matrices A^c, B^c, C^c and M^c; a size k of the reduced system; a size N of the coarse system; a total number Nt of time steps; a time step dt

OUTPUT approximated output q of a reduced system

Step 1 Set $i = 1, P - zero\ \ N \times N\ \ matrix$ (the initial controllability Gramian), $Q - zero\ \ N \times N\ \ matrix$ (the initial observability Gramian), $p_r - zero\ \ k \times 1\ \ column$ (the solution of the reduced system).

Step 2 For $i \leq Nt$ do Steps 3-12.

Step 3 Update A^c, B^c, C^c, M^c.

Step 4 $Bs = (M^c + dt * A^c) \setminus B^c$; $S = (M^c + dt * A^c) \setminus M^c$. (Calculate auxiliary terms.)

Step 5 $P = S * P * S' + Bs * Bs'$. (Update the controllability Gramian)

Step 6 Set $j = 1$.

Step 7 For $j \leq Nt - i + 1$ do Step 8.

Step 8 $Q = S' * Q * S + Cs' * Cs$. (Update the observability Gramian)

Step 9 Find T - the matrix of the right eigenvectors of the matrix $P * Q$, where the eigenvalues of $P * Q$ are sorted in a descending order.

Step 10 $A_r = T' * A^c * T; B_r = T' * B^c; C_r = C^c * T; F_r = T' * F^c; M_r = T' * M^c * T$. Next, cut all matrices A_r, B_r, C_r, F_r, C_r by eliminating $(k+1)th, ..., Nth$

columns and rows. (Calculate reduced matrices.)

Step 11 Solve $(p_r)_t = -A^r(t)p_r + B^r(t)u$.

Step 12 Calculate $q = C_r * p_r$. (Calculate the final output.)

Computational advantage.

First, note that the original method for computing Gramians involves SVD calculations (using Cholesky factors) and it has computational cost $O(n^3)$. While the new method significantly reduce the number of calculations due to its structure.

Second, for varying right hand side F^c the Gramians P and Q can be reused. Therefore, P and Q can be calculated once and at each time step we need to calculate the output.

The numerical illustration is proposed in the next section.

5.2 Numerical results

In order to study the implementation of the proposed balanced truncation approach for time-varying systems we consider three numerical cases. The first choice of permeability $\kappa(x,t) = \kappa_0(x)\cdot\exp(t)$ is referred as a separable case, where the initial permeability field $\kappa_0(x) = \kappa(x,0)$ has the structure shown in Figure 5.1. Notice that this permeability field $\kappa_0(x)$ exhibits a high-contrast structure, where the contrast is taken to be $\kappa_{\max}/\kappa_{\min} \approx 10^5 - 10^6$. The second and the third cases are non-separable examples. We use Nt as a total number of time steps.

The first non-separable case has the following permeability at a time step s:

$$\kappa_1(x,t) = \exp(c\cdot\frac{s}{Nt}),\ \text{where}\ \begin{cases} c = 15 & \text{for a high-contrast region,} \\ c = 1 & \text{otherwise.} \end{cases}$$

The second non-separable case has a permeability at a time step s as following:

$$\kappa_2(x,t) = \kappa_0(x) \cdot \begin{cases} \exp(1 + 4.5 \cdot \frac{s}{Nt}) & \text{for a high-contrast region,} \\ 1 & \text{otherwise.} \end{cases}$$

The separable permeability coefficient is illustrated in Figure 5.2 and non-separable cases are represented in Figures 5.3-5.4 for some time instants.

A domain, boundary and initial conditions, a right hand side and an output are chosen as in the first set of examples in Section 4. We consider the two-dimensional unit domain $D = [0,1]\times[0,1]$, we assume a homogeneous boundary condition $p = 0$ and as the initial condition we use $p_0(x) = 0$. We let $f(u)$ be a piecewise constant forcing parameter on a coarse grid, such that the j^{th} column of B is defined as

$$B_{(:,j)} = \begin{cases} 1 & \text{if } x \in K_j, \\ 0 & \text{otherwise,} \end{cases}$$

where we use K_j to denote the j^{th} coarse element of the domain. In essence, the matrix B is a mapping of a coarse-dimensional parameter u to the fine grid. For all examples u is taken to be a multi-dimensional random parameter subordinated to the coarse grid such that each component is uniformly distributed on $[0,1]$. Additionally, we assume that the output selection matrix is defined by $C = B^T$. For simplicity of computations, the output selection matrix and the right hand side don't depend on time. And finally, we discretize D into a 100×100 fine element mesh. In all cases 10×10 coarse mesh is considered with online space of dimension $N_c = 564$.

In order to justify GMsFEM, we list the illustration of the pressure fields for the fine-scale system (see Figure 5.5) and for the coarse-scale system (see Figure 5.6) for a non-separable case 1.

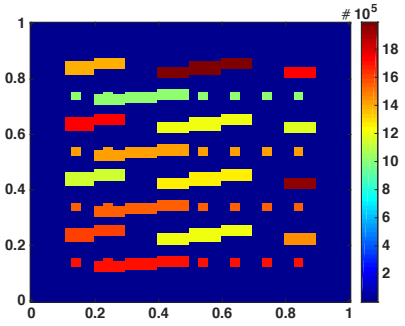


Figure 5.1: Initial high-contrast permeability coefficient κ_0

The rest of numerical results are devoted to comparing the outputs from the fine-scale system and the coarse-scale system. We take a discrete, relative l^2-norm scaled by the coarse output norm. At each time step we wish to choose the dimension such that the fine to coarse errors $\|q^{f,i} - q^{c,i}\|_2$ and the coarse to reduced errors $\|q^{c,i} - q^{r,i}\|_2$ are comparable (where i is the time instant).

Table 5.1 compares the errors between the outputs of the coarse-scale system and of the reduced system using a variety of time steps and a variety of reduced system size for a separable case. Tables 5.2 and 5.3 contain numerical results for a non-separable case 1 and a non-separable case 2, respectively. We test BT reduced dimensions of $N_r = 5, 10, 20$ and time steps of $Nt = 5, 10, 20, 50$ for the same time interval $dt = 0.01$. We see that the fine to coarse system error decreases as the number of time steps increases and the coarse to reduced system error decreases as the BT reduced dimensions increases. We remind that the size of coarse space is 564 and the 'optimal' size of reduced space is 20, while the fine-scale system is of size

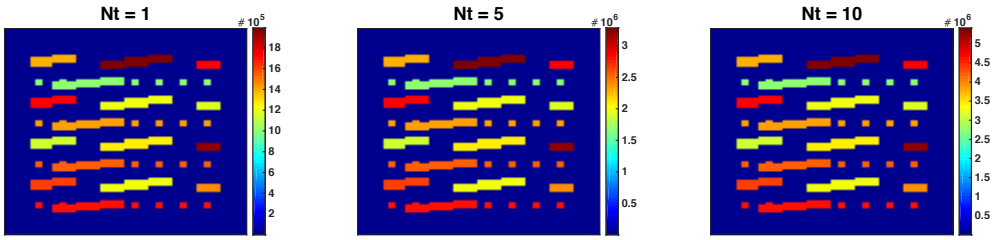


Figure 5.2: High-contrast permeability coefficients for a separable case, 10 time steps

$N_f = 10201$.

The next sets of numerical results verify the computational efficiency of the proposed method. Tables 5.4, 5.5, 5.6, 5.7, 5.8 and 5.9 contain computational times corresponding to the fine, coarse, and reduced dimension solution procedures. All calculations were completed serially on a dual-core desktop workstation where each core uses two Intel Core i3 2.6 GHz processors, each with 8GB of RAM. We note that a parallel implementation of the suggested method would likely provide the online GMsFEM computations negligible, but still computational savings are significant. Tables 5.4, 5.5 are devoted to a separable case; Tables 5.6, 5.7 are devoted to a non-separable case 1 and Tables 5.8, 5.9 are devoted to a non-separable case 2. Since all above tables show similar numerical observations, we concentrate on Table 5.4 for a separable case with 10 time steps. The first row shows the required time to calculate a solution for the given fine-scale system. Next row is the actual time of finding a solution by the online GMsFEM approach. The last time is obtained by adding three components: a time for the construction of the GMsFEM online

Time steps [Nt]	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size [N_r]
5	1.54	4.37	5
5	1.54	2.33	10
5	1.54	1.12	20
10	1.52	4.45	5
10	1.52	2.34	10
10	1.52	1.12	20
20	1.52	4.49	5
20	1.52	2.36	10
20	1.52	1.12	20
50	1.51	4.52	5
50	1.51	2.37	10
50	1.51	1.12	20

Table 5.1: Measurable output errors for a variety of reduced model dimensions at steady state: 10×10 coarse mesh ($\dim(V_{\text{on}})$ $[N_c] = 564$, a separable case)

Time steps [Nt]	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size [N_r]
5	1.82	4.60	5
5	1.82	2.27	10
5	1.82	1.12	20
10	2.09	4.89	5
10	2.09	2.66	10
10	2.09	1.23	20
20	2.27	8.87	5
20	2.27	2.97	10
20	2.27	1.38	20
50	2.39	11.31	5
50	2.39	3.24	10
50	2.39	1.47	20

Table 5.2: Measurable output errors for a variety of reduced model dimensions at steady state: 10×10 coarse mesh ($\dim(V_{\text{on}})$ $[N_c] = 564$, a non-separable case 1)

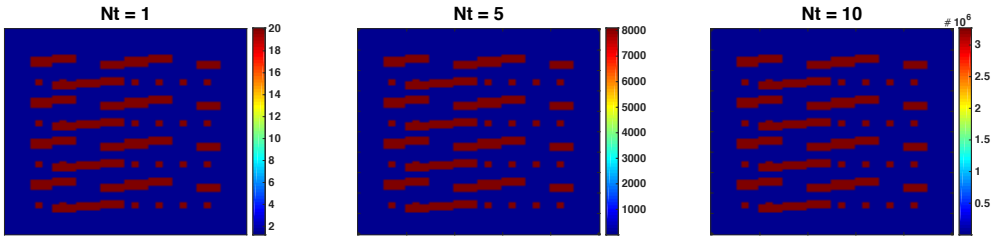


Figure 5.3: High-contrast permeability coefficients for a non-separable case 1, 10 time steps

space, a time for solving the Lyapunov equations, and a time for finding a solution for the reduced GMsFEM-BT system. This numerical example shows a significant reduction in the computational cost. In particular, a computational time of 563.23 seconds for the fine-scale solve can be serially reduced to a time of 7.09 seconds for the reduced model. Figures 5.7, 5.8 and 5.9 give an illustration of fine, coarse and reduced outputs for a separable case, a non-separable case 1 and non-separable case 2, respectively.

Time steps $[Nt]$	Fine-Coarse Error (%)	Coarse-BT Error (%)	BT Size $[N_r]$
5	1.34	6.15	5
5	1.34	2.98	10
5	1.34	1.29	20
10	1.34	5.97	5
10	1.34	2.90	10
10	1.34	1.25	20
20	1.33	5.88	5
20	1.33	2.86	10
20	1.33	1.24	20
50	1.33	5.82	5
50	1.33	2.83	10
50	1.33	1.23	20

Table 5.3: Measurable output errors for a variety of reduced model dimensions at steady state: 10×10 coarse mesh ($\dim(V_{\text{on}})$ $[N_c] = 564$, a non-separable case 2)

System	Size	Time (*sec*)
Fine	10201	**563.23**
GMsFEM (Online)	564	3.71
BT	100	3.38
GMsFEM-BT	100	**7.09**

Table 5.4: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: a separable case, 10 time steps

System	Size	Time (*sec*)
Fine	10201	**3160.72**
GMsFEM (Online)	564	21.65
BT	100	28.71
GMsFEM-BT	100	**50.36**

Table 5.5: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: a separable case, 50 time steps

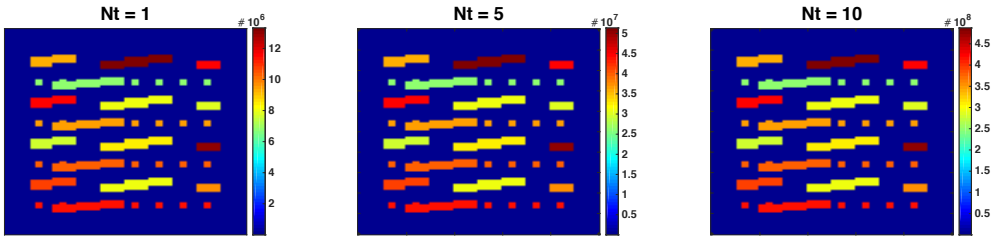


Figure 5.4: High-contrast permeability coefficients for a non-separable case 2, 10 time steps

System	Size	Time (*sec*)
Fine	10201	**553.04**
GMsFEM (Online)	564	5.04
BT	20	3.41
GMsFEM-BT	20	**8.59**

Table 5.6: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: a non-separable case 1, 10 time steps

5.3 Conclusions

In this section we consider a time-varying system that represents a typical source of computational expense. A new robust and computationally efficient method is suggested, and it is based on combing local and global model reduction techniques. As a local model reduction, we use the GMsFEM and as a global model reduction a special form of time-varying BT. We would to remark that such form of time-varying BT doesn't require calculations of "expensive" Lyapunov equations in each time interval.

System	Size	Time (*sec*)
Fine	10201	**2708.89**
GMsFEM (Online)	564	23.33
BT	20	27.75
GMsFEM-BT	20	**63.01**

Table 5.7: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: 10×10 coarse mesh, a non-separable case 1, 50 time steps

System	Size	Time (*sec*)
Fine	10201	**566.67**
GMsFEM (Online)	564	4.48
BT	20	2.87
GMsFEM-BT	20	**7.35**

Table 5.8: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: a non-separable case 2, 10 time steps

System	Size	Time (*sec*)
Fine	10201	**2885.71**
GMsFEM (Online)	564	21.36
BT	20	28.05
GMsFEM-BT	20	**49.41**

Table 5.9: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms: 10×10 coarse mesh, a non-separable case 2, 50 time steps

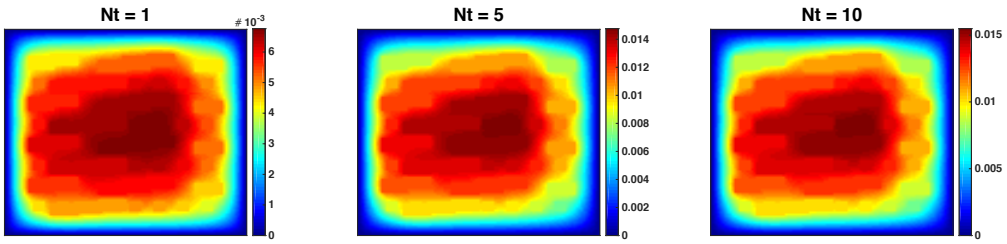


Figure 5.5: Pressure fields for a non-separable case 1, 10 time steps, a fine-scale system

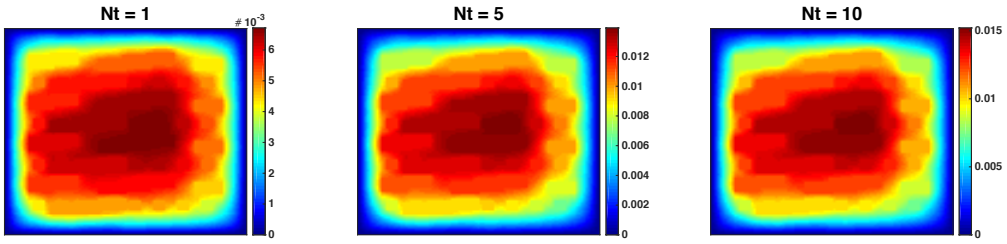
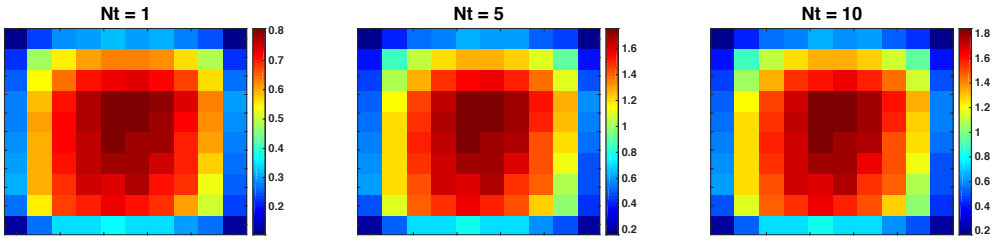
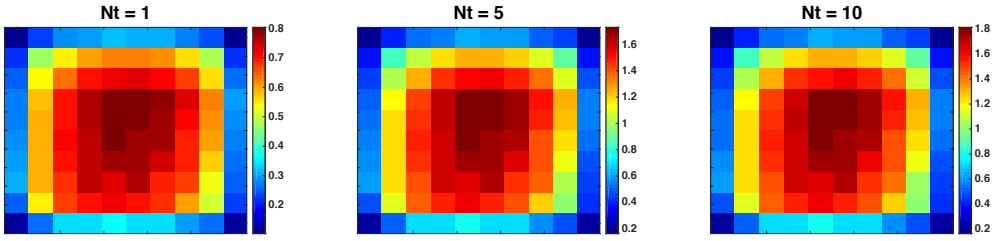


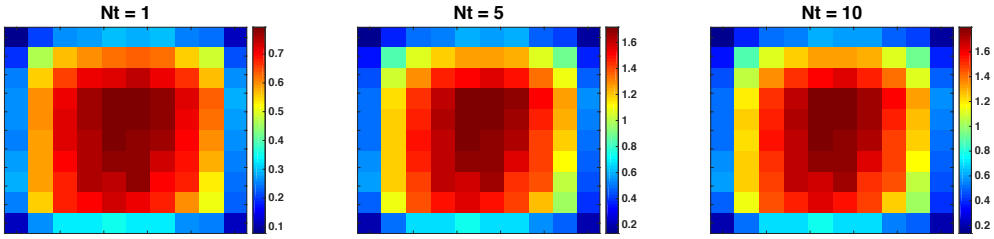
Figure 5.6: Pressure fields for a non-separable case 1, 10 time steps, a coarse-scale system



(a) Fine ($N_f = 10201$)

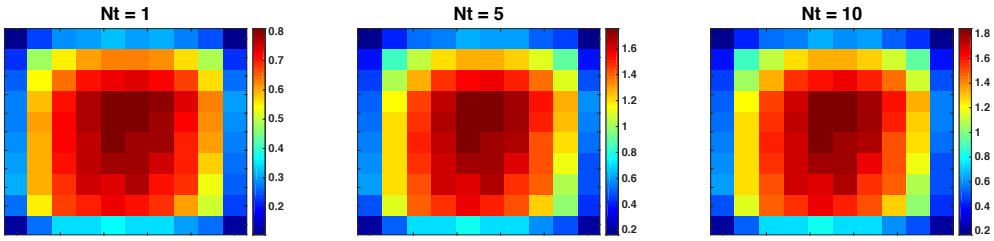


(b) GMsFEM ($N_c = 564$)

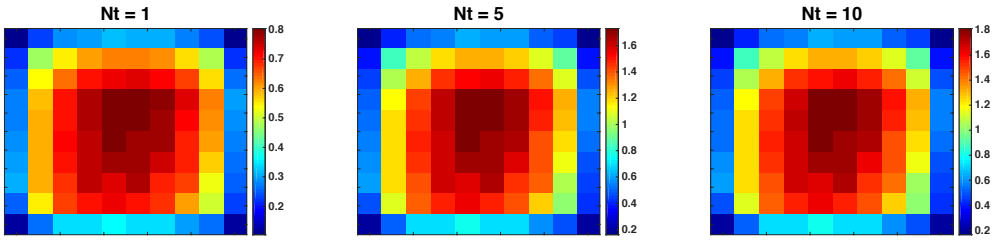


(c) BT ($N_r = 20$)

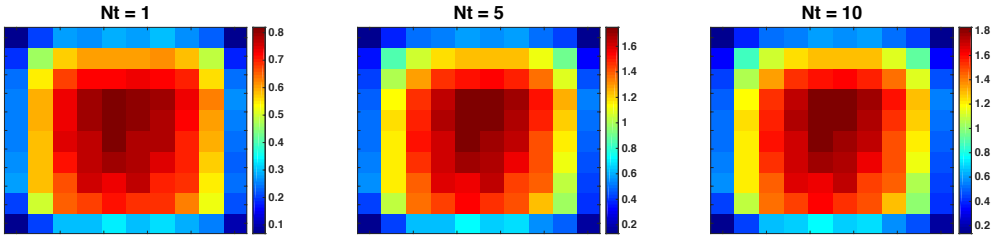
Figure 5.7: Time variant output comparisons for a separable case, 10 time steps



(a) Fine ($N_f = 10201$)

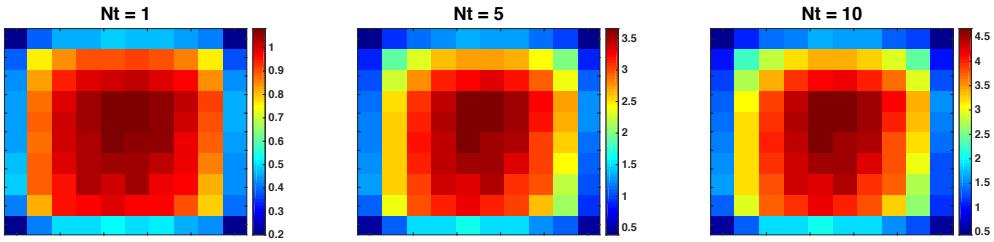


(b) GMsFEM ($N_c = 564$)

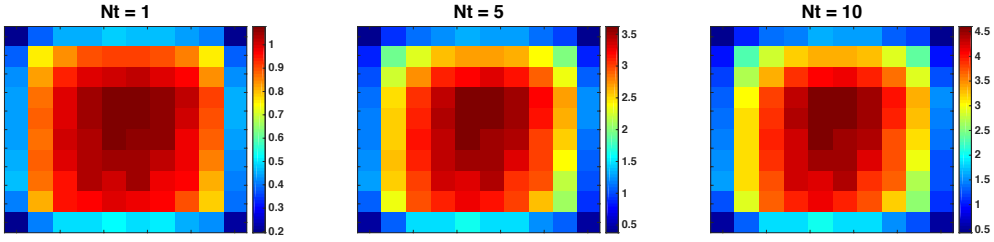


(c) BT ($N_r = 20$)

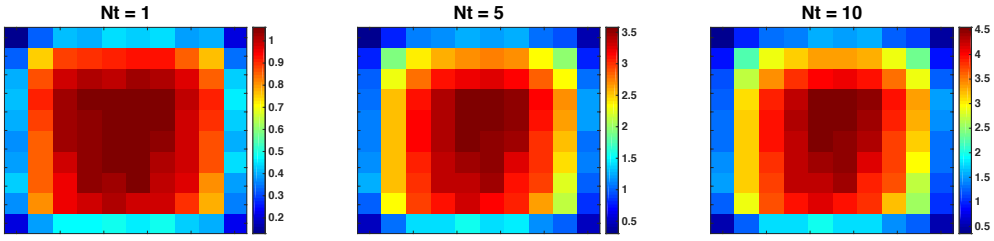
Figure 5.8: Time variant output comparisons for a non-separable case 1, 10 time steps



(a) Fine ($N_f = 10201$)



(b) GMsFEM ($N_c = 564$)



(c) BT ($N_r = 20$)

Figure 5.9: Time variant output comparisons for a non-separable case 2, 10 time steps

6. BALANCED TRUNCATION METHOD FOR COUPLED FLOW AND TRANSPORT

6.1 Local-global model reduction

In this section we consider a new local-global model reduction technique based on BT approach for solving coupled flow and transport equations. A suggested method combines the GMsFEM with BT approach for a mixed formulation. The mixed formulation provides a mass conservative velocity field and is important for subsurface applications. In this section, we use coarse-grid models for both flow and transport equations. The mixed formulation used in the section is presented in [16].

6.2 Generalized Multiscale Finite Element Method for coupled flow and transport

6.2.1 Multiscale solution space V_H^v

First, we present the construction for the snapshot space V_{s}^v. As in [16], we define the local problem as following. For each $E_i \in \mathcal{E}^H$, on its neighboring coarse cell K, find $p_j^i \in Q_h$, $\psi_{v,j}^i \in V_h^v$, such that

$$\begin{aligned} \psi_{v,j}^i + \kappa \nabla p_j^i &= 0 \qquad \text{in } K, \\ \nabla \cdot \psi_{v,j}^i &= \alpha_j^i \qquad \text{in } K, \\ \psi_{v,j}^i \cdot n &= 0 \qquad \text{on } \partial K \backslash E_i, \\ \psi_{v,j}^i \cdot n &= \delta_j^i \qquad \text{on } E_i, \end{aligned} \tag{6.1}$$

where j varies over all fine-grid edges on E_i, α_j^i is a constant on K, n is a fixed unit-normal vector on ∂K, and $\delta_j^i(x)$ is defined on E_i with respect to the fine-grid

edges such that

$$\delta_j^i = \begin{cases} 1, & \text{on } e_j^i, \\ 0, & \text{on the other fine-grid edges.} \end{cases}$$

Here, Q_h is piecewise constant basis functions.

Next we solve the local problems with respect to E_i and a local snapshot space can be obtained as $V_{\mathrm{s}}^{v,i} = \mathrm{span}\{\psi_{v,j}^i : 1 \le j \le J_i\}$, where J_i is the number of the fine-grid edges on E_i. The snapshot space is constructed as following

$$V_{\mathrm{s}}^v = \bigcup_{E_i} V_{\mathrm{s}}^{v,i} = \mathrm{span}\{\psi_{v,j}^i : 1 \le j \le J_i, 1 \le i \le N_e\}. \tag{6.2}$$

After renumerating the snapshot functions by a single index, we obtain the coarse-to-fine snapshot matrix $R_{\mathrm{s}}^v = \left[\psi_1^v, \psi_2^v, \cdots, \psi_{L_{\mathrm{s}}}^v\right]$, where $L_{\mathrm{s}} = \sum_{i=1}^{N_e} J_i$.

Next, we construct the offline space V_{off}^v. We define the eigenvalue problem in $V_{\mathrm{s}}^{v,i}$ as following:

$$A_v^{\mathrm{off}} \Psi_k^v = \lambda_k M_v^{\mathrm{off}} \Psi_k^v, \tag{6.3}$$

where $A_v^{\mathrm{off}} = \int_{E_i} (\psi_{v,s}^i \cdot n)(\psi_{v,t}^i \cdot n)$, and $M_v^{\mathrm{off}} = \int_{\omega_i} \frac{1}{\kappa} \psi_{v,s}^i \cdot \psi_{v,t}^i$.

In order to compute the offline space, we need to select L_{off}^i eigenvectors of (6.3) corresponding to the smallest L_{off}^i eigenvalues, and then calculate the element $\phi_{v,k}^i = \sum_{j=1}^{J_i} \Psi_{k,j}^v \psi_{v,j}^i$ for $k = 1, 2, \cdots, L_{\mathrm{off}}^i$. Finally, the local offline space is constructed as $V_{\mathrm{off}}^{v,i} = \mathrm{span}\{\phi_{v,k}^i : 1 \le k \le L_{\mathrm{off}}^i\}$ and the offline space is

$$V_{\mathrm{off}}^v = \bigcup_{E_i} V_{\mathrm{off}}^{v,i} = \mathrm{span}\{\phi_{v,k}^i : 1 \le k \le L_{\mathrm{off}}^i,\ 1 \le i \le N_e\}. \tag{6.4}$$

After renumerating the basis functions by a single index, the offline matrix can be obtained as $R_{\mathrm{off}}^v = \left[\phi_1^v, \phi_2^v, \cdots, \phi_{L_{\mathrm{off}}}^v\right]$, where $L_{\mathrm{off}} = \sum_{i=1}^{N_e} L_{\mathrm{off}}^i$.

We define the multiscale solution space as $V_H^v = V_{\text{off}}^v$. In order to approximate v_H, we need to solve the following problem. Find $p_H \in Q_H$, $v_H \in V_H^v$ such that

$$\begin{aligned} \int_D \frac{1}{\kappa} v_H \cdot \tilde{v} &= \int_D (\nabla \cdot \tilde{v}) p_H \qquad \text{for } \forall \tilde{v} \in V_H^v, \\ \int_D (\nabla \cdot v_H) \tilde{w} &= 0 \qquad \text{for } \forall \tilde{w} \in Q_H, \end{aligned} \tag{6.5}$$

where the boundary condition $p_H = g$.

6.2.2 Multiscale solution space V_H^w

The process is similar as for the multiscale solution space V_H^v ([16]). A local problem corresponding to each coarse-grid edge $E_i \in \mathcal{E}^H$ is defined as following. Find $c_j^i \in Q_h$, $\psi_{w,j}^i \in V_h^w$, such that

$$\begin{aligned} \psi_{w,j}^i + G \nabla c_j^i - v_H c_j^i &= 0 \qquad \text{in } K, \\ \nabla \cdot \psi_{w,j}^i &= \alpha_j^i \qquad \text{in } K, \\ \psi_{w,j}^i \cdot n &= 0 \qquad \text{on } \partial K \backslash E_i, \\ \psi_{w,j}^i \cdot n &= \delta_j^i \qquad \text{on } E_i. \end{aligned} \tag{6.6}$$

The snapshot space is constructed as

$$V_{\text{s}}^w = \bigcup_{E_i} V_{\text{s}}^{w,i} = \text{span}\{\psi_{w,j}^i : 1 \le j \le J_i,\ 1 \le i \le N_e\} \tag{6.7}$$

and the coarse-to-fine snapshot matrix as $R_{\text{s}}^w = \left[\psi_1^w, \psi_2^w, \cdots, \psi_{L_{\text{s}}}^w\right]$.

In order to find a snapshot space $W_{\rm s}^w$, we solve the adjoint problem of (6.6)

$$\begin{aligned} z_{w,j}^i - G\nabla p_j^i &= 0 \qquad \text{in } K, \\ \nabla \cdot z_{w,j}^i + \frac{1}{G} v_H \cdot z_{w,j}^i &= \alpha_j^i \qquad \text{in } K, \\ z_{w,j}^i \cdot n &= 0 \qquad \text{on } K \backslash E_i, \\ z_{w,j}^i \cdot n &= \delta_j^i \qquad \text{on } E_i. \end{aligned} \tag{6.8}$$

The snapshot space $W_{\rm s}^w$ is formed by the snapshot functions $z_{w,j}^i$ for all $E_i \in \mathcal{T}^H$.

As before, the local offline space $V_{\rm off}^{w,i}$ for w_H can be obtained by solving the following eigenvalue problem in $V_{\rm s}^{w,i}$:

$$A_w^{\rm off} \Psi_k^w = \lambda_k M_w^{\rm off} \Psi_k^w, \tag{6.9}$$

where $A_w^{\rm off} = \int_{E_i} (\psi_{w,s}^i \cdot n)(\psi_{w,t}^i \cdot n)$, $M_w^{\rm off} = \int_{\omega_i} \tilde{G} \psi_{w,s}^i \cdot \psi_{w,t}^i$, $\tilde{G} = \frac{1+|v_H|}{G}$. Then the offline space is $V_{\rm off}^w = \bigcup_{E_i} V_{\rm off}^{w,i} = {\rm span}\{\phi_{w,k}^i : 1 \le k \le L_{\rm off}^i, 1 \le i \le N_e\}$ and the offline matrix is $R_{\rm off}^w = \left[\phi_1^w, \phi_2^q, \cdots, \phi_{L_{\rm off}}^w\right]$.

We constructed $V_H^w = V_{\rm off}^w$ and the testing space

$$W_H^w = \{z \in W_{\rm s}^w : (z-w) \cdot n|_E = 0, \ \text{ for all } E \in \mathcal{E}^H, \ \text{ for some } \ w \in V_H^w\}, \tag{6.10}$$

where n is the unit-normal vector of the coarse-grid edge E. Finally, the GMsFEM solution c_H, w_H can be computed by (2.21).

6.3 Novel model reduction approach based on Balanced Truncation

We write the discretized formulation of (2.17) on a fine grid as following:

$$\left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} + \frac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot \begin{bmatrix} w(t_{n+1}) \\ c(t_{n+1}) \end{bmatrix} =$$

$$= \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} - \frac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot \begin{bmatrix} w(t_n) \\ c(t_n) \end{bmatrix} + \triangle t \cdot \begin{bmatrix} 0 \\ f(t_{n+1}) \end{bmatrix}.$$

After performing GMsFEM, we denote a trial and a testing snapshot matrices as $R_{trial} = \begin{bmatrix} R_w & 0 \\ 0 & R_c^{(1)} \end{bmatrix}$ and as $R_{test} = \begin{bmatrix} R_w & 0 \\ 0 & R_c^{(2)} \end{bmatrix}$, respectively. Therefore, the discretized formulation on a coarse grid can be written as

$$R_{test}^T \cdot \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} + \frac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot R_{trial} \cdot \begin{bmatrix} w(t_{n+1}) \\ c(t_{n+1}) \end{bmatrix} =$$

$$R_{test}^T \cdot \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} - \frac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot R_{trial} \cdot \begin{bmatrix} w(t_n) \\ c(t_n) \end{bmatrix} +$$

$$+ \triangle t \cdot R_{test}^T \cdot \begin{bmatrix} 0 \\ f(t_{n+1}) \end{bmatrix},$$

and the coarse system can be written in a compact form for BT as

$$P_H^{n+1} := \begin{bmatrix} w_H(t_{n+1}) \\ c_H(t_{n+1}) \end{bmatrix} = A_H \cdot \begin{bmatrix} w_H(t_n) \\ c_H(t_n) \end{bmatrix} + B_H \cdot \begin{bmatrix} 0 \\ f(t_{n+1}) \end{bmatrix},$$

where

$$A_H = \left(R_{test}^T \cdot \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} + \tfrac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot R_{trial} \right) \cdot R_{trial} \Big)^{-1} \cdot$$

$$\cdot R_{test}^T \cdot \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} - \frac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot R_{trial},$$

$$B_H = \left(R_{test}^T \cdot \left(\begin{bmatrix} 0 & 0 \\ 0 & M_c \end{bmatrix} + \tfrac{\triangle t}{2} \cdot \begin{bmatrix} M_w & C+G \\ -C^T & 0 \end{bmatrix} \right) \cdot R_{trial} \right)^{-1} \cdot \triangle t \cdot R_{test}^T,$$

and w_H, c_H denote the coarse scale solution. Similarly, an approximation of the measurable output may be written as $q_H = CRP_H = C_H P_H$.

We denote $F_H = B_H \cdot \begin{bmatrix} 0 \\ f_H(t_{n+1}) \end{bmatrix}$ and we have the following coarse system

$$\begin{aligned} P_H^{n+1} &= A_H P_H^n + F_H, \\ q_H^{n+1} &= C_H P_H^{n+1}. \end{aligned} \tag{6.11}$$

The next step is to apply BT approach. We calculate the Gramians by solving the following Lyapunov equations:

$$\begin{aligned} W_{\text{ob}} A_H + \big(A_H\big)^T W_{\text{ob}} + (C_H)^T C_H &= 0, \\ A_H W_{\text{co}} + W_{\text{co}} \big(A_H\big)^T + F_H (F_H)^T &= 0, \end{aligned} \tag{6.12}$$

where we denote the observability Gramian as W_{ob} and the controllability Gramian as W_{co}. Then having Cholesky decompositions of Gramians: $W_{\text{ob}} = L_{\text{ob}}^T L_{\text{ob}}$ and $W_{\text{co}} = L_{\text{co}} L_{\text{co}}^T$, we can solve the spectral problems

$$L_{\text{co}}^T W_{\text{ob}} L_{\text{co}}\, \xi_i = \rho_i \xi_i \quad \text{for} \quad i = 1, \ldots, N_r, \tag{6.13}$$

where N_r is the number of first ordered eigenvalues and normalized eigenvectors of $L_{\text{co}}^T W_{\text{ob}} L_{\text{co}}$ and we can choose N_r according to the desired error. The eigenvalues and (row) eigenvectors of $L_{\text{ob}}^T W_{\text{co}} L_{\text{ob}}$ are given by $\{\rho_i\}$ and $\{\zeta_i\} = \{\sigma_i^{-1} \xi_i^T L_{\text{co}}^T L_{\text{ob}}^T\}$ $(i = 1, \ldots, N_r)$, where $\sigma_i = \rho_i^{1/2}$. With the respective eigenvalues and eigenvectors in place we form the matrices U and V such that

$$V = L_{\text{co}} \left[\xi_1 \sigma_1^{-1/2} \cdots \xi_{N_r} \sigma_{N_r}^{-1/2}\right] \quad \text{and} \quad U = \begin{bmatrix} \sigma_1^{-1/2} \zeta_1 \\ \vdots \\ \sigma_{N_r}^{-1/2} \zeta_{N_r} \end{bmatrix} L_{\text{ob}},$$

with $UV = I_{N_r}$. The reduced order model of Eq. (6.11) is finally formed by creating the matrices $A_R = U A_H V$, $F_R = U F_H$, and $C_R = C_H V$. Finally, we arrive at the reduced dimension model:

$$\begin{aligned} P_R^{n+1} &= A_R P_R^n + F_R, \\ q_R^{n+1} &= C_R P_R^{n+1}. \end{aligned} \tag{6.14}$$

We would like the output q_R to approximate q_H.

Note that since the method is applied directly to the online coarse-scale system (which has already been reduced using local GMsFEM), the computational cost can be significantly reduced.

6.4 Numerical results

In order to study the performance of the proposed GMsFEM-BT approach for approximating coupled flow and transport equations we consider two numerical cases. For both cases we have the two-dimensional unit domain $D = [0, 1] \times [0, 1]$ and the diffusivity $G = 1$. The final time is taken to be $T = 0.1$. We use 10 time steps $(\Delta t = 0.01)$.

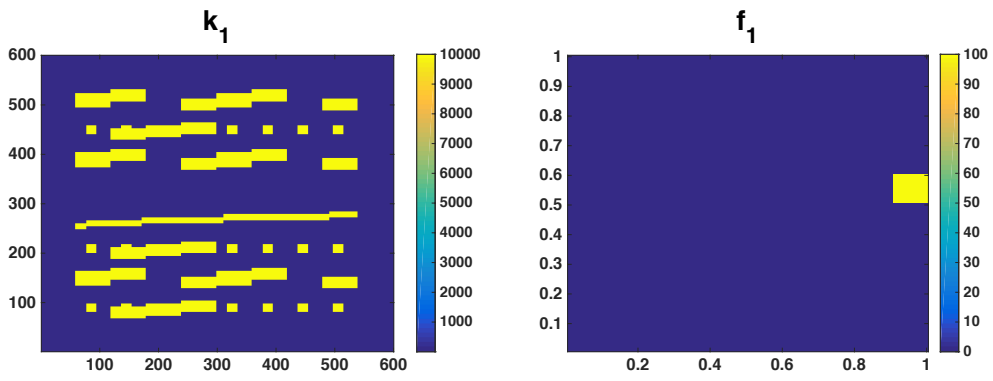


Figure 6.1: High-contrast permeability coefficient κ_1 and the source term f_1 for the first case

The first case consists of a permeability $\kappa_1(x)$, the source term $f = f_1$ as shown in Figure 6.1. Note that this permeability field $\kappa_1(x)$ exhibits a random high-contrast structure, where the contrast is taken to be $\kappa_1^{\max}/\kappa_1^{\min} \approx 10^4$. The boundary condition is $g = -7xy$. We discretize D into a 100×100 fine element mesh.

The second case consists of a permeability $\kappa_2(x)$, the source term $f = f_2$ as shown in Figure 6.2. The boundary condition for this case is $g = 10(x - y)$. We discretize D into a 200×200 fine-grid element mesh. We define the output coefficients for both cases to be the average concentration in the coarse elements. The velocity fields are shown in Figures 6.3 and 6.4 (see [16]).

In order to show the accuracy of the GMsFEM, we list the show the concentrations for fine-scale system (see Figure 6.5) and for coarse-scale system (see Figure 6.6) for the first case at different time instants. We need to make a note about the fine-scale

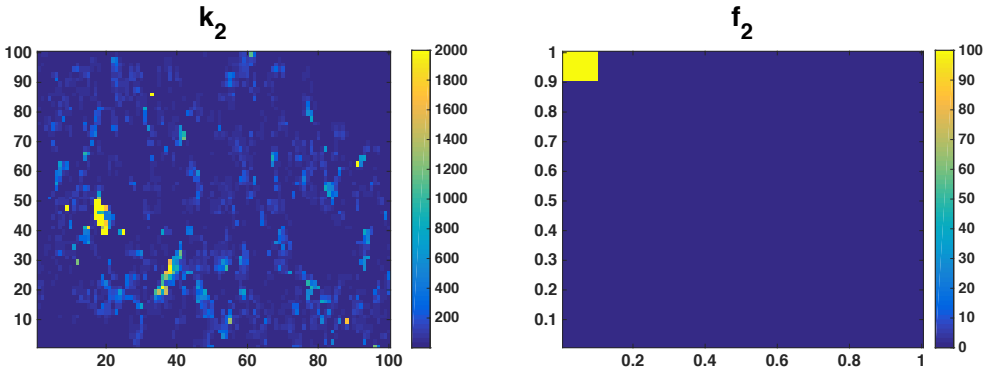


Figure 6.2: High-contrast permeability coefficient κ_2 and the source term f_2 for the second case

solution we used in Figure 6.5. The approximation of the fine-scale solution for the velocity field was calculated having one basis function for the pressure field and three basis functions for the velocity field.

The rest of numerical results are devoted to comparing the outputs from the fine-scale system, the coarse-scale system and the reduced system. We denote the fine-scale solution at the last time step as P_F, the GMsFEM solution as P_H, and the reduced solution obtained by BT as P_R, where $P = (w, c)'$ as before. Then we can calculate all outputs $q_F = C_F P_F$, $q_H = C_H P_H$, and $q_R = C_R P_R$. Notice that all inputs are calculated at the specific time instant (for simplicity, we omit the "time index"). Applying the triangle inequality, we have

$$\|q_F - q_R\|_2 \leq \|q_F - q_H\|_2 + \|q_H - q_R\|_2. \tag{6.15}$$

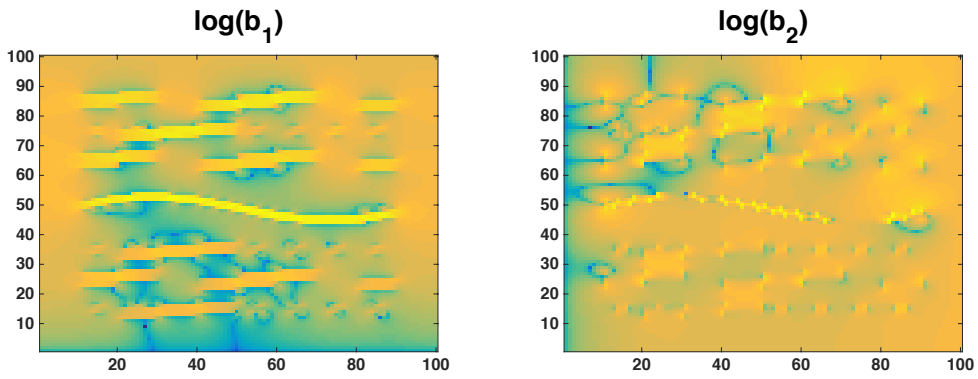


Figure 6.3: Velocity $(-7b_1, -7b_2)$ for the first case

We would like to choose the dimension such that the errors $\|q_F - q_H\|_2$ and $\|q_H - q_R\|_2$ are comparable. For further computations we take a discrete, relative l^2-norm scaled by the coarse output norm.

For the first case, a 10×10 coarse mesh is considered and for the second case, a 20×20 coarse mesh is considered with online spaces of dimension $N_c = 760$ and $N_c = 2920$, respectively. Tables 6.1 and 6.2 compare the errors between the outputs of coarse-scale system and the reduced system using a variety of reduced system size. For the first case, we test BT reduced dimensions of $N_r = 3, 4, 5$ and we observe that the error between the coarse-scale system and the reduced coarse-scale system decreases as the BT reduced dimension increases. We remind that the size of coarse space is 760 and the 'optimal' size (defined as the global reduced-order system size, which yields an error comparable to the error between the coarse-grid

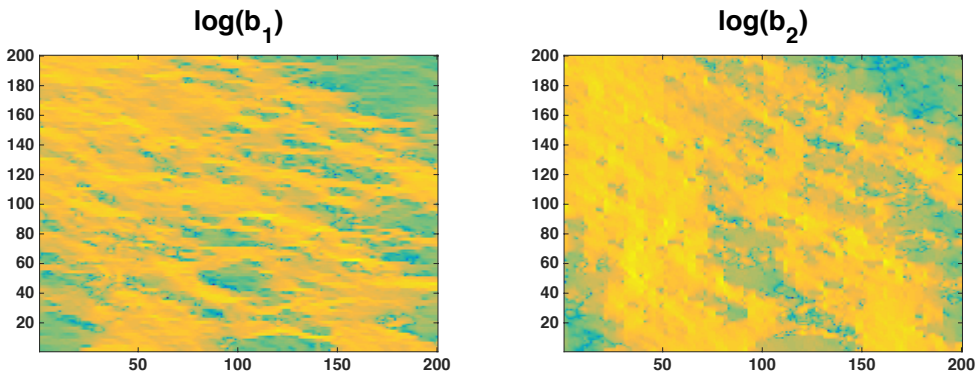


Figure 6.4: Velocity $(-10b_1, -10b_2)$ for the second case

and fine-grid systems) of reduced space is $3-4$, while the fine-scale system is of size $N_f = 30200$. For the second case, the optimal 'size' for the reduced system is $4-5$ global modes, while the fine-scale system is of size $N_f = 120400$.

Table 6.3 contains a set of computational times corresponding to the fine, coarse, and reduced dimension solution procedures. All calculations were completed serially on a dual-core desktop workstation where each core uses two Intel Core i3 2.6 GHz processors, each with 8GB of RAM. We notice that a parallel implementation of the suggested method would likely provide the online GMsFEM computations negligible, but still computational savings are significant. The first row shows the required time to calculate a solution for the given fine-scale system. Next, the actual time of finding a solution by the online GMsFEM approach. The last time is obtained by adding three components: a time for the construction of the GMsFEM online space,

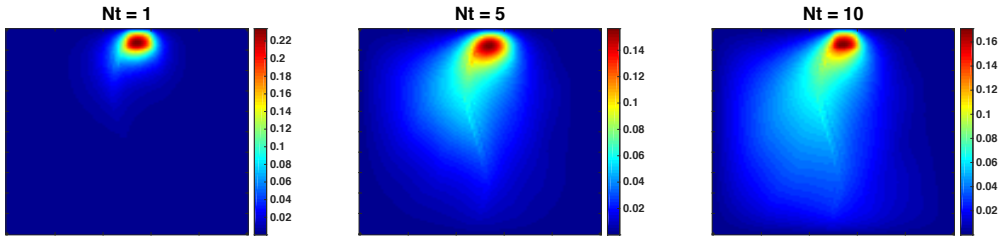


Figure 6.5: Fine-scale solution c for the first case at 1, 5 and 10 time steps

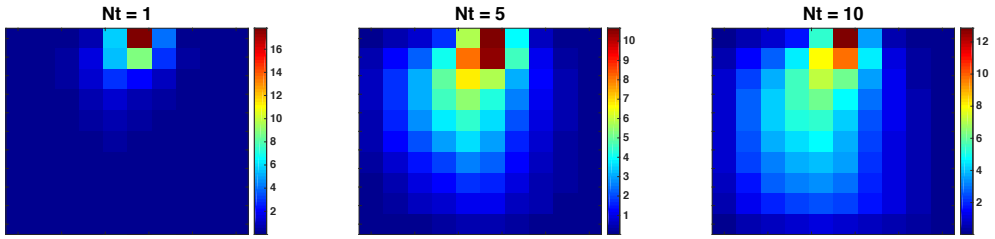


Figure 6.6: Coarse-scale solutions c for the first case at 1, 5 and 10 time steps

a time for solving the Lyapunov equations, and a time for finding a solution for the reduced GMsFEM-BT system. We can observe that the computational time for the GMsFEM is small compared to the BT. Also, the computational time of performing BT is very small. Performing BT on the fine grid is very expensive. Figures 6.7 and 6.8 give an illustration of the fine, coarse and reduced outputs for the first and the second cases, respectively. We observe from these figures that BT coarse-grid output approximation is close to the coarse-grid output approximation.

BT Dimension $[N_r]$	Fine-Coarse Error (%)	Coarse-BT Error (%)
3	5.93	9.66
4	5.93	4.79
5	5.93	1.28

Table 6.1: Measurable output errors for the first case

BT Dimension [N_r]	Fine-Coarse Error (%)	Coarse-BT Error (%)
3	10.68	20.59
4	10.68	15.19
5	10.68	6.96

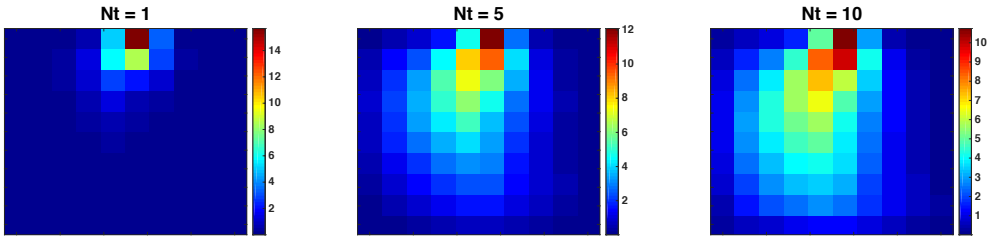
Table 6.2: Output errors for the second case

System	Size	Time (*sec*)
Fine	30200	**298.63**
GMsFEM (Online)	760	0.92
BT	4	4.22
GMsFEM-BT	4	**5.14**

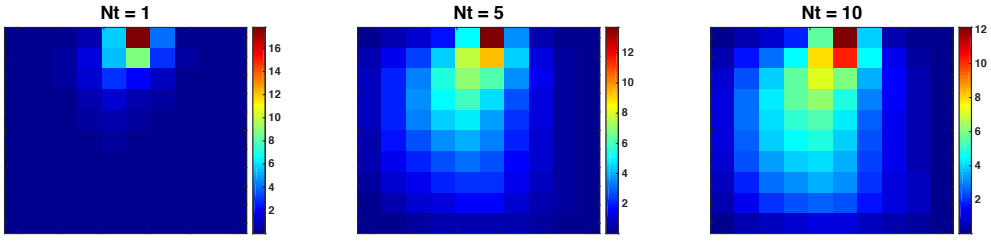
Table 6.3: Computational timing comparisons between the fine, GMsFEM online, and BT algorithms for the first case

6.5 Conclusions

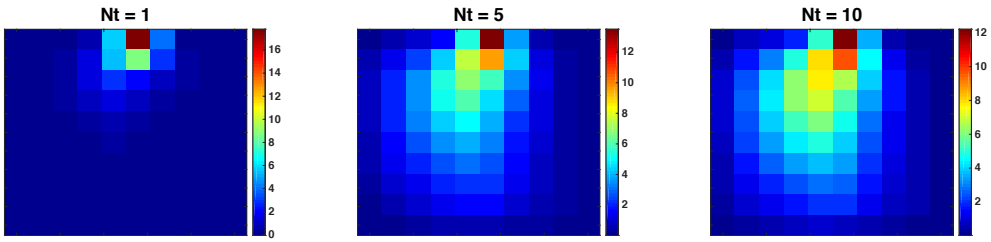
In this section we develop a reduced-order modeling technique for a coupled flow and transport equation. In order to have mass conservation we consider a mixed formulation for both flow and transport equations. GMsFEM constructs the multiscale spaces for the flux and the velocity by choosing the snapshot spaces and implementing local spectral decompositions. For further model reduction we use BT approach on a coarse-scale system for a mixed formulation. From the numerical examples, we observe that suggested coupled GMsFEM and BT technique allows to reduce the model significantly.



(a) Fine ($N_f = 30200$)

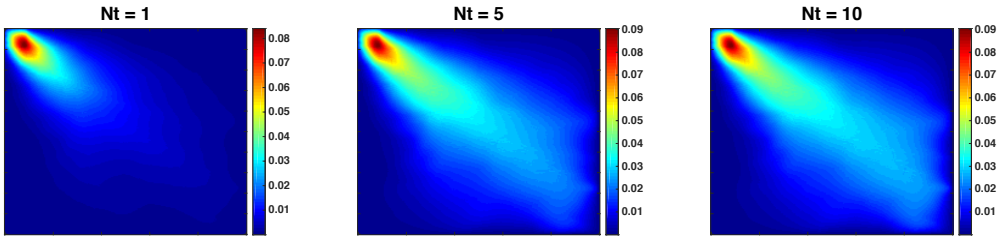


(b) GMsFEM ($N_c = 760$)

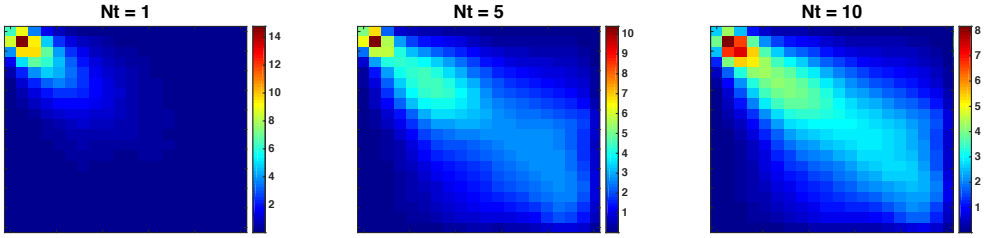


(c) BT ($N_r = 4$)

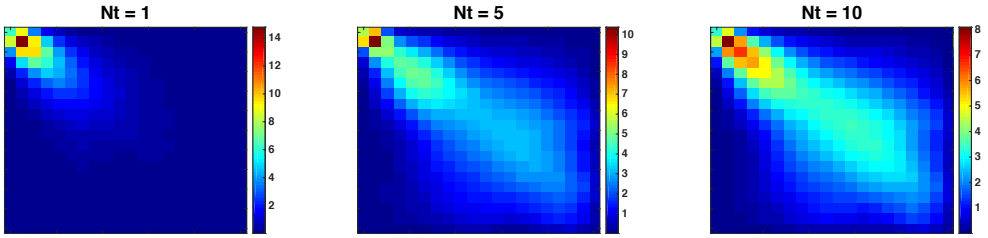
Figure 6.7: Output comparisons for the first case at 1, 5, and 10 time step instants



(a) Fine ($N_f = 120400$)



(b) GMsFEM ($N_c = 2920$)



(c) BT ($N_r = 5$)

Figure 6.8: Output comparisons for the first case at 1, 5, and 10 time step instants

7. CONCLUSIONS

Problems involving heterogeneous porous media applications entail prohibitively expensive computations due to the large-scale fields and extremely complex geometry. The proposed work is devoted to creating computationally efficient methods based on combining model reduction techniques such as Balanced Truncation (BT) approach, Generalized Multiscale Finite Element Method (GMsFEM), homogenization techniques. Four different model settings are considered.

The first case contains a model for flows in porous media with separable scales. In this section, two main objectives are achieved: a robust and inexpensive hierarchical local-global reduced-order modeling technique to solve problems involving heterogeneous porous media applications is developed and a modeling technique is further applied to obtain a solution for a single-phase compressible flow.

The second case is devoted to a model for a parameter-dependent, single-phase flow in general heterogeneous porous media. The input-output formulation of the problem is considered. The main result is a creating the offline-online local-global model reduction technique, where GMsFEM is used as a local model reduction tool and BT as a global model reduction tool. The localized multiscale basis function computations are cast in the framework of an offline-online procedure in which a respective set of eigenvalue problems are used to capture the underlying behavior of the system. Since the computation of the snapshots and offline space accounts for a one-time preprocessing step, the online coarse space may be cheaply constructed for a fixed input state.

In the next section, the second case is extended to a time-dependent permeability problem. The main challenge of this problem is a computational cost. The special

time-varying GMsFEM-BT approach allows solving the problem and obtaining time reduction due to the use of time-varying BT.

The last case is devoted to a convection-dominated flow and transport coupled system. We consider a mixed formulation for both flow and transport equations in order to preserve the mass conservation. As the first step of the process, the coarse grid solver from [16] is applied. This solver uses only a few multiscale basis functions in each coarse block to solve the coupled system. As the second step, the BT technique is applied on the obtained coarse system. The main contribution is a development of a robust model reduction technique that allows to significantly reduce the size of the system and save the computational timing.

REFERENCES

[1] J. E. Aarnes, S. Krogstad, K.-A. Lie, A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform grids, SIAM Multiscale Modelling and Simulation, Vol. 5(2), pp. 337–363, 2006.

[2] V.M. Adamjan, D.Z. Arov, M.G. Krein, Infinite block Hankel matrices and related extension problems, American Mathematical Society Transactions, Vol. 111, pp. 133-156, 1978.

[3] G. Allaire, Homogenization and two-scale convergence, SIAM Journal on Mathematical Analysis, Vol. 23, pp. 1482–1518, 1992.

[4] A.C. Antoulas, D.C. Sorensen, Approximation of large-scale dynamical systems: An overview, Applied Mathematics and Computer Science, Vol. 11(5), pp. 1093-1122, 2001.

[5] A.C. Antoulas, Approximation of large-scale dynamical systems, SIAM Press, Philadelphia, 2005.

[6] T. Arbogast, G. Pencheva, M. F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, SIAM Multiscale Modelling and Simulation, Vol. 6, pp. 319–346, 2007.

[7] I. Babuska, R. Lipton, Optimal local approximation spaces for generalized finite element methods with application to multiscale problems, SIAM Multiscale Modelling and Simulation, Vol. 9(1), pp. 373-406, 2011.

[8] I. Babuška, V. Nistor, N. Tarfulea, Generalized finite element method for second-order elliptic operators with Dirichlet boundary conditions, Journal of Computational and Applied Mathematics, Vol. 218, pp. 175–183, 2008.

[9] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathématique. Académie des Sciences. Paris, Vol. 339(9), pp. 667-672, 2004.

[10] R.H. Bartels, G.W. Stewart, Solution of the matrix equation $AX + XB = C$: Algorithm 432, Communications of the ACM, Vol. 15(9), pp. 820-826, 1972.

[11] P. Benner, J.-R. Li, T. Penzl, Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems, Numerical Linear Algebra with Applications, Vol. 15, pp. 755-777, 2008.

[12] S. Boyaval, Reduced-basis approach for homogenization beyond the periodic setting, SIAM Multiscale Modelling and Simulation, Vol. 7(1), pp. 466-494, 2008.

[13] D. Brown, V.H. Hoang, Y. Efendiev, An efficient hierarchical multiscale finite element method for Stokes equations in slowly varying media, Multiscale Modelling and Simulation, Vol. 11(1), pp. 30-58, 2013.

[14] C.T. Chen, Linear system theory and design (2nd. Edition), Holt, Rinehart and Winston, 1984.

[15] H. Cheng, K. Dehghani, T. Billiter, A structured approach for probabilistic-assisted history matching using evolutionary algorithms: Tengiz Field Applications, paper SPE 116212 presented at SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA, 21-24 September, 2008.

[16] E. T. Chung, Y. Efendiev, W. T. Leung, J. Ren, Multiscale simulations for coupled flow and transport using the Generalized Multiscale Finite Element Method, Computation, Vol. 3(4), pp. 670-686, 2015.

[17] C.L. Cipolla, X. Weng, M. G. Mack, Integrating microseismic mapping and complex fracture modelling to characterize hydraulic fracture complexity, paper SPE 140185 presented at SPE Hydraulic Fracturing Technology Conference, The Woodlands, Texas, USA, 24-26 January, 2011.

[18] B. N. Datta, Krylov subspace methods in control: an overview, Decision and Control, Proceedings of the 36th IEEE Conference, Vol. 4, pp. 3844-3848, 1997.

[19] J. F. M. van Doren, R. Markovinovic, J.D. Jansen, Reduced-order optimal control of water flooding using proper orthogonal decomposition, Computational Geosciences, Vol. 10, pp. 137-158, 2006.

[20] R. Dorf, R.H. Bishop, Modern control systems, Prentice Hall, 11/E, 2010.

[21] L. J. Durlofsky, Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media, Water Resources Research, Vol. 27, pp. 699-708, 1991.

[22] Y. Efendiev, J. Galvis, E. Gildin, Local-global multiscale model reduction for flows in highly heterogeneous media, Journal of Computational Physics, Vol. 231(24), pp. 8100-8113, 2012.

[23] Y. Efendiev, J. Galvis, G. Li, M. Presho, Generalized Multiscale Finite Element methods, Journal of Computational Physics, Vol. 251, pp. 116-135, 2013.

[24] Y. Efendiev, J. Galvis, F. Thomines, A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications, SIAM Multiscale Modelling and Simulation, Vol. 10(4), pp. 1317-1343, 2012.

[25] Y. Efendiev, J. Galvis, X. Wu, Multiscale finite element methods for high-contrast problems using local spectral basis functions, Journal of Computational

Physics, Vol. 230, pp. 937–955, 2011.

[26] Y. Efendiev, V. Ginting, T. Hou, R. Ewing, Accurate multiscale finite element methods for two-phase flow simulations, Journal of Computational Physics, Vol. 220(1), pp. 155-174, 2006.

[27] Y. Efendiev, T. Hou, Multiscale finite element methods: theory and applications, Springer, New York, 2009.

[28] Y. Efendiev, T. Hou, V. Ginting, Multiscale finite element methods for nonlinear problems and their applications, Communications in Mathematical Sciences, Vol. 2(4), pp. 553-589, 2004.

[29] E. Gildin, M. Ghasemi, A. Protasov, Y. Efendiev, Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media, paper SPE 163618 presented at SPE Annual Technical Conference and Exhibition, Woodlands, Texas, USA, 18-20 February, 2013.

[30] V. Ginting, F. Pereira, M. Presho, Michael, S. Wo, Application of the two-stage Markov chain Monte Carlo method for characterization of fractured reservoirs using a surrogate flow model, Computational Geosciences, Vol. 15(4), pp. 691-707, 2011.

[31] K. Glover, All optimal Hankel-norm approximations of linear multivariable systems and their L^{∞}-error bounds , International Journal of Control, Vol. 39, pp. 1115-1193, 1984.

[32] E.J. Grimme, Krylov projection methods for model reduction, PhD Dissertation, ECE Department, University of Illinois, Urbana-Champaign, 1997.

[33] S. Guercin, A. Antoulas, A survey of model reduction by balanced truncation and some new results, International Journal of Control, Vol. 77(8), pp. 748-766,

2004.

[34] S. Gugercin, D.C. Sorensen, A.C. Antoulas, A modified low-rank Smith method for large-scale Lyapunov equations, Numerical Algorithms, Vol. 32, pp. 27-55, 2003.

[35] S. Hammarling, Numerical solution of the stable, non-negative definite Lyapunov equation, IMA Journal of Numerical Analysis, Vol. 2, pp. 303-323, 1982.

[36] M. Heinkenschloss, T. Reis, A.C. Antoulas, Balanced truncation model reduction for systems with inhomogeneous initial conditions, Automatica, Vol. 47, pp. 559-564, 2011.

[37] J. P. Hespanha, Linear systems theory, Princeton University Press, 2009.

[38] M. Hinze, S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control, Dimension Reduction of Large-Scale Systems, pp. 261-306, 2005.

[39] V. H. Hoang, High-dimensional finite elements for periodic monotone nonlinear elliptic problems with multiple scales, SIAM Multiscale Modelling and Simulations, Vol. 7, pp. 1042-1072, 2008.

[40] V. H. Hoang, Ch. Schwab, High dimensional finite elements for elliptic problems with multiple scales, SIAM Multiscale Modelling and Simulations, Vol. 3, pp. 168-194, 2005.

[41] T. Hou, X.H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, Journal of Computational Physics, Vol. 134, pp. 169–189, 1997.

[42] T. Hughes, G. R. Feijóo, Gonzalo, L. Mazzei, J. -B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Computer Methods in Applied Mechanics and Engineering, Vol. 166(1-2), pp. 3-24, 1998.

[43] P. Jenny, S. H. Lee, H.A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, Journal of Computational Physics, Vol. 187(1), pp. 47-67, 2003.

[44] V. Jikov, S. Kozlov, O. Oleinik, Homogenization of differential operators and integral functionals, Springer-Verlag, 1994, Translated from Russian.

[45] T. Kailath, Linear systems, Prentice-Hall, New Jersey, 1980.

[46] W. J. Lee, C. W. Hopkins, Characterization of tight reservoirs, SPE Journal of Petroleum Technology, Vol. 46(11), pp. 956-964, 1994.

[47] E.B. Lee, L. Markus, Foundations of optimal control theory, The SIAM Series in Applied Mathematics, John Wiley & Sons, 1967.

[48] T. Li, P. Chang-Yi Weng, E. King-wah Chu, Wen-Wei Lin, Large-scale Stein and Lyapunov equations, Smith method, and applications, Numerical Algorithms, Vol. 63, pp. 727-752, 2013.

[49] N. C. Nguyen, A multiscale reduced-basis method for parameterized elliptic partial differential equations with multiple scales, Journal of Computational Physics, Vol. 227(23), pp. 9807-9822, 2008.

[50] M. Presho, A. Protasov, E. Gildin, Local-global model reduction of parameter-dependent, single-phase flow models via balanced truncation, Journal of Computational and Applied Mathematics, Vol. 271, pp. 163-179, 2014.

[51] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parameterized elliptic coercive partial dif-

ferential equations: application to transport and continuum mechanics, Archives of Computational Methods in Engineering, Vol. 15(3), pp. 229-275, 2008.

[52] J. Sabino, Solution of large-scale Lyapunov equations via the block modified Smith method, PhD Dissertation, Rice University, Houston, 2006.

[53] H. Sandberg, Model reduction for linear time-varying systems, Department of Automatic Control Lund Institute of Technology, PhD Dissertation, 2004.

[54] H. Sandberg, A. Rantzer, Balanced truncation of linear-varying systems, IEEE transactions on automatic control, Vol. 49(2), pp. 217-229, 2004.

[55] E. D. Sontag, Mathematical control theory: deterministic finite dimensional systems (2nd. edition), Text in Applied Mathematics, 6, Springer-Verlag, New York, 1998.

[56] X.H. Wu, Y. Efendiev, T.Y. Hou, Analysis of upscaling absolute permeability, Discrete and Continuous Dynamical Systems Series B, Vol. 2, pp. 185-204, 2002.

[57] K. Zhou, John C. Doyle, K. Glover, Robust and optimal control, Prentice-Hall, New Jersey, 1996.