Article

# Generalized Multiscale Finite Element Method and Balanced Truncation for Parameter-Dependent Parabolic Problems 

Shan Jiang ${ }^{1, *(D)}$, Yue Cheng ${ }^{1}$, Yao Cheng ${ }^{2(D)}$ and Yunqing Huang ${ }^{3,4,5}$<br>1 School of Science, Nantong University, Nantong 226019, China; 2023085@ntu.edu.cn<br>2 School of Mathematical Sciences, Suzhou University of Science and Technology, Suzhou 215009, China; ycheng@usts.edu.cn<br>3 School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China; huangyq@xtu.edu.cn<br>4 Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan 411105, China<br>5 Key Laboratory of Intelligent Computing Information Processing of Ministry of Education, Xiangtan 411105, China<br>* Correspondence: jiangshan@ntu.edu.cn

Citation: Jiang, S.; Cheng, Y.; Cheng, Y.; Huang, Y. Generalized Multiscale Finite Element Method and Balanced Truncation for Parameter-Dependent Parabolic Problems. Mathematics 2023, 11, 4965. https://doi.org/10.3390/ math11244965

Academic Editor: Dongfang Li
Received: 24 October 2023
Revised: 13 November 2023
Accepted: 27 November 2023
Published: 15 December 2023


Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).


#### Abstract

We propose a generalized multiscale finite element method combined with a balanced truncation to solve a parameter-dependent parabolic problem. As an updated version of the standard multiscale method, the generalized multiscale method contains the necessary eigenvalue computation, in which the enriched multiscale basis functions are picked up from a snapshot space on users' demand. Based upon the generalized multiscale simulation on the coarse scale, the balanced truncation is applied to solve its Lyapunov equations on the reduced scale for further savings while ensuring high accuracy. A $\theta$-implicit scheme is utilized for the fully discretization process. Finally, numerical results validate the uniform stability and robustness of our proposed method.


Keywords: generalized multiscale finite element; balanced truncation; eigenvalue computation; fully discretization; uniform stability

MSC: 35K20; 65M12; 65M60

## 1. Introduction

Many scientific and engineering problems in porous media flow and composite material involve inherently multiscale model equations. Basically, the model coefficient often exhibits high-contrast and parameter-dependent behaviors in a sense. Conventional methods such as the finite difference method, or finite element method, which are applied to solve these problems on a fully-resolved scale would become unbearably expensive, especially when a large amount of computations may be consumed for uncertainty parameter quantification, model reduction and optimization. So an urgent demand for developing multiscale model reductions with properties of both accuracy and efficiency is of particular interest.

In the research history, the multiscale finite element method (MsFEM) was first proposed by Hou and Chen in [1,2]. Then it evolved to a generalized multiscale version and had many applications in scientific fields, see [3-7]. For example, in [3] the generalized multiscale finite element (GMsFEM) approximated the solution space locally by employing the enriched multiscale basis functions. This approximation constructed a proper snapshot space and a local spectral decomposition to realize an efficient model reduction. Chung, Efendiev and Hou proposed an adaptive multiscale model reduction with GMsFEM in [5], an adaptively local approach was utilized to update the global snapshot when online computations proceeded. The local-global couple is shown to be effective in many applications. As for the parameter-dependent model and the thermoelastic model with phase flow,
the strategy of multiscale model reduction was addressed in [8-12]. Besides, there were other techniques such as multiscale reduced-basis method [8,10,13,14], localized orthogonal decomposition [15], weak Galerkin GMsFEM [16], mixed GMsFEM [17], multiscale-spectral GFEM [18], meshfree GMsFEM [19], higher-order three-scale method [20] and local discontinuous Galerkin method [21], were applied in a large number of interested fields.

In literature [22-24], a balanced truncation (BT) is directly addressed to their problems. The aim of BT is to study the principal component analyses in linear systems and in the time or frequency domains, and it identifies the weakly controllable and observable components which are used for model reductions. Moreover, the eigenvalues of the Lyapunov solutions in frequency-limited BT often decay faster than those in the standard BT. However, as for the parameter-dependent parabolic problem, it would be annoyingly costing or infeasible since Lyapunov equations are hard to solve. In particular, solving these equations from the high-contrast permeability, boundary condition, and forcing term on the fine scale would quickly become prohibitive. Therefore, in this work we propose the joint usages of a generalized multiscale and balanced truncation to obtain accurate and economical results.

The paper is characterized by the following highlights: As for the parameter-dependent parabolic problem, we present a combination strategy of the generalized multiscale finite element method with the balanced truncation, namely GMsFEM-BT. The GMsFEM is skilled in the random-parameterized model and different coefficients and boundary conditions, it constructs an enriched projection matrix in the global reduction from the snapshot space. And it offers a good balance for the accuracy and the efficiency, which is compared with the fully-resolved FEM and the coarse version of MsFEM. Based on this, the BT is addressed on the correspondingly reduced scale for further reduction. Then moving from a current time step to the next time step, an effective implicit full-discretization process is utilized for the spatial-temporal problem. The uniform stability and convergent results are to be validated in our work.

The paper proceeds as follows. In Section 2, a preliminary description of the parameterdependent parabolic problem is provided and we brief the idea of the local-global multiscale reduction. In Section 3, the GMsFEM is applied to the spatial variables through an eigenvalue computation in the spectral sub-problems, and an offline-online coupling is accomplished by a projection matrix. In Section 4, the detail of BT is offered for further reduction for the model. We present a fully discretization process both for time and for space in Section 5. In Section 6 numerical experiments validate the ability of the combined GMsFEM-BT strategy. Finally, the conclusion is presented in Section 7.

## 2. Preliminary Description

### 2.1. Parameter-Dependent Parabolic Model

We investigate a parameter-dependent parabolic problem in the two-dimensional heterogeneous media,

$$
\begin{cases}u_{t}-\operatorname{div}(\kappa(\mathbf{x} ; \mu) \nabla u)=f & \text { in } \Omega \times[0, T]  \tag{1}\\ u=g & \text { on } \partial \Omega \times[0, T] \\ u=u_{0} & \text { at } t=0 \text { and in } \Omega\end{cases}
$$

where $\kappa(\mathbf{x} ; \mu)$ is a high-contrast coefficient with spatial variables $\mathbf{x}$ in the two-dimensional square domain $\Omega, \mu$ is used to represent the parameter dependence and may be random. $T$ is the time termination, $f(\mathbf{x}, t)$ and $g(\mathbf{x}, t)$ are given functions, and $u_{0}(\mathbf{x})$ is an initial function at $t=0$. The numerical solution with the most accuracy, efficiency and stability to $u(\mathbf{x}, t)$ is our target in this work.

In the finite-dimensional discretization, the spatial domain $\Omega$ is divided into many quadrilateral finite elements on coarse grids and then divided on fine grids, which is denoted by $\mathcal{T}^{H}$ and $\mathcal{T}^{h}$, respectively. For a clearer illustration see Figure 1, firstly we divide the domain on coarse grids, in this way coarse elements $K$ are formed. Denote the coarse nodes by $\left\{y_{i}\right\}_{i}^{N_{c}}$ ( $N_{c}$ is the number of coarse nodes), and denote the coarse neighborhood of $y_{i}$ by

$$
\begin{equation*}
\omega_{i}=\bigcup\left\{K_{j} \in \mathcal{T}^{H} ; y_{i} \in \overline{K_{j}}\right\} \tag{2}
\end{equation*}
$$

In the following, it will be applied in our generalized multiscale finite element method. Secondly, on the coarse grids $\mathcal{T}^{H}$ we refine every coarse element $K$ into more fine elements $k$, which is denoted by $\mathcal{T}^{h}$ on those fine grids. In this paper, we denote the coarse and the fine (of every coarse element) partition number in each spatial direction by $N$ and $M$, thus $H\left(=\frac{1}{N}\right)$ and $h\left(=\frac{H}{M}\right)$ are used to denote the coarse mesh size and the fine one, respectively.

It is known that the classical finite element method is manipulated on very fine grids $\mathcal{T}^{h}$ to achieve its maximum accuracy. The functional space is spanned by the piecewise bilinear polynomials $\mathbb{Q}^{1}(k)$, which are subordinated to $\mathcal{T}^{h}$ as

$$
\begin{equation*}
V^{h}=\left\{v \in H^{1}(\Omega): v_{\left.\right|_{k}} \in \mathbb{Q}^{1}(k), \forall k \in \mathcal{T}^{h}\right\} . \tag{3}
\end{equation*}
$$

In Galerkin weak form, it is applied to find $u_{h} \in V^{h}$ such that

$$
\begin{equation*}
\left(\frac{\partial u_{h}}{\partial t}, v_{h}\right)+a\left(u_{h}, v_{h}\right)=\left(f, v_{h}\right), \quad \forall v_{h} \in V^{h} \tag{4}
\end{equation*}
$$

where

$$
\begin{gathered}
\left(\frac{\partial u}{\partial t}, v\right)=\int_{\Omega} \frac{\partial u}{\partial t} v \mathrm{~d} \mathbf{x} \\
a(u, v)=\int_{\Omega} \kappa(\mathbf{x} ; \mu) \nabla u \cdot \nabla v \mathrm{~d} \mathbf{x} \\
(f, v)=\int_{\Omega} f v \mathrm{~d} \mathbf{x} .
\end{gathered}
$$

As for the initial-boundary value parabolic problem (1), the above (4) produces a discrete system to solve

$$
\begin{equation*}
M \vec{u}_{t}+A(\mu) \vec{u}=\vec{b}, \tag{5}
\end{equation*}
$$

where $M=\left[m_{i j}\right]=\int_{\Omega} \psi_{i} \psi_{j} \mathrm{~d} \mathbf{x}$ is a mass matrix, $A(\mu)=\left[a(\mu)_{i j}\right]=\int_{\Omega} \kappa(\mathbf{x} ; \mu) \nabla \psi_{i} \cdot \nabla \psi_{j} \mathrm{~d} \mathbf{x}$ is a parameter-dependent stiffness matrix, $\vec{b}=\left[b_{i}\right]=\int_{\Omega} f \psi_{i} \mathrm{dx}$ is a right-side vector, and $\psi_{i}$ is the bilinear basis function in the 2D problem.


Figure 1. Illustration of coarse grids, elements and neighborhoods.
As a consequence, in the finite element scheme with a very large partition number, a huge-size discrete system is obliged to solve

$$
\begin{equation*}
M_{f} \vec{u}_{f t}+A_{f}(\mu) \vec{u}_{f}=\vec{b}_{f}, \tag{6}
\end{equation*}
$$

where $M_{f}=\Sigma \int_{k} \psi_{i} \psi_{j} \mathrm{~d} \mathbf{x}, A_{f}(\mu)=\Sigma \int_{k} \kappa(\mathbf{x} ; \mu) \nabla \psi_{i} \cdot \nabla \psi_{j} \mathrm{~d} \mathbf{x}$, and $\vec{b}_{f}=\Sigma \int_{k} f \psi_{i} \mathrm{~d} \mathbf{x}$. The subscript $f$ means that the matrices and vectors are processed from all of the fine elements,
which is different from the subscript $c / r$ from the coarse/reduced elements in the following situations. For instance, the stiffness matrix $A_{f}(\mu)$ is assembled from all of the fine elements, whose size is $N_{f} \times N_{f}$, where $N_{f}=(N M+1)^{2}$ for the 2D spatial discretization. In the paper, we denote the nodal matrices by $B, C$ (which are transposed to each other $B=C^{T}$, and the size of $B$ is $N_{f} \times N^{2}$ ), which are allocated to the domain boundary vertices.

### 2.2. Local-Global Multiscale Reduction

In this subsection, we begin with the process of semi-discretization in the space. Different from the above-mentioned finite element method (5) on very fine grids, we present two types of local-global multiscale reductions on coarse grids by contrast.

First, we recall the standard multiscale finite element method (see [1,2,25-27]). A homogeneous sub-problem is proposed for solving the multiscale basis functions $\phi_{i}$ in every quadrilateral coarse element $K$ as

$$
\begin{cases}-\operatorname{div}\left(\kappa(\mathbf{x} ; \mu) \nabla \phi_{i}\right)=0 & \text { in } K \in \mathcal{T}^{H}  \tag{7}\\ \phi_{i}=l_{i} & \text { on } \partial K\end{cases}
$$

where the boundary condition $l_{i}$ satisfies that $\delta_{j}\left(y_{i}\right)=1(i=j), \delta_{j}\left(y_{i}\right)=0(i \neq j)$. It is known that useful local information is delivered through the multiscale basis functions into a global system. However, we also know that this type of standard multiscale scheme has its deficiencies, such as the relatively low accuracy, slow convergence even divergence in former literature.

Second, we track the generalized multiscale finite element method (see [3-5]). Another version of the non-homogeneous local problem in every coarse neighborhood is solved for the (updated) multiscale basis functions, via the necessary eigenvalue computation. In this way, plenty of parameter-dependent details are captured through the over-sampling technique and this would be solved via a singular value decomposition.

Thus in contrast to the finite element method in the fine version (6), the generalized multiscale method in the coarse version is

$$
\begin{equation*}
M_{c} \vec{u}_{c t}+A_{c}(\mu) \vec{u}_{c}=\vec{b}_{c} . \tag{8}
\end{equation*}
$$

In this multiscale scheme, the size of the stiffness matrix $A_{c}(\mu)$ is just $N_{c} \times N_{c}$, where $N_{c}=(N+1)^{2}$ to realize a preliminary computation reduction. It also builds the counterpart of nodal matrices, which are denoted by $B_{c}, C_{c}$ with $B_{c}=R B, C_{c}=C R^{T}$. The projection matrix $R$ with sufficient microscopic information is constructed from the multiscale basis functions, which would be provided in the following Formula (21).

If it is necessary for a further decline in computational costs, we could carry out a further matrix reduction with a strategy of balanced truncation in Section 4, which is in the reduced-version like

$$
\begin{equation*}
M_{r} \vec{u}_{r t}+A_{r}(\mu) \vec{u}_{r}=\vec{b}_{r} . \tag{9}
\end{equation*}
$$

The size of $A_{r}$ is $N_{r} \times N_{r}$, and the number $N_{r}\left(\ll N_{c} \ll N_{f}\right)$ can be artificially chosen by users later on. Singular value decompositions would be employed to form the matrices $U, V$ in (29) with $U V=I_{N_{r}}$. Set $B_{r}=U B_{c}, C_{r}=C_{C} V$, thus the matrices are further reduced to save consumptions. At the same time, accuracy and efficiency are granted in the novel method.

In a word, we are constantly motivated to chase a better cost-effective balance for the complicated multiscale parameter-dependent parabolic model.

## 3. Mechanism of GMsFEM

The GMsFEM is an extended framework that generalizes the MsFEM. Its key mechanism relies on that the multiple basis functions in every coarse neighborhood could be systematically updated, by flexibly enriching the functional space through necessary eigenvalue computations.

The GMsFEM has two stages: offline and online. At the offline stage, firstly a snapshot space is created then based on it a small dimensional construction is required for solving local spectral decompositions. One of the advantages of the GMsFEM is that the created snapshot space can be re-used to capture the necessary fine-scale features. After that, a space reduction is performed and the dominant modes (including sufficient eigenvalues and eigenvectors) are selected as the multiscale basis functions, for any input parameters which are fitting our parameter-dependent problem ideally. At the online stage, when the corresponding boundary condition and force term are defined, the available offline basis functions are utilized to pursue a high-precision numerical solution with the potential least costs. Users can also adaptively determine the various number of eigenfunctions needed in the coarse neighborhoods, to fulfill a proper balance of efficiency and accuracy.

The mathematical theories behind the offline and online spaces are the construction of snapshot space and the design of spectral sub-problems, we will present the details in the following subsections in turn. Since plenty of parameter-dependent multiscale problems may have small eigenvalues, we would like to comprise enough of the eigenvectors associated with the small eigenvalues to be enriched into the coarse functional space surely. As a consequence, the multiscale basis functions could be achieved by multiplying the dominant eigenmodes by the partition of unity functions.

### 3.1. Offline Stage

At this stage, we first create a snapshot space subordinated to each coarse neighborhood $\omega_{i}$ in (2). The construction of snapshot space involves solving the local problems from a set of parameter-dependent inputs. Being different from the standard multiscale finite element method solving the local problem (7), an eigenvalue computation is utilized in this generalized multiscale finite element method.

First, we solve for the snapshot functions from the spectral sub-problems: seeking $\left(\lambda_{i, l}^{\omega_{i}, \text { snap }}, \phi_{i, l}^{\omega_{i}, \text { snap }}\right) \in \mathbb{R} \times V^{\omega_{i}, \text { snap }}$ such that

$$
\begin{equation*}
-\operatorname{div}\left(\kappa(\mathbf{x} ; \mu) \nabla \phi_{i, l}^{\omega_{i}, \text { snap }}\right)=\lambda_{i, l}^{\omega_{i}, \text { snap }} \phi_{i, l}^{\omega_{i} \text {,snap }} \quad \text { in } \quad \omega_{i} \in \mathcal{T}^{H}, \tag{10}
\end{equation*}
$$

where $l=1, \cdots, L_{i}$ ( $L_{i}$ is the chosen number of eigenfunctions from small to large that correspond to the dominant eigenvalues). Note that we define zero Neumann boundary condition for (10), except in the discontinuous case the Dirichlet condition is defined instead. It is known that solving all the eigenfunctions of a large matrix is exhausting, so we choose the $L_{i}$ eigenfunctions to construct a snapshot space. The snapshot space $V^{\omega_{i}}$,snap is associated to the neighborhood $\omega_{i}$, which is produced from the multiscale eigenfunctions as

$$
\begin{equation*}
V^{\omega_{i}, \text { snap }}=\operatorname{span}\left\{\phi_{i, l}^{\omega_{i}, \text { snap }}: i=1, \cdots, 4, l=1, \cdots, L_{i}\right\} . \tag{11}
\end{equation*}
$$

Next, the snapshot functions are reordered by using a single index as the columns to create a projection matrix in a snapshot

$$
\begin{equation*}
R^{\omega_{i}, \text { snap }}=\left[\phi_{1}^{\omega_{i}, \text { snap }}, \cdots, \phi_{L_{\text {snap }}}^{\omega_{i}, \text { snap }}\right] \tag{12}
\end{equation*}
$$

where $L_{\text {snap }}$ denotes the number of functions in the snapshot construction. It should be pointed out that $L_{\text {snap }}$ is not only dependent on the chosen number $L_{i}$, but also it is related to the number of parameter $\mu$. For brevity, we omit the superscript $\omega_{i}$ to brief as $V^{\text {snap }}, R^{\text {snap }}$. However, keep in mind that throughout the paper, the offline and online eigenvalue decompositions are localized for the corresponding coarse neighborhoods.

The discretized form of (10) is

$$
\begin{equation*}
A(\mu) \phi_{i, l}^{\omega_{i}, \text { snap }}=\lambda_{i, l}^{\omega_{i}, \text { snap }} M(\mu) \phi_{i, l}^{\omega_{i, \text { snap }}} \quad \text { in } \omega_{i} \in \mathcal{T}^{H} \tag{13}
\end{equation*}
$$

where $M(\mu)=\Sigma \int_{\omega_{i}} \widetilde{\kappa}(\mathbf{x} ; \mu) \psi_{i} \psi_{j} \mathrm{~d} \mathbf{x}$, and $\widetilde{\kappa}$ will be defined in the followed Formula (18).

Then, we perform an elementary dimension reduction of the snapshot space by applying an auxiliary spectral decomposition to build an offline space. We seek a subspace of the original snapshot space so that it may properly approximate any neighborhood. The key is to efficiently build a set of multiscale basis functions for each parameter $\mu$. We reiterate the offline bilinear forms to be parameter-independent, then there is no need to rebuild the offline space for each value $\mu$. Then the eigenvalue problem can be described as

$$
\begin{equation*}
A^{\text {off }} \Phi_{l}^{\text {off }}=\lambda_{l}^{\text {off }} M^{\text {off }} \Phi_{l}^{\text {off }} \tag{14}
\end{equation*}
$$

where

$$
\begin{gathered}
A^{\text {off }}=\int_{\omega_{i}} \bar{\kappa}(\mathbf{x} ; \mu) \nabla \phi_{i}^{\text {snap }} \cdot \nabla \phi_{j}^{\text {snap }}=R^{\text {snap }} \bar{A}(\mu)\left(R^{\text {snap }}\right)^{T}, \\
M^{\text {off }}=\int_{\omega_{i}} \overline{\widetilde{\kappa}}(\mathbf{x} ; \mu) \phi_{i}^{\text {snap }} \phi_{j}^{\text {snap }}=R^{\text {snap }} \bar{M}(\mu)\left(R^{\text {snap }}\right)^{T},
\end{gathered}
$$

where $\bar{\kappa}(\mathbf{x} ; \mu)$ and $\overline{\widetilde{\kappa}}(\mathbf{x} ; \mu)$ are averaged parameter-dependent coefficients. $\bar{A}$ and $\bar{M}$ denote similar fine scale matrices as in (13), except that averaged coefficients of them are used. The smallest number of eigenvalues $L_{\text {off }}$ are picked up in (14) and form the corresponding eigenvectors as $\phi_{l}^{\text {off }}=\sum_{j} \Phi_{l, j}^{\text {off }} \phi_{j}^{\text {snap }}\left(l=1, \cdots, L_{\text {off }}, j=1, \cdots, L_{\text {snap }}\right)$, where $\Phi_{l, j}^{\text {off }}$ are the coordinates of the vector $\Phi_{l}^{\text {off }}$. In this way, the offline projection matrix is constructed as

$$
\begin{equation*}
R^{\text {off }}=\left[\phi_{1}^{\text {off }}, \cdots, \phi_{L_{\text {off }}}^{\text {off }}\right] . \tag{15}
\end{equation*}
$$

### 3.2. Online Stage

With the offline space available, next we build an online space on every coarse neighborhood for a specified parameter $\mu$. For the computational efficiency, it is assumed that the online space is a subspace with small dimensions. At the online stage, the reducedorder bilinear forms are chosen to be parameter-dependent. The eigenvalue problem is ready to solve

$$
\begin{equation*}
A^{\mathrm{on}}(\mu) \Phi_{l}^{\mathrm{on}}=\lambda_{l}^{\mathrm{on}} M^{\mathrm{on}}(\mu) \Phi_{l}^{\mathrm{on}}, \tag{16}
\end{equation*}
$$

where

$$
\begin{gathered}
A^{\mathrm{on}}(\mu)=\int_{\omega_{i}} \kappa(\mathbf{x} ; \mu) \nabla \phi_{i}^{\text {off }} \cdot \nabla \phi_{j}^{\text {off }}=R^{\text {off }} A(\mu)\left(R^{\text {off }}\right)^{T}, \\
M^{\mathrm{on}}(\mu)=\int_{\omega_{i}} \widetilde{\kappa}(\mathbf{x} ; \mu) \phi_{i}^{\text {off }} \phi_{j}^{\text {off }}=R^{\mathrm{off}} M(\mu)\left(R^{\mathrm{off}}\right)^{T},
\end{gathered}
$$

where $\kappa(\mathbf{x} ; \mu)$ and $\widetilde{\kappa}(\mathbf{x} ; \mu)$ now rely on the specified value $\mu, A(\mu)$ and $M(\mu)$ are specified fine-scale matrices in (13).

To construct the online space we pick the smallest number of eigenvalues $L_{\mathrm{on}}$ from (16) and finalize the corresponding eigenvectors as $\phi_{l}^{\mathrm{on}}=\sum_{j} \Phi_{l, j}^{\mathrm{on}} \phi_{j}^{\text {off }}\left(l=1, \cdots, L_{\mathrm{on}}\right.$, $\left.j=1, \cdots, L_{\text {off }}\right)$, where $\Phi_{l, j}^{\text {on }}$ are the coordinates of the vector $\Phi_{l}^{\text {on }}$. It should be noted that the matrices size in (16) solely relies on the dimension of the offline subspace. This results in the declining consumption of the online space, which in contrast to the original snapshot space would cost more expenses.

### 3.3. Offline-Online Coupling

In order to enrich the online basis functions into a reduced formula of (1), we denote $\chi_{i}$ as the multiscale partition of unity functions which are from another local problem

$$
\begin{cases}-\operatorname{div}\left(\kappa(\mathbf{x} ; \mu) \nabla \chi_{i}\right)=0 & \text { in } K \in \omega_{i},  \tag{17}\\ \chi_{i}=g_{i} & \text { on } \partial K,\end{cases}
$$

where $g_{i}$ is a continuous and bilinear boundary condition. A summed pointwise energy is defined as

$$
\begin{equation*}
\tilde{\kappa}=\kappa \sum_{i=1}^{N_{c}} H^{2}\left|\nabla \chi_{i}\right|^{2} . \tag{18}
\end{equation*}
$$

As above, $H$ is the coarse mesh size and $N_{c}$ is the number of coarse nodes. To construct the global coarse-grid solution space, the partition of unity functions are multiplied by the online eigenfunctions to obtain the final multiscale bases

$$
\begin{equation*}
\phi_{i, l}=\chi_{i} \phi_{l}^{\mathrm{on}}, \quad i=1, \cdots, N_{c}, l=1, \cdots, L_{\mathrm{on}} . \tag{19}
\end{equation*}
$$

Then it follows the online space

$$
\begin{equation*}
V^{\mathrm{on}}=\operatorname{span}\left\{\phi_{i, l}: i=1, \cdots, N_{c}, l=1, \cdots, L_{\mathrm{on}}\right\}, \tag{20}
\end{equation*}
$$

which is quite different from the large-scale finite dimensional space in (3). Using a single index notation we denote $V^{\text {on }}=\operatorname{span}\left\{\phi_{k}\right\}_{k=1}^{N_{c}}$. Recalling in (8) now we finalize the projection matrix $R$ to be utilized in the global reduction

$$
\begin{equation*}
R=\left[\phi_{1}, \cdots, \phi_{N_{c}}\right] \tag{21}
\end{equation*}
$$

where $\phi_{k}\left(k=1, \cdots, N_{c}\right)$ represents the nodal vector of the generalized multiscale bases. The matrix $R$ (whose size is $N_{c} \times N_{f}$ ) is now plugged in (8), which could be enriched with the dominant eigenfunctions and meaningful microscopic behaviors. This would lead to strengthening the numerical abilities later on.

Now it completes the coarse model reduction in the GMsFEM. In the next section, a further model reduction will be carried on a balanced truncation (BT).

## 4. Mechanism of Balanced Truncation

### 4.1. Related Lyapunov Equations

First recalling that for the standard Lyapunov equation

$$
\begin{equation*}
A W+W A^{T}=Z \tag{22}
\end{equation*}
$$

where the matrix $W$ is dense even though the system is sparse, and generally the eigenvalues of $W$ decay quickly when the right-side $Z$ has a low rank. This decay is typically associated with the approximation error from the balanced truncation.

Next, a joint measure of controllability and observability is offered for solving Lyapunov equations. It should be pointed out that these equations are not based on the above-mentioned fine matrices, but instead, they are based on the coarse matrices $A_{c}(\mu)$ defined in (8) and $B_{c}, C_{c}$ below that. Then the related Lyapunov equations are in the form of

$$
\begin{align*}
& A_{c}(\mu) W_{\mathrm{con}}+W_{\mathrm{con}}\left(A_{c}(\mu)\right)^{T}+B_{c}\left(B_{c}\right)^{T}=0  \tag{23}\\
& \left(A_{c}(\mu)\right)^{T} W_{\mathrm{obs}}+W_{\mathrm{obs}} A_{c}(\mu)+\left(C_{c}\right)^{T} C_{c}=0, \tag{24}
\end{align*}
$$

where $W_{\text {con }}$ is a controllability matrix, $W_{\text {obs }}$ is an observability one, and $Z$ is dependent on the $B_{c}, C_{c}$. The solutions of these Lyapunov equations play a key role in the model reduction and in the balanced truncation particularly.

Generally, the computations of dense matrices are of order $O\left(n^{3}\right)$. Some techniques have been evolved to efficiently compute the solutions by reducing the complexity. However, most of these techniques are often iterative and low-rank approximations, while in our high-contrast situation, it is hard to provide enough satisfactory approximations. In Matlab code lyap, the matrices are transformed into the complex Schur forms, and it computes the resulting triangular systems. In this way, the high-rank approximations are available and the solutions of Lyapunov equations have a close relationship to the dominant eigenvalues. Then

$$
\begin{aligned}
W_{\text {con }} & =\operatorname{lyap}\left(A_{c}(\mu), B_{c}\left(B_{c}\right)^{T}\right) \\
W_{\mathrm{obs}} & =\operatorname{lyap}\left(\left(A_{c}(\mu)\right)^{T},\left(C_{c}\right)^{T} C_{c}\right)
\end{aligned}
$$

are used to realize the numerical advantages of the coupled Lyapunov equations. Since the matrices $A_{c}(\mu), B_{c}, C_{c}$ are based on the coarse-scale, then the sizes of $W_{\text {con }}, W_{\text {obs }}$ are just $N_{c} \times N_{c}$.

Once we obtain the two matrices $W_{\text {con }}, W_{\text {obs }}$, which may be expressed in the form of Cholesky singular value decompositions to get $L_{\text {con }}, L_{\mathrm{obs}}$ as

$$
\begin{align*}
& W_{\mathrm{con}}=L_{\mathrm{con}} L_{\mathrm{con}}^{T} \\
& W_{\mathrm{obs}}=L_{\mathrm{obs}} L_{\mathrm{obs}}^{T} \tag{25}
\end{align*}
$$

and

$$
L_{\mathrm{obs}}^{T} L_{\mathrm{con}}=U \Sigma V^{T}=\left[\begin{array}{ll}
U_{1} & U_{2}
\end{array}\right]\left[\begin{array}{cc}
\Sigma_{1} & O  \tag{26}\\
O & O
\end{array}\right]\left[\begin{array}{c}
V_{1}^{T} \\
V_{2}^{T}
\end{array}\right]=U_{1} \Sigma_{1} V_{1}^{T}
$$

where $U, V$ are to be specified soon in (29). Set $S_{\text {con }}=L_{\text {con }} V_{1} \Sigma_{1}^{-1 / 2}, S_{\text {obs }}=\Sigma_{1}^{-1 / 2} U_{1}^{T} L_{\mathrm{obs}}{ }^{\prime}$ where $U_{1}^{T} U_{1}=V_{1}^{T} V_{1}=I_{N_{r}}$, and the subscript $r$ means the reduced case.

Note that our BT is based on the GMsFEM. An independent set of online basis functions are constructed in which the coarse-scale solution may be sought. As a consequence, the global system would be processed on a coarse scale to avoid applying the BT on a fine scale directly. After the offline and online stages in the GMsFEM, it produces a subspace with small dimensions for each fixed parameter $\mu$. It serves to further diminish the computational cost, which is associated with the balanced truncation in the model reductions.

### 4.2. Further Reduction

Based on coarse matrices (compared with the fine ones), now we have the platform to fulfill the potential further reduction. In our work, we use the first $N_{r}$ ordered eigenvalues and the normalized eigenvectors of the matrix $L_{\mathrm{con}}^{T} W_{\mathrm{obs}} L_{\mathrm{con}}$, by solving

$$
\begin{equation*}
L_{\mathrm{con}}^{T} W_{\mathrm{obs}} L_{\mathrm{con}} \xi_{i}=\lambda_{i} \xi_{i}, \quad i=1, \cdots, N_{r} . \tag{27}
\end{equation*}
$$

The sets of eigenvalues and eigenvectors are given by

$$
\begin{equation*}
\left\{\lambda_{i}\right\}, \quad \text { and } \quad\left\{\zeta_{i}\right\}=\left\{\sigma_{i}^{-1} \xi_{i}^{T} L_{\mathrm{con}}^{T} L_{\mathrm{obs}}^{T}\right\}, \tag{28}
\end{equation*}
$$

where $\sigma_{i}=\lambda_{i}^{1 / 2}$ is the singular values.
The balanced truncation is validated to be more efficient in the numerical simulation. Since their a-priori error estimates rely on the quick decay of eigenvalues in (27) and the magnitude of all truncated eigenvalues summation. On the other hand, the balanced truncation has its weaknesses too. The Lyapunov Equations (23) and (24) require expensive computations for unknown $W_{\text {con }}$ and $W_{\text {obs, }}$, and the singular value decompositions are costing too, especially for the very fine case.

With the eigenvalues and corresponding eigenvectors in place, the above-mentioned matrices $U$ and $V$ are constructed from the reduced matrices in (9) and $B_{r}, C_{r}$ below that as

$$
U=\left[\begin{array}{c}
\sigma_{1}^{-1 / 2} \zeta_{1}  \tag{29}\\
\vdots \\
\sigma_{N_{r}}^{-1 / 2} \zeta_{N_{r}}
\end{array}\right] L_{\mathrm{obs},} \quad V=L_{\mathrm{con}}\left[\xi_{1} \sigma_{1}^{-1 / 2} \cdots \xi_{N_{r}} \sigma_{N_{r}}^{-1 / 2}\right]
$$

whose size is $N_{r} \times N_{c}$ and $N_{c} \times N_{r}$, respectively. And the property of $U V=I_{N_{r}}$ is ensured.
As a consequence, the reduced version is $A_{r}(\mu)=U A_{c}(\mu) V$, where $r$ depends on the eigenvalue decay and is determined by users in the numerical experiment. Now we achieve the reduced scale since $A_{r}(\mu)$ has a size of $N_{r} \times N_{r}$, compared with the coarse scale $A_{c}(\mu)$ with a size of $N_{c} \times N_{c}$, while the fine-scale $A_{f}(\mu)$ with a size of $N_{f} \times N_{f}$, whose $N_{r} \ll N_{c} \ll N_{f}$.

## 5. Fully Discretization for Time and Space

As for the parabolic problem, we are now in the procedure of fully discretization. A $\theta$-implicit scheme is presented to the original Formula (5), we have

$$
\begin{aligned}
& M \frac{\vec{u}\left(\mathbf{x}, t_{m+1}\right)-\vec{u}\left(\mathbf{x}, t_{m}\right)}{\Delta t}+\theta A(\mu) \vec{u}\left(\mathbf{x}, t_{m+1}\right)+(1-\theta) A(\mu) \vec{u}\left(\mathbf{x}, t_{m}\right) \\
& =\theta \vec{b}\left(\mathbf{x}, t_{m+1}\right)+(1-\theta) \vec{b}\left(\mathbf{x}, t_{m}\right) .
\end{aligned}
$$

In the form of $\theta$-explicit scheme, it shows

$$
\begin{aligned}
& \left(\frac{M}{\Delta t}+\theta A(\mu)\right) \vec{u}\left(\mathbf{x}, t_{m+1}\right) \\
& =\theta \vec{b}\left(\mathbf{x}, t_{m+1}\right)+(1-\theta) \vec{b}\left(\mathbf{x}, t_{m}\right)+\frac{M}{\Delta t} \vec{u}\left(\mathbf{x}, t_{m}\right)-(1-\theta) A(\mu) \vec{u}\left(\mathbf{x}, t_{m}\right) .
\end{aligned}
$$

We know the influence of the temporal parameter $\theta$ on the fully discretization. In the following numerical experiments, we set $\theta=1$, namely the backward Euler difference is applied.

Define $\tilde{A}(\mu)=\frac{M}{\Delta t}+\theta A(\mu), \tilde{\vec{b}}\left(\mathbf{x}, t_{m+1}\right)=\theta \vec{b}\left(\mathbf{x}, t_{m+1}\right)+(1-\theta) \vec{b}\left(\mathbf{x}, t_{m}\right)+\left[\frac{M}{\Delta t}-(1-\theta) A(\mu)\right] \vec{u}\left(\mathbf{x}, t_{m}\right)$, then we get

$$
\begin{equation*}
\tilde{A}(\mu) \cdot \vec{u}\left(\mathbf{x}, t_{m+1}\right)=\tilde{\vec{b}}\left(\mathbf{x}, t_{m+1}\right) \tag{30}
\end{equation*}
$$

Solving the above ordinary differential equations, the corresponding numerical results are obtained from different schemes.

Thus in the paper, a proposed combination of GMsFEM-BT approach for the parameterdependent parabolic problem is evolved for both the spatial and the temporal discretizations. It avoids a huge system computation, while maintaining an ideal balance between accuracy and efficiency at the same time. The generalized multiscale method is addressed in an offline-online coarse scale system, which is further reduced via using the balanced truncation, whose computational cost would be saved greatly.

## 6. Numerical Experiment

Now numerical results are ready to reveal the strength of our proposed method. Matlab codes are executed on a Windows desktop with Intel(R) Core(TM) i9-10900K CPU 3.70 GHz and 32 GB RAM. We set the right side $f=1$, so there is no exact solution or analytical solution for this random-parameterized parabolic problem. To solve the problem (1), first we divide the square domain $\Omega=[0,1] \times[0,1]$ into $N M \times N M=200 \times 200$ fine elements and use the classical FEM to solve on space and the FDM to solve on time for (6). This approach is the most expensive, and it serves as a reference solution.

A random-parameterized coefficient is combined like

$$
\begin{equation*}
\kappa(\mathbf{x} ; \mu)=\mu_{1} \kappa_{1}(\mathbf{x})+\mu_{2} \kappa_{2}(\mathbf{x})+\mu_{3} \kappa_{3}(\mathbf{x})+\mu_{4} \kappa_{4}(\mathbf{x}) \tag{31}
\end{equation*}
$$

where the parameter $\mu=\left(\mu_{1}, \mu_{2}, \mu_{3}, \mu_{4}\right)$ may be randomly picked from $[0,1]^{4}$,

$$
\begin{aligned}
& \kappa_{1}(\mathbf{x})=10 \cdot\left(\frac{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{1}}\right)}{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{2}}\right)}+\frac{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{1}}\right)}{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{2}}\right)}\right), \\
& \kappa_{2}(\mathbf{x})=10 \cdot\left(\frac{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{3}}\right)}{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{4}}\right)}+\frac{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{3}}\right)}{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{4}}\right)}\right), \\
& \kappa_{3}(\mathbf{x})=10 \cdot\left(\frac{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{5}}\right)}{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{6}}\right)}+\frac{2+1.8 \cos \left(\frac{2 \pi y}{\varepsilon_{5}}\right)}{2+1.8 \sin \left(\frac{2 \pi x}{\varepsilon_{6}}\right)}\right), \\
& \kappa_{4}(\mathbf{x})=100 \cdot\left(4+2.5 \sin \left(\frac{2 \pi \cdot \tanh (20(r-0.3))}{\varepsilon_{7}}\right)\right),
\end{aligned}
$$

and $r=\sqrt{\left(x-\frac{1}{2}\right)^{2}+\left(y-\frac{1}{2}\right)^{2}}$, and set $\varepsilon_{1}=0.2, \varepsilon_{2}=0.08, \varepsilon_{3}=0.125, \varepsilon_{4}=0.0078125$, $\varepsilon_{5}=0.012, \varepsilon_{6}=0.004, \varepsilon_{7}=0.8$. A resulting permeability field expresses the structure with high contrast and quasi-periodicity, see Figure 2. The homogeneous boundary condition $g=0$ and the initial function $u_{0}=0$ are defined for simplicity.


Figure 2. Coefficient $\kappa(\mathbf{x} ; \mu)$.
In the coarse-scale schemes, the MsFEM and the updated GMsFEM for (8) are applied to obtain the coarse-scale outputs. In both schemes, we set the double refinement of the coarse partition $N=5,10,20$ in turn and fix the fine partition $M=10$ in each coarse element. However the latter GMsFEM, with the contributions of eigenvalue computation and enriched multiscale basis functions, would offer more accurate simulations. Finally, we utilize the BT on the reduced scale for (9) to test the behaviors of our GMsFEM-BT approach.

To be more specifical, in the GMsFEM to create the snapshot space first for the offlineonline stages, three equally spaced points in each dimension are set and the eigenfunctions number $L_{i}=10$ is kept in (11), then the number of snapshot functions is $L_{\text {snap }}=3^{j} \times$ $L_{i}=810$ in (12), where $j=4$ is the number of random parameter $\mu_{j}$. In each coarse neighborhood, the online-stage eigenfunctions number $L_{\text {on }}$ in (20) is smaller than the original $L_{\text {snap }}$. Through a further reduction in the BT, after solving the Lyapunov Equations in (23) and (24) for a controllability $W_{\text {con }}$ and observability $W_{\text {obs }}$ on the coarse scale, we use the singular value decompositions to build $U, V$ in (29), whose size is dependent on a smaller number $N_{r}$ on the reduced-scale.

We are quite interested in comparing the simulation behaviors from the fully-resolved fine system, the coarse system and the reduced system. We set the refinement of $N=5$, 10,20 , and the fine partition is kept as $M=10$ in each coarse element, then on both $x, y$ directions in the 2D problem the number of coarse vertices is $N_{c}=(N+1)^{2}=36,121,441$. Then the matrix size of the fine system is $N_{f}=(N M+1)^{2}=2601,10,201,40,401$, while the corresponding matrix size of the generalized coarse system is $207,988,4293$, which is much smaller than $N_{f}$ and is a bit larger than $N_{c}$. In contrast, the matrix size of a reduced system is $N_{r}$, which would be chosen as smaller integers in the following.

Figure 3 shows the discrete errors of MsFEM and GMsFEM, both are compared to the reference solution. Note that the reference is solved from a fully-resolved fine system on $N M \times N M=200 \times 200$, while the solutions of MsFEM and GMsFEM are obtained from their coarse system. It is certified that with the contributions of eigenvalue computation for the dominant eigenfunctions and multiscale enrichments, the GMsFEM performs better than the MsFEM. We list the norm errors from the MsFEM and the GMsFEM with the refinement of the partition number in Table 1 and Figure 4. The error percentages indicate
the fast convergence and stability of GMsFEM, which is much superior to the behavior of MsFEM.


Figure 3. Errors of MsFEM and GMsFEM on coarse partition $N=20$.


Figure 4. Percentages of $L^{2}$ and $H^{1}$ norm errors from MsFEM and GMsFEM on log-log scale.
Table 1. Percentages of norm error from MsFEM and GMsFEM, on different partition numbers $N$.

| Coarse Partition $\boldsymbol{N}$ | Norm | MsFEM Error (\%) | GMsFEM Error (\%) |
| :---: | :---: | :---: | :---: |
| 5 | $L^{2}$ | 7.35 | 0.81 |
|  | $H^{1}$ | 29.35 | 8.97 |
| 10 | $L^{2}$ | 3.00 | 0.31 |
| 20 | $H^{1}$ | 18.39 | 5.44 |
|  | $L^{2}$ | 2.30 | 0.08 |
|  | $H^{1}$ | 15.75 | 2.90 |

In Figure 5, a series of outputs are provided for the reference on the very fine partition $N M=200$, the GMsFEM on the quite coarse partition $N=10$, and the BT on the several reductions $N_{r}$. In Table 2 we observe that on small reductions $N_{r}$ such as 20, the GMsFEMBT does not behave well. However, with the increase of $N_{r}$ it shows the ability to deliver more necessary details of the reference, and its error percentage is less than $0.04 \%$ in the end. Note that the reduced size of GMsFEM-BT on $N_{r}=260$, which is compared to
the generalized coarse size of GMsFEM with $N_{c}=988$ and the fine size of FEM with $N_{f}=40,401$. In the meantime, in fact the GMsFEM-BT works based on the GMsFEM's coarse scale, and it would take some time to solve the Lyapunov equations and the singular value decompositions. However on the contrary, if the GMsFEM-BT works based on the FEM's fully-resolved fine scale, the computational cost would increase inevitably.


Figure 5. Outputs on coarse partition $N=20$ of reference, GMsFEM coarse and GMsFEM-BT with different reduced numbers $N_{r}$, respectively.

Table 2. Percentages of $H^{1}$ norm error from GMsFEM and GMsFEM-BT. on coarse partition number $N=20$.

| GMsFEM Error (\%) | GMsFEM-BT Error on Different $N_{r}(\%)$ |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | $N_{r}=\mathbf{2 0}$ | $N_{r}=\mathbf{1 0 0}$ | $N_{r}=\mathbf{1 8 0}$ | $\mathbf{N}_{r}=\mathbf{2 6 0}$ |
| 2.90 | 2.16 | 1.04 | 0.51 | 0.38 |

Figure 6 shows the error percentages between the fine and coarse, and between the coarse and reduced, respectively. We find that with time goes by, it is reasonable that the error percentages between the reference and the GMsFEM with a coarse partition number $N=20$ (which is fixed during the process) would increase with the time step, but the error remains at a relatively low level overall (less than $0.16 \%$ ). However, it is exciting to observe that the error percentages between the generalized coarse and the reduced $N_{r}=340$ would decrease and remain stable at $0.2 \%$ during the time elapses, which validates the strength of GMsFEM-BT to simulate the real application with costing just an amount of computational resource.

Finally, we provide the CPU time among different methods based on their corresponding matrix sizes. In Table 3 it is apparent that the FEM consumes the maximum time on a fully fine matrix size of $N_{f}=40,401$ to solve for the reference solution. The MsFEM (on a coarse size of $N_{c}=441$ ) and the GMsFEM (on a coarse size of $N_{c}=4293$ ) spend less time to reach their numerical results. However, note that the GMsFEM has to construct the snapshot space (which could be reused) and couple the offline-online stages with the singular value decompositions. This process consumes some time, so three parts of time (for offline, online and solving stages, respectively) are shown in the table for GMsFEM, but it would bring good accuracy. A Lyapunov solver based on the GMsFEM with the generalized coarse size of $N_{c}=4393$ offers its strength. After that, the reduced matrix size
of BT is chosen as the user's demand, and the satisfactory balance both for accuracy and efficiency of the GMsFEM-BT is shown in the numerical results.


Figure 6. With time elapsing, error percentages between fine and coarse $N=20$ (left), and between coarse and reduced number $N_{r}=340$ (right).

Table 3. Comparisons of CPU time.

|  | FEM | MsFEM | GMsFEM | BT |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| size | 40401 | 441 | 4293 | 20 | 100 | 180 | 260 |
| time (s) | 572.78 | 1.48 | $16.59+5.78+7.47$ | 0.0039 | 0.0082 | 0.011 | 0.025 |
| Lyapunov solve |  |  |  |  |  |  |  |
| time (s) | GMsFEM-BT |  |  |  |  |  |  |

## 7. Conclusions

In the paper, a parameter-dependent parabolic model is addressed by a combination strategy of GMsFEM-BT. Neither the FEM works on the fully-resolved fine scale, nor the MsFEM works on the coarse scale, the GMsFEM acts on the updated coarse scale by introducing the necessary eigenvalue computation to create the snapshot space and couple the offline-online stages. In this way, the selected dominant eigenfunctions are plugged into the projection matrix. The balanced truncation is then presented to solve the Lyapunov equations, whose results are served in the reduced matrices for further reduction. The fully discretization for time and space is applied to the parabolic problem. In the numerical experiments, the capabilities of the GMsFEM-BT are demonstrated convincingly. Its simulation is quite close to the reference result, and the error percentages and CPU time of the GMsFEM-BT are encouraging. Our method could be extended to other random parameter cases in the multiscale simulations.

Author Contributions: Conceptualization, S.J.; Methodology, S.J. and Y.C. (Yue Cheng); Validation, Y.C. (Yao Cheng); Formal analysis, Y.C. (Yao Cheng) and Y.H. All authors have read and agreed to the published version of the manuscript.
Funding: Supported by NSFC (11771224) to Shan Jiang, NSFC (11801396) to Yao Cheng, and NSFC (11971410), National Key R\&D Program of China (2020YFA0713500) to Yunqing Huang.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare no conflict of interest.

## References

1. Hou, T.Y.; Wu, X.H. A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. 1997, 134, 169-189. [CrossRef]
2. Chen, Z.M.; Hou, T.Y. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. Math. Comput. 2002, 72, 541-576. [CrossRef]
3. Efendiev, Y.; Galvis, J.; Hou, T.Y. Generalized multiscale finite element methods (GMsFEM). J. Comput. Phys. 2013, 251, 116-135. [CrossRef]
4. Calo, V.M.; Efendiev, Y.; Galvis, J.; Li, G.L. Randomized oversampling for generalized multiscale finite element methods. Multiscale Model. Simul. 2016, 14, 482-501. [CrossRef]
5. Chung, E.; Efendiev, Y.; Hou, T.Y. Adaptive multiscale model reduction with generalized multiscale finite element methods. J. Comput. Phys. 2016, 320, 69-95. [CrossRef]
6. Presho, M.; Hill, M. A conservative generalized multiscale finite volume/element method for modeling two-phase flow with capillary pressure. J. Comput. Appl. Math. 2021, 381, 113026. [CrossRef]
7. Kalachikova, U.; Vasilyeva, M.; Harris, I.; Chung, E. Generalized multiscale finite element method for scattering problem in heterogeneous media. J. Comput. Appl. Math. 2023, 424, 114977. [CrossRef]
8. Nguyen, N.C. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. J. Comput. Phys. 2008, 227, 9807-9822. [CrossRef]
9. Presho, M.; Protasov, A.; Gildin, E. Local-global model reduction of parameter-dependent, single-phase flow models via balanced truncation. J. Comput. Appl. Math. 2014, 271, 163-179. [CrossRef]
10. Jiang, L.J.; Li, Q.Q. Reduced multiscale finite element basis methods for elliptic PDEs with parameterized inputs. J. Comput. Appl. Math. 2016, 301, 101-120. [CrossRef]
11. Jiang, S.; Protasov, A.; Sun, M.L. Balanced truncation based on generalized multiscale finite element method for the parameterdependent elliptic problem. Adv. Appl. Math. Mech. 2018, 10, 1527-1548. [CrossRef]
12. Ammosov, D.A.; Vasilev, V.I.; Vasileva, M.V.; Stepanov, S.P. Multiscale model reduction for a thermoelastic model with phase change using a generalized multiscale finite-element method. Theor. Math. Phys. 2022, 211, 595-610. [CrossRef]
13. Abdulle, A.; Bai, Y. Adaptive reduced basis finite element heterogeneous multiscale method. Comput. Methods Appl. Mech. Eng. 2013, 257, 203-220. [CrossRef]
14. Wang, Z.W.; Chen, M.X.; Chen, J.R. Solving multiscale elliptic problems by sparse radial basis function neural networks. J. Comput. Phys. 2023, 492, 112452. [CrossRef]
15. Henning, P.; Malqvist, A. Localized orthogonal decomposition techniques for boundary value problems. SIAM J. Sci. Comput. 2014, 36, A1609-A1634. [CrossRef]
16. Mu, L.; Wang, J.P.; Ye, X. A weak Galerkin generalized multiscale finite element method. J. Comput. Appl. Math. 2016, 305, 68-81. [CrossRef]
17. Spiridonov, D.; Huang, J.; Vasilyeva, M.; Huang, Y.; Chung, E.T. Mixed generalized multiscale finite element method for Darcy-Forchheimer model. Mathematics 2019, 7, 1212. [CrossRef]
18. Babuska, I.; Lipton, R.; Sinz, P.; Stuebner, M. Multiscale-Spectral GFEM and optimal oversampling. Comput. Methods Appl. Mech. Eng. 2020, 364, 112960. [CrossRef]
19. Nikiforov, D. Meshfree generalized multiscale finite element method. J. Comput. Phys. 2023, 474, 111798. [CrossRef]
20. Dong, H. Computationally efficient higher-order three-scale method for nonlocal gradient elasticity problems of heterogeneous structures with multiple spatial scales. Appl. Math. Model. 2022, 109, 426-454. [CrossRef]
21. Cheng, Y.; Jiang, S.; Stynes, M. Supercloseness of the local discontinuous Galerkin method for a singularly perturbed convectiondiffusion problem. Math. Comput. 2023, 92, 2065-2095. [CrossRef]
22. Reis, T.; Stykel, T. Balanced truncation model reduction of second-order systems. Math. Comput. Model. Dyn. Sys. 2008, 14, 391-406. [CrossRef]
23. Benner, P.; Kurschner, P.; Saak, J. Frequency-limited balanced truncation with low-rank approximations. SIAM J. Sci. Comput. 2016, 38, A471-A499. [CrossRef]
24. Benner, P.; Werner, W.R. Frequency- and time-limited balanced truncation for large-scale second-order systems. Linear Algebra Appl. 2021, 623, 68-103. [CrossRef]
25. Jiang, S.; Presho, M.; Huang, Y.Q. An adapted Petrov-Galerkin multiscale finite element for singularly perturbed reaction-diffusion problems. Int. J. Comput. Math. 2016, 93, 1200-1211. [CrossRef]
26. Jiang, S.; Sun, M.L.; Yang, Y. Reduced multiscale computation on adapted grid for the convection-diffusion Robin problem. J. Appl. Anal. Comput. 2017, 7, 1488-1502.
27. Jiang, S.; Ding, X.; Sun, M.L. Parameter-uniform superconvergence of multiscale computation for singular perturbation exhibiting twin boundary layers. J. Appl. Anal. Comput. 2023, 13, 3330-3351. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

