Journal of Computational Physics 444 (2021) 110572

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Wavelet-based edge multiscale parareal algorithm for parabolic equations with heterogeneous coefficients and rough initial data

Guanglian Li [a,*,1], Jiuhua Hu [b]

[a] Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong
[b] Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA

ARTICLE INFO

Article history:
Available online 19 July 2021

Keywords:
Multiscale
Heterogeneous
Wavelets
Parareal
Rough initial data
Parabolic

ABSTRACT

We propose in this paper the Wavelet-based Edge Multiscale Parareal (WEMP) Algorithm to solve parabolic equations with heterogeneous coefficients efficiently. This algorithm combines the advantages of multiscale methods that can deal with heterogeneity in the spatial domain effectively, and the strength of parareal algorithms for speeding up time evolution problems when sufficient processors are available. We derive the convergence rate of this algorithm in terms of the mesh size in the spatial domain, the level parameter used in the multiscale method, the coarse-scale time step and the fine-scale time step. Extensive numerical tests are presented to demonstrate the performance of our algorithm, which verify our theoretical results perfectly.

1. Introduction

We consider in this paper a new efficient multiscale parareal algorithm for parabolic problems with heterogeneous coefficients. We first formulate the heterogeneous parabolic problems to present our new multiscale methods. Let $D \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ be an open bounded Lipschitz domain. We seek a function $u(\cdot, t) \in V := H_0^1(D)$ such that

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (\kappa \nabla u) &= f \qquad \text{in } D \times (0, T] \\ u(\cdot, 0) &= u_0 \qquad \text{in } D \\ u &= 0 \qquad \text{on } \partial D \times [0, T], \end{aligned} \tag{1.1}$$

where the force term $f \in L^\infty([0, T]; \dot{H}^2(D))$ satisfying $\partial_t f \in L^1([0, T]; L^2(D))$, the initial data $u_0 \in L^2(D)$ and the permeability coefficient $\kappa \in C^\infty(D)$ with $\alpha \le \kappa(x) \le \beta$ almost everywhere for some lower bound $\alpha > 0$ and upper bound $\beta > \alpha$. Here, $\dot{H}^s(D) \subset L^2(D)$ is a Hilbert space to be defined in (2.1). We denote by $\Lambda := \frac{\beta}{\alpha}$ the ratio of these bounds, which reflects the contrast of the coefficient κ. To simplify the notation, let $I := [0, T]$. Note that the existence of multiple scales in the coefficient κ rends directly solving Problem (1.1) challenging, since resolving the problem to the finest scale would incur huge computational cost.

* Corresponding author.
E-mail addresses: lotusli0707@gmail.com, lotusli@hku.hk (G. Li), jiuhuahu115@gmail.com (J. Hu).

[1] GL acknowledges the support from the Royal Society (London, United Kingdom, grant number AL 201006) through a Newton Fellowship Alumni follow on funding.

https://doi.org/10.1016/j.jcp.2021.110572
0021-9991/

The accurate description of many important applications, e.g., composite materials, porous media and reservoir simulation, involves mathematical models with heterogeneous coefficients. In order to adequately describe the intrinsic complex properties in practical scenarios, the heterogeneous coefficients can have both multiple inseparable scales and high-contrast. Due to this disparity of scales, the classical numerical treatment becomes prohibitively expensive and even intractable for many multiscale applications. Nonetheless, motivated by the broad spectrum of practical applications, a large number of multiscale model reduction techniques, e.g., multiscale finite element methods (MsFEMs), heterogeneous multiscale methods (HMMs), variational multiscale methods, flux norm approach, generalized multiscale finite element methods (GMsFEMs) and localized orthogonal decomposition (LOD), have been proposed in the literature [5,8–10,20,21,24,31] over the last few decades. They have achieved great success in the efficient and accurate simulation of heterogeneous problems. Recently, a so-called Wavelet-based Edge Multiscale Finite Element Method (WEMsFEM), cf. Algorithm 1, was proposed within the framework of GMsFEMs [10] that facilitates deriving a rigorous convergence rate with merely mild assumptions [15,16,23]. A key ingredient in GMsFEMs is designing local multiscale basis function in local regions with good approximation properties. However, deriving such local approximation is nontrivial. The main idea of WEMsFEM is to utilize wavelets as the basis functions over the coarse edges, and transform the approximation properties from the edges to each local region. Then the Partition of Unity Method (PUM) [32] is applied to derive the global convergence rate. The motivation for using wavelets as the ansatz space over the coarse edges origins from the low regularity of the solution to (1.1) in the spatial domain D due to the existence of heterogeneity in the coefficient κ, which makes its approximation use the standard basis functions, e.g., the element-wise polynomials, infeasible or even prohibitive. Further, the multiresolution analysis enables the approximation of functions with low regularities using wavelets. We will apply this method in this paper to handle the heterogeneity in the spatial domain D.

Furthermore, motivated by the great demand for an efficient solver with high accuracy as well as a reasonable wall-clock time in many practical applications, e.g., financial mathematics [4], fluid mechanics and fluid-structure interaction [12–14], oceanography [26], chemistry [28,6] and quantum chemistry [29], and the increasing computational capacity of current computers, a variety of efficient numerical schemes exploiting parallel computing architectures emerge during the last few decades. Among them, the parareal algorithm is one of the most popular and successful algorithms. The parareal algorithm facilitates speeding up the numerical solver to time dependent equations on the condition of sufficient processors [3], which is an iterative solver based on a cheap inaccurate sequential coarse-scale time solver and expensive accurate fine-scale time solvers that can be performed in parallel. If it converges sufficiently fast, then the parareal algorithm could result in less wall-clock time than sequentially computing. The parareal algorithm was introduced by Lions, Maday and Turinici [25] and extended to the linear problems in [2,4,30]. Its convergence for nonlinear system of ordinary differential equations and partial differential equations is derived in [4,17]. Convergence properties are investigated for three fine propagators in [34]: the trapezoidal rule, the third-order diagonal implicit Runge-Kutta method, and the fourth-order Gauss Runge-Kutta method. A critical condition was derived, which guarantees the fast convergence of the parareal algorithm using these aforementioned fine propagators. Recently, new parareal algorithms are developed to solve problems involving discontinuous right-hand sides [18,19]. Coupling of parareal algorithm and other techniques has been developed in many literatures, see [1,7,11,22]. Among them is the coupling of parareal algorithm with the model reduction techniques. In [22], a micro-macro parareal algorithm for the time-parallel integration of multiscale-in-time systems is introduced to solve singularly perturbed ordinary differential equations. One contribution of this paper is that the fast variables are eliminated from the coarse propagator, therefore, the resulting algorithm only evolves with the slow variables. A new coupling strategy to compute high oscillatory solutions to a class of ODEs is introduced in [1], where multiscale integrators are coupled with fully resolved fine scale integrators for parallel in time.

In this paper, we incorporate the parareal algorithm into WEMsFEM to numerically calculate the time evolution problems efficiently. This new algorithm is called WEMP Algorithm, cf. Algorithm 2. This algorithm is divided into two steps: a multiscale space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ based on WEMsFEM with ℓ as the wavelets level parameter is constructed in the first step, and then we apply the parareal algorithm by using $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ as the ansatz space in the second step to obtain the solution more efficiently. The convergence analysis of this algorithm is presented in Theorem 4.1. We proved

$$\begin{aligned}\left\|u(\cdot,T^n)-U_k^n\right\|_{L^2(D)} \lesssim & \left(H\left\|\widetilde{\kappa}\right\|_{L^\infty(D)}^{1/2}+2^{-\ell/2}\|\kappa\|_{L^\infty(\mathcal{F}_H)}+\delta t\right)\frac{1}{T^n}\|u_0\|_{L^2(D)}\\ &+\left(H\left\|\widetilde{\kappa}\right\|_{L^\infty(D)}^{1/2}+2^{-\ell/2}\|\kappa\|_{L^\infty(\mathcal{F}_H)}\right)T^n\sup_{s\le T^n}|f(\cdot,s)|_2\\ &+\delta t\left(T^n\sup_{s\le T^n}|f(\cdot,s)|_2+\int_0^{T^n}\|\partial_s f(\cdot,s)\|_{L^2(D)}\mathrm{d}s\right)\\ &+\left(\Pi_{j=0}^k\frac{1}{T^{n-j}}\right)\Delta T^{k+1}\|u_0\|_{L^2(D)},\end{aligned}$$

where $u(\cdot,T^n)$ and U_k^n are the exact solution and numerical solution derived from WEMP algorithm at $T^n=n\times\Delta T$ for $n=2,\cdots$. The notations ΔT and δt represent the coarse time step size and fine time step size, respectively. H, ℓ and k are the spatial domain mesh size, the level parameter and iteration number. $\mathcal{F}_H$ denotes the collection of all edges in the

coarse mesh $\mathcal{T}_H$. We refer to Section 2 for more details. $\widetilde{\kappa}$ is a weighted coefficient to be defined in (2.8). This implies that taking $\ell = [-2\log_2 H]$ and $k = [-\log_{\Delta T}\delta t]$, we recover $\mathcal{O}(H+\delta t)$ error, which actually is the error for the backward Euler conforming Galerkin method. Here, [.] denotes taking integer part of a number. Furthermore, the singularity of the solution for $t \to 0$ is reflected in the coefficient of the last term, namely, $\left(\Pi_{j=0}^{k}\frac{1}{T^{n-j}}\right)$. Note that similar convergence of parareal algorithm was derived for parabolic equations under the condition that $n \gg k$ in [4]. Our result can be applied to small time step n.

To demonstrate the performance of our proposed algorithm we present several numerical tests using backward Euler and Crank-Nicolson schemes for the fine time step solver, respectively. Our numerical tests indicate similar convergence as derived in the theoretical results. Furthermore, we take different coarse time steps and observe similar convergence behavior.

The paper is organized as follows. We summarize the basics on the fully discretization of Problem (1.1), the framework of WEMsFEMs in Section 2. Our main proposed algorithm is presented in Section 3. The convergence of WEMsFEM and WEMP algorithms are derived in Section 4. Extensive numerical tests are presented in Section 5. Finally, we complete our paper with concluding remarks in Section 6.

2. Problem setting and the construction of multiscale space

In this section, we will mainly introduce the full discretization of problem (1.1), and its multiscale model reduction in the spatial domain D.

2.1. Full discretization

We present in this subsection the discretization of problem (1.1). Firstly, we define the Hilbert space $\dot{H}^s(D)$, which is analogous to [33, Chapter 3].

Let $\{(\lambda_m, \phi_m)\}_{m=1}^{\infty}$ be the eigenpairs of the following eigenvalue problems with the eigenvalues arranged in a nondecreasing order,

$$\begin{aligned} \mathcal{L}\phi_m := -\nabla\cdot(\kappa\nabla\phi_m) &= \lambda_m\phi_m \qquad \text{in } D \\ \phi_m &= 0 \qquad \text{on } \partial D. \end{aligned}$$

Note that the eigenfunctions $\{\phi_m\}_{m=1}^{\infty}$ form an orthonormal basis in $L^2(D)$, and consequently, each $v \in L^2(D)$ admits the representation $v = \sum_{m=1}^{\infty}(v,\phi_m)_D\phi_m$ with $(\cdot,\cdot)_D$ being the inner product in $L^2(D)$. The Hilbert space $\dot{H}^s(D) \subset L^2(D)$ is defined by

$$\dot{H}^s(D) = \{v \in L^2(D) : \sum_{m=1}^{\infty}\lambda_m^s|(v,\phi_m)_D|^2 < \infty\}. \tag{2.1}$$

The associated norm in $\dot{H}^s(D)$ is $|v|_s = (\sum_{m=1}^{\infty}\lambda_m^s|(v,\phi_m)_D|^2)^{1/2}$.

Remark 2.1. Since the initial data $u_0 \in \dot{H}^3(D)\cap H_1^0(D)$, we obtain

$$\|\mathcal{L}u_0\|_{L^2(D)} = |u_0|_2. \tag{2.2}$$

Indeed, u_0 allows the expression

$$u_0 = \sum_{m=1}^{\infty}(u_0,\phi_m)_D\phi_m.$$

Taking $L^2(D)$-norm after operating $\mathcal{L}$ on both sides and utilize the definition (2.1), we obtain the desired assertion (2.2).

To discretize problem (1.1), we first introduce fine and coarse grids. Let $\mathcal{T}_H$ be a regular partition of the domain D into finite elements (triangles, quadrilaterals, tetrahedral, etc.) with a mesh size H. We refer to this partition as coarse grids, and its elements as the coarse elements. Then each coarse element is further partitioned into a union of connected fine grid blocks. The fine-grid partition is denoted by $\mathcal{T}_h$ with h being its mesh size. Let $\mathcal{F}_h$ (or $\mathcal{F}_H$) be the collection of all edges in $\mathcal{T}_h$ (or $\mathcal{T}_H$). Over the fine mesh $\mathcal{T}_h$, let V_h be the conforming piecewise linear finite element space:

$$V_h := \{v \in V : v|_E \in \mathcal{P}_1(E) \text{ for all } E \in \mathcal{T}_h\},$$

where $\mathcal{P}_1(E)$ denotes the space of linear polynomials on the fine element $E \in \mathcal{T}_h$.

The time interval $I := [0,T]$ is decomposed into a sequence of coarse subintervals $[T^n, T^{n+1}]$ for $n = 0, 1, \cdots, M_\Delta$ of size ΔT with $\Delta T := T/M_\Delta$ for some $M_\Delta \in \mathbb{N}_+$ and $T^0 := 0$. Each coarse time interval $[T^n, T^{n+1}]$ is further discretized

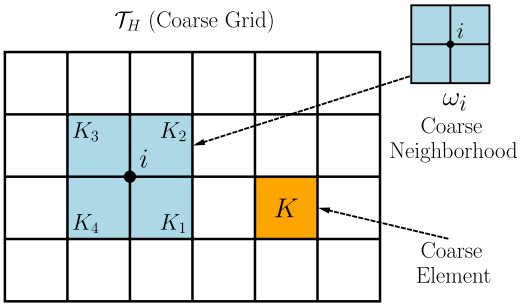

Fig. 1. Illustration of a coarse neighborhood and coarse element.

with a fine time step δt. Let $t_n = n \times \delta t$ for $n = 0, 1, \cdots, M_\delta$ with $M_\delta := T \times \delta t^{-1}$. Note that $\Delta T \gg \delta t$. To simplify the notations, backward Euler method is utilized to discretize the time variable, and we use conforming Galerkin method for the discretization in the spatial variable throughout this paper. Then the fine-scale solution $U_h^n \in V_h$ for $n = 1, 2, \cdots, M_\delta$ satisfies

$$\begin{cases} (\dfrac{U_h^n - U_h^{n-1}}{\delta t}, v_h)_D + a(U_h^n, v_h) = (f(\cdot, t_n), v_h)_D \quad \text{for all } v_h \in V_h, \\ U_h^0 = I_h u_0. \end{cases} \tag{2.3}$$

Here, the bilinear form $a(\cdot, \cdot)$ on $V \times V$ is defined by

$$a(v_1, v_2) := \int_D \kappa \nabla v_1 \cdot \nabla v_2 \, dx \text{ for all } v_1, v_2 \in V.$$

I_h is a proper projection from V to V_h. Furthermore, we define the energy norm $\|v\|_{H^1_\kappa(D)} := \sqrt{a(v, v)}$ for all $v \in V$.

The fine-scale solution u_h^n will serve as a reference solution in Section 5. Note that due to the presence of multiple scales in the coefficient κ, the fine-scale mesh size h should be commensurate with the smallest scale and thus it can be very small in order to obtain an accurate solution. This necessarily involves huge computational complexity, and more efficient methods are in great demand.

2.2. Multiscale model reduction in the spatial domain D

We present in the section the multiscale model reduction to Problem (2.3) in the spatial domain D.

2.2.1. Multiscale solver in the spatial domain

The multiscale method we are investigating aim at solving Problem (1.1) on the coarse mesh $\mathcal{T}_H$, which, meanwhile, maintains a certain accuracy compared to the fine-scale solution U_h^n to Problem (2.3). To provide a brief overview, we first recap a few definitions.

The vertices of $\mathcal{T}_H$ are denoted by $\{O_i\}_{i=1}^N$, with N being the total number of coarse nodes. The coarse neighborhood associated with the node O_i is denoted by

$$\omega_i := \bigcup \{K_j \in \mathcal{T}_H : \quad O_i \in \overline{K}_j\}. \tag{2.4}$$

We refer to Fig. 1 for an illustration of neighborhoods and elements subordinated to the coarse discretization $\mathcal{T}_H$. Throughout, we use ω_i to denote a coarse neighborhood. Furthermore, let $\mathcal{F}_h(\partial\omega_i)$ (or $\mathcal{F}_H(\partial\omega_i)$) be the restriction of $\mathcal{F}_h$ on $\partial\omega_i$ (or $\mathcal{F}_H$ on $\partial\omega_i$).

Let V_{ms} be the multiscale finite element space to be defined in Section 2.2.2. The multiscale solution $U_{\text{ms}}^n \in V_{\text{ms}}$ for $n = 1, \cdots, M_\delta$ satisfies

$$\begin{cases} (\dfrac{U_{\text{ms}}^n - U_{\text{ms}}^{n-1}}{\delta t}, v_{\text{ms}})_D + a(U_{\text{ms}}^n, v_{\text{ms}}) = (f(\cdot, t_n), v_{\text{ms}})_D \quad \text{for all } v_{\text{ms}} \in V_{\text{ms}}, \\ U_{\text{ms}}^0 = I_{\text{ms}} u_0, \end{cases} \tag{2.5}$$

where V_{ms} denotes the multiscale space spanned by these multiscale basis functions and I_{ms} is a $L^2(D)$-projection operator from V to V_{ms}.

Note that we need a very tiny fine-scale time step δt to guarantee a reasonable approximation property of u_{ms}^n to $u(\cdot, t_n)$ for $n = 1, \cdots, M_\delta$ due to, e.g., the singularity of the solution $u(\cdot, t)$ at $t = 0$ arising from the rough initial data u_0 or when

the source term f fails to have certain regularity. Consequently, the computational complexity of the multiscale method (2.5) can be extremely expensive. For this reason, we present in Section 3 a multiscale algorithm incorporated with the parareal algorithm to reduce further this part of computational cost.

We end this section with assumptions on the permeability field κ, which is required to obtain approximation properties of the multiscale finite element space V_{ms} in the energy norm, cf., (4.1):

Assumption 2.1 *(Structure of D and κ).* Let D be a domain with a $C^{1,\alpha}$ $(0<\alpha<1)$ boundary ∂D, and $\{D_i\}_{i=1}^m \subset D$ be m pairwise disjoint strictly convex open subsets, each with a $C^{1,\alpha}$ boundary $\Gamma_i := \partial D_i$, and denote $D_0 = D\backslash\overline{\cup_{i=1}^m D_i}$. Let the permeability coefficient κ be piecewise regular function defined by

$$\kappa = \begin{cases} \eta_i(x) & \text{in } D_i, \\ 1 & \text{in } D_0. \end{cases} \tag{2.6}$$

Here $\eta_i \in C^{\mu}(\bar{D}_i)$ with $\mu \in (0,1)$ for $i = 1, \cdots, m$. Denote $\eta_{\min} := \min_i\{\min_{x\in D_i}\{\eta_i(x)\}\} \geq 1$ and $\eta_{\max} := \max_i\{\|\eta_i\|_{C_0(D_i)}\}$.

2.2.2. Multiscale space construction

This subsection is concerned with the construction of the multiscale space by means of the Wavelet-based Edge Multiscale Finite Element Methods (WEMsFEM) [23,16]. The algorithm is presented in Algorithm 1. Given the level parameter $\ell \in \mathbb{N}$, and the type of wavelets on each edge of the coarse neighborhood ω_i, one can obtain the local multiscale space $V_{i,\ell}$ on ω_i by solving $2^{\ell+2}$ local problems in Step 2 using the fine-scale mesh and its associated proper finite element space. Those local problems $\mathcal{L}_i$ are homogeneous elliptic operators coupled with wavelets Dirichlet data $V_{i,\ell}$. In Step 3, we can use these local multiscale space to build the global multiscale space $V_{\text{ms},\ell}^{\text{EW}}$ by multiplying the partition of unity functions χ_i. Finally, we can solve (2.5) by backward Euler conforming Galerkin scheme using this global multiscale space, coupled with $I_\ell u_0$ as the initial condition. Here, I_ℓ denotes the $L^2(D)$-projection from $L^2(D)$ to $V_{\text{ms},\ell}^{\text{EW}}$.

The weighted coefficient appears in Step 3 is

$$\widetilde{\kappa} = H^2\kappa\sum_{i=1}^{N}|\nabla\chi_i|^2. \tag{2.7}$$

Further, its inverse $\widetilde{\kappa}^{-1}$ is

$$\widetilde{\kappa}^{-1}(x) := \begin{cases} \widetilde{\kappa}^{-1}, & \text{when } \widetilde{\kappa}(x) \neq 0, \\ 1, & \text{otherwise}, \end{cases} \tag{2.8}$$

which will be utilized in the analysis. The partition of unity functions χ_i used in Step 4 are the standard multiscale basis functions defined coarse elementwise. On each coarse element $K \in \mathcal{T}_H$, it satisfies

$$\begin{aligned} -\nabla\cdot(\kappa(x)\nabla\chi_i) &= 0 \quad \text{in } K, \\ \chi_i &= g_i \quad \text{on } \partial K, \end{aligned} \tag{2.9}$$

where g_i is affine over ∂K with $g_i(O_j) = \delta_{ij}$ for all $i, j = 1, \cdots, N$. Recall that $\{O_j\}_{j=1}^N$ are the set of coarse nodes on $\mathcal{T}_H$. By its definition, χ_i is locally supported,

$$\text{supp}(\chi_i) \subset \omega_i.$$

Algorithm 1 Wavelet-based Edge Multiscale Finte Element Method (WEMsFEM).

Input: The level parameter $\ell \in \mathbb{N}$; coarse neighborhood ω_i and its four coarse edges $\Gamma_{i,k}$ with $k = 1,2,3,4$, i.e., $\cup_{k=1}^4\Gamma_{i,k} = \partial\omega_i$; the subspace $V_{\ell,k}^i \subset L^2(\Gamma_{i,k})$ up to level ℓ on each coarse edge $\Gamma_{i,k}$.
Output: Multiscale solution $u_{\text{ms},\ell}^{\text{EW}}$.

1: Denote $V_{i,\ell} := \oplus_{k=1}^4 V_{\ell,k}^i$. Then the number of basis functions in $V_{i,\ell}$ is $4\times 2^\ell = 2^{\ell+2}$. Denote these basis functions as v_k for $k = 1, \cdots, 2^{\ell+2}$.

2: Calculate local multiscale basis $\mathcal{L}_i^{-1}(v_k)$ for all $k = 1, \cdots, 2^{\ell+2}$. Here, $\mathcal{L}_i^{-1}(v_k) := v$ satisfies: $\begin{cases} \mathcal{L}_i v := -\nabla\cdot(\kappa\nabla v) = 0 & \text{in } \omega_i, \\ v = v_k & \text{on } \partial\omega_i. \end{cases}$

3: Solve one local problem.

$$\begin{cases} -\nabla\cdot(\kappa\nabla v^i) = \dfrac{\widetilde{\kappa}}{\int_{\omega_i}\widetilde{\kappa}\,\mathrm{d}x} & \text{in } \omega_i, \\ -\kappa\dfrac{\partial v^i}{\partial n} = |\partial\omega_i|^{-1} & \text{on } \partial\omega_i. \end{cases}$$

4: Build global multiscale space. $V_{\text{ms},\ell}^{\text{EW}} := \text{span}\{\chi_i\mathcal{L}_i^{-1}(v_k), \chi_i v^i :\ 1\leq i\leq N,\ 1\leq k\leq 2^{\ell+2}\}$.

5: Solve for (2.5) by backward Euler conforming Galerkin method in $V_{\text{ms},\ell}^{\text{EW}}$ with $U_{\text{ms},\ell}^{\text{EW},1} = I_\ell u_0$ to obtain $U_{\text{ms},\ell}^{\text{EW},n}$ for $n = 1, \cdots, M_\delta$.

3. Wavelet-based edge multiscale parareal algorithm

We construct in this section the Wavelet-based Edge Multiscale Parareal (WEMP) Algorithm, cf. Algorithm 2, which is divided into two main steps. In the first step, the multiscale space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ for $\ell \in \mathbb{N}_+$ is built based on Section 2.2.2. This multiscale space serves as the trial space and test space for our conforming Galerkin method, cf. (2.5). Then the parareal algorithm is utilized in the second step to solve the problem.

We first recap a few terminologies commonly appeared in parareal algorithm.

The one step coarse solver on the time domain $(0, T)$ is

$$\begin{aligned} U^{n+1} &= E_\Delta(T^n, U^n), \quad U^0 = I_h u_0, \\ U_{\mathrm{ms},\ell}^{n+1} &= E_\Delta^{\mathrm{ms},\ell}(T^n, U_{\mathrm{ms},\ell}^n), \quad U_{\mathrm{ms},\ell}^0 = I_{\mathrm{ms}} u_0, \end{aligned} \tag{3.1}$$

which yields U^{n+1} (or $U_{\mathrm{ms},\ell}^{n+1}$) as a coarse approximation to $u(\cdot, T^{n+1})$, provided with an approximation U^n (or $U_{\mathrm{ms},\ell}^n$) of $u(\cdot, T^n)$. In matrix form, it reads

$$\begin{aligned} U^{n+1} &= (M + \Delta T \times A)^{-1} M (U^n + \Delta T \times F^{n+1}), \\ U_{\mathrm{ms},\ell}^{n+1} &= \Phi_{\mathrm{ms},\ell} (\Phi_{\mathrm{ms},\ell}^T M \Phi_{\mathrm{ms},\ell} + \Delta T \times \Phi_{\mathrm{ms},\ell}^T A \Phi_{\mathrm{ms},\ell})^{-1} \Phi_{\mathrm{ms},\ell}^T M (U_{\mathrm{ms},\ell}^n + \Delta T \times F^{n+1}). \end{aligned}$$

Here, A and M are the mass matrices and stiffness matrices corresponding to the discretization of the elliptic operator $-\nabla \cdot (\kappa \nabla \cdot)$ in the finite element space $V_h := \mathrm{span}\{\phi_1, \cdots, \phi_{\mathrm{dof}_h}\}$. Here, dof_h denotes the dimension of V_h. $(F^{n+1})_i := \int_D f(\cdot, t_{n+1}) \phi_i \, \mathrm{d}x$ for all $i = 1, \cdots, \mathrm{dof}_h$. $\Phi_{\mathrm{ms},\ell}$ denotes a matrix with columns composed of the coefficients of multiscale basis functions in $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ in the finite element space V_h.

The one step fine solver

$$\begin{aligned} \psi &= \mathcal{F}_\delta(s, \sigma, \phi), \\ \psi_{\mathrm{ms},\ell} &= \mathcal{F}_\delta^{\mathrm{ms},\ell}(s, \sigma, \phi), \end{aligned} \tag{3.2}$$

yields an approximation $\psi(\cdot, s+\sigma)$ (or $\psi_{\mathrm{ms},\ell}(\cdot, s+\sigma)$) to the solution $u(\cdot, s+\sigma)$ with the initial condition $\psi(\cdot, s) = \phi$ (or $\psi_{\mathrm{ms},\ell}(\cdot, s) = \mathcal{P}_\ell(\phi)$) and a uniform discrete time step δ for all $s \in (0, T)$ and $\sigma \in (0, T-s)$ in the infinite dimensional space V (or in the ansatz space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$) with $s/\delta t \in \mathbb{N}_+$.

We also define the semi-discretization in space solver

$$u_{\mathrm{ms},\ell}(\cdot, s+\sigma) = \mathcal{F}^{\mathrm{ms},\ell}(s, \sigma, \phi), \tag{3.3}$$

which yields an approximation $u_{\mathrm{ms},\ell}(\cdot, s+\sigma)$ to the solution $u(\cdot, s+\sigma)$ with initial condition $u_{\mathrm{ms},\ell}(\cdot, s) = \mathcal{P}_\ell(\phi)$ for all $s \in (0, T)$, $\sigma \in (0, T-s)$ in the ansatz space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$. We will denote $\bar{E}_\Delta^{\mathrm{ms},\ell}(T^n, U_{\mathrm{ms},\ell}^n)$ as the one step coarse solver with $f = 0$. We will define $\bar{\mathcal{F}}^{\mathrm{ms},\ell}(s, \sigma, \phi)$ and $\bar{\mathcal{F}}_\delta^{\mathrm{ms},\ell}(s, \sigma, \phi)$ analogously.

Note that the cheap multiscale coarse solver $E_\Delta^{\mathrm{ms},\ell}$ is sequentially utilized over the global time interval I to provide a rough approximation to $u(\cdot, T^{n+1})$, while the expensive accurate multiscale fine solver $\mathcal{F}_\delta^{\mathrm{ms},\ell}$ is applied in each subinterval $[T^n, T^{n+1}]$ for $n = 0, 1, \cdots, M_\Delta - 1$ independently. This local fine solver will embed more detailed information to the approximation of $u(\cdot, T^{n+1})$, which usually differs from the one obtained from the global coarse solver. In the process of parareal algorithm, a correction operator is very important to improve the approximation to $u(\cdot, T^{n+1})$ based on the discrepancy between the coarse solver and fine solver, which is defined by

$$\mathcal{S}(T^n, U_{\mathrm{ms},\ell}^n) := \mathcal{F}_\delta^{\mathrm{ms},\ell}(T^n, \Delta T, U_{\mathrm{ms},\ell}^n) - E_\Delta^{\mathrm{ms},\ell}(T^n, U_{\mathrm{ms},\ell}^n) \quad \text{and} \quad U_{\mathrm{ms},\ell}^0 = I_{\mathrm{ms}} u_0$$

for all $n = 0, 1, \cdots, M_\Delta - 1$.

Now we are ready to present our main algorithm, i.e., Algorithm 2. To obtain a good approximation to the solution of (1.1) at discrete time points $\{T^n\}$ for $n = 1, \cdots, M_\Delta$, we first construct a proper multiscale space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ based on the WEMsFEM, i.e., Algorithm 1, which corresponds to Steps 1 to 3. This allows one to solve (2.5) using the constructed multiscale space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ and obtain an intermediate solution $U_{\mathrm{ms},\ell}^{\mathrm{EW},n}$ with certain accuracy depending on the spatial coarse mesh size H and level parameter ℓ. This solution will only be utilized in the convergence analysis.

In order to further reduce the computational cost, we apply the parareal algorithm in the following. Given the iteration parameter k, we apply the global coarse solver (3.1) in Step 6 to obtain U_k^{n+1}, which is an approximation to the intermediate solution $U_{\mathrm{ms},\ell}^{\mathrm{EW},n+1}$ from Algorithm 1. Using the coarse solution U_k^n as the initial condition, the fine solver (3.2) subsequently is used to calculate the fine solution U_k^{n+1} in parallel on each local time subinterval $[T^n, T^{n+1}]$. Then we calculate the discrepancy between the coarse solution and the fine solution in Step 8 on each discrete coarse time point T^n for $n = 1, 2, \cdots, M_\Delta$, and denote it as $\mathcal{S}(T^{n-1}, U_k^{n-1})$. Subsequently, this jump term is utilized in Step 9 to update the coarse solution via the global coarse solver (3.1). This process will be performed iteratively until certain tolerance on the jump terms is satisfied.

The multiscale solution in Algorithm 1 is calculated using a fine time step $\delta t \ll \Delta T$, which solves a linear system of size N_{ms} a number of $\frac{T}{\delta t}$ times, where N_{ms} denotes the dimension of the multiscale space $V_{\text{ms},\ell}^{\text{EW}}$. In each iteration, a linear system of size N_{ms} is solved a number of $\frac{T}{\Delta T}$ times sequentially and a number of $\frac{\Delta T}{\delta t}$ times simultaneously using $\frac{T}{\Delta T}$ processors. In total, Algorithm 2 involves solving a linear system of size N_{ms} a number of $k \times \frac{T}{\Delta T}$ times sequentially after k iteration. Algorithm 2 converges within a shorter wall-clock time compared with Algorithm 1 when $k \ll \frac{T}{\Delta T}$, which, indeed, can be supported by extensive numerical experiments presented in Section 5.

Algorithm 2 Wavelet-based Edge Multiscale Parareal (WEMP) algorithm.

Input: The initial data u_0, the source term f; tolerance ϵ; the level parameter $\ell \in \mathbb{N}$; coarse neighborhood ω_i and its four coarse edges $\Gamma_{i,j}$ with $j = 1, 2, 3, 4$, i.e., $\cup_{j=1}^{4} \Gamma_{i,j} = \partial\omega_i$; the subspace $V_{\ell,j}^{i} \subset L^2(\Gamma_{i,j})$ up to level ℓ on each coarse edge $\Gamma_{i,j}$.
Output: U.

1: Denote $V_{i,\ell} := \oplus_{k=1}^{4} V_{\ell,k}^{i}$. Then the number of basis functions in $V_{i,\ell}$ is $4 \times 2^{\ell} = 2^{\ell+2}$. Denote these basis functions as v_k for $k = 1, \cdots, 2^{\ell+2}$.

2: Calculate local multiscale basis $\mathcal{L}_i^{-1}(v_m)$ for all $m = 1, \cdots, 2^{\ell+2}$. Here, $\mathcal{L}_i^{-1}(v_m) := v$ satisfies: $\begin{cases} \mathcal{L}_i v := -\nabla \cdot (\kappa \nabla v) = 0 & \text{in } \omega_i, \\ v = v_m & \text{on } \partial\omega_i. \end{cases}$

3: Build global multiscale space. $V_{\text{ms},\ell}^{\text{EW}} := \text{span}\{\chi_i \mathcal{L}_i^{-1}(v_k), \chi_i v^i : \ 1 \le i \le N, \ \ 1 \le k \le 2^{\ell+2}\}$.

4: $k = 0$, err$= 1$.

5: **while** err$> \epsilon$ **do**

6: Compute U_k^{n+1} for $n = 0, \cdots, M_\Delta - 1$:

$$U_k^{n+1} = E_\Delta^{\text{ms},\ell}(T^n, U_k^n),$$
$$U_k^0 = \mathcal{P}_\ell u_0.$$

7: Compute u_k^{n+1} for $n = 0, \cdots, M_\Delta - 1$ on each local time subinterval $[T^n, T^{n+1}]$:

$$u_k^{n+1} = \mathcal{F}_\delta^{\text{ms},\ell}(T^n, \Delta T, U_k^n).$$

8: Compute the jumps for $n = 1, \cdots, M_\Delta$:

$$\mathcal{S}(T^{n-1}, U_k^{n-1}) := u_k^n - U_k^n.$$

9: Compute the corrected coarse solutions U_{k+1}^{n+1} for $n = 0, \cdots, M_\Delta - 1$:

$$U_{k+1}^{n+1} = \mathcal{S}(T^n, U_k^n) + E_\Delta^{\text{ms},\ell}(T^n, U_{k+1}^n),$$
$$U_{k+1}^0 = \mathcal{P}_\ell u_0.$$

10: Calculate:

$$\text{err} := 1/M_\Delta \sum_{n=1}^{M_\Delta} \|U_{k+1}^n - U_k^n\|_{\ell_2}.$$

$k \leftarrow k + 1$

11: **end while**

12: $U_n := U_k^n$ and $U := [U_0, \cdots, U_{M_\Delta}]$.

4. Convergence study

This section is concerned with the theoretical study of Algorithm 1 and Algorithm 2. The proof of the former follows from [33, Theorems 7.7 and 8.5], where the approximation properties of the multiscale space $V_{\text{ms},\ell}^{\text{EW}}$ and the convergence rate of the associated solver $\mathcal{L}_\ell^{-1}$ are needed. The error of the latter can be decomposed as the summation of the error from the multiscale space and the parareal error.

4.1. Convergence for Algorithm 1

We first derive in the section the properties of the numerical operator that approximate the differential operator in Algorithm 1, then present the approximation properties of this numerical operator in the multiscale space $V_{\text{ms},\ell}^{\text{EW}}$.

Let $\mathcal{L} := -\nabla \cdot (\kappa \nabla \cdot)$ be the elliptic operator defined on V, and let its discrete operator $\mathcal{L}_\ell : V_{\text{ms},\ell}^{\text{EW}} \to L^2(D)$ be

$$(\mathcal{L}_\ell w_\ell, v_\ell) := (\mathcal{L} w_\ell, v_\ell) = a(w_\ell, v_\ell) \text{ for all } v_\ell \text{ and } w_\ell \in V_{\text{ms},\ell}^{\text{EW}}.$$

Then the inverse operator $\mathcal{L}_\ell^{-1}$ exists, which is self-adjoint, positive semi-definite on $L^2(D)$, and positive definite on $V_{\text{ms},\ell}^{\text{EW}}$. Further, let $\mathcal{R}_\ell$ be the Riesz operator associated to $\mathcal{L}$ in the multiscale space $V_{\text{ms},\ell}^{\text{EW}}$, i.e.,

$$\forall v \in V \text{ and } w_{\mathrm{ms}} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}} : a(v - \mathcal{R}_\ell v, w_{\mathrm{ms}}) = 0.$$

Then it holds

$$\mathcal{L}_\ell^{-1} = \mathcal{R}_\ell \mathcal{L}^{-1}.$$

The approximation property of $\mathcal{L}_\ell^{-1}$ in energy norm is derived in [15, Proposition 5.2]. For any $v \in L^2(D)$, it holds

$$\left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{H_\kappa^1(D)} \lesssim \left(H \|\widetilde{\kappa}\|_{L^\infty(D)}^{1/2} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} \right) \|v\|_{L^2(D)}. \tag{4.1}$$

Together with the duality argument, we can derive the approximation property of $\mathcal{L}_\ell^{-1}$ in $L^2(D)$-norm.

Lemma 4.1 *(Approximation property of $\mathcal{L}_\ell^{-1}$ in $L^2(D)$-norm). For all $v \in L^2(D)$, there holds*

$$\left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{L^2(D)} \lesssim \left(H \|\widetilde{\kappa}\|_{L^\infty(D)}^{1/2} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} \right) \left\| \mathcal{L}^{-1} v \right\|_{H_\kappa^1(D)}. \tag{4.2}$$

Proof. This assertion can be derived from the duality argument together with (4.1). Indeed, let $w \in V$ and $w_{\mathrm{ms}} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ satisfy

$$\begin{aligned} \mathcal{L} w &= \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \\ \mathcal{L}_\ell w_{\mathrm{ms}} &= \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v. \end{aligned}$$

Then it holds

$$\begin{aligned} \left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{L^2(D)}^2 &= a(w - w_{\mathrm{ms}}, \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v) \\ &\le \|w - w_{\mathrm{ms}}\|_{H_\kappa^1(D)} \left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{H_\kappa^1(D)} \end{aligned}$$

Together with the estimate (4.1), we derive

$$\begin{aligned} &\left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{L^2(D)}^2 \\ &\le \left(H \|\widetilde{\kappa}\|_{L^\infty(D)}^{1/2} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} \right) \left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{L^2(D)} \left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{H_\kappa^1(D)}. \end{aligned} \tag{4.3}$$

The stability of Riesz projection $\mathcal{R}_\ell$ implies

$$\left\| \mathcal{L}^{-1} v - \mathcal{L}_\ell^{-1} v \right\|_{H_\kappa^1(D)} \le \left\| \mathcal{L}^{-1} v \right\|_{H_\kappa^1(D)},$$

then together with (4.3), this shows the desired assertion. □

Using the properties of the discrete operator $\mathcal{L}_\ell$, we can obtain the error estimate of Algorithm 1.

Proposition 4.1 *(Pointwise-in-time error estimate in $L^2(D)$-norm for Algorithm 1). For all $m = 1, 2, \cdots, M_\delta$, there holds*

$$\begin{aligned} \left\| u(\cdot, t_m) - U_{ms,\ell}^{EW,m} \right\|_{L^2(D)} &\lesssim \left(H \|\widetilde{\kappa}\|_{L^\infty(D)}^{1/2} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} + \delta t \right) t_m^{-1} \|u_0\|_{L^2(D)} \\ &\quad + \left(H \|\widetilde{\kappa}\|_{L^\infty(D)}^{1/2} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} \right) t_m \sup_{s \le t_m} |f(\cdot, s)|_2 \\ &\quad + \delta t \left(t_m \sup_{s \le t_m} |f(\cdot, s)|_2 + \int_0^{t_m} \|\partial_s f(\cdot, s)\|_{L^2(D)} \mathrm{d}s \right). \end{aligned} \tag{4.4}$$

Proof. This result can be obtained from [33, Theorems 7.7 and 8.5]. □

4.2. Convergence for Algorithm 2

We present in this section the convergence analysis for Algorithm 2. To this end, we first prove the boundedness and Lipschitz continuity properties of the coarse solver $E_{\Delta}^{\mathrm{ms},\ell}$ and the jump operator $\mathcal{S}$ in the multiscale space $V_{\mathrm{ms},\ell}^{\mathrm{EW}}$:

Lemma 4.2. *For all $n \in \{1, \cdots, M_{\Delta} - 1\}$, the following properties hold.*

1. *The one step coarse solver $E_{\Delta}^{ms,\ell}$ is Lipschitz in $V_{ms,\ell}^{\mathrm{EW}}$. For all $v_1, v_2 \in V_{ms,\ell}^{\mathrm{EW}}$, there holds*

$$\left\| E_{\Delta}^{ms,\ell}(T^n, v_1) - E_{\Delta}^{ms,\ell}(T^n, v_2) \right\|_{L^2(D)} \leq \alpha \left\| v_1 - v_2 \right\|_{L^2(D)}$$

with

$$\alpha := \frac{1}{1 + \Delta T / \mathrm{C}_{\mathrm{poin}}(D)}.$$

Here, $\mathrm{C}_{\mathrm{poin}}(D)$ denotes the Poincarè constant over the domain D.

2. *The jump operator $\mathcal{S}$ is an approximation of order 1 with Lipschitz regularity. For all $v_1, v_2 \in V_{ms,\ell}^{\mathrm{EW}} \cap \dot{H}^2(D)$ and any $\epsilon > 0$, there holds*

$$\left\| \mathcal{S}(T^n, v_1) - \mathcal{S}(T^n, v_2) \right\|_{L^2(D)} \leq \Delta T \frac{2}{\epsilon} (\Delta T)^{\epsilon/2} |v_1 - v_2|_{2+\epsilon}. \tag{4.5}$$

Proof. 1. Let $e_{\mathrm{ms}}^{n+1} := E_{\Delta}^{\mathrm{ms},\ell}(T^n, v_1) - E_{\Delta}^{\mathrm{ms},\ell}(T^n, v_2)$, then it holds

$$\forall w_{\mathrm{ms}} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}} : \int_D e_{\mathrm{ms}}^{n+1} w_{\mathrm{ms}} \, \mathrm{d}x + \Delta T \int_D \kappa \nabla e_{\mathrm{ms}}^{n+1} \cdot \nabla w_{\mathrm{ms}} \, \mathrm{d}x = \int_D (v_1 - v_2) w_{\mathrm{ms}} \, \mathrm{d}x.$$

Choosing $w_{\mathrm{ms}} := e_{\mathrm{ms}}^{n+1}$ leads to

$$\left\| e_{\mathrm{ms}}^{n+1} \right\|_{L^2(D)}^2 + \Delta T \left\| e_{\mathrm{ms}}^{n+1} \right\|_{H_{\kappa}^1(D)}^2 = \int_D e_{\mathrm{ms}}^{n+1} (v_1 - v_2) \, \mathrm{d}x.$$

Finally an application of the Young's inequality and the Poincarè inequality proves the first assertion.
2. To prove the second assertion, let

$$\begin{aligned}
e_{\mathrm{ms}}^{n+1} &:= \mathcal{S}(T^n, v_1) - \mathcal{S}(T^n, v_2) \\
&= \left(\mathcal{F}_{\delta}^{\mathrm{ms},\ell}(T^n, \Delta T, v_1) - \mathcal{F}_{\delta}^{\mathrm{ms},\ell}(T^n, \Delta T, v_2) \right) - \left(E_{\Delta}^{\mathrm{ms},\ell}(T^n, v_1) - E_{\Delta}^{\mathrm{ms},\ell}(T^n, v_2) \right) \\
&= \bar{\mathcal{F}}_{\delta}^{\mathrm{ms},\ell}(T^n, \Delta T, v_1 - v_2) - \bar{E}_{\Delta}^{\mathrm{ms},\ell}(T^n, v_1 - v_2) \\
&= \left(\bar{\mathcal{F}}_{\delta}^{\mathrm{ms},\ell}(T^n, \Delta T, v_1 - v_2) - \bar{\mathcal{F}}^{\mathrm{ms},\ell}(T^n, \Delta T, v_1 - v_2) \right) - \left(\bar{E}_{\Delta}^{\mathrm{ms},\ell}(T^n, v_1 - v_2) \right. \\
&\quad \left. - \bar{\mathcal{F}}^{\mathrm{ms},\ell}(T^n, \Delta T, v_1 - v_2) \right) \\
&=: e_{\mathrm{ms},\delta}^{n+1} - e_{\mathrm{ms},\Delta}^{n+1}.
\end{aligned}$$

To estimate e_{ms}^{n+1}, we only need to derive the estimate for $e_{\mathrm{ms},\delta}^{n+1}$ and $e_{\mathrm{ms},\Delta}^{n+1}$, separately.

To this end, let $v_{\mathrm{ms},i}^{n+1} := v_{\mathrm{ms},i}(\cdot, T^{n+1}) := \mathcal{F}^{\mathrm{ms},\ell}(T^n, \Delta T, v_i)$ for $i = 1, 2$, we first construct the equation for $e_{\mathrm{ms},\Delta}^{n+1}$ by the definitions of the coarse solver (3.1) and fine solver (3.2). There holds

$$\forall w_{\mathrm{ms}} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}} : \int_D e_{\mathrm{ms},\Delta}^{n+1} w_{\mathrm{ms}} \, \mathrm{d}x + \Delta T \int_D \kappa \nabla e_{\mathrm{ms},\Delta}^{n+1} \cdot \nabla w_{\mathrm{ms}} \, \mathrm{d}x = \int_D w_0 \cdot w_{\mathrm{ms}} \, \mathrm{d}x.$$

Here,

$$\begin{aligned}
w_0 &:= \Delta T \Big(- \partial_t v_{\mathrm{ms},1}|_{t=T^{n+1}} + \frac{v_{\mathrm{ms},1}^{n+1} - v_1}{\Delta T} + \partial_t v_{\mathrm{ms},2}|_{t=T^{n+1}} - \frac{v_{\mathrm{ms},2}^{n+1} - v_1}{\Delta T} \Big) \\
&= - \int_{T^n}^{T^{n+1}} (s - T^n) \partial_{ss} (v_{\mathrm{ms},1} - v_{\mathrm{ms},2})(\cdot, s) \, \mathrm{d}s.
\end{aligned}$$

Note that

$$\|w_0\|_{L^2(D)} \leq \Delta T \int_{T^n}^{T^{n+1}} \left\|\partial_{ss}(v_{\mathrm{ms},1} - v_{\mathrm{ms},2})(\cdot, s)\right\|_{L^2(D)} \mathrm{d}s.$$

An adaptation of the proof to [33, Lemma 3.2] shows

$$\left\|\partial_{tt}(v_{\mathrm{ms},1} - v_{\mathrm{ms},2})(\cdot, t)\right\|_{L^2(D)} \lesssim (t - T^n)^{-1+\epsilon/2} |v_1 - v_2|_{2+\epsilon} \text{ for all } t > 0.$$

Consequently, we derive

$$\|w_0\|_{L^2(D)} \leq \Delta T \frac{2}{\epsilon} (\Delta T)^{\epsilon/2} |v_1 - v_2|_{2+\epsilon}.$$

Choosing $w_{\mathrm{ms}} := e^{n+1}_{\mathrm{ms},\Delta}$ leads to

$$\left\|e^{n+1}_{\mathrm{ms},\Delta}\right\|^2_{L^2(D)} + \Delta T \left\|e^{n+1}_{\mathrm{ms},\Delta}\right\|^2_{H^1_k(D)} = \int_D e^{n+1}_{\mathrm{ms},\Delta} w_0 \,\mathrm{d}x.$$

Consequently, an application of the Young's inequality implies

$$\left\|e^{n+1}_{\mathrm{ms},\Delta}\right\|_{L^2(D)} \leq \Delta T \frac{2}{\epsilon} (\Delta T)^{\epsilon/2} |v_1 - v_2|_{2+\epsilon}.$$

Analogously, we can obtain the estimate for $e^{n+1}_{\mathrm{ms},\delta}$, which reads

$$\left\|e^{n+1}_{\mathrm{ms},\delta}\right\|_{L^2(D)} \leq \delta t \frac{2}{\epsilon} (\delta t)^{\epsilon/2} |v_1 - v_2|_{2+\epsilon}.$$

Note that $\delta t \ll \Delta T$, then a combination of the two estimates above with the triangle inequality, shows the second assertion. □

We present in the next theorem the convergence rate of Algorithm 2 to Problems (1.1) in pointwise-in-time in $L^2(D)$-norm. To derive it, we first decompose the error from Algorithm 2 as a summation of the error from WEMsFEM and the error from parareal algorithm. Then we estimate the former by Proposition 4.1, and the latter can be estimated by mathematical induction. This result relies on the following assumption.

Assumption 4.1. Let m be a positive integer such that $T^n = t_m$ for some integer n. For $\epsilon > 0$ be sufficiently small, we assume the following inequality holds

$$|U^{\mathrm{EW},m}_{\mathrm{ms},\ell} - U^n_k|_{2+\epsilon} \leq (\frac{2}{e})^4 (T^n)^{-1} \left\|U^{\mathrm{EW},m}_{\mathrm{ms},\ell} - U^n_k\right\|_{L^2(D)}.$$

We remark here this assumption is provable for the continuous problem [33, Lemma 3.2].

Theorem 4.1. *[Pointwise-in-time error estimate in $L^2(D)$-norm for Algorithm 2] Let Assumptions 2.1 and 4.1 hold. Assume that the source term $f \in L^\infty([0,T]; \dot{H}^2(D))$ satisfying $\partial_t f \in L^1([0,T]; L^2(D))$ and initial data $u_0 \in L^2(D)$. Let $\ell \in \mathbb{N}_+$ be the level parameter. The coarse time step size and fine time step size are ΔT and δt. Let $u(\cdot, t) \in V$ be the solution to Problem (1.1) and let U^n_k be the solution from Algorithm 2 with a small iteration $k \in \mathbb{N}$. There holds*

$$\begin{aligned}
\left\|u(\cdot, T^n) - U^n_k\right\|_{L^2(D)} \lesssim & \left(H \|\widetilde{\kappa}\|^{1/2}_{L^\infty(D)} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)} + \delta t\right) \frac{1}{T^n} \|u_0\|_{L^2(D)} \\
& + \left(H \|\widetilde{\kappa}\|^{1/2}_{L^\infty(D)} + 2^{-\ell/2} \|\kappa\|_{L^\infty(\mathcal{F}_H)}\right) T^n \sup_{s \leq T^n} |f(\cdot, s)|_2 \\
& + \delta t \left(T^n \sup_{s \leq T^n} |f(\cdot, s)|_2 + \int_0^{T^n} \|\partial_s f(\cdot, s)\|_{L^2(D)} \mathrm{d}s\right) \\
& + \left(\Pi^k_{j=0} \frac{1}{T^{n-j}}\right) \Delta T^{k+1} \|u_0\|_{L^2(D)}.
\end{aligned}$$

Proof. We first define the multiscale solution to Problem (1.1) using Algorithm 1. Find $U_{\mathrm{ms},\ell}^{\mathrm{EW},m} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}}$ for $m = 1, \cdots, M_\delta$, satisfying

$$\forall w_{\mathrm{ms}} \in V_{\mathrm{ms},\ell}^{\mathrm{EW}} : (\frac{U_{\mathrm{ms},\ell}^{\mathrm{EW},m} - U_{\mathrm{ms},\ell}^{\mathrm{EW},m-1}}{\delta t}, w_{\mathrm{ms}})_D + a(U_{\mathrm{ms},\ell}^{\mathrm{EW},m}, w_{\mathrm{ms}}) = (f(\cdot, t_m), w_{\mathrm{ms}})_D \tag{4.6}$$

$$U_{\mathrm{ms},\ell}^{\mathrm{EW},0} = I_\ell(u_0).$$

Then we only need to estimate $\left\|u(\cdot, T^n) - U_{\mathrm{ms},\ell}^{\mathrm{EW},m}\right\|_{L^2(D)}$ and $\left\|U_{\mathrm{ms},\ell}^{\mathrm{EW},m} - U_k^n\right\|_{L^2(D)}$ for $m := \Delta T/\delta t \times n$. Note that $T^n = t_m$. Therefore, we can replace T^n with t_m. Similarly, let $m' := \Delta T/\delta t \times (n-1)$, then it holds $t_{m'} = T^{n-1}$. The first term $\left\|u(\cdot, t_m) - U_{\mathrm{ms},\ell}^{\mathrm{EW},m}\right\|_{L^2(D)}$ can be estimated by Proposition 4.1. The second term $\left\|U_{\mathrm{ms},\ell}^{\mathrm{EW},m} - U_k^n\right\|_{L^2(D)}$ corresponds to the error induced by parareal algorithm in the multiscale method, and we will prove by mathematical induction:

$$e_k^n := \left\|U_{\mathrm{ms},\ell}^{\mathrm{EW},m} - U_k^n\right\|_{L^2(D)} \le C_{\mathrm{o}}\Big(\Pi_{j=0}^{k'} \frac{1}{T^{n-j}}\Big)\Delta T^{k+1} \left\|u_0\right\|_{L^2(D)}. \tag{4.7}$$

Here, C_{o} is a positive constant independent of H, M_Δ or T with its value changes from context.

Obviously, the inequality (4.7) holds when $k = 0$. Assume that it holds for iteration k for some $k \in \mathbb{N}_+$. We will show that it holds for the next iteration $k+1$.

We can obtain from Algorithm 2:

$$e_{k+1}^n = \|\mathcal{S}(T^{n-1}, U_{\mathrm{ms},\ell}^{\mathrm{EW},m'}) - \mathcal{S}(T^{n-1}, U_k^{n-1}) + E_\Delta^{\mathrm{ms},\ell}(T^{n-1}, U_{\mathrm{ms},\ell}^{\mathrm{EW},m'}) - E_\Delta^{\mathrm{ms},\ell}(T^{n-1}, U_{k+1}^{n-1})\|_{L^2(D)}.$$

Consequently, an application of Lemma 4.2 and Assumption 4.1 lead to

$$\begin{aligned}
e_{k+1}^n &\le \left\|\mathcal{S}(T^{n-1}, U_{\mathrm{ms},\ell}^{\mathrm{EW},m'}) - \mathcal{S}(T^{n-1}, U_k^{n-1})\right\|_{L^2(D)} + \left\|E_\Delta^{\mathrm{ms},\ell}(T^{n-1}, U_{\mathrm{ms},\ell}^{\mathrm{EW},m'}) - E_\Delta^{\mathrm{ms},\ell}(T^{n-1}, U_{k+1}^{n-1})\right\|_{L^2(D)} \\
&\le \Delta T \frac{2}{\epsilon}(\Delta T)^{\epsilon/2} |U_{\mathrm{ms},\ell}^{\mathrm{EW},m'} - U_k^{n-1}|_{2+\epsilon} + \alpha \left\|U_{\mathrm{ms},\ell}^{\mathrm{EW},m'} - U_{k+1}^{n-1}\right\|_{L^2(D)} \\
&\le (1-\alpha)\frac{\Delta T}{T^{n-1}} \|U_{\mathrm{ms},\ell}^{\mathrm{EW},m'} - U_k^{n-1}\|_{L^2(D)} + \alpha \left\|U_{\mathrm{ms},\ell}^{\mathrm{EW},m'} - U_{k+1}^{n-1}\right\|_{L^2(D)} \\
&= (1-\alpha)\frac{\Delta T}{T^{n-1}} e_k^{n-1} + \alpha e_{k+1}^{n-1}.
\end{aligned} \tag{4.8}$$

In the last inequality we choose the positive parameter ϵ satisfying

$$\epsilon \ge (\frac{2}{\mathrm{e}})^4 \frac{2}{1-\alpha}.$$

Note that $e_{k+1}^n = 0$ for all $n \le k+1$. We can obtain by using (4.8) repeatedly

$$\begin{aligned}
e_{k+1}^n &\le (1-\alpha) \sum_{j=1}^{n-(k+1)} \alpha^{j-1} \frac{\Delta T}{T^{n-j}} e_k^{n-j}. \\
&\le (1-\alpha) \sum_{j=1}^{n-(k+1)} \alpha^{j-1} C_{\mathrm{o}} \frac{\Delta T}{T^{n-j}} \Big(\Pi_{i=0}^{k} \frac{1}{T^{n-j-i}}\Big)\Delta T^{k+1} \left\|u_0\right\|_{L^2(D)} \\
&= C_{\mathrm{o}}\Big(\Pi_{j=0}^{k+1} \frac{1}{T^{n-j}}\Big)\Delta T^{k+2} \left\|u_0\right\|_{L^2(D)} \times (1-\alpha) \sum_{j=1}^{n-(k+1)} \alpha^{j-1} \Big(\Pi_{i=0}^{k+1} \frac{T^{n-i}}{T^{n-j-i}}\Big) \frac{T^{n-j-k-1}}{T^{n-j}} \\
&\le C_{\mathrm{o}}\Big(\Pi_{j=0}^{k+1} \frac{1}{T^{n-j}}\Big)\Delta T^{k+2} \left\|u_0\right\|_{L^2(D)} \times (1-\alpha) \sum_{j=1}^{n-(k+1)} \alpha^{j-1} \Big(\Pi_{i=0}^{k+1} \frac{T^{n-i}}{T^{n-j-i}}\Big).
\end{aligned}$$

We only need to prove $A(n,k)$ as defined in the following, is bounded.

$$\begin{aligned}
A(n,k) &:= (1-\alpha) \sum_{j=1}^{n-(k+1)} \alpha^{j-1} \Big(\Pi_{i=0}^{k+1} \frac{T^{n-i}}{T^{n-j-i}}\Big) \\
&= (1-\alpha) \sum_{m=k+1}^{n-1} \alpha^{n-1-m} \Big(\Pi_{i=0}^{k+1} \frac{n-i}{m-i}\Big).
\end{aligned}$$

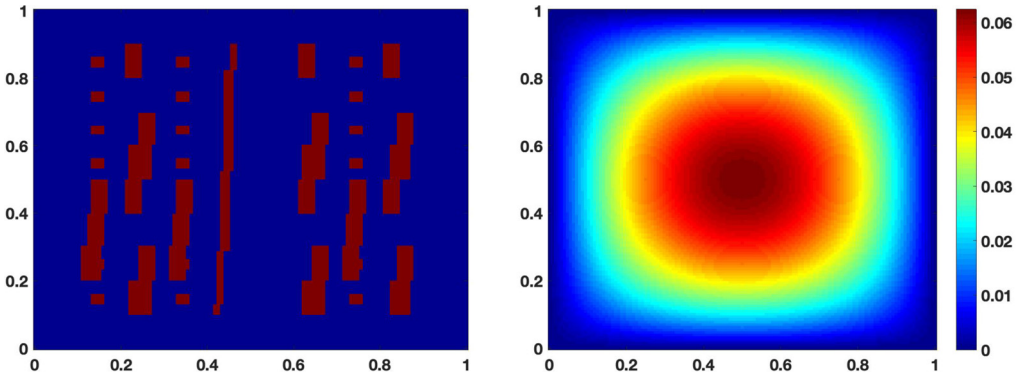

Fig. 2. The heterogeneous permeability field κ and the initial data $u_0 = x(1-x)y(1-y)$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

We can further express $A(n,k)$ as

$$\begin{aligned} A(n,k) &= (1-\alpha)C_{n-1}^{k+1}\sum_{m=k+1}^{n-1}\frac{\alpha^{n-1-m}}{C_m^{k+1}} \\ &=: (1-\alpha)C_{n-1}^{k+1}\sum_{m=k+1}^{n-1} a_m. \end{aligned} \tag{4.9}$$

Note that

$$\begin{aligned} &\frac{a_{m+1}}{a_m} = \frac{m-k}{m+1}\alpha^{-1} \le \frac{n-1-k}{n}\alpha^{-1} := \eta\alpha^{-1} \\ &a_{k+1} = \alpha^{n-k-2}. \end{aligned}$$

Consequently, we obtain

$$A(n,k) \le (1-\alpha)C_{n-1}^{k+1}(n-k-2)\max\{\eta,\alpha\}^{n-k-2}.$$

Since k is small, $A(n,k)$ is bounded for any n.

This proves estimate (4.7) corresponding to the case with iteration $k+1$. Hence, the estimate (4.7) is proved. Finally, a combination with (4.4) results in the desired estimate. □

Theorem 4.1 indicates that the pointwise-in-time error estimate of Algorithm 2 to Problems (1.1) in $L^2(D)$-norm will deteriorate when the time step approaches the original $t = 0$. This blow-up of error is produced by the parareal algorithm (Step 2 in the proof to Theorem 4.1), which essentially arises from the approximation property of the jump operator (4.5).

Remark 4.1. Algorithm 2 outweighs Algorithm 1 only when the former achieves similar accuracy to the latter within a very few iteration $k \ll M_\Delta$. Therefore, we are not interested in the case when $k \ge M_\Delta$ or the error at time level T^n with $k \ge n$.

5. Numerical results

In this section, we perform a series of numerical experiments to demonstrate the performance of the proposed WEMP Algorithm. In particular, we compare the performance of Algorithms 1 and 2 for each experiment. Furthermore, we investigate whether replacing backward Euler scheme by Crank-Nicolson scheme would reduce the iteration number. Motivated by the critical condition proposed in [34], we choose different values of $\frac{\Delta T}{\delta t}$ to test how they will influence the iteration number. It can be seen from Equation (4.6) that WEMP Algorithm would generate a solution of better accuracy when the source term being 0. In the last subsection, we conduct experiments to verify this.

We consider the parabolic equation (1.1) in the space domain $D := [0,1]^2$ and the time domain $[0,T] = [0,1]$. The permeability coefficient κ we choose has two distinct value: 1 and 1000. It is high-contrast and heterogeneous. We refer to Fig. 2 (left figure) for an illustration. The initial data tested in our numerical experiments is chosen to be a smooth function $u_0 := x(1-x)y(1-y)$. We refer to Fig. 2 (right figure) for an illustration.

Let $\mathcal{T}_H$ be a decomposition of the domain D into non-overlapping shape-regular rectangular elements with maximal mesh size $H := 2^{-4}$. These coarse rectangular elements are further partitioned into a collection of connected fine rectangular elements $\mathcal{T}_h$ using fine mesh size $h := 2^{-7}$. Similarly, we define V_h to be a conforming piecewise affine finite element

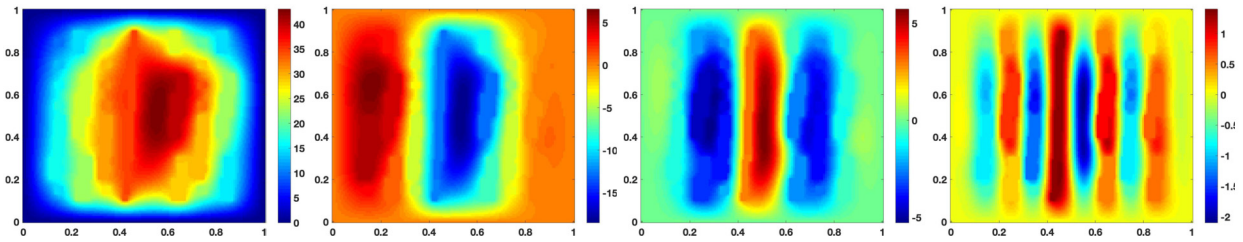

Fig. 3. Numerical solution U_h^n to (2.3) for $n = 10^3, 3\times 10^3, 5\times 10^3$ and 10^4 with $\delta t = 10^{-4}$.

associated with $\mathcal{T}_h$. In our numerical experiments, space meshes $\mathcal{T}_H$ and $\mathcal{T}_h$ are fixed. To keep our presentation concise, we will only present the numerical results with a fixed level parameter $\ell := 2$. The temporal discretization is presented in Section 2 with $T := 1$. The coarse time step size and fine time step size are ΔT and δt. Note that $\delta t \ll \Delta T$.

We introduce the following notations to calculate the errors. The relative errors for the multiscale solution in $L^2(D)$-norm and $H^1_\kappa(D)$-norm are

$$\mathrm{Rel}^{\mathrm{EW}}_{L^2}(t_m) := \frac{\left\| U_h^m - U^{\mathrm{EW},m}_{\mathrm{ms},\ell} \right\|_{L^2(D)}}{\left\| U_h^m \right\|_{L^2(D)}} \times 100 \quad \text{and} \quad \mathrm{Rel}^{\mathrm{EW}}_{H^1_\kappa}(t_m) := \frac{\left\| U_h^m - U^{\mathrm{EW},m}_{\mathrm{ms},\ell} \right\|_{H^1_\kappa(D)}}{\left\| U_h^m \right\|_{H^1_\kappa(D)}} \times 100.$$

Analogously, the relative errors for our proposed algorithm with iteration $k \in \mathbb{N}$ in $L^2(D)$-norm and $H^1_\kappa(D)$-norm are

$$\mathrm{Rel}^{k}_{L^2}(T^n) := \frac{\left\| U_h^m - U_k^n \right\|_{L^2(D)}}{\left\| U_h^m \right\|_{L^2(D)}} \times 100 \quad \text{and} \quad \mathrm{Rel}^{k}_{H^1_\kappa}(T^n) := \frac{\left\| U_h^m - U_k^n \right\|_{H^1_\kappa(D)}}{\left\| U_h^m \right\|_{H^1_\kappa(D)}} \times 100$$

with $m := \Delta T/\delta t \times n$.

Our numerical experiments include testing nonzero source term in section 5.1 and zero source term in section 5.2. We investigate the influence of different tempo discretization schemes, e.g., backward Euler scheme and Crank-Nicolson Galerkin scheme, on the performance of our algorithm.

5.1. Numerical tests with nonzero source term

To define nonzero source term, we take time-dependent smooth function

$$f(x, y, t) := 200\pi^2 \sin(\pi x)\sin(\pi y)\sin(10\pi t x).$$

Since there is no analytic solution to system (1.1), we need to find an approximation of the exact solutions. To this end, we take time step size $\delta t = 10^{-4}$ and use backward Euler Galerkin Method in (2.3) to obtain the reference solutions U_h^n. Note that we use a much finer time step size to simulate the reference solution. We plot the reference solutions U_h^n for $n = 10^3, 3\times 10^3, 5\times 10^3$ and 10^4 in Fig. 3. In the rest of this subsection, we will present numerical tests using backward Euler scheme with $\frac{\Delta T}{\delta t} = 100$ in Experiment 1, Crank-Nicolson scheme with $\frac{\Delta T}{\delta t} = 100$ in Experiment 2 and backward Euler scheme with $\frac{\Delta T}{\delta t} = 10$ in Experiment 3. For all the three experiments, our proposed algorithm, i.e. Algorithm 2, can generate numerical solutions by a few iterations at least of the same accuracy as the multiscale solutions from Algorithm 1. For the brevity of the paper, we only present numerical solutions U_k^n from Algorithm 2 with iteration number $k = 0, 1$ and 2 and multiscale solutions $U^{\mathrm{EW},m}_{\mathrm{ms},\ell}$ from Algorithm 1 in Experiment 1.

Experiment 1: backward Euler with $\frac{\Delta T}{\delta t} = 100$

We test in this experiment the performance of Algorithm 2 with a fine time step size $\delta t = 10^{-3}$ and a coarse time step size $\Delta T = 0.1$. The backward Euler scheme is utilized for the time discretization.

We present the numerical solutions U_k^n for $n = 1, 3, 5, 10$ from Algorithm 2 with iteration number $k = 0, 1$ and 2 in Fig. 4. One can observe that U_k^n converges to the multiscale solution $U^{\mathrm{EW},n}_{\mathrm{ms},\ell}$ as the iteration k increases.

The convergence history of Algorithm 2 in relative $L^2(D)$-norm and relative $H^1_\kappa(D)$-norm are presented in Fig. 5.

One observes from Fig. 5 that 4 iterations is sufficient for Algorithm 2 to attain the same accuracy as Algorithm 1 for all discrete time steps under the $L^2(D)$-norm, while 2 iterations under the $H^1_\kappa(D)$-norm. Each iteration involves a number of $1/\Delta T = 10$ sequential solver and $\Delta T/\delta t = 100$ parallel solver. In comparison, Algorithm 1 involves a number of $1/\delta t = 1000$ sequential solver. Consequently, Algorithm 2 involves much less wall clock time with the aid of a sufficient number of processors.

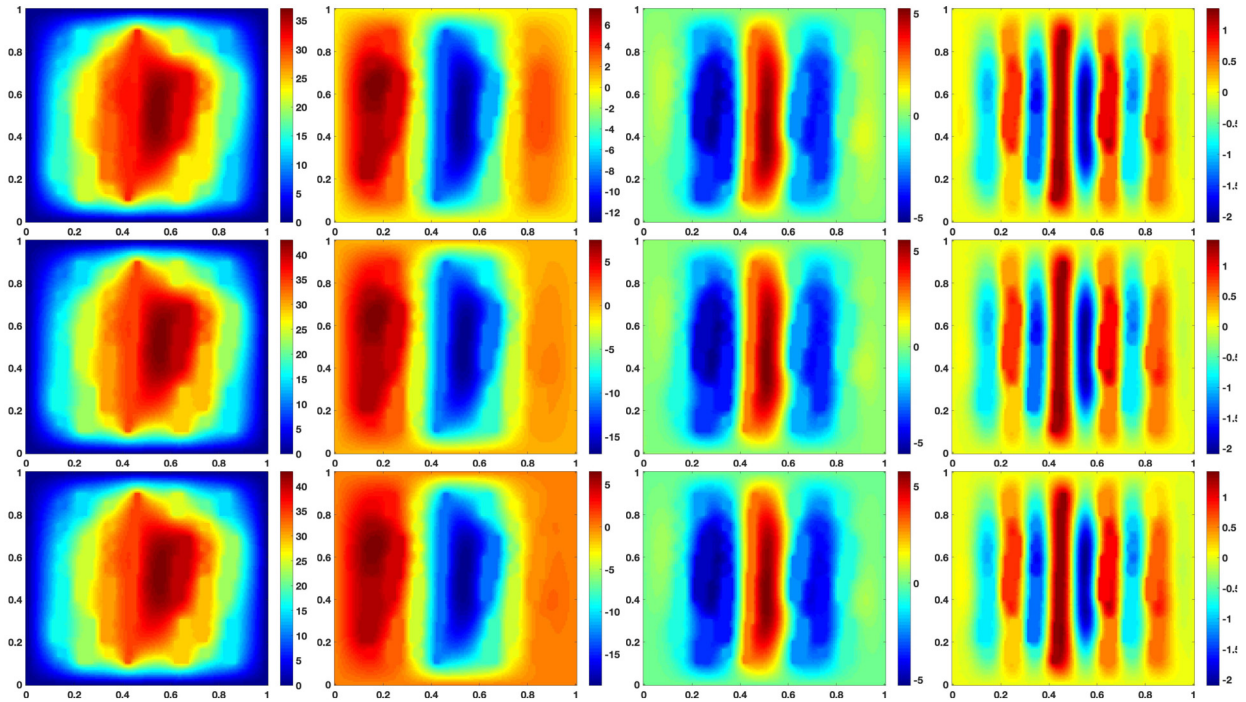

Fig. 4. Numerical solutions U_k^n for $n = 1, 3, 5, 10$ from Algorithm 2 with $\Delta T = 0.1$ and $\delta t = 10^{-3}$, backward Euler scheme: iteration number $k = 0$ (top), $k = 1$ (middle) and $k = 2$ (bottom).

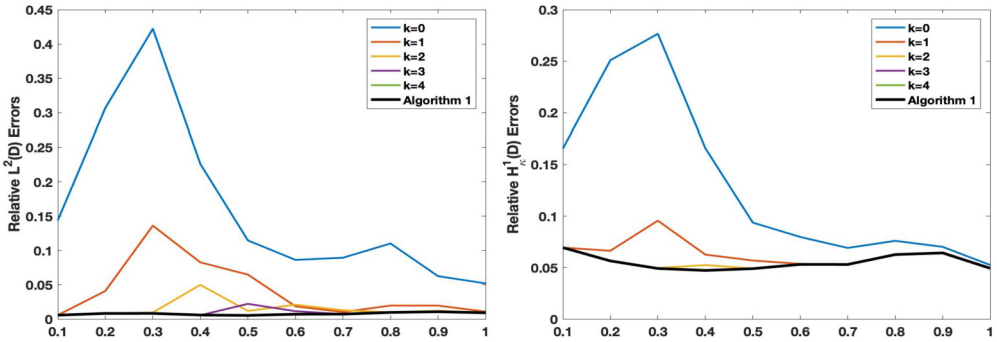

Fig. 5. Convergence history of Algorithm 2 in relative $L^2(D)$ error and relative $H_\kappa^1(D)$ error for Experiment 1: backward Euler scheme with $\Delta T = 0.1$ and $\delta t = 10^{-3}$.

Experiment 2: Crank-Nicolson with $\frac{\Delta T}{\delta t} = 100$

Since the backward Euler scheme is only first order accurate, a higher order accurate scheme can improve the performance of Algorithm 1 and Algorithm 2. This can be seen from the proof to Theorem 4.1. In this section, we will present the numerical tests with Crank-Nicolson scheme for both algorithms.

A direct application of Crank-Nicolson scheme as a time discretization fails to maintain second order accuracy due to the possible blow up of the eigenvalues of the elliptic operator $-\nabla \cdot (\kappa \nabla \cdot)$ when $\eta_{\max} \to \infty$. To improve its performance and maintain second order convergence rate, we use 3 steps of backward Euler scheme before Crank-Nicolson scheme kicks in [27,33].

The convergence history of Algorithm 2 in $L^2(D)$-norm and $H_\kappa^1(D)$-norm is presented in Fig. 6. Similar to Experiment 1, we observe that 4 iterations is sufficient for Algorithm 2 to reach the same accuracy as Algorithm 1 at all discrete time levels under the $L^2(D)$-norm, while 2 iterations under the $H_\kappa^1(D)$-norm. Comparing Fig. 5 with Fig. 6, one observes that Algorithm 2 with Crank-Nicolson scheme outperforms that with backward Euler scheme under $L^2(D)$-norm.

Experiment 3: backward Euler with $\frac{\Delta T}{\delta t} = 10$

We are also interested in studying how the coarse solver and fine solver affect the performance of our proposed WEMP algorithm. To this end, we choose $\Delta T = 10^{-2}$, $\delta t = 10^{-3}$ and utilize backward Euler scheme in time discretization. Note that the ratio between the coarse time step and fine time step is smaller than that in Experiment 1.

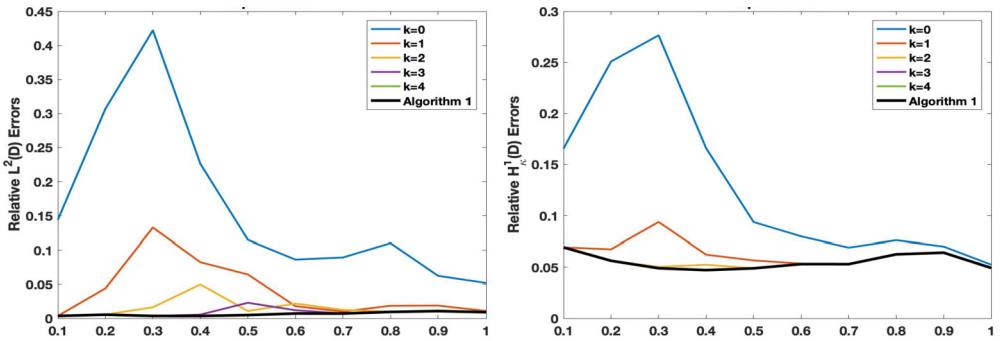

Fig. 6. Convergence history of Algorithm 2 in relative $L^2(D)$-norm and relative $H^1_\kappa(D)$-norm for Experiment 2: Crank-Nicolson scheme with $\Delta T = 0.1$ and $\delta t = 10^{-3}$.

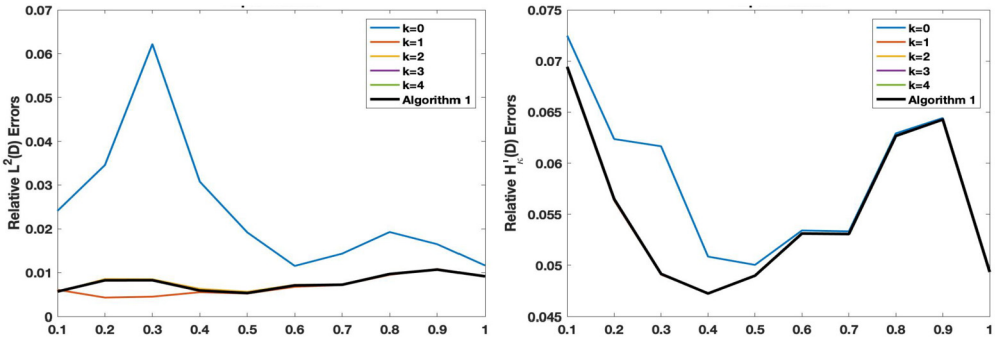

Fig. 7. Convergence history of Algorithm 2 in relative $L^2(D)$ error and relative $H^1_\kappa(D)$ error for Experiment 3: backward Euler scheme with $\Delta T = 10^{-2}$ and $\delta t = 10^{-3}$.

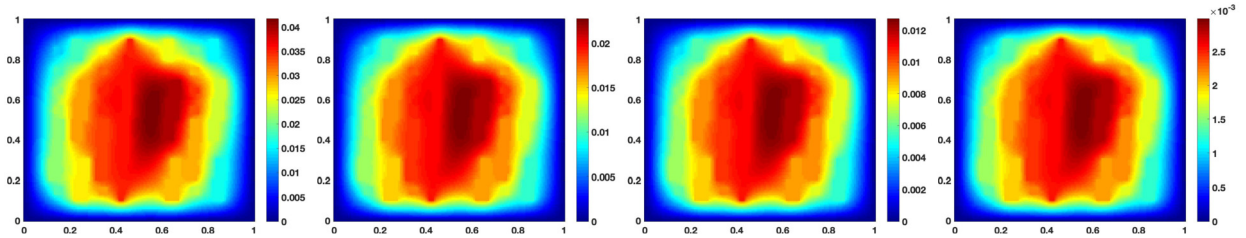

Fig. 8. Numerical solutions U^n_h to (2.3) with $f = 0$ for $n = 10, 30, 50$ and 100 with $\delta t = 10^{-3}$.

The convergence history of Algorithm 2 in $L^2(D)$-norm and $H^1_\kappa(D)$-norm is presented in Fig. 7. Comparing Fig. 5 with Fig. 7, one can see 1 iteration is sufficient for the numerical solutions from Algorithm 2 to reach the same accuracy as multiscale solutions from Algorithm 1 under $L^2(D)$-norm and $H^1_\kappa(D)$-norm when the coarse time step $\Delta T = 10^{-2}$ becomes smaller. However, this involves more coarse solvers for each iteration. Furthermore, a decreased coarse time step is only practical when sufficient processors are available.

5.2. Numerical tests with a vanishing source term

To avoid the complicated requirement on the source term in Theorem 4.1, we test in this section the performance of Algorithm 2 for Problem (1.1) with a vanishing source term $f := 0$ using a backward Euler scheme and Crank-Nicolson scheme. Consequently, the solution decays rapidly to 0. To generate solutions with reasonable size, we set the final time $T = 0.1$, the coarse time step $\Delta T := 10^{-2}$ and the fine time step $\delta t = 10^{-3}$. The initial data and permeability are the same as in the previous section. We use backward Euler scheme with time step 10^{-3} to obtain the reference solutions U^n_h. The reference solutions U^n_h for $n = 10, 30, 50, 100$ are plotted in Fig. 8.

We present the numerical solutions U^n_k for $n = 1, 3, 5, 10$ from Algorithm 2 with iteration number $k = 0, 1, 2$ in Fig. 9. One can observe U^n_k converges to $U^{\mathrm{EW},n}_{\mathrm{ms},\ell}$ as the iteration number k increases.

The convergence history of Algorithm 2 in $L^2(D)$-norm and $H^1_\kappa(D)$-norm is presented in Fig. 10. From the figure, one can see that 1 iteration is sufficient for the numerical solutions from Algorithm 2 with backward Euler to converge under

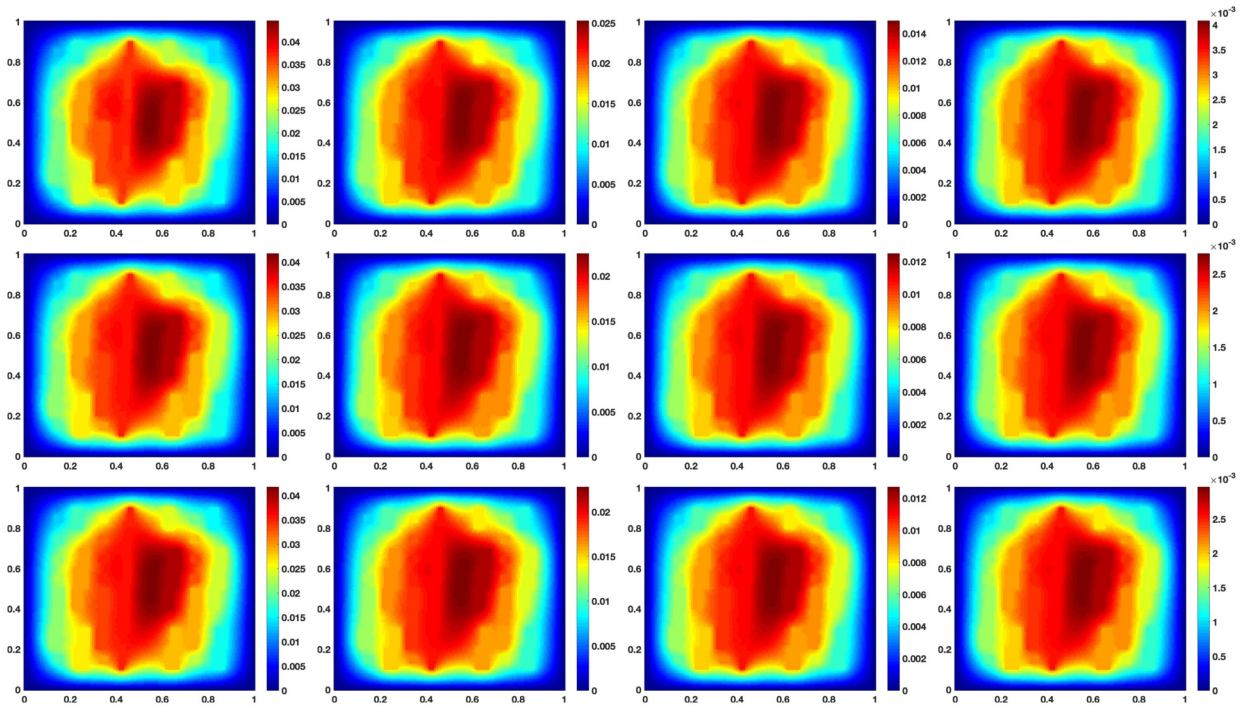

Fig. 9. Numerical solutions U_k^n for $n = 1, 3, 5, 10$ from Algorithm 2 with $\Delta T = 10^{-2}$ and $\delta t = 10^{-3}$, backward Euler scheme: iteration number $k = 0$ (top), $k = 1$ (middle) and $k = 2$ (bottom).

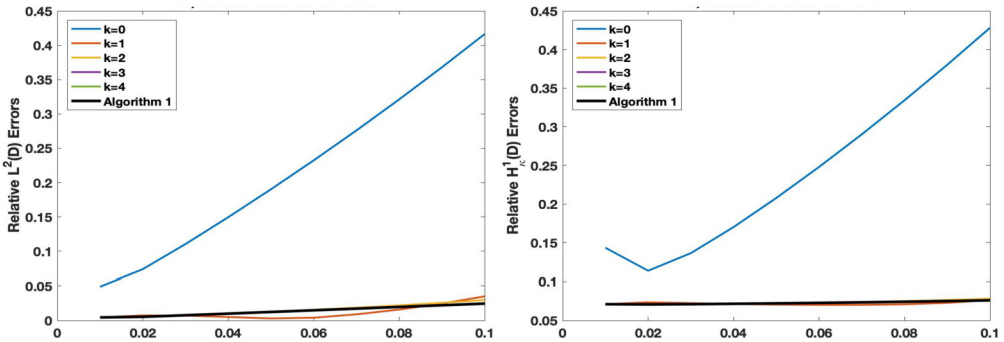

Fig. 10. Convergence history of Algorithm 2 in relative $L^2(D)$-norm and relative $H_\kappa^1(D)$-norm for $f = 0$: backward Euler scheme with $\Delta T = 10^{-2}$ and $\delta t = 10^{-3}$.

$L^2(D)$-norm and $H_\kappa^1(D)$-norm. We can conclude that our proposed algorithm with backward Euler scheme is effective in solving Problem (1.1) with zero source term.

Our last experiment is replacing backward Euler scheme by Crank-Nicolson scheme for the above problem. We observe the same convergence behavior as in the previous experiment that the numerical solutions from Algorithm 2 converges to the multiscale solutions from Algorithm 1. For brevity of the paper, we do not present these figures.

The convergence history of Algorithm 2 in $L^2(D)$-norm and $H_\kappa^1(D)$-norm is presented in Fig. 11. One observes that it takes 4 iterations to converge under $L^2(D)$-norm and 3 iterations to converge under $H_\kappa^1(D)$-norm when using Algorithm 2 with Crank-Nicolson scheme. Comparing Fig. 10 with Fig. 11, we can see that Algorithm 2 with the Crank-Nicolson scheme yields a better accuracy than that with the backward Euler scheme. We conclude that Algorithm 2 with backward Euler scheme converges faster than that with Crank-Nicolson scheme, while Algorithm 2 with Crank-Nicolson scheme generate solutions with a higher accuracy for Problem (1.1) with zero source term.

6. Conclusion

We propose in this paper a new efficient algorithm for parabolic problems with heterogeneous coefficients. This algorithm is named as the Wavelet-based Edge Multiscale Parareal (WEMP) Algorithm, which incorporates parareal algorithm

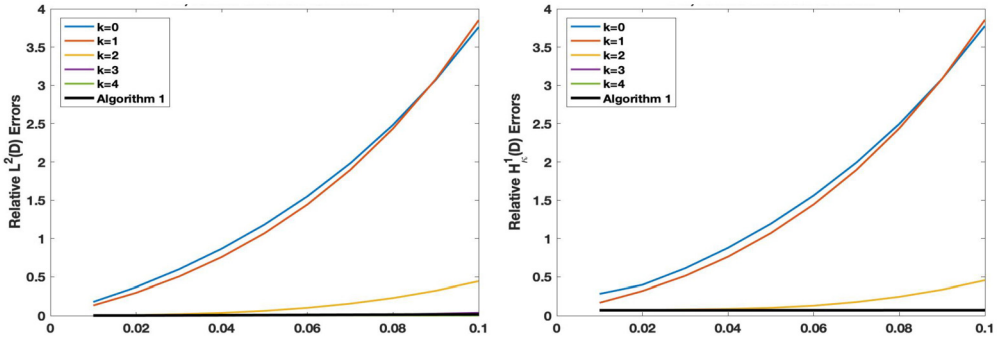

Fig. 11. Convergence history of Algorithm 2 in relative $L^2(D)$-norm and relative $H^1_\kappa(D)$-norm for $f = 0$: Crank-Nicolson scheme with $\Delta T = 10^{-2}$ and $\delta t = 10^{-3}$.

into the Wavelet-based Edge Multiscale Finite Element Methods (WEMsFEMs). Therefore, it can handle parabolic problems with heterogeneous coefficients more efficiently if multiple processors are available. We derive the convergence rate of this algorithm, and verify its performance by several numerical tests. Numerical experiments demonstrate WEMP Algorithm with coarse and fine propagators of higher order accuracy, e.g., Crank-Nicolson scheme, can enhance its performance. However, a detailed study of such algorithm is beyond the scope of this paper, which demands a comprehensive spectral study on the approximating operator $\mathcal{L}_\ell$ and its dependence on the multiple scales, especially when the coefficient κ is heterogeneous. Further, we also observe that the decay of the ratio between the coarse time step and fine time step $\frac{\Delta T}{\delta t}$ leads to faster convergence, yet more computational complexity is involved in each iteration. Moreover, the application of our proposed WEMP algorithm to time-fractional diffusion problems with heterogeneous coefficients in a large time domain is nontrivial since the fractional derivative is nonlocal, which makes single step scheme used in our algorithm infeasible. This is our interest in the future.

CRediT authorship contribution statement

Jiuhua Hu worked mainly on the numerical part. Guanglian Li proposed the main method and worked mainly on the analysis part.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Ariel, S.J. Kim, R. Tsai, Parareal multiscale methods for highly oscillatory dynamical systems, SIAM J. Sci. Comput. 38 (6) (2016) A3540–A3564.
[2] L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zérah, Parallel-in-time molecular-dynamics simulations, Phys. Rev. E 66 (Nov 2002) 057701.
[3] G. Bal, On the convergence and the stability of the parareal algorithm to solve partial differential equations, in: Domain Decomposition Methods in Science and Engineering, Springer, 2005, pp. 425–432.
[4] G. Bal, Y. Maday, A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put, in: Recent Developments in Domain Decomposition Methods, Zürich, 2001, in: Lect. Notes Comput. Sci. Eng., vol. 23, Springer, Berlin, 2002, pp. 189–202.
[5] L. Berlyand, H. Owhadi, Flux norm approach to finite dimensional homogenization approximations with non-separated scales and high contrast, Arch. Ration. Mech. Anal. 198 (2) (2010) 677–721.
[6] A. Blouza, L. Boudin, S.M. Kaber, Parallel in time algorithms with reduction methods for solving chemical kinetics, Commun. Appl. Math. Comput. Sci. 5 (2) (2010) 241–263.
[7] A.J. Christlieb, C.B. Macdonald, B.W. Ong, Parallel high-order integrators, SIAM J. Sci. Comput. 32 (2) (2010) 818–835.
[8] E. Chung, Y. Efendiev, T.Y. Hou, Adaptive multiscale model reduction with generalized multiscale finite element methods, J. Comput. Phys. 320 (2016) 69–95.
[9] W. E, B. Engquist, The heterogeneous multiscale methods, Commun. Math. Sci. 1 (1) (2003) 87–132.
[10] Y. Efendiev, J. Galvis, T. Hou, Generalized multiscale finite element methods, J. Comput. Phys. 251 (2013) 116–135.
[11] R.D. Falgout, S. Friedhoff, T.V. Kolev, S.P. MacLachlan, J.B. Schroder, Parallel time integration with multigrid, SIAM J. Sci. Comput. 36 (6) (2014) C635–C661.
[12] C. Farhat, M. Chandesris, Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications, Int. J. Numer. Methods Eng. 58 (9) (2003) 1397–1434.
[13] C. Farhat, J. Cortial, C. Dastillung, H. Bavestrello, Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses, Int. J. Numer. Methods Eng. 67 (5) (2006) 697–724.
[14] P.F. Fischer, F. Hecht, Y. Maday, A parareal in time semi-implicit approximation of the Navier-Stokes equations, in: Domain Decomposition Methods in Science and Engineering, in: Lect. Notes Comput. Sci. Eng., vol. 40, Springer, Berlin, 2005, pp. 433–440.
[15] S. Fu, E. Chung, G. Li, Edge multiscale methods for elliptic problems with heterogeneous coefficients, J. Comput. Phys. 369 (1) (2019) 228–242.
[16] S. Fu, G. Li, R. Craster, S. Guenneau, Wavelet-based edge multiscale finite element method for Helmholtz problems in perforated domains, 2019, accepted by Multiscale Model. Simul.

[17] M.J. Gander, E. Hairer, Nonlinear convergence analysis for the parareal algorithm, in: Domain Decomposition Methods in Science and Engineering XVII, Springer, 2008, pp. 45–56.
[18] M.J. Gander, I. Kulchytska-Ruchka, I. Niyonzima, S. Schöps, A new parareal algorithm for problems with discontinuous sources, SIAM J. Sci. Comput. 41 (2) (2019) B375–B395.
[19] M.J. Gander, I. Kulchytska-Ruchka, S. Schöps, A New Parareal Algorithm for Time-Periodic Problems with Discontinuous Inputs. ArXiv e-prints, Oct. 2018.
[20] T. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput. Phys. 134 (1) (1997) 169–189.
[21] T. Hughes, G. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1–2) (1998) 3–24.
[22] F. Legoll, T. Lelièvre, G. Samaey, A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations, SIAM J. Sci. Comput. 35 (4) (2013) A1951–A1986.
[23] G. Li, On the convergence rates of gmsfems for heterogeneous elliptic problems without oversampling techniques, Multiscale Model. Simul. 17 (2) (2019) 593–619.
[24] G. Li, D. Peterseim, M. Schedensack, Error analysis of a variational multiscale stabilization for convection-dominated diffusion equations in two dimensions, IMA J. Numer. Anal. 38 (3) (2018) 1229–1253.
[25] J.-L. Lions, Y. Maday, G. Turinici, Résolution d'EDP par un schéma en temps "pararéel", C. R. Acad. Sci. Paris Sér. I Math. 332 (7) (2001) 661–668.
[26] Y. Liu, J. Hu, Modified propagators of parareal in time algorithm and application to Princeton ocean model, Int. J. Numer. Methods Fluids 57 (12) (2008) 1793–1804.
[27] M. Luskin, R. Rannacher, On the smoothing property of the Crank-Nicolson scheme, Appl. Anal. 14 (2) (1982) 117–135.
[28] Y. Maday, Parareal in time algorithm for kinetic systems based on model reduction, in: High-Dimensional Partial Differential Equations in Science and Engineering, in: CRM Proc. Lecture Notes, vol. 41, Amer. Math. Soc., Providence, RI, 2007, pp. 183–194.
[29] Y. Maday, J. Salomon, G. Turinici, Monotonic parareal control for quantum systems, SIAM J. Numer. Anal. 45 (6) (2007) 2468–2482.
[30] Y. Maday, G. Turinici, A parareal in time procedure for the control of partial differential equations, C. R. Math. Acad. Sci. Paris 335 (4) (2002) 387–392.
[31] A. Målqvist, D. Peterseim, Localization of elliptic multiscale problems, Math. Comput. 83 (290) (2014) 2583–2603.
[32] K.C.T. Strouboulis, I. Babuška, The design and analysis of the generalized finite element method, Comput. Methods Appl. Mech. Eng. 181 (2000) 43–69.
[33] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, vol. 1054, Springer, 2006.
[34] S.-L. Wu, T. Zhou, Convergence analysis for three parareal solvers, SIAM J. Sci. Comput. 37 (2) (2015) A970–A992.