Parallel in time partially explicit splitting scheme for high contrast multiscale problems

Zhengya Yang * Yating Wang † Wing Tat Leung ‡

November 15, 2024

Abstract

Solving multiscale diffusion problems is often computationally expensive due to the spatial and temporal discretization challenges arising from high-contrast coefficients. To address this issue, a partially explicit temporal splitting scheme is proposed in [8]. By appropriately constructing multiscale spaces, the spatial multiscale property is effectively managed, and it has been demonstrated that the temporal step size is independent of the contrast. To enhance simulation speed, we propose a parallel algorithm for the multiscale flow problem that leverages the partially explicit temporal splitting scheme. The idea is first to evolve the partially explicit system using a coarse time step size, then correct the solution on each coarse time interval with a fine propagator, for which we consider both the sequential solver and all-at-once solver. This procedure is then performed iteratively till convergence. We analyze the stability and convergence of the proposed algorithm. The numerical experiments demonstrate that the proposed algorithm achieves high numerical accuracy for high-contrast problems and converges in a relatively small number of iterations. The number of iterations stays stable as the number of coarse intervals increases, thus significantly improving computational efficiency through parallel processing.

1 Introduction

Numerous scientific problems and models exhibit multiscale properties, such as flow in heterogeneous porous media, the diffusion of pollutants in the atmosphere, turbulent transport in high Reynolds number flows, and so on. These models often involve significant variations in media properties, commonly referred to as high contrast. The presence of high contrast introduces stiffness

*School of Mathematics and Statistic, Xi'an Jiaotong University, Xi'an, People's Republic of China.

†School of Mathematics and Statistic, Xi'an Jiaotong University, Xi'an, People's Republic of China. Corresponding author.

‡Department of Mathematics, City University of Hong Kong, Hong Kong Special Administrative Region.

to the system, which makes spatial and temporal discretization challenging for traditional numerical methods due to their high computational demands.

There have been many existing approaches in the literature to handle spatial multiscale problems, such as numerical homogenization (NH) [12, 1], multiscale finite element methods (MsFEM) [15, 28, 4], generalized multiscale finite element methods (GMsFEM) [5, 10, 11], constraint energy minimizing GMsFEM (CEM-GMsFEM) [21, 6, 26], localized orthogonal decomposition (LOD) [33, 25] and nonlocal multi-continua method (NLMC) [9, 19, 29]. Among which, CEM-GMsFEM is a multiscale finite element method used to effectively address multiscale problems with high-contrast parameters. It constructs multiscale basis functions by minimizing energy constraints, which can achieve contrast independent convergence rates. Based on CEM-GMsFEM, NLMC is proposed to construct local basis functions that automatically identify physical properties in each local region and provides non-local transmissibilities in the global formulation.

For non-stationary multiscale problems, the high-contrast will lead to very small time steps when treating explicitly. The partially explicit temporal splitting scheme [8] originates from the solution decomposition [14] and splitting algorithms [13]. The main idea of the method is to divide the solution space into two parts, the dominant part and the complementary part, such that the time step size is independent of the high-contrast. The method is successfully employed in solving wave equation, nonlinear diffusion, and time-fractional diffusion equations[7, 16, 22, 29] and extended to an adaptive algorithm[29]. In this paper, we follow the concept developed in [8] for linear equations.

All the existing literatures are based on the sequential method for solving the partially explicit temporal splitting scheme, which is easy to implement but might be inefficient when the temporal mesh partition is small enough and a long time simulation is needed. For this reason, we introduce a parareal algorithm to enhance computational efficiency. The parareal algorithm was proposed by Lions, Maday et al in [24]. Its core idea is to divide the entire time interval into subintervals and compute simultaneously and independently on each subinterval. There has been many research work on the analysis and applications of the parareal method, see [20, 2, 31, 27, 32]. However, the existing literature is based on Euler or Runge-Kutta method as fine solver in each subinterval, which could be time consuming if one takes small time steps. To address this issue, we further introduce the waveform relaxation (WR) method via the diagonalization technique based on all-at-once algorithm[23, 18, 3, 30] as the fine solver in the parareal framework. All-at-once algorithm is a global method that solves the problem over the entire time interval simultaneously instead of solving it step by step. It naturally fits for parallel computation and can significantly improve computational efficiency.

The main contributions of this paper are as follows.

- The parareal algorithm for solving the partially explicit temporal splitting scheme is presented. The convergence of the proposed algorithm is proved.
- The WR method via the diagonalization technique based on the all-at-

once method is introduced into the parareal algorithm to speed up the computation for the fine propagator.

- The proposed algorithm can achieve high accuracy, and convergence can be reached with a small number of iterations. As the number of coarse interval (processors) increases, the number of iteration needed to achieve the error tolerance is quite stable, thus significantly saving the computational cost with parallel computation.

The rest of this paper is organized as follows. In section 2, we present preliminaries. In section 3, we give in detail the construction of multiscale spaces. The parareal all-at-once algorithm and the error estimate of full discretization are given in section 4. Numerical experiments and conclusion are given in section 5 and section 6, respectively.

2 Preliminaries

In this paper, we consider the diffusion equation of the form

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot (\kappa \nabla u) = f, \text{ in } \Omega \times (0,T] \\ \qquad\qquad\qquad\quad u = 0, \text{ on } \partial\Omega \times (0,T] \\ \qquad\qquad\qquad\quad u = u_0, \text{ on } \partial\Omega \times \{0\} \end{cases} \tag{2.1}$$

where Ω is a bounded domain, $\kappa \in L^\infty(\Omega)$ is a high contrast parameter and $f \in L^2(0,T;L^2(\Omega))$ is the source term.

We first present the fully-discretized problem for (2.1), with finite element method in spatial discretization and backward Euler method for temporal discretization. Next, we derive the partially explicit temporal splitting scheme. Then we briefly introduce the framework of the parareal algorithm.

We now introduce some notations. Denote by $(u,v) := \int_\Omega uvd\Omega$ the inner product of $L^2(\Omega)$ whose norm is defined by $\|\cdot\|_\Omega$. We use $H_0^1(\Omega)$ to denote the Sobolev spaces with zero boundary values. To simplify the notation, Ω may be dropped in the notations. We denote C a generic positive constant independent of any function and of any discretization parameters.

We write the problem (2.1) in the weak formulation: find $u(\cdot,t) \in H_0^1(\Omega)$ such that

$$(\frac{\partial u}{\partial t}, v) + a(u,v) = (f,v), \ \forall v \in H_0^1(\Omega),$$

where the bilinear form $a(\cdot,\cdot)$ is given by

$$a(u,v) = \int_\Omega \kappa \nabla u \cdot \nabla v,$$

and define the energy norm $\| u \|_a = a(u,u)^{\frac{1}{2}}$.

Consider a coarse spatial partition $\mathcal{T}_H$ of Ω with mesh size H, we will construct multiscale basis functions on $\mathcal{T}_H$ and form a multiscale space V_H which

has good approximation power. For the approximation space $V_H \in H_0^1(\Omega)$, the semi-discretization in space leads to seeking $u_H(t) \in V_H$ such that

$$\begin{aligned} &(\frac{\partial u_H(t)}{\partial t}, v) + a(u_H(t), v) = (f, v),\ \forall v \in V_H,\ 0 < t \leqslant T, \\ &u_H(0) = u_{H,0}. \end{aligned} \tag{2.2}$$

where $u_{H,0}$ is the projection of u_0 in V_H.

Let Δt be the time step size and $t_n = n\Delta t, n = 0, 1, \cdots, N, T = N\Delta t$. Then the full discretization with Backward Euler scheme reads

$$(\frac{u_H^{n+1} - u_H^n}{\Delta t}, v) + a(u_H^{n+1}, v) = (f^{n+1}, v),\ \forall v \in V_H, \tag{2.3}$$

where $u_H^{n+1} \approx u(\cdot, t_{n+1})$.

2.1 Partially explicit temporal splitting scheme

Now we introduce the partially explicit temporal splitting scheme [8]. Assume that V_H can be decomposed into two subspaces $V_{H,1}$ and $V_{H,2}$, that is

$$V_H = V_{H,1} + V_{H,2}.$$

Then a partially explicit temporal splitting scheme reads: finding $\{u_n\}_{n=1}^N \in V_{H,1}$ and $\{w_n\}_{n=1}^N \in V_{H,2}$ such that

$$(\frac{u_{n+1} - u_n}{\Delta t}, v_1) + (\frac{w_n - w_{n-1}}{\Delta t}, v_1) + a(u_{n+1} + w_n, v_1) = (f^{n+1}, v_1), \tag{2.4}$$

$$(\frac{w_{n+1} - w_n}{\Delta t}, v_2) + (\frac{u_n - u_{n-1}}{\Delta t}, v_2) + a(u_{n+1} + w_n, v_2) = (f^{n+1}, v_2), \tag{2.5}$$

$\forall v_1 \in V_{H,1}, \forall v_2 \in V_{H,2}$. Initial conditions are projected onto corresponding subspaces. Thus, the solution at time step $n+1$ will be $u_H^{n+1} = u_{n+1} + w_{n+1}$.

It has been proved in [8] that with a suitable choice of $V_{H,2}$ the partially explicit temporal splitting scheme is stable.

2.2 Parareal algorithm

To enhance the computational efficiency of the partially explicit temporal splitting scheme (2.4)-(2.5), we will introduce a temporal parallel algorithm. We first describe the basic flow of the parareal algorithm by considering the following initial value problem

$$\begin{aligned} &\frac{du}{dt} = F(t, u),\ t \in (0, T], \\ &u(0) = u_0, \end{aligned} \tag{2.6}$$

We first divide $(0, T]$ into N elements as described in the (2.3). Let $\mathcal{F}$ be a fine solver that can achieve desired accuracy but has a high computational cost. We

also introduce a coarse solver $\mathcal{G}$, which have lower accuracy but provide results at a lower computational costs compared to the fine solver. For example, one can use the same numerical scheme for both $\mathcal{F}$ and $\mathcal{G}$ but $\mathcal{F}$ using a small step size δt, while $\mathcal{G}$ is utilizing a bigger step size Δt ($\Delta t \gg \delta t$).

Denote by $\mathcal{F}_{\Delta t}(u, t_n, t_{n+1})$ the result obtained by integrating u from t_n to t_{n+1} using $\mathcal{F}_{\delta t}$ (fine solver with time step size δt) and $\mathcal{G}_{\Delta t}(u, t_n, t_{n+1})$ denoting the similar integration forward in time using the coarse solver. At the zero iteration of the parareal method, we get $\{u_n^0\}_{n=0}^N$ using the coarse solver $\mathcal{G}$. Denote by u_n^k the approximation for $u(t_n)$ at the kth iteration. Then the solution of the $k+1$th iteration is obtained from the following formula: for all $0 \leqslant n \leqslant N-1$ and $k = 0, 1, 2, \cdots$,

$$u_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(u_n^{k+1}, t_n, t_{n+1}) + \mathcal{F}_{\Delta t}(u_n^k, t_n, t_{n+1}) - \mathcal{G}_{\Delta t}(u_n^k, t_n, t_{n+1}). \tag{2.7}$$

One can terminate the parareal algorithm if the maximum number of iterations is met ($k = n_{\max}$), or if

$$\| u_n^k - u_n^{k-1} \|_\infty \leqslant \epsilon, \tag{2.8}$$

where ϵ is a given tolerance.

Notice that the evolution of the fine solver $\mathcal{F}$ only requires the initial value u_n^k, which depends on the previous iteration. Therefore, for each k, the $\mathcal{F}$ can be computed in parallel.

The solution of (2.7) converges under suitable assumptions, i.e., $u_n^k \to u_n^*$, where u_n^* is the solution obtained from $\mathcal{F}$ with time step δt throughtout the whole temporal domain.

3 Multiscale space construction

In this section, we first briefly describe the construction of multiscale spaces based on CEM-GMsFEM [6, 8, 7]. We then present the construction of the spaces $V_{H,1}$ and $V_{H,2}$ based on nonlocal multicontinuum (NLMC) method [9, 34].

3.1 CEM-GMsFEM

Denote by $\{K_i\}$ the set of coarse blcoks in $\mathcal{T}_H$, and denote $V = H_0^1(\Omega)$. For each $K_i \in \mathcal{T}_H$, we have to build a collection of auxiliary based in $V(K_i)$, where $V(K_i)$ be the restriction of V on K_i. We solve the following eigenvalue problem

$$\int_{K_i} \kappa \nabla \psi_j^{(i)} \cdot \nabla v = \lambda_j^{(i)} s_i(\psi_j^{(i)}, v), \ \forall v \in V(K_i), \tag{3.1}$$

where

$$s_i(u, v) = \int_{K_i} \widetilde{\kappa} u v, \ \widetilde{\kappa} = \kappa \sum_i | \nabla \chi_i |^2 \ \text{ or } \widetilde{\kappa} = \kappa H^{-2},$$

$\{\chi_i\}$ is a partition of unity functions corresponding to an overlapping partition of the domain. Then we collect the first J_i eigenfunctions corresponding to the first J_i smallest eigenvaules to form the auxiliary spaces

$$V_{\text{aux}}^{(i)} := \text{span}\{\psi_j^{(i)} : 1 \leqslant j \leqslant J_i\}.$$

Define a projection operator $\Pi : L^2(\Omega) \to V_{\text{aux}} \subset L^2(\Omega)$

$$s(\Pi u, v) = s(u, v),\ \forall v \in V_{\text{aux}} := \sum_{i=1}^{N_e} V_{\text{aux}}^{(i)},$$

where $s(u, v) = \sum_{i=1}^{N_e} s_i(u|_{K_i}, v|_{K_i})$ and N_e being the number of coarse elements. Define K_i^+ be an oversampling domain obtained by enlarging K_i by a few coarse grid layers. For each $\psi_j^{(i)}$, we search for a local basis function $\phi_j^{(i)} \in V(K_i^+)$ such that for some $\mu_j^{(i)} \in V(K_i^+)$

$$\begin{aligned} a(\phi_j^{(i)}, v) + s(\mu_j^{(i)}, v) = 0,\ \forall v \in V(K_i^+), \\ s(\phi_j^{(i)}, \nu) = s(\psi_j^{(i)}, \nu),\ \forall \nu \in V_{\text{aux}}(K_i^+). \end{aligned} \tag{3.2}$$

Then we define the space V_{cem} as

$$V_{cem} := \text{span}\{\phi_j^{(i)} : 1 \leqslant i \leqslant N_e, 1 \leqslant j \leqslant J_i\}. \tag{3.3}$$

Thus we can choose V_{cem} to be $V_{H,1}$ and construct a complementary space $V_{H,2}$[5, 10].

3.2 Construction of two multiscale subspaces

For channelized/fractured media, the construction of multiscale subspaces can be simplified. That is, denote by the computational domain $\Omega = \Omega_m \bigoplus_{l=1}^{s} d_l \Omega_{f,l}$, where m and f denote the matrix and fractures. In the fracture regions $\Omega_{f,l}$, the scalar d_l and s denote the aperture and the number of the discrete fracture networks, respectively. Since the value of the permeabilities in the matrix and fracture regions can differ in magnitudes, thus we can construct constraint energy minimizing basis functions via NLMC, such that the obtained basis functions can automatically separate continua such as matrices and fracture. Specifically, for a given coarse block K_i, we use constants for each individual fracture network and then a constant for the matrix to form a auxiliary space. That is to say, for any coarse block K_i, we write $K_i = K_{i,f} \cup K_{i,m}$, where $K_{i,f} := \{f_j^{(i)}, j = 1, \cdots, m_i\}$ is the high-contrast channelized region, m_i is the number of non-connected fractures in K_i, $K_{i,m}$ is its complement in K_i. Then we define two auxiliary spaces

$$V_{\text{aux},1}^{(i)} := \text{span}\{\phi_{\text{aux},k}^{(i)} \mid \phi_{\text{aux},k}^{(i)} = 0 \text{ in } K_{i,m}, \phi_{\text{aux},k}^{(i)} = \delta_{jk} \text{ in } f_j^{(i)}, k = 1, \cdots, m_i\}, \tag{3.4}$$

$$V_{\text{aux},2}^{(i)} := \text{span}\{\phi_{\text{aux},0}^{(i)} \mid \phi_{\text{aux},0}^{(i)} = 1 \text{ in } K_{i,m}, \phi_{\text{aux},0}^{(i)} = 0 \text{ in } K_{i,f}\}. \tag{3.5}$$

Then the NLMC basis functions are obtained by finding $\psi_m^{(i)} \in V_0(K_i^+)$ and $\mu_0^{(j)}, \mu_n^{(j)} \in \mathbb{R}$ from the following localized constraint energy minimizing problem

$$\begin{cases} a(\psi_m^{(i)}, v) + \sum\limits_{K_j \subset K_i^+} (\mu_0^{(j)} \int_{K_{j,m}} v + \sum\limits_{1 \leqslant n \leqslant m_j} \mu_n^{(j)} \int_{f_n^{(j)}} v) = 0, \ \forall v \in V_0(K_i^+), \\ \int_{K_{j,m}} \psi_m^{(i)} = \delta_{ij}\delta_{m0}, \ \forall K_j \subset K_i^+, \\ \int_{f_n^{(i)}} \psi_m^{(i)} = \delta_{ij}\delta_{mn}, \ \forall f_n^{(j)} \in \mathcal{F}_j, \ \forall K_j \subset K_i^+. \end{cases} \tag{3.6}$$

Then the NLMC basis functions are $\{\psi_m^{(i)}, 0 \leqslant m \leqslant m_i, 1 \leqslant i \leqslant N_e\}$. The above problems are posed in infinite-dimensional spaces, but for numerical computations, we solve the discretized the system on the fine grid using standard finite elements to obtain the solutions and use them as our basis.

Denote the average of all NLMC basis by

$$\overline{\psi} := \frac{1}{L} \sum_{i=1}^{N_e} \sum_{m=0}^{m_i} \psi_m^{(i)}, \quad L = \sum_{i=1}^{N} m_i. \tag{3.7}$$

We then let $\widetilde{\psi}_m^{(i)} = \psi_m^{(i)} - \frac{s(\psi_m^{(i)}, \overline{\psi})}{s(\overline{\psi}, \overline{\psi})} \overline{\psi}, 0 \leqslant m \leqslant m_i, 1 \leqslant i \leqslant N_e$. In order to simplify the notations, we omit the double script in $\widetilde{\psi}_m^{(i)}$ and denote the set of bases by $\{\widetilde{\psi}_k, k = 1, \cdots, L\}$. Thus, we define the space $V_{H,1}$ as

$$V_{H,1} = \text{span}\{\widetilde{\psi}_k, 1 \leqslant k \leqslant L-1\}.$$

The basis functions corresponding to the matrix and the basis $\overline{\psi}$ will be included in the second subspace $V_{H,2}$, that is,

$$V_{H,2} = \text{span}\{\overline{\psi}, \psi_0^{(i)}, 1 \leqslant i \leqslant N_c\}.$$

By this construction, $V_{H,1}$ contains a basis representing the high-contrast fractures only, and $V_{H,2}$ includes a basis representing the background matrix and the constant basis.

Next, we will introduce some notations. Let $\dim(V_{H,1}) = d_1$, $\dim(V_{H,2}) = d_2$, $\dim(V_H) = D$, and let $\Psi_1 \in \mathbb{R}^{D \times d_1}$ and $\Psi_2 \in \mathbb{R}^{D \times d_2}$ be the matrices whose columns are the bases of $V_{H,1}$ and $V_{H,2}$, respectively. Denote M_f and A_f be the fine scale mass matrix and stiffness matrix, define the following coarse scale matrices

$$\begin{aligned} M_{11} &= \Psi_1^T M_f \Psi_1, \ A_{11} = \Psi_1^T A_f \Psi_1, \\ M_{22} &= \Psi_2^T M_f \Psi_2, \ A_{22} = \Psi_2^T A_f \Psi_2, \\ M_{12} &= \Psi_1^T M_f \Psi_2, \ A_{12} = \Psi_1^T A_f \Psi_2, \\ F_1^n &= \Psi_1^T f^n, \qquad F_2^n = \Psi_2^T f^n. \end{aligned} \tag{3.8}$$

4 Parallel in time for partially explicit temporal splitting scheme

In this section, we describe in detail parareal algorithm for the partially explicit temporal splitting scheme. First, we introduce the WR method via the diagonalization technique based on the all-at-once system for the partially explicit scheme in subsection 4.1 and give the convergence of the method. Then we propose our main algorithm, the parareal all-at-once partially explicit temporal splitting algorithm, in section 4.2, where we adopt the WR method via diagonalization technique as the fine solver. Finally, we carry out the error analysis for the proposed algorithm.

4.1 WR method via diagonalization for all-at-once system

Let us look back at the splitting scheme (2.4)-(2.5), and solve it using the iterative all-at-once method. We write

$$(\frac{u_{n+1}^j - u_n^j}{\Delta t}, v_1) + a(u_{n+1}^j, v_1) = (f^{n+1}, v_1) - (\frac{w_n^{j-1} - w_{n-1}^{j-1}}{\Delta t}, v_1) - a(w_n^{j-1}, v_1), \tag{4.1}$$

$$(\frac{w_{n+1}^j - w_n^j}{\Delta t}, v_2) + (\frac{u_n^j - u_{n-1}^j}{\Delta t}, v_2) + a(u_{n+1}^j + w_n^j, v_2) = (f^{n+1}, v_2) \tag{4.2}$$

where $v_1 \in V_{H,1}, v_2 \in V_{H,2}$ and j will be the iteration index. Let $U^j = (u_1^j, u_2^j, \cdots, u_N^j)$ and $W^j = (w_1^j, w_2^j, \cdots, w_N^j)$, therefore (4.1)-(4.2) is a direct discretization of the following scheme

$$(\partial_t U^j(t), v_1) + a(U^j(t), v_1) = -(\partial_t U^{j-1}(t), v_1) - a(W^{j-1}(t), v_1), \ \forall v_1 \in V_{H,1}, \tag{4.3}$$

$$(\partial_t W^j(t), v_2) + a(W^j(t), v_2) = -(\partial_t U^j(t), v_2) - a(U^j(t), v_2), \ \forall v_2 \in V_{H,2}. \tag{4.4}$$

Notice that the (4.1) can be written as an all-at-once system

$$(B \otimes M_{11} + I_t \otimes A_{11})U^j = F \tag{4.5}$$

with initial condition

$$u_0^j = u_0 + \alpha(u_N^j - u_N^{j-1}),$$

where B is a periodic-like matrix with the parameter $\alpha \in (0, 1)$

$$B = \frac{1}{\Delta t} \begin{pmatrix} 1 & & & -\alpha \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \tag{4.6}$$

and $F = (\widetilde{f}^1, \widetilde{f}^2, \cdots, \widetilde{f}^N)$, and

$$\widetilde{f}^1 = F_1^1 - \frac{1}{\Delta t} M_{12} w_0^{j-1} - A_{12} w_0^{j-1} + \frac{1}{\Delta t} M_{11}(u_0 - \alpha u_N^{j-1}),$$

$$\widetilde{f}^s = F_1^s - \frac{1}{\Delta t} M_{12}(w_s^{j-1} - w_{s-1}^{j-1}) - A_{12} w_s^{j-1},$$

for $s = 2, 3, \cdots, N$. Now, if the matrix B is diagonalizable, such as

$$B = SDS^{-1}, D = \mathrm{diag}(d_1, d_2, \cdots, d_N),$$

we can factorize the coefficient matrix in (4.5) as follows

$$B \otimes M_{11} + I_t \otimes A_{11} = (S \otimes I_M)(D \otimes M_{11} + I_t \otimes A_{11})(S^{-1} \otimes I_M), \tag{4.7}$$

where I_M is a identity matrix. This implies that we can solve (4.5) at the j-th iteration in three steps

$$\begin{aligned} &(a)\ (S \otimes I_M)P = F^j, \\ &(b)\ (D \otimes M_{11} + I_t \otimes A_{11})Q = P, \\ &(c)\ (S^{-1} \otimes I_M)U^j = Q. \end{aligned} \tag{4.8}$$

Note that the matrix S can be further decomposed into $S = \Lambda V$, where $\Lambda = \mathrm{diag}\{1, \alpha^{-\frac{1}{N}}, \cdots, \alpha^{-\frac{N-1}{N}}\}$, V is the discrete Fourier matrix. Thus, the Fast Fourier Transform (FFT) can be employed to speed up the implementation of (4.8). In addition, the second step of (4.8) is to solve N independent equations, thus it can be done in parallel.

The problem (4.1)-(4.2) will be solved with the following steps:

1. Solve (4.5) by (4.8), obtain $U^j = (u_1^j, \cdots, u_N^j)$, which is the part of the solution in $V_{H,1}$ at all time steps.

2. Plug the solution U in (4.2), solve for $W^j = (w_1^j, \cdots, w_N^j)$ in a sequential manner.

3. Iterate the above process until converge.

The following theorem gives the convergence result for the WR method at each subinterval (t_n, t_{n+1}). We remark that we will adopt the WR method as the fine propagator at each subinterval (t_n, t_{n+1}) within the parareal framework. It is trivial to extend the following results on the whole temporal domain $(0, T]$.

Theorem 4.1. *Let $u(t)$ be the exact solution of (2.1) and satisfy $u(t) = U(t) + W(t)$, $U^j(t)$ and $W^j(t)$ are the solution of (4.3)-(4.4) at j-th iteration respectively. Then it holds for $j = 1, 2, \cdots$ and $n = 0, 1, 2, \cdots, N-1$ that*

$$\begin{aligned} &\sup_{t \in (t_n, t_{n+1})} \| U^j(t) - U(t) \| + \sup_{t \in (t_n, t_{n+1})} \| W^j(t) - W(t) \| \\ &\leqslant C\gamma^{2(j-1)} \sup_{t \in (t_n, t_{n+1})} \| A_{22}^{-1} \partial_t (W^0(t) - W(t)) + (W^0(t) - W(t)) \|_\infty, \end{aligned} \tag{4.9}$$

where γ is defined by

$$\gamma = \frac{|(v_1, v_2)|}{\| v_1 \| \| v_2 \|}, \ \forall v_1 \in V_{H,1}, \ \forall v_2 \in V_{H,2}. \tag{4.10}$$

Proof: We consider

$$\begin{aligned}
&(\partial_t U(t), v_1) + a(U(t), v_1) = -(\partial_t U(t), v_1) - a(W(t), v_1), \ \forall v_1 \in V_{H,1}, \\
&(\partial_t W(t), v_2) + a(W(t), v_2) = -(\partial_t U(t), v_2) - a(U(t), v_2), \ \forall v_2 \in V_{H,2}
\end{aligned}$$

We now define $P_1 : V_{H,2} \to V_{H,1}$, $\Pi_1 : V_{H,2} \to V_{H,1}$ and $P_2 : V_{H,1} \to V_{H,2}$, $\Pi_2 : V_{H,1} \to V_{H,2}$ such that

$$\begin{aligned}
(P_1 v_2, v_1) =& (v_2, v_1), \ \forall v_1 \in V_{H,1}, \ (P_2 v_1, v_2) = (v_1, v_2), \ \forall v_2 \in V_{H,2}, \\
a(\Pi_1 v_2, v_1) =& a(v_2, v_1), \ \forall v_1 \in V_{H,1}, \ a(\Pi_2 v_1, v_2) = a(v_1, v_2), \ \forall v_2 \in V_{H,2},
\end{aligned}$$

and define the errors

$$e_u^j(t) = U^j(t) - U(t), \ e_w^j(t) = W^j(t) - W(t).$$

Then for $\forall v_1 \in V_{H,1}$ and $\forall v_2 \in V_{H,2}$, we have

$$(\partial_t e_u^j(t), v_1) + a(e_u^j(t), v_1) = -(\partial_t e_w^{j-1}(t), P_2 v_1) - a(e_w^{j-1}(t), \Pi_2 v_1), \tag{4.11}$$

$$(\partial_t e_w^j(t), v_2) + a(e_w^j(t), v_2) = -(\partial_t e_u^j(t), P_1 v_2) - a(e_u^j(t), \Pi_1 v_2). \tag{4.12}$$

with initial condition $e_u^j(0) = e_w^j(0) = 0$. Then from (4.11)-(4.12) we obtain

$$\begin{aligned}
&(\partial_t e_w^j(t), v_2) + a(e_w^j(t), v_2) + a(e_u^j(t), \Pi_1 v_2) \\
=& a(e_u^j(t), P_1 v_2) + (\partial_t e_w^{j-1}(t), P_2 P_1 v_2) + a(e_w^{j-1}(t), \Pi_2 P_1 v_2).
\end{aligned} \tag{4.13}$$

From (4.12), a directly calculation yields the following recurrence relation

$$\begin{aligned}
&- (\partial_t e_w^{j-1}(t), v_2) \\
=& a(e_w^{j-1}(t), v_2) + (\partial_t e_u^{j-1}(t), P_1 v_2) + a(e_u^{j-1}(t), \Pi_1 v_2) \\
=& a(e_w^{j-1}(t), v_2) - a(e_u^{j-1}(t), P_1 v_2) - (\partial_t e_w^{j-2}(t), P_2 P_1 v_2) \\
&- a(e_w^{j-2}(t), \Pi_2 P_1 v_2) + a(e_u^{j-1}(t), \Pi_1 v_2) \\
=& a(e_w^{j-1}(t), v_2) - a(e_w^{j-2}(t), \Pi_2 P_1 v_2) + a(e_u^{j-1}(t), \Pi_1 v_2 - P_1 v_2) \\
&- (\partial_t e_w^{j-2}(t), P_2 P_1 v_2) \\
=& a(e_w^{j-1}(t), v_2) + a(e_u^{j-1}(t), \Pi_1 v_2 - P_1 v_2) + a(e_u^{j-1}(t), P_2 P_1 v_2 - \Pi_2 P_1 v_2) \\
&+ a(e_u^{j-2}(t), \Pi_1 P_2 P_1 v_2 - P_1 P_2 P_1 v_2) - a(e_w^{j-3}(t), \Pi_2 P_1 P_2 P_1 v_2) \\
&- (\partial_t e_w^{j-3}(t), P_2 P_1 P_2 P_1 v_2).
\end{aligned}$$

Denote by $P_{21} = P_2 P_1$ and $P_{12} = P_1 P_2$, then the following recurrence relation can be written as

$$\begin{aligned}
&- (\partial_t e_w^{j-1}(t), v_2) = a(e_w^{j-1}(t), v_2) + a(e_u^{j-1}(t), \Pi_1 v_2 - P_1 v_2) \\
&+ \sum_{i=0} \left(a(e_w^{j-2-i}(t), P_{21}^{i+1} v_2 - \Pi_2 P_1 P_{21}^i v_2) + a(e_u^{j-2-i}(t), \Pi_1 P_{21}^{i+1} v_2 - P_1 P_{21}^{i+1} v_2) \right) \\
&- a(e_w^0(t), \Pi_2 P_1 P_{21}^{j-2} v_2) - (\partial_t e_w^0(t), P_{21}^{j-1} v_2).
\end{aligned}$$

Substitute the above equation into (4.11) and (4.13) respectively, we have

$$\begin{aligned}
&(\partial_t e_u^j(t), v_1) + a(e_u^j(t), v_1) \\
=&\sum_{i=0} \big(a(e_w^{j-1-i}(t), (P_2 - \Pi_2)P_{12}^i v_1) + a(e_u^{j-1-i}(t), (\Pi_1 - P_1)P_2 P_{12}^i v_1)\big) \\
&- a(e_w^0(t), \Pi_2 P_{12}^{j-1} v_1) - (\partial_t e_w^0(t), P_2 P_{12}^{j-1} v_1)
\end{aligned} \tag{4.14}$$

and

$$\begin{aligned}
&(\partial_t e_w^j(t), v_2) + a(e_w^j(t), v_2) + a(e_u^j(t), \Pi_1 v_2) \\
=&\sum_{i=0} \big(a(e_w^{j-1-i}(t), (\Pi_2 - P_2)P_1 P_{21}^i v_2) + a(e_u^{j-1-i}(t), (P_1 - \Pi_1)P_{21}^{i+1} v_2)\big) \\
&+ a(e_w^0(t), \Pi_2 P_1 P_{21}^{j-1} v_2) + (\partial_t e_w^0(t), P_{21}^j v_2).
\end{aligned} \tag{4.15}$$

We then write the above operator into the matrix form

$$\begin{aligned}
&\partial_t e_u^j(t) + A_{11} e_u^j(t) \\
=&\sum_{i=0} \big(P_{12}^{i*}(P_2^* - \Pi_1)A_{22} e_w^{j-1-i}(t) + P_{12}^{i*} P_2^* (\Pi_2 - P_1^*) A_{11} e_u^{j-1-i}(t)\big) \\
&- (P_{12}^{j-1})^* \Pi_1 A_{22} e_w^0(t) - (P_{12}^{j-1})^* P_2^* \partial_t e_w^0(t)
\end{aligned} \tag{4.16}$$

and

$$\begin{aligned}
&\partial_t e_w^j(t) + A_{22} e_w^j(t) + (\Pi_2 - P_1^*) A_{11} e_u^j(t) \\
=&\sum_{i=0} \big(P_{21}^{i*} P_1^* (P_2^* - \Pi_1) A_{22} e_w^{j-1-i}(t) + (P_{21}^{i+1})^* (\Pi_2 - P_1^*) A_{11} e_u^{j-1-i}(t)\big) \\
&+ (P_{21}^{j-1})^* P_1^* \Pi_1 A_{22} e_w^0(t) + (P_{21}^j)^* \partial_t e_w^0(t).
\end{aligned} \tag{4.17}$$

Denote by

$$\begin{aligned}
f_w^j(t) &= \sum_{i=0} P_{12}^{i*}(P_2^* - \Pi_1) A_{22} e_w^{j-1-i}(t), \\
f_u^j(t) &= \sum_{i=0} P_{12}^{i*} P_2^* (\Pi_2 - P_1^*) A_{11} e_u^{j-1-i}(t),
\end{aligned}$$

then with the above two notations, we can rewrite (4.16) and (4.17) as follows

$$\partial_t e_u^j(t) + A_{11} e_u^j(t) = f_w^j(t) + f_u^j(t) - (P_{12}^{j-1})^* \Pi_1 A_{22} e_w^0(t) - (P_{12}^{j-1})^* P_2^* \partial_t e_w^0(t) \tag{4.18}$$

and

$$\begin{aligned}
&\partial_t e_w^j(t) + A_{22} e_w^j(t) + (\Pi_2 - P_1^*) A_{11} e_u^j(t) \\
=&P_1^* f_w^j(t) + P_1^* f_u^j(t) - P_1^* (P_{12}^{j-1})^* \Pi_1 A_{22} e_w^0(t) - P_1^* (P_{12}^{j-1})^* P_2^* \partial_t e_w^0(t).
\end{aligned} \tag{4.19}$$

We consider $\| A_{11}^{-1}P_2^*A_{22} \| \leqslant C$ and $\| A_{22}^{-1}P_1^*A_{11} \| \leqslant C$, where C is independent of contrast, and let λ_p and ϕ_p be the eigenvalues and eigenfunctions such that

$$P_{12}^*\phi_p = \lambda_p\phi_p,\ p = 1, 2, \cdots .$$

Thus, for $\widetilde{\phi}_p = A_{11}^{-1}\phi_p$, we have $A_{11}^{-1}P_{12}^*A_{11}\widetilde{\phi}_p = \lambda_i\widetilde{\phi}_p$. Let $\widetilde{\Phi}$ and D be the matrix $\widetilde{\Phi} = [\widetilde{\phi}_1, \widetilde{\phi}_2, \cdots]$ and $D = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots\}$, then

$$\begin{aligned}
f_w^j(t) &= \sum_{i=0} P_{12}^{i*}(P_2^* - \Pi_1)A_{22}e_w^{j-1-i}(t) \\
&= \sum_{i=0} P_{12}^{i*}A_{11}(A_{11}^{-1}P_2^*A_{22} - \Pi_1)e_w^{j-1-i}(t) \\
&= \sum_{i=0} P_{12}^{i*}A_{11}\widetilde{\Phi}\widetilde{\Phi}^{-1}(A_{11}^{-1}P_2^*A_{22} - \Pi_1)e_w^{j-1-i}(t) \\
&= \sum_{i=0} A_{11}\widetilde{\Phi}D^i\widetilde{\Phi}^{-1}(A_{11}^{-1}P_2^*A_{22} - \Pi_1)e_w^{j-1-i}(t)
\end{aligned}$$

Since $\lambda_p \leqslant \gamma^2, p = 1, 2, \cdots$, where $\gamma = \frac{|(v_1, v_2)|}{\|v_1\|\|v_2\|}$, then we have

$$\begin{aligned}
\| A_{11}^{-1}f_w^j(t) \| &= \| \sum_{i=0} \widetilde{\Phi}D^i\widetilde{\Phi}^{-1}(A_{11}^{-1}P_2^*A_{22} - \Pi_1)e_w^{j-1-i}(t) \| \\
&\leqslant C\sum_{i=0} \max_p\{\lambda_p\}^i \| e_w^{k-1-i} \| \\
&\leqslant C\frac{1}{1-\gamma^2} \max_{0\leqslant i\leqslant j-1} \| e_w^i(t) \| . \qquad (4.20)
\end{aligned}$$

Similarly we have

$$\| A_{11}^{-1}f_u^j(t) \| \leqslant C\frac{1}{1-\gamma^2} \max_{0\leqslant i\leqslant j-1} \| e_u^i(t) \|, \qquad (4.21)$$

$$\| A_{22}^{-1}P_1^*f_w^j(t) \| \leqslant C\frac{1}{1-\gamma^2} \max_{0\leqslant i\leqslant j-1} \| e_w^i(t) \|, \qquad (4.22)$$

$$\| A_{22}^{-1}P_1^*f_u^j(t) \| \leqslant C\frac{1}{1-\gamma^2} \max_{0\leqslant i\leqslant j-1} \| e_w^i(t) \| . \qquad (4.23)$$

Then we turn to estimate the last two terms of (4.16) and (4.17),we obtain

$$\begin{aligned}
&\| A_{11}^{-1}\left((P_{12}^{j-1})^*\Pi_1A_{22}e_w^0(t) + (P_{12}^{j-1})^*P_2^*\partial_te_w^0(t)\right) \| \\
\leqslant& C\gamma^{2(k-1)} \| A_{11}^{-1}(P_2^*\partial_te_w^0(t) + \Pi_1A_{22}e_w^0(t)) \|, \qquad (4.24) \\
&\| A_{22}^{-1}P_1^*\left((P_{12}^{j-1})^*\Pi_1A_{22}e_w^0(t) + (P_{12}^{j-1})^*P_2^*\partial_te_w^0(t)\right) \| \\
\leqslant& C\gamma^{2(k-1)} \| A_{22}^{-1}P_1^*(P_2^*\partial_te_w^0(t) + \Pi_1A_{22}e_w^0(t)) \| . \qquad (4.25)
\end{aligned}$$

Now we consider the following matrices

$$\widetilde{A} = \begin{pmatrix} A_{11} & O \\ (\Pi_2 - P_1^*)A_{11} & A_{22} \end{pmatrix},\ A = \begin{pmatrix} A_{11} & O \\ O & A_{22} \end{pmatrix}$$

then we have

$$\widetilde{A}^{-1} = \begin{pmatrix} A_{11}^{-1} & O \\ -A_{22}^{-1}(\Pi_2 - P_1^*) & A_{22}^{-1} \end{pmatrix}, \widetilde{A}^{-1}A = \begin{pmatrix} I & O \\ -A_{22}^{-1}(\Pi_2 - P_1^*)A_{11} & I \end{pmatrix}$$

Thus, we can easily get $\| \widetilde{A}^{-1}A \| \leqslant C$ and

$$\| \int_0^t e^{-(t-s)\tilde{A}} A \| = \| (I - e^{-t\tilde{A}})\widetilde{A}^{-1}A \| \leqslant \| I - e^{-t\tilde{A}} \| \| \widetilde{A}^{-1}A \| \leqslant Ct.$$

Therefore, given any $\Delta t > 0$, $t_n = n\Delta t$, and with the help of the (4.20)-(4.25), we have the following estimates

$$\begin{aligned} \sup_{t\in(t_n,t_{n+1})} \| e_u^j(t) \| \leqslant & \| e_u^j(t_n) \| + C\Delta t\gamma^{2(j-1)} \| A_{11}^{-1}(P_2^*\partial_t e_w^0(t) + \Pi_1 A_{22} e_w^0(t)) \| \\ & + C\Delta t \frac{1}{1-\gamma^2} \left(\max_{0\leqslant i\leqslant j-1} \sup_{t\in(t_n,t_{n+1})} \| e_w^i(t) \| + \max_{0\leqslant i\leqslant j-1} \sup_{t\in(t_n,t_{n+1})} \| e_u^i(t) \| \right), \end{aligned} \tag{4.26}$$

$$\begin{aligned} \sup_{t\in(t_n,t_{n+1})} \| e_w^j(t) \| \leqslant & \| e_w^j(t_n) \| + C\Delta t\gamma^{2(j-1)} \| A_{22}^{-1}P_1^*(P_2^*\partial_t e_w^0(t) + \Pi_1 A_{22} e_w^0(t)) \| \\ & + C\Delta t \frac{1}{1-\gamma^2} \left(\max_{0\leqslant i\leqslant j-1} \sup_{t\in(t_n,t_{n+1})} \| e_w^i(t) \| + \max_{0\leqslant i\leqslant j-1} \sup_{t\in(t_n,t_{n+1})} \| e_u^i(t) \| \right). \end{aligned} \tag{4.27}$$

We set $\mu = C\Delta t\frac{1}{1-\gamma^2} \leqslant 1$, then add (4.26) and (4.27). With the condition $\| A_{11}^{-1}P_2^*A_{22} \| \leqslant C$ and $\| A_{22}^{-1}P_1^*A_{11} \| \leqslant C$, we can get

$$\begin{aligned} & \sup_{t\in(t_n,t_{n+1})} \| e_u^j(t) \| + \sup_{t\in(t_n,t_{n+1})} \| e_w^j(t) \| \\ \leqslant & \mu \max_{0\leqslant i\leqslant j-1} \left(\sup_{t\in(t_n,t_{n+1})} \| e_u^i(t) \| + \sup_{t\in(t_n,t_{n+1})} \| e_w^i(t) \| \right) \\ & + C\sum_{i=0}^{1} \Delta t\mu^i\gamma^{2(j-1-i)} \sup_{t\in(t_n,t_{n+1})} \| A_{22}^{-1}\partial_t e_w^0(t) + e_w^0(t) \| \\ \leqslant & C\sum_{i=0}^{j-1} \Delta t\mu^i\gamma^{2(j-1-i)} \sup_{t\in(t_n,t_{n+1})} \| A_{22}^{-1}\partial_t e_w^0(t) + e_w^0(t) \|, \end{aligned}$$

for $n = 1, 2, \cdots, N-1$. Finally we consider $\Delta t \to 0$, thus $\mu \to 0$, then we get the desired result. □

4.2 Parareal all-at-once partially explicit temporal splitting algorithm

In this subsection, we describe in detail the use of parareal algorithm to solve the partially explicit temporal splitting scheme and give its specific algorithm.

We further divide each time slice $[t_n, t_{n+1}]$ into M subintervals with $\delta t = \Delta t/M$, $\widehat{t} = m\delta t$, $m = 0, 1, \cdots, M$. According to the framework of the parareal algorithm, we should first compute a series of initial solutions based on the coarse solver $\mathcal{G}$. To do this, we write the (2.4)-(2.5) as follows

$$(\frac{u_{n+1}^k - u_n^k}{\Delta t}, v_1) + (\frac{w_{n+1}^k - w_n^k}{\Delta t}, v_1) + a(u_{n+1}^k + w_n^k, v_1) = (f^{n+1}, v_1), \quad (4.28)$$

$$(\frac{w_{n+1}^k - w_n^k}{\Delta t}, v_2) + (\frac{u_{n+1}^k - u_n^k}{\Delta t}, v_2) + a(u_{n+1}^k + w_n^k, v_2) = (f^{n+1}, v_2), \quad (4.29)$$

where the subscript k denote the kth iteration in the parareal algorithm. Now, the splitting scheme need to be solved together at each new time step, i.e., for all $n = 0, 1, \cdots, N-1$ and $k = 0$, we have

$$\begin{pmatrix} u_{n+1}^0 \\ w_{n+1}^0 \end{pmatrix} = \mathcal{G}_{\Delta t}(\begin{pmatrix} u_n^0 \\ w_n^0 \end{pmatrix}, t_n, t_{n+1}). \quad (4.30)$$

In this way, we can compute the initial solution by (4.30). Then on each time slice $[t_n, t_{n+1}]$,$n = 0, 1, \cdots, N-1$, we use all-at-once method to solve (4.1)-(4.2), this leads to our main algorithm: the parareal all-at-once partially explicit temporal splitting algorithm.

We present in Algorithm 1 the main ingredient of the parareal all-at-once algorithm for (2.4)-(2.5), which include the parallel computation in Step 3 and the sequential propagation in Step 4.

4.3 Convergence of the main algorithm

This subsection is concerned with the convergence analysis for the Algorithm 1 in subsection 4.2. To begin with, we introduce some lemmas that will be used in theoretical proof.

Lemma 1. *Let u the be solution of (2.1), $u_H(t)$ be the solution of (2.2), if $f_t \in L^1(0,T;L^2(\Omega))$ and $u_{tt} \in L^1(0,T;L^2(\Omega))$, then there holds[21]*

$$\begin{aligned} \| u(\cdot,T) - u_H(T) \| \leqslant CH^2\Lambda^{-1}\kappa_0^{-\frac{1}{2}}(\max_{0\leqslant t\leqslant T} \| \kappa^{-\frac{1}{2}}(f - u_t) \| \\ + \| \kappa^{-\frac{1}{2}}(f_t - u_{tt}) \|_{L^1(0,T;L^2(\Omega))}) + \| u_0 - u_{H,0} \|, \end{aligned} \quad (4.31)$$

where $\kappa_0 = \max \kappa^{-\frac{1}{2}}$ and $\Lambda = \min_{1\leqslant j\leqslant N_e} \lambda_{J_i+1}^{(i)}$.

This lemma gives an error estimate in the spatial semi-discrete scheme.

Lemma 2. *The coarse solve $\mathcal{G}$ is Lipschitz, it holds[17]*

$$\| \mathcal{G}_{\Delta t}(u, t_n, t_{n+1}) - \mathcal{G}_{\Delta t}(v, t_n, t_{n+1}) \| \leqslant (1 + C\Delta t) \| u - v \|. \quad (4.32)$$

Lemma 3. *Define $\mathcal{S}_{\Delta t}(u, t_n, t_{n+1}) = \mathcal{F}_{\Delta t}(u, t_n, t_{n+1}) - \mathcal{G}_{\Delta t}(u, t_n, t_{n+1})$. Then it has following property[17]*

$$\| \mathcal{S}_{\Delta t}(u, t_n, t_{n+1}) \| \leqslant C(\Delta t)^{m+1} \| u \|. \quad (4.33)$$

Algorithm 1 Parareal all-at-once partially explicit temporal splitting algorithm

Input: initial date u_0, source term f, tolerance ϵ, coarse matrices (3.8), Ψ_1 and Ψ_2.

Result: $u_H^{n,k}$ for $n = 1, 2, \cdots, N$.

Compute a series of initial solution $\{u_n^0\}_{n=1}^N$ and $\{w_n^0\}_{n=1}^N$

$$\begin{pmatrix} u_{n+1}^0 \\ w_{n+1}^0 \end{pmatrix} = \mathcal{G}_{\Delta t}(\begin{pmatrix} u_n^0 \\ w_n^0 \end{pmatrix}, t_n, t_{n+1}),\ n = 0, 1, \cdots, N-1.$$

for $k = 1, 2, \cdots$, **do**

  Parallel compute $\widehat{u}_{n+1}^k$ and $\widehat{w}_{n+1}^k$ on each time slice $[t_n, t_{n+1}]$

$$\begin{pmatrix} \widehat{u}_{n+1}^k \\ \widehat{w}_{n+1}^k \end{pmatrix} = \mathcal{F}_{\Delta t}(\begin{pmatrix} u_n^{k-1} \\ w_n^{k-1} \end{pmatrix}, t_n, t_{n+1})$$

  Sequentially compute the corrected solution $\{u_n^{k+1}\}_{n=1}^N$ and $\{w_n^{k+1}\}_{n=1}^N$

$$\begin{pmatrix} u_{n+1}^{k+1} \\ w_{n+1}^{k+1} \end{pmatrix} = \mathcal{G}_{\Delta t}(\begin{pmatrix} u_n^{k+1} \\ w_n^{k+1} \end{pmatrix}, t_n, t_{n+1}) + \begin{pmatrix} \widehat{u}_{n+1}^k \\ \widehat{w}_{n+1}^k \end{pmatrix} - \mathcal{G}_{\Delta t}(\begin{pmatrix} u_n^k \\ w_n^k \end{pmatrix}, t_n, t_{n+1})$$

  Determine whether the given condition is met.

  if $\| u_n^k - u_n^{k-1} \| < \epsilon$ **then**

    return $u_H^{n,k} = u_n^k + w_n^k, n = 1, \cdots, N$

    break

  end if

end for

Remark 1 Notice that in parareal algorithm, fine solver $\mathcal{F}$ is usually considered exact, so m in Lemma 3 is the order of the coarse solver $\mathcal{G}$. Due to the fact that the coarse solver $\mathcal{G}$ is a first order scheme in our algorithm, thus we can safety replace m by 1 in the later analysis.

With the aid of above lemmas, we have the following error results in full discrete scheme.

Theorem 4.2. *Let u be the solution of (2.1), $u_H(t)$ be the solution of (2.2) and $u_H^{N,k}$ be the solution of (2.4)-(2.5) computed by Algorithm 1, then we have*

$$\begin{aligned}\| u(\cdot,T) - u_H^{N,k} \| \leqslant & CH^2\Lambda^{-1}\kappa_0^{-\frac{1}{2}}(\max_{0\leqslant t\leqslant T} \| \kappa^{-\frac{1}{2}}(f-u_t) \| \\ & + \| \kappa^{-\frac{1}{2}}(f_t-u_{tt}) \|_{L^1(0,T;L^2(\Omega))}) + \| u_0 - u_{H,0} \| + C(\Delta t)^k \| u_{H,0} \|\end{aligned}$$

Proof: Using triangle inequality

$$\begin{aligned}\| u(\cdot,T) - u_H^{N,k} \| =& \| u(\cdot,T) - u_H(T) + u_H(T) - u_H^{N,k} \| \\ \leqslant& \| u(\cdot,T) - u_H(T) \| + \| u_H(T) - u_H^{N,k} \| .\end{aligned} \tag{4.34}$$

Notice that for the first term of the right hand side(RHS) of (4.34), it can be estimated by Lemma 1, thus, we aim at estimating the second term. Following the framework of the Parareal algorithm, assume that the fine solver $\mathcal{F}$ is exact, i.e., $\forall n = 0, 1, \cdots, N-1$,

$$u_H(t_{n+1}) = \mathcal{F}_{\Delta t}(u_H(t_n), t_n, t_{n+1}).$$

Therefore, we have following equation

$$\begin{aligned}& u_H(T) \\ =& \mathcal{G}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) + \mathcal{F}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) - \mathcal{G}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) \\ =& \mathcal{G}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) + \mathcal{S}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T).\end{aligned}$$

On the other hand, by (2.7) we have

$$\begin{aligned}u_H^{N,k} =& \mathcal{G}_{\Delta t}(u_H^{N-1,k}, t_{N-1}, T) + \mathcal{F}_{\Delta t}(u_H^{N-1,k-1}, t_{N-1}, T) - \mathcal{G}_{\Delta t}(u_H^{N-1,k-1}, t_{N-1}, T) \\ =& \mathcal{G}_{\Delta t}(u_H^{N-1,k}, t_{N-1}, T) + \mathcal{S}_{\Delta t}(u_H^{N-1,k-1}, t_{N-1}, T).\end{aligned}$$

Then by triangle inequality again, the second term of RHS of the equation (4.34) can be written as follows

$$\begin{aligned}\| u_H(T) - u_H^{N,k} \| =& \| (\mathcal{G}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) - \mathcal{G}_{\Delta t}(u_H^{N-1,k}, t_{N-1}, T)) \\ & + (\mathcal{S}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) - \mathcal{S}_{\Delta t}(u_H^{N-1,k-1}, t_{N-1}, T)) \| \\ \leqslant& \theta(t_{N-1}) + \rho(t_{N-1}),\end{aligned} \tag{4.35}$$

where

$$\begin{aligned}\theta(t_{N-1}) =& \| \mathcal{G}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) - \mathcal{G}_{\Delta t}(u_H^{N-1,k}, t_{N-1}, T) \|, \\ \rho(t_{N-1}) =& \| \mathcal{S}_{\Delta t}(u_H(t_{N-1}), t_{N-1}, T) - \mathcal{S}_{\Delta t}(u_H^{N-1,k-1}, t_{N-1}, T) \| .\end{aligned}$$

With the help of the Lemma 2 and Lemma 3, we get the following result

$$\| u_H(T) - u_H^{N,k} \| \leqslant C(\Delta t)^k \| u_{H,0} \| . \tag{4.36}$$

Now we turn to prove (4.36) by mathematics induction. First, from the classical result in [18], we know it holds for $k = 1$, i.e.,

$$\| u_H(t_N) - u_H^{N,1} \| \leqslant C(\Delta t) \| u_{H,0} \| .$$

Assume it holds for k, then we prove it holds for $k+1$. For all $n = 0, 1, \cdots, N-1$, similar to (4.35) we have

$$\| u_H(t_{n+1}) - u_H^{n+1,k+1} \| \leqslant \theta(t_n) + \rho(t_n).$$

Utilizing Lemma 2 for $\theta(t_n)$ and Lemma 3 for $\rho(t_n)$, we obtain

$$\begin{aligned} &\| u_H(t_{n+1}) - u_H^{n+1,k+1} \| \\ \leqslant &(1 + C\Delta t) \| u_H(t_n) - u_H^{n,k+1} \| + C(\Delta t)^{k+1} \| u_{H,0} \| . \end{aligned} \tag{4.37}$$

Repeatedly utilizing the recursive (4.37) and sum all $n = 0, 1, \cdots, N-1$, we get the result

$$\| u_H(T) - u_H^{N,k+1} \| \leqslant C(\Delta t)^{k+1} \| u_{H,0} \| .$$

Thus we have completed the estimation of the second term of the RHS of (4.34). Finally, the desired result can be obtained directly from (4.31), (4.34) and (4.36).□

5 Numerical experiments

In this section, we perform some numerical experiments to verify the feasibility and effectiveness of the Algorithm 1. In all examples, we let $\Omega = [0, 1] \times [0, 1]$ and $T = 0.005$. The coarse scale and fine scale spatial mesh size are $H = \frac{1}{10}$ and $h = \frac{1}{100}$. We consider zero Dirichlet boundary conditions and zero initial conditions. The coarse scale and fine scale time step size satisfy the relation $\delta t = \frac{\Delta t}{N}$, i.e., $N = M$. In all examples, we set the tolerance $\epsilon = 10^{-14}$. The relative error is defined as follows:

$$\frac{\| u_h^N - u_H^N \|_{L^2(\Omega)}}{\| u_h^N \|_{L^2(\Omega)}} \tag{5.1}$$

where u_h denote the reference solution obtained by finite element method in space and WR method in time.

Example 1 The medium parameter κ and the source term f are shown in Figure 1. As we see that the permeability field is heterogeneous with high contrast. The contrast is 10^4. The reference solution and the solution obtained from Algorithm 1 at $t = T$ are presented in Figure 2.

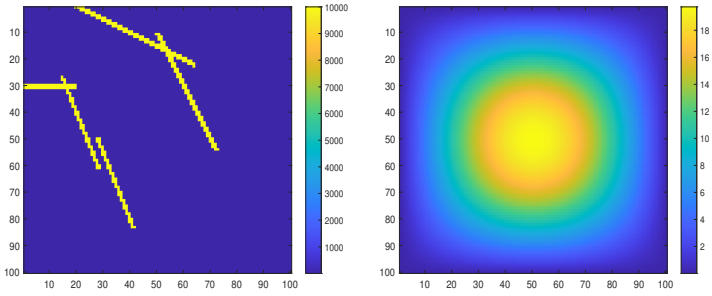

Figure 1: Example 1. **Left:** κ; **Right:** f

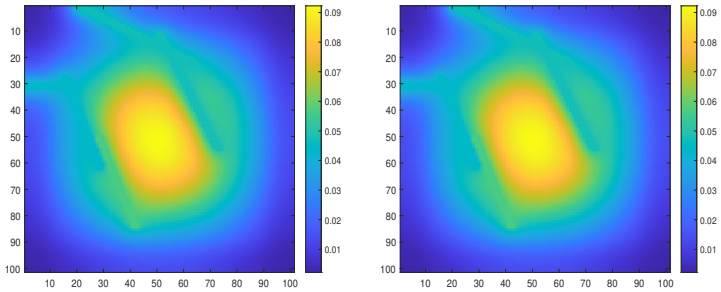

Figure 2: Example 1. **Left:** reference solution; **Right:** Algorithm 1 solution.

We take $N = 50$ and $\alpha = 0.5$ as an example. In the left of Figure 3, we give the max differences computed by (2.8). It is clear that as the number of iterations increase, the max differences becomes smaller and eventually stabilises. The convergence rate of the Algorithm 1 is shown in the right of Figure 3. It is evident that the error converges quickly. Table 1 gives the number of iterations needed to reach the tolerance ϵ when we take different N. We observe that as N becomes larger, the number of iterations needed to meet the tolerance are similar. Clearly, the algorithm will be sufficiently efficient as the number of processors increases.

Table 1: Example 1. Algorithm 1 iteration steps.

N	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 60$
iteration	17	16	14	13	13

Example 2 In this numerical test, we take more complicated permeability field (more high conductivity streaks) and point source term as shown in Figure 4. We also depict the reference solution and the Algorithm 1 solution at the final time with $N = 50$ in Figure 5.

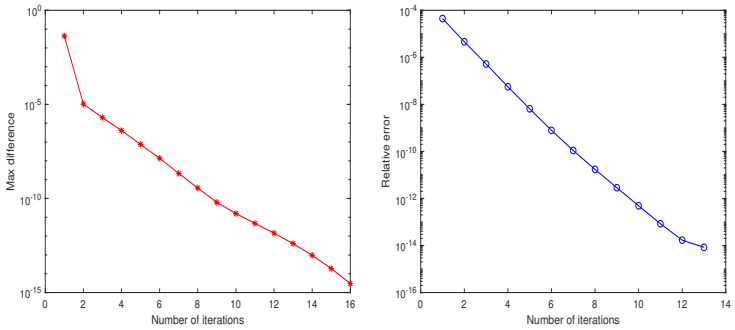

Figure 3: Example 1. **Left:** Max differences between the Algorithm 1 solution. **Right:** Convergence rate.

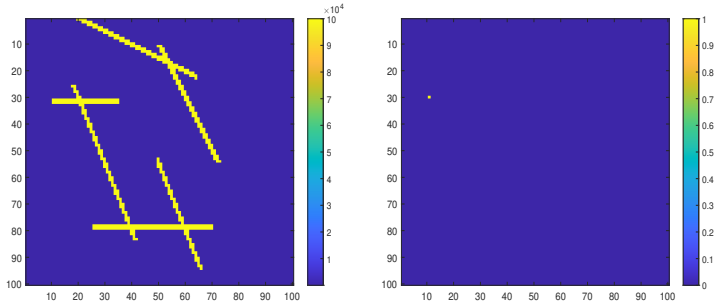

Figure 4: Example 2. **Left:** κ; **Right:** f

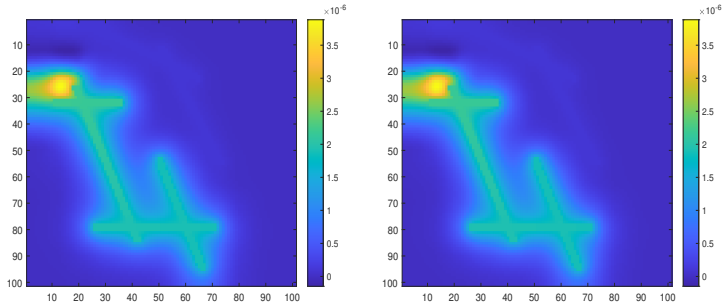

Figure 5: Example 2. **Left:** reference solution; **Right:** Algorithm 1 solution.

Similar as in Example 1, we take $N = 50$ and $\alpha = 0.6$, and give the max differences between the solution obtained by Algorithm 1 and convergence rate in Figure 6, respectively. we can see that the differences reach the tolerance after a few of iteration steps, and the error decays quickly, showing fast convergence.

Table 2 shows the number of the iterations required to meet the tolerance ϵ when we take different N. The results suggest that the larger N is, a slightly fewer iterations are needed.

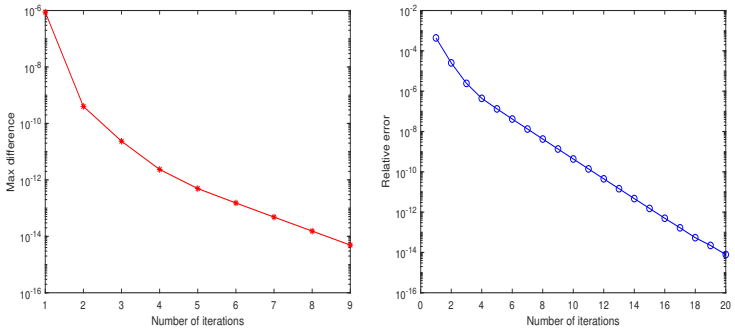

Figure 6: Example 2. **Left:** Max differences between the Algorithm 1 solution. **Right:** Convergence rate.

Table 2: Example 2. Algorithm 1 iteration steps.

N	$N = 20$	$N = 30$	$N = 40$	$N = 50$	$N = 60$
iteration	20	19	19	18	20

6 Conclusion

In this paper, we consider the diffusion equation with high contrast coefficient, and propose a parareal algorithm for the partially explicit temporal splitting scheme. In the parareal algorithm, we propose a one-step partially explicit temporal splitting scheme as the coarse solver, and utilize the all-at-once method as the fine solver to efficiently improve the computational efficiency. The convergences of the all-at-once method and the proposed parareal algorithm are given. An error estimate for the full discretization is given. Numerical experiments show that the proposed algorithm is computationally fast and accurate. The algorithm and the analysis in this paper are based on the case of a linear model. We will consider generalizing the algorithm to the nonlinear case in future work.

Acknowledgments

Y. Wang's work is partially supported by the NSFC grant 12301559. W.T. Leung is partially supported by the Hong Kong RGC Early Career Scheme 21307223.

References

[1] Grégoire Allaire and Robert Brizzi. A multiscale finite element method for numerical homogenization. *Multiscale Modeling & Simulation*, 4(3):790–812, 2005.

[2] Gil Ariel, Seong Jun Kim, and Richard Tsai. Parareal multiscale methods for highly oscillatory dynamical systems. *SIAM Journal on Scientific Computing*, 38(6):A3540–A3564, 2016.

[3] Daniele Bertaccini and Fabio Durastante. Block structured preconditioners in tensor form for the all-at-once solution of a finite volume fractional diffusion equation. *Applied Mathematics Letters*, 95:92–97, 2019.

[4] Jinru Chen and Junzhi Cui. A multiscale rectangular element method for elliptic problems with entirely small periodic coefficients. *Applied mathematics and computation*, 130(1):39–52, 2002.

[5] Eric T. Chung, Yalchin Efendiev, and Shubin Fu. Generalized multiscale finite element method for elasticity equations. *GEM - International Journal on Geomathematics*, 5(2):225–254, 2014.

[6] Eric T Chung, Yalchin Efendiev, and Wing Tat Leung. Constraint energy minimizing generalized multiscale finite element method. *Computer Methods in Applied Mechanics and Engineering*, 339:298–319, 2018.

[7] Eric T Chung, Yalchin Efendiev, Wing Tat Leung, and Wenyuan Li. Contrast-independent, partially-explicit time discretizations for nonlinear multiscale problems. *Mathematics*, 9(23):3000, 2021.

[8] Eric T Chung, Yalchin Efendiev, Wing Tat Leung, and Petr N Vabishchevich. Contrast-independent partially explicit time discretizations for multiscale flow problems. *Journal of Computational Physics*, 445:110578, 2021.

[9] Eric T Chung, Yalchin Efendiev, Wing Tat Leung, Maria Vasilyeva, and Yating Wang. Non-local multi-continua upscaling for flows in heterogeneous fractured media. *Journal of Computational Physics*, 372:22–34, 2018.

[10] Yalchin Efendiev, Juan Galvis, and Thomas Y. Hou. Generalized multiscale finite element methods (gmsfem). *Journal of Computational Physics*, 251(23):116–135, 2013.

[11] Yalchin Efendiev, Juan Galvis, Raytcho D Lazarov, M. Moon, and Marcus V Sarkis. Generalized multiscale finite element method. symmetric interior penalty coupling. *Journal of Computational Physics*, 2013.

[12] Yalchin Efendiev and Alexander Pankov. Numerical homogenization of monotone elliptic operators. *Multiscale Modeling & Simulation*, 2:62–79, 2006.

[13] Yalchin Efendiev, Sai-Mang Pun, and Petr N Vabishchevich. Temporal splitting algorithms for non-stationary multiscale problems. *Journal of Computational Physics*, 439:110375, 2021.

[14] Yalchin Efendiev and Petr N Vabishchevich. Splitting methods for solution decomposition in nonstationary problems. *Applied Mathematics and Computation*, 397:125785, 2021.

[15] Thomas Y. Hou and Xiao Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169–189, 1997.

[16] Jiuhua Hu, Anatoly Alikhanov, Yalchin Efendiev, and Wing Tat Leung. Partially explicit time discretization for time fractional diffusion equation. *Fractional Calculus and Applied Analysis*, 25(5):1908–1924, 2022.

[17] David E Keyes and Olof B Widlund. *Domain decomposition methods in science and engineering XVI*. Springer, 2007.

[18] Fande Kong. Parallel memory-efficient all-at-once algorithms for the sparse matrix triple products in multigrid methods. *arXiv preprint arXiv:1905.08423*, 2019.

[19] Wing Tat Leung and Yating Wang. Multirate partially explicit scheme for multiscale flow problems. *SIAM Journal on Scientific Computing*, 44(3):A1775–A1806, 2022.

[20] Guanglian Li. Wavelet-based edge multiscale parareal algorithm for subdiffusion equations with heterogeneous coefficients in a large time domain. *Journal of Computational and Applied Mathematics*, 440:115608, 2024.

[21] Mengnan Li, Eric Chung, and Lijian Jiang. A constraint energy minimizing generalized multiscale finite element method for parabolic equations. *Multiscale Modeling & Simulation*, 17(3):996–1018, 2019.

[22] Wenyuan Li, Anatoly Alikhanov, Yalchin Efendiev, and Wing Tat Leung. Partially explicit time discretization for nonlinear time fractional diffusion equations. *Communications in Nonlinear Science and Numerical Simulation*, 113:106440, 2022.

[23] Xue-lei Lin and Michael Ng. An all-at-once preconditioner for evolutionary partial differential equations. *SIAM Journal on Scientific Computing*, 43(4):A2766–A2784, 2021.

[24] Jacques Louis Lions, Yvon Maday, and Gabriel Turinici. A "parareal" in time discretization of pde's. *Comptes Rendus de l Académie des Sciences - Series I - Mathematics*, 332(7), 2001.

[25] Axel Målqvist and Daniel Peterseim. Localization of elliptic multiscale problems. *Mathematics of Computation*, 83(290):2583–2603, 2014.

[26] Leonardo A Poveda, Shubin Fu, Eric T Chung, and Lina Zhao. Convergence of the cem-gmsfem for compressible flow in highly heterogeneous media. *Computers & Mathematics with Applications*, 151:153–163, 2023.

[27] Wu Shulin, Wang Zhiyong, and Huang Chengming. Analysis of mean-square stability of the parareal algorithm. *Mathematica Numerica Sinica*, 33(2):113, 2011.

[28] Thomas, Y., Hou, Xiao-Hui, Wu, Zhiqiang, and Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comp*, 1999.

[29] Yating Wang and Wing Tat Leung. An adaptive space and time method in partially explicit splitting scheme for multiscale flow problems. *Computers & Mathematics with Applications*, 144:100–123, 2023.

[30] Shuonan Wu and Zhi Zhou. A parallel-in-time algorithm for high-order bdf methods for diffusion and subdiffusion equations. *SIAM Journal on Scientific Computing*, 43(6):A3627–A3656, 2021.

[31] Qinwu Xu, Jan S Hesthaven, and Feng Chen. A parareal method for time-fractional differential equations. *Journal of Computational Physics*, 293:173–183, 2015.

[32] Dandan Xue, Yanren Hou, and Wenjia Liu. Analysis of the parareal method with spectral deferred correction method for the stokes/darcy equations. *Applied Mathematics and Computation*, 387:124625, 2020.

[33] Kuokuo Zhang, Weibing Deng, and Haijun Wu. A combined multiscale finite element method based on the lod technique for the multiscale elliptic problems with singularities. *Journal of Computational Physics*, 469:111540, 2022.

[34] Lina Zhao and Eric T Chung. An analysis of the nlmc upscaling method for high contrast problems. *Journal of Computational and Applied Mathematics*, 367:112480, 2020.