Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2025-0010

Vol. **19**, No. 1, pp. 20-38
February 2026

A Space-Time Extrapolation Cascadic Multigrid Method for 2D Linear Parabolic Problems

Huaqing Wang[1,2], Kejia Pan[1], Jin Li[1] and Hongling Hu[3,*]

[1] *School of Mathematics and Statistics, HNP-LAMA, Central South University, Changsha 410083, China*
[2] *School of Mathematics and Big Data, Jining University, Qufu 273155, China*
[3] *MOE-LCSM, School of Mathematics and Statistics, Hunan Normal University, Changsha 410081, China*

Received 27 January 2025; Accepted (in revised version) 1 August 2025

Abstract. We present a new space-time extrapolation cascadic multigrid method with SSOR preconditioned GMRES smoother for linear parabolic equations. This method simultaneously solves all time steps of the system using an all-at-once approach, employing Crank-Nicolson discretization in time and central difference discretization in space. The key components of the new algorithm are the Richardson extrapolation and Lagrange interpolation operators. By utilizing these techniques with numerical solutions of current and previous grids, we generate a good initial guess for the iterative solution on the next finer grid, greatly reducing the number of required iterations and computational time. Finally, we explain how to implement the new multigrid method and show its efficiency through numerical experiments.

AMS subject classifications: 65M10, 78A48

Key words: Parabolic problems, Crank-Nicolson scheme, all-at-once discretization, Richardson extrapolation, multigrid.

1. Introduction

In this paper, we seek an approximate solution for the general second-order parabolic equation under the initial and Dirichlet boundary conditions

$$\begin{aligned} \partial_t u(\mathbf{x},t) &= \nabla\cdot\big(a(\mathbf{x})\nabla u(\mathbf{x},t)\big) - b(\mathbf{x})\cdot\nabla u(\mathbf{x},t) \\ &\quad - c(\mathbf{x})u(\mathbf{x},t) + f(\mathbf{x},t), && (\mathbf{x},t)\in\Omega\times[0,T], \qquad (1.1)\\ u(\mathbf{x},t) &= g(\mathbf{x},t), && (\mathbf{x},t)\in\partial\Omega\times(0,T], \\ u(\mathbf{x},0) &= u_0(\mathbf{x}), && \mathbf{x}\in\Omega, \qquad (1.2) \end{aligned}$$

*Corresponding author. *Email address:* honglinghu@hunnu.edu.cn (H.L. Hu)

where Ω is a rectangular domain in R^2, and T is a positive constant. Here the coefficients a, b, c and forcing function f, and the unknown function u are sufficiently smooth functions and have required regularities for the computation and analysis. The functions u_0 and g represent the initial and boundary conditions, respectively, which characterize the exact solution u.

The classical time-stepping method solves parabolic PDEs one-time step after one-time step (i.e., in a fully sequential manner), which would be time-consuming if the number of time steps is large. This challenge has driven the advancement of parallel-in-time (PinT) numerical methods for solving evolutionary PDEs, for example, the parareal algorithm [15, 24, 33], the multigrid-reduction-in-time (MGRIT) method [9, 12, 19, 35], the preconditioned Krylov subspace method [13, 16, 20, 28, 36, 39] and the space-time multigrid (STMG) method [3, 10, 14, 21, 27, 29, 37]. For an in-depth summary of the past 50 years of parallel-in-time research, please refer to [11]. In this paper, we focus on the STMG method.

Multigrid solvers serve as optimal preconditioners for elliptic problems and, with certain considerations, have demonstrated efficiency for space-time discretizations of parabolic problems like the heat equation. By treating time as an additional spatial dimension, STMG extends traditional multigrid paradigms to the temporal domain, enabling simultaneous coarsening and refinement in space and time. Hackbusch [18] first introduced the parabolic multigrid method, which achieved excellent convergence when coarsening in space but struggled with temporal coarsening. Lubich and Ostermann [26] later developed a continuous variant using waveform relaxation as a smoother, which, when discretized, aligned with Hackbusch's original method. Then, Horton and Vandewalle [21] were the first to introduce a general multigrid method that supports coarsening in both space and time, using semi-coarsening strategies to address the strong anisotropy in the time direction. To overcome the time coarsening challenges caused by anisotropy, Gander and Neumuller [14] proposed an STMG method with a block Jacobi smoother for the heat equation, enabling full space-time coarsening to solve the global discrete system at once, with excellent convergence and scalability properties. In recent years, multigrid methods have been used to solve various complex problems, demonstrating remarkable flexibility and efficiency, see [22, 34, 38] for details.

Inspired by the progress mentioned above, we aim to investigate a new space-time extrapolation cascadic multigrid (EXCMG) method for solving all-at-once (AaO) systems derived from full space-time discretization. The EXCMG approach is known for its exceptional efficiency in handling large linear systems, outperforming classical multigrid methods, especially in the context of elliptic problems [4, 6, 30]. In this paper, we first introduce a space-time extrapolation cascadic multigrid (STEXCMG) method to solve parabolic PDEs across all time levels simultaneously. This method utilizes full space-time discretization, resulting in an all-at-once system. Given the popularity of the implicit Crank-Nicolson (CN) scheme [7] for parabolic PDEs due to its unconditional convergence properties, we adopted this method in our study. Unlike existing multigrid methods (see Fig. 1), our method generates a high-quality initial guess for the pre-

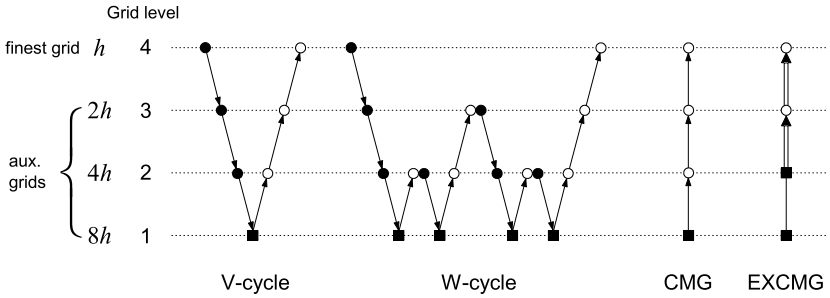


Figure 1: The four-level structure of the V- and W-cycles, CMG, and EXCMG methods [31].

conditioned GMRES smoother on the next finer grid by applying quadratic Lagrange interpolation and completed Richardson extrapolation to the numerical solutions on two levels of coarse grids (current and previous), significantly reducing the number of iterations required at each grid level. As a result, AaO systems are solved efficiently and robustly using the STEXCMG method.

The remainder of this paper is organized as follows. In the next section, we discretize the entire space-time domain for parabolic PDEs, resulting in an AaO linear system. In Section 3, we develop an STEXCMG method to solve the AaO linear system. Section 4 contains the numerical results to demonstrate the high efficiency and accuracy of the proposed method. Finally, the conclusions are given in Section 5.

2. The CN scheme and its AaO system

To solve Eqs. (1.1)-(1.2), we employ discretization in both spatial and temporal domains. For clarity, we adopt a central finite difference scheme for spatial discretization and the CN method for temporal discretization.

2.1. Temporal discretization

We divide the time interval $[0,T]$ into N_t equal subintervals of size $\tau = T/N_t$. Denote $t_n = n\tau$ and $t_{n+1/2} = (n+1/2)\tau$. We discretize the temporal domain using the CN method on a set of time levels with a constant time step size τ

$$\frac{U^{n+1}-U^n}{\tau} = \frac{1}{2}\left[LU^{n+1} + LU^n\right] + f^{n+1/2},$$

where

$$L = \nabla\cdot(a(\mathbf{x})\nabla) - b(\mathbf{x})\cdot\nabla - c(\mathbf{x})$$

is the spatial operator on the right-hand side of the equation. U^n represents the approximation of $u(\cdot,t_n)$ at all points on the spatial grid. Similarly, $f^{n+1/2}$ represents the approximation of $f(\cdot,t_{n+1/2})$ at all points of the spatial grid.

2.2. Spatial discretization

Let the patial domain $\Omega = [x_k, x_l] \times [y_k, y_l]$. We construct a uniform grid as follows:

$$\begin{aligned} x_i &= x_k + ih, \quad i = 0, 1, \ldots, N_x, \\ y_j &= y_k + jh, \quad j = 0, 1, \ldots, N_y \end{aligned}$$

with

$$h = \frac{x_l - x_k}{N_x} = \frac{y_l - y_k}{N_y}.$$

For function u, we denote

$$U_{ij} = u(x_i, y_j), \quad U_{i+1/2,j} = \frac{1}{2}(U_{i,j} + U_{i+1,j}).$$

Let

$$\begin{aligned} \delta_x U_{i+1/2,j} &= \frac{U_{i+1,j} - U_{i,j}}{h}, \\ \delta_x U_{ij} &= \frac{U_{i+1/2,j} - U_{i-1/2,j}}{h}, \\ \delta_{0x} U_{ij} &= \frac{U_{i+1,j} - U_{i-1,j}}{2h}, \end{aligned}$$

$\delta_y U_{i,j+1/2}$, $\delta_y U_{ij}$ and $\delta_{0y} U_{ij}$ are defined analogously. Derivation of the finite difference formula for $\delta(a\delta U)_{i,j}$,

$$\begin{aligned} \delta(a\delta U)_{ij} &= \delta_x(a\delta_x U)_{ij} + \delta_y(a\delta_y U)_{ij} \\ &= \frac{1}{h}\left[a_{i+1/2,j}\delta_x U_{i+1/2,j} - a_{i-1/2,j}\delta_x U_{i-1/2,j}\right] \\ &\quad + \frac{1}{h}\left[a_{i,j+1/2}\delta_y U_{i,j+1/2} - a_{i,j-1/2}\delta_y U_{i,j-1/2}\right]. \end{aligned}$$

The second-order approximation of L is as follows:

$$\begin{aligned} L_h U_{ij} &= \delta(a\delta U)_{ij} - b_{ij}(\delta_{0x} U_{ij} + \delta_{0y} U_{ij}) - c_{ij} U_{ij} \\ &= \delta_x(a\delta_x U)_{ij} + \delta_y(a\delta_y U)_{ij} - b_{ij}(\delta_{0x} U_{ij} + \delta_{0y} U_{ij}) - c_{ij} U_{ij} \\ &= \frac{1}{h}\left[a_{i+1/2,j}\delta_x U_{i+1/2,j} - a_{i-1/2,j}\delta_x U_{i-1/2,j}\right] \\ &\quad + \frac{1}{h}\left[a_{i,j+1/2}\delta_y U_{i,j+1/2} - a_{i,j-1/2}\delta_y U_{i,j-1/2}\right] \\ &\quad - b_{ij}\left[\frac{U_{i+1,j} - U_{i-1,j}}{2h} + \frac{U_{i,j+1} - U_{i,j-1}}{2h}\right] - c_{ij} U_{ij} \\ &= A_{i,j-1} U_{i,j-1} + A_{i-1,j} U_{i+1,j} + A_{ij} U_{ij} + A_{i+1,j} U_{i+1,j} + A_{i,j+1} U_{i,j+1}, \end{aligned} \tag{2.1}$$

where

$$A_{i,j-1} = \frac{a_{i,j-1/2}}{h^2} + \frac{b_{ij}}{2h}, \quad A_{i-1,j} = \frac{a_{i-1/2,j}}{h^2} + \frac{b_{ij}}{2h},$$

$$A_{i+1,j} = \frac{a_{i+1/2,j}}{h^2} - \frac{b_{ij}}{2h}, \quad A_{i,j+1} = \frac{a_{i,j+1/2}}{h^2} - \frac{b_{ij}}{2h},$$
$$A_{i,j} = -\left(\frac{a_{i+1/2,j}}{h^2} + \frac{a_{i-1/2,j}}{h^2} + \frac{a_{i,j+1/2}}{h^2} + \frac{a_{i,j-1/2}}{h^2} + c_{ij}\right).$$

We denote U_{ij}^n as the numerical approximation of $u(x_i, y_j, t_n)$, $f_{ij}^{n+1/2}$ denotes $f(x_i, y_j, t_{n+1/2})$. The detailed numerical scheme for Eqs. (1.1)-(1.2) is as follows:

$$\begin{aligned} &\frac{U_{ij}^{n+1} - U_{ij}^n}{\tau} = \frac{1}{2}\left[L_h U_{ij}^{n+1} + L_h U_{ij}^n\right] + f_{ij}^{n+1/2}, \\ &U_{ij}^n = g(x_i, y_j, t_n), \quad n = 0, 1, \ldots, N_t - 1, \\ &U_{ij}^0 = u_0(x_i, y_j). \end{aligned} \tag{2.2}$$

The unconditional stability and convergence of the CN scheme are given by the following lemmas.

Lemma 2.1 ([1, 2]). *The CN scheme* (2.2) *is unconditionally stable, and satisfies*

$$\|U^n - u(\cdot, t_n)\|_\infty \le C(\tau^2 + h^2), \quad n = 0, 1, \ldots, N_t - 1,$$

where C is a positive constant.

2.3. AaO system

Let $A \in R^{N\times N}$ be the spatial discretization matrix of L, derived from Eq. (2.1). Substituting (2.1) into (2.2), we derive the matrix form of the CN scheme for Eqs. (1.1)-(1.2) as shown below

$$\frac{U^{n+1} - U^n}{\tau} = A\frac{U^{n+1} + U^n}{2} + f^{n+1/2}, \quad n = 0, 1, \ldots, N_t - 1.$$

For simplicity, we let $\mathbf{0}$ denote a zero vector or matrix of appropriate dimensions, and I_s represent an identity matrix of size N. Denote

$$Q = I_s - \frac{\tau}{2}A, \quad \tilde{Q} = I_s + \frac{\tau}{2}A,$$

then, the AaO system can be written as

$$K\mathbf{U} := \begin{bmatrix} Q & & & \\ -\tilde{Q} & Q & & \\ & \ddots & \ddots & \\ & & -\tilde{Q} & Q \end{bmatrix} \begin{bmatrix} U^1 \\ U^2 \\ \vdots \\ U^{N_t} \end{bmatrix} = \begin{bmatrix} \tilde{Q}U^0 + \tau f^{1/2} \\ \tau f^{3/2} \\ \vdots \\ \tau f^{N_t - 1/2} \end{bmatrix} =: \mathbf{b},$$

where K is a coefficient matrix and $\mathbf{b}$ represents the source term on the right-hand side.

It is obvious that the scheme often leads to a system of equations with a nonsymmetric coefficient matrix K for general anisotropic cases, and the resulting scheme is a 10-point one. A general sparse direct solver can be computationally expensive, particularly for high-dimensional problems on fine meshes. Therefore, developing fast and accurate methods to solve the large AaO systems arising from CN discretizations is both crucial and challenging. This topic will be explored in the next section.

3. Space-time extrapolation cascadic multigrid method

This section outlines the key components of the new space-time multigrid method. To simplify the explanation, we focus on the one-dimensional parabolic equation, which corresponds to a two-dimensional space-time problem. The operators for the three-dimensional space-time problem are defined similarly and are provided as follows.

3.1. Extrapolation and quadratic interpolation

The extrapolation technique is widely recognized as an effective approach for enhancing the accuracy of numerical solutions in various computational problems. In the current and subsequent subsections, we assume that the exact solution u is sufficiently smooth. We explain how to employ extrapolation and high-order interpolation to construct a third-order accurate approximation $\overset{(l)}{W}$ to the FD solution on the next finer grid. Furthermore, we illustrate the construction of a fourth-order accurate extrapolated solution $\tilde{\overset{(l)}{U}}$ for problem (1.1)-(1.2).

3.1.1. Extrapolation for the FD solution

We begin with a two-dimensional space-time element on the uniform grid. Consider a three-level nested grid Z_l with step-length $\tau = h = h_l = h_0/2^l$, $l = 0, 1, 2, \ldots, L$ and corresponding numerical solutions $\overset{(l)}{U}$, as depicted in Fig. 2.

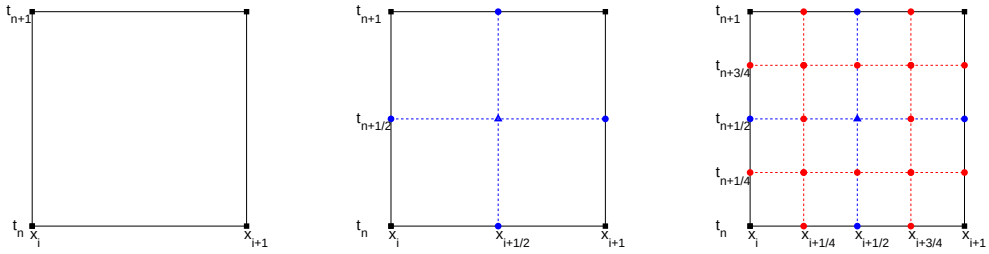
Assume $u \in C^4$ and denote the error $\overset{(l)}{e} = \overset{(l)}{U} - u$ and the nodal value $\overset{(l)}{e}_j = e(x_j)$. There is an asymptotic expansion

$$\overset{(l)}{e}_j = (\overset{(l)}{U} - u)(x_j) = A(x_j)h_l^2 + \mathcal{O}\big(h_l^4\big), \quad l = 0, 1, 2, \tag{3.1}$$

where $A(x)$ denotes a sufficiently smooth function independent of h_l.

Consider the coarse element $S_{s,w} = [x_i, x_{i+1}] \times [t_n, t_{n+1}]$ shown in Fig. 2. Assume that $\overset{(0)}{U}$ and $\overset{(1)}{U}$ are given as follows:

$$\left\{ \overset{(0)}{U}{}_s^w, \;\; s = i, i+1, \;\; w = n, n+1 \right\} \quad \text{on } \mathbb{Z}_0,$$



(a) Z_0 (b) Z_1 (c) Z_2

Figure 2: Three-level nested mesh.

$$\left\{ \overset{(1)}{U}{}_s^w,\ \overset{(1)}{U}{}_{i+1/2}^w,\ \overset{(1)}{U}{}_s^{n+1/2},\ \overset{(1)}{U}{}_{i+1/2}^{n+1/2},\ s=i,i+1,\ w=n,n+1 \right\} \quad \text{on } \mathbb{Z}_1.$$

Next, based on the midpoint extrapolation formula proposed in [4, 6], we can construct a third-order approximation $\overset{(2)}{W}$ to the numerical solution $\overset{(2)}{U}$.

- Corner points ■. The approximated results on 4 corner points '■' determined using an extrapolation formula

$$\overset{(2)}{W}{}_s^w = \frac{5}{4}\overset{(1)}{U}{}_s^w - \frac{1}{4}\overset{(0)}{U}{}_s^w, \quad s=i,i+1, \quad w=n,n+1. \tag{3.2}$$

- Midpoints ●. The approximated results on 4 midpoints '●' are derived through the midpoint extrapolation approach

$$\begin{cases} \overset{(2)}{W}{}_{i+1/2}^w = \overset{(1)}{U}{}_{i+1/2}^w + \frac{1}{8}(\overset{(1)}{U}{}_i^w - \overset{(0)}{U}{}_i^w + \overset{(1)}{U}{}_{i+1}^w - \overset{(0)}{U}{}_{i+1}^w), & w=n,n+1, \\ \overset{(2)}{W}{}_s^{n+1/2} = \overset{(1)}{U}{}_s^{n+1/2} + \frac{1}{8}(\overset{(1)}{U}{}_s^n - \overset{(0)}{U}{}_s^n + \overset{(1)}{U}{}_s^{n+1} - \overset{(0)}{U}{}_s^{n+1}), & s=i,i+1. \end{cases} \tag{3.3}$$

- Central points ▲. The central points '▲' are treated as midpoints of two face diagonals, yielding two distinct extrapolated values through midpoint extrapolation. These values are arithmetically averaged to determine the estimated values at the central points '▲'

$$\begin{aligned} \overset{(2)}{W}{}_{i+1/2}^{n+1/2} = \overset{(1)}{U}{}_{i+1/2}^{n+1/2} &+ \frac{1}{16}(\overset{(1)}{U}{}_i^n - \overset{(0)}{U}{}_i^n + \overset{(1)}{U}{}_{i+1}^{n+1} - \overset{(0)}{U}{}_{i+1}^{n+1}) \\ &+ \frac{1}{16}(\overset{(1)}{U}{}_i^{n+1} - \overset{(0)}{U}{}_i^{n+1} + \overset{(1)}{U}{}_{i+1}^n - \overset{(0)}{U}{}_{i+1}^n). \end{aligned} \tag{3.4}$$

- Quarter points ●. Once the extrapolated values on the finest grid are obtained, then the interpolated values at the 16 quarter-division points '●' are derived using

bi-quadratic Lagrange interpolation, for example,

$$
\begin{aligned}
\overset{(2)}{W}{}^{n}_{i+1/4} &= \frac{3}{8}\overset{(2)}{U}{}^{n}_{i} + \frac{3}{4}\overset{(2)}{U}{}^{n}_{i+1/2} - \frac{1}{8}\overset{(2)}{U}{}^{n}_{i+1}, \\
\overset{(2)}{W}{}^{n+1/4}_{i+1/4} &= \frac{1}{8}\left(\frac{9}{8}\overset{(2)}{U}{}^{n}_{i} + \frac{9}{4}\overset{(2)}{U}{}^{n}_{i+1/2} - \frac{3}{8}\overset{(2)}{U}{}^{n}_{i+1} + \frac{9}{4}\overset{(2)}{U}{}^{n+1/2}_{i}\right. \\
&\qquad \left. + \frac{9}{2}\overset{(2)}{U}{}^{n+1/2}_{i+1/2} - \frac{3}{4}\overset{(2)}{U}{}^{n+1/2}_{i+1} - \frac{3}{8}\overset{(2)}{U}{}^{n+1}_{i} - \frac{3}{4}\overset{(2)}{U}{}^{n+1}_{i+1/2} + \frac{1}{8}\overset{(2)}{U}{}^{n+1}_{i+1}\right).
\end{aligned}
\tag{3.5}
$$

The expressions for the remaining points can be derived in a similar manner.

Once the initial guesses $\overset{(2)}{W}$ at the corner points ■, midpoints ●, and central points ▲ in Fig. 2 are obtained from Eqs. (3.2)-(3.4), the initial guesses at the 16 quarter-division points ● can be obtained using bi-quadratic interpolation, as in Eq. (3.5). This extrapolation-interpolation procedure defines a prolongation operator EXP_{CN} as follows:

$$\overset{(2)}{W} \Leftarrow \mathrm{EXP}_{CN}(\overset{(1)}{U}, \overset{(0)}{U}).$$

Based on the theory of polynomial interpolation, it is straightforward to demonstrate that the extrapolation expressions (Eq. (3.5)) offer a third-order approximation to the finite difference solution, i.e.,

$$
\begin{aligned}
\overset{(2)}{W}{}^{n}_{i+1/4} &= \overset{(2)}{U}{}^{n}_{i+1/4} + \mathcal{O}\left(h_0^3\right), \\
\overset{(2)}{W}{}^{n+1/4}_{i+1/4} &= \overset{(2)}{U}{}^{n+1/4}_{i+1/4} + \mathcal{O}\left(h_0^3\right),
\end{aligned}
$$

see numerical verification of Tables 1-6 in the next section.

Remark 3.1. Our extrapolation technique relies on the asymptotic error expansion (3.1), which requires certain regularity of the exact solution to guarantee high-order accuracy. However, as numerically demonstrated in [30], the proposed method still performs well when the solution has lower regularity.

3.1.2. 3D case

The operators for the three-dimensional space-time problem on the embedded hexahedral grid, as illustrated in Fig. 3, can also be derived using a similar approach to that in [31], allowing us to obtain a reliable initial approximation on the finer mesh for the iterative solver.

The procedure for constructing the approximation $\overset{(2)}{W}$ is as follows:

Corner points (1, 5, 21, 25, 101, 105, 121, 125): The approximated results on 8 corner points '●' determined using an extrapolation formula (Eq. (3.2))

$$\overset{(2)}{W}{}^{w}_{s,r} = \frac{5}{4}\overset{(1)}{U}{}^{w}_{s,r} - \frac{1}{4}\overset{(0)}{U}{}^{w}_{s,r}, \quad s = i, i+1, \quad r = j, j+1, \quad w = n, n+1.$$

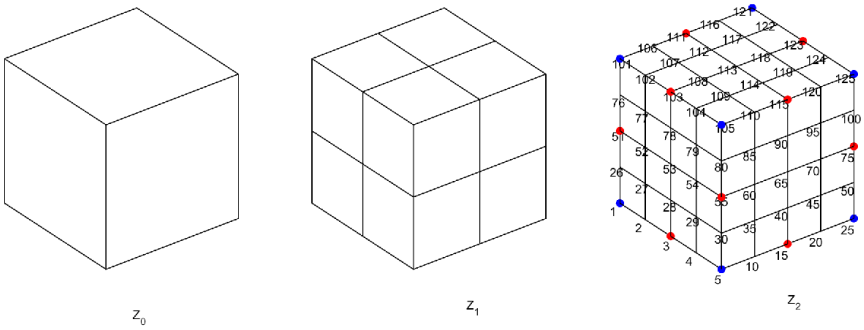


Figure 3: 3D nested rectilinear meshes.

Edge center (3, 11, 15, 23, 51, 55, 71, 75, 103, 111, 115, 123): The approximated results on 12 midpoints '•' can be determined using the extrapolation formulas (Eq. (3.3))

$$\begin{cases} \overset{(2)}{W}{}^{w}_{i+1/2,r} = \overset{(1)}{W}{}^{w}_{i+1/2,r} + \frac{1}{8}(\overset{(1)}{U}{}^{w}_{i,r} - \overset{(0)}{U}{}^{w}_{i,r} + \overset{(1)}{U}{}^{w}_{i+1,r} - \overset{(0)}{U}{}^{w}_{i+1,r}), & r = j, j+1, \quad w = n, n+1, \\ \overset{(2)}{W}{}^{w}_{s,j+1/2} = \overset{(1)}{U}{}^{w}_{s,j+1/2} + \frac{1}{8}(\overset{(0)}{U}{}^{w}_{s,j} - \overset{(0)}{U}{}^{w}_{s,j} + \overset{(1)}{U}{}^{w}_{s,j+1} - \overset{(0)}{U}{}^{w}_{s,j+1}), & s = i, i+1, \quad w = n, n+1, \\ \overset{(2)}{W}{}^{n+1/2}_{s,r} = \overset{(1)}{U}{}^{n+1/2}_{s,r} + \frac{1}{8}(\overset{(1)}{U}{}^{n}_{s,r} - \overset{(0)}{U}{}^{n}_{s,r} + \overset{(1)}{U}{}^{n+1}_{s,r} - \overset{(0)}{U}{}^{n+1}_{s,r}), & s = i, i+1, \quad r = j, j+1. \end{cases}$$

Remaining 105 points: The remaining 105 $(5^3 - 20)$ grid points can be approximated through tri-quadratic serendipity interpolation, utilizing the known values of 20 nodes, which include 8 corner nodes and 12 mid-side nodes. This procedure of constructing initial guess on grid Z_2 defines an prolongation operator EXP_{CN} as follows:

$$\overset{(2)}{W} \Leftarrow \mathrm{EXP}_{CN}(\overset{(1)}{U}, \overset{(0)}{U}).$$

In Cartesian coordinates, the tri-quadratic serendipity interpolation function can be expressed as

$$w(\xi, \eta, \zeta) = \sum_m N_m(\xi, \eta, \zeta) w_m,$$

where the interpolation functions N_m are defined as follows [25]:

$$N_m(\xi, \eta, \zeta) = \begin{cases} \frac{1}{8}(1+\xi_m\xi)(1+\eta_m\eta)(1+\zeta_m\zeta)(\xi_m\xi + \eta_m\eta + \zeta_m\zeta - 2), \\ \qquad\qquad m = 1, 5, 21, 25, 101, 105, 121, 125, \\ \frac{1}{4}(1-\xi^2)(1+\eta_m\eta)(1+\zeta_m\zeta), \quad m = 3, 23, 103, 123, \\ \frac{1}{4}(1-\eta^2)(1+\xi_m\xi)(1+\zeta_m\zeta), \quad m = 11, 15, 111, 115, \\ \frac{1}{4}(1-\zeta^2)(1+\xi_m\xi)(1+\eta_m\eta), \quad m = 51, 55, 71, 75 \end{cases}$$

with (ξ_m, η_m, ζ_m) denotes coordinate position for the point m.

3.1.3. Extrapolation for the true solution

Based on the asymptotic expansion (3.1), the Richardson extrapolated solution $\tilde{\overset{(1)}{U}}{}_s^w$ at points (x_s, t_w) $(s = i, i+1, w = n, n+1)$ can be written as

$$\tilde{\overset{(1)}{U}}{}_s^w := \frac{4\overset{(1)}{U}{}_s^w - \overset{(0)}{U}{}_s^w}{3} = u(x_s, t_w) + \mathcal{O}(h^4), \quad s = i, i+1, \quad w = n, n+1. \tag{3.6}$$

At fine grid points, Chen and Lin [5] proposed midpoint extrapolation formulas

$$\begin{aligned}
\tilde{\overset{(1)}{U}}{}_{i+/2}^w &:= \overset{(1)}{U}{}_{i+1/2}^w + \frac{1}{6}\left(\overset{(1)}{U}{}_i^w - \overset{(0)}{U}{}_i^w + \overset{(1)}{U}{}_{i+1}^w - \overset{(0)}{U}{}_{i+1}^w\right) \\
&= u(x_{i+1/2}, t_w) + \mathcal{O}(h^4), \quad w = n, n+1, \\
\tilde{\overset{(1)}{U}}{}_s^{n+1/2} &:= \overset{(1)}{U}{}_s^{n+1/2} + \frac{1}{6}\left(\overset{(1)}{U}{}_s^n - \overset{(0)}{U}{}_s^n + \overset{(1)}{U}{}_s^{n+1} - \overset{(0)}{U}{}_s^{n+1}\right) \\
&= u(x_s, t_{n+1/2}) + \mathcal{O}(h^4), \quad s = i, i+1, \\
\tilde{\overset{(1)}{U}}{}_{i+1/2}^{n+1/2} &= \overset{(1)}{U}{}_{i+1/2}^{n+1/2} + \frac{1}{12}(\overset{(1)}{U}{}_i^n - \overset{(0)}{U}{}_i^n + \overset{(1)}{U}{}_i^{n+1} - \overset{(0)}{U}{}_i^{n+1}) \\
&\quad + \frac{1}{12}(\overset{(1)}{U}{}_{i+1}^n - \overset{(0)}{U}{}_{i+1}^n + \overset{(1)}{U}{}_{i+1}^{n+1} - \overset{(0)}{U}{}_{i+1}^{n+1}).
\end{aligned} \tag{3.7}$$

The extrapolation formulas (3.6)-(3.7) can be easily generalized to 3D case [31]. We denote this procedure by

$$\tilde{\overset{(1)}{U}}{}_s^n \Leftarrow \mathrm{EXP}_{ture}(\overset{(1)}{U}, \overset{(0)}{U}).$$

3.2. Description of STEXCMG algorithm

The core components of the STEXCMG method are extrapolation and quadratic interpolation (see Fig. 1), which are employed to generate a more accurate initial approximation for the iterative solver on the subsequent finer grid compared to the one obtained through linear interpolation in CMG. The steps of the algorithm are as follows.

Algorithm 3.1 Space-Time Extrapolation Cascadic Multigrid Method

1: **procedure** STEXCMG(m_L, L), m_L denotes the maximum number of iterations on the finest grid Z_L.
2: Calculate the solutions $\overset{(0)}{U}$ and $\overset{(1)}{U}$ by a direct solver.
3: **for** $l = 2$ to L **do**
4: Calculate an initial guess: $\overset{(l)}{W} \Leftarrow \mathrm{EXP}_{\mathrm{CN}}(\overset{(l-1)}{U}, \overset{(l-2)}{U})$, EXP_{CN} is the prolongation operator.

5: $m_l = m_L \cdot \gamma^{L-l}$.
6: Run the GMRES method for m_l times: $\overset{(l)}{U} \Leftarrow \text{GMRES}_{pre}(\overset{(l)}{K}, \overset{(l)}{\mathbf{b}}, \overset{(l)}{W})$.
7: Calculate extrapolated solution: $\overset{\tilde{(l)}}{U} \Leftarrow \text{EXP}_{true}(\overset{(l)}{U}, \overset{(l-1)}{U})$. Optional step EXP_{true} is the extrapolation operator.
8: **end for**
9: **return** $\overset{(L)}{U}$ and $\overset{\tilde{(L)}}{U}$.
10: **end procedure**

In Algorithm 3.1, a direct method is applied to the first two coarse grids because the linear systems involved are relatively small. Using the numerical solutions $\overset{(l-1)}{U}$ and $\overset{(l-2)}{U}$, Richardson extrapolation and quadratic interpolation provide a third-order approximation $\text{EXP}_{CN}(\overset{(l-1)}{U}, \overset{(l-2)}{U})$ for the finite difference solution $\overset{(l)}{U}$. This approximation is used as the initial guess for the GMRES solver, helping to reduce the number of iterations. Meanwhile, $\text{EXP}_{true}(\overset{(l)}{U}, \overset{(l-1)}{U})$ offers a more accurate extrapolation on the finest grid.

Generally, the number of GMRES iteration m_l is the same as in the CMG method [32]

$$m_l = m_L \cdot \gamma^{L-l}, \quad l = 2, 3, \ldots, L,$$

m_L represents the maximum number of iterations on the finest grid, where $2 \leq \gamma \leq 2^d$, with d denoting the spatial dimension. In such a case, optimal multigrid complexity is easily achieved.

Remark 3.2. In Algorithm 3.1, GMRES with an SSOR preconditioner is used as the multigrid smoother, replacing the conjugate gradient iteration in the traditional cascadic multigrid method [30]. The SSOR preconditioner is computationally efficient, requiring no additional storage, and is well-suited for the non-symmetric linear systems (2.3) arising from the CN scheme, as shown in Section 4.

Remark 3.3. The matrix-vector product KU required in each iteration of GMRES can be efficiently computed by exploiting the sparsity of K. Specifically, K is stored in compressed sparse row (CSR) format, and the sparse matrix-vector multiplication (SpMV) operation is optimized to reduce both memory usage and computational time. This ensures that the cost per iteration remains low even for large-scale AaO systems established in Section 2.3.

4. Numerical experiments

In this section, six typical numerical examples without particular settings are discussed to demonstrate the convergence rates and computational efficiency of our multigrid solver for one- and two-dimensional linear parabolic partial differential equations. The SSOR-preconditioned GMRES smoother (with a restart number of 50) is employed

to solve the discretized AaO system, using the initial guess W constructed in Section 3.1. The iteration is terminated based on a stopping criterion that the relative residual norm is reduced below the tolerance tol. By "Its" we denote the number of iterations of the preconditioned GMRES smoother in step 6 of Algorithm 3.1. It can be observed that the proposed solver is simple and easy to implement. Our code is written in MATLAB R2023a and all programs are carried out on a server with Intel(R) Xeon(R) Gold 6248R CPU (3.0 GHz) and 192 GB RAM.

The order of convergence of the method is computed by

$$\text{order} = \log_2 \frac{\|U_h - u\|}{\|U_{h/2} - u\|},$$

where $\|\cdot\|$ denotes some norm (for instance, L^2-norm or the L^∞-norm), u and U_h are the exact and the numerical solutions, respectively. In all computational experiments, the initial and Dirichlet boundary conditions are extracted from the exact solution.

4.1. Convergence behavior

4.1.1. One-dimensional parabolic equation

Example 4.1 ([8])**.** Consider $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = -1$, $c(\mathbf{x}) = 0$, $f(\mathbf{x}, t) = 0$ in Eqs. (1.1)-(1.2). We get the following linear parabolic equation with constant coefficient:

$$\partial_t u = u_{xx} + u_x, \quad 0 \le x \le 1, \quad 0 \le t \le 1$$

with exact solution

$$u(x, t) = e^{-t} \sin(x - t).$$

Using a six-level hierarchy of embedded spatial grids, ranging from the coarsest grid of 64×64 to the finest grid of 2048×2048, the numerical results obtained from the STEXCMG method are summarized in Table 1. In this table, "Its" represents the

Table 1: Convergence of the STEXCMG method for Example 4.1 with $tol = 10^{-10}$ and $\tau = h$.

$N_x \times N_t$	Its	$\|U - u\|_\infty$	Order	$\|W - U\|_2$	Order
64×64	15	2.17e-06		1.71e-06	
128×128	53	5.42e-07	2.0015	2.16e-07	2.9840
256×256	126	1.36e-07	2.0001	2.72e-08	2.9918
512×512	32	3.39e-08	2.0000	3.41e-09	2.9972
1024×1024	20	8.56e-09	1.9859	4.17e-10	3.0292
2048×2048	1	2.23e-09	1.9434	2.49e-11	4.0660
305.31 s[a]	10.19 WU[b]				

[a] The time taken by the STEXCMG method to solve the entire system all at once.
[b] WU (work unit) is defined as the computational cost of performing one SSOR-GMRES iteration on the finest grid. Here, the STEXCMG computational cost $= 1 + 20 \times 2^{-2} + 32 \times 2^{-4} + 126 \times 2^{-6} + 53 \times 2^{-8} + 15 \times 2^{-10} = 10.19$.

number of iterations required by the iterative solver to achieve the specified relative residual at all time steps. The computational cost of our solver is typically expressed as the equivalent work required for one relaxation sweep on the finest grid. The notations used in all subsequent tables are consistent with those described here.

Table 1 lists the iteration numbers, the errors between the numerical solution and the exact solution ($\|U-u\|_\infty$), and the errors between the numerical solution and its initial values ($\|W-U\|_2$) for the STEXCMG method. It is clear that the numerical solution U achieves second-order approximation and the initial value W, which is obtained by Richardson extrapolation and quadratic interpolation, achieves almost third-order approximation to the numerical solution U. On the finest mesh with partition 2048×2048, the maximum error between the initial guess W and the numerical solution U is $\mathcal{O}(10^{-11})$, and only one iteration is needed to process the numerical solution with second-order convergence, which is particularly important for solving large linear systems efficiently. In terms of the efficiency of the STEXCMG method, one can find that the computational cost of performing one SSOR-GMRES iteration on the finest grid is 10.19 WU and the time required to solve the entire system all at once is less than 6 minutes.

Example 4.2 ([8])**.** Consider problem (1.1)-(1.2) with $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = -\sin(x)$, $c(\mathbf{x}) = -\cos(x)$ and $f(\mathbf{x},t) = e^{-t}\sin(2x)$, we get the following 1D equation with variable coefficient:

$$\partial_t u = u_{xx} + \sin(x)u_x + \cos(x)u + f, \quad 0 \le x \le 1, \quad 0 \le t \le 1$$

with exact solution

$$u(x,t) = e^{-t}\sin(x).$$

Again, from Table 2, one can see that the numerical solution U achieves second-order accuracy for all mesh. It is once again proved that the smooth property of the initial condition is vital to derive high-accuracy approximation without too many iterations. On the finest mesh with partition 2048×2048, it happens to be the same that only one iteration is needed to obtain the second-order convergence rate because the initial guess W already is an extremely good approximation of the numerical solution U, which significantly reduces the computational time.

Table 2: Convergence of the STEXCMG method for Example 4.2 with $tol = 10^{-10}$ and $\tau = h$.

$N_x \times N_t$	Its	$\|U-u\|_\infty$	Order	$\|W-U\|_2$	Order
64×64	15	7.32e-07		8.66e-07	
128×128	46	1.83e-07	2.0005	1.10e-07	2.9829
256×256	89	4.57e-08	2.0005	1.38e-08	2.9914
512×512	25	1.14e-08	2.0008	1.73e-09	2.9962
1024×1024	3	3.02e-09	1.9207	1.86e-10	3.2116
2048×2048	1	8.93e-10	1.7576	2.31e-11	3.0139
108.34 s	4.90 WU				

4.1.2. Two-dimensional parabolic equation

Example 4.3 ([8]). When $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = 1$, $c(\mathbf{x}) = 0$ and $f(\mathbf{x}, t) = 0$ then Eqs. (1.1)-(1.2) becomes parabolic equation with constant coefficient given by

$$\partial_t u = \Delta u - u_x - u_y, \quad 0 \le x, y \le 1, \quad 0 \le t \le 1$$

with exact solution

$$u(x, y, t) = e^{-2t} \sin(x + y - 2t).$$

Using 5 embedded space-time grids starting from the coarsest grid $32 \times 32 \times 32$, the corresponding numerical results by the STEXCMG method are tabulated in Table 3. It is evident that the numerical solution U reaches second-order accuracy, and the initial guess W is almost a third-order approximation to the numerical solution U, which is in agreement with the theoretical results in Section 3.1. This significantly reduces the number of iterations required, thereby lowering the computational cost necessary to achieve the desired accuracy.

Table 3: Convergence of the STEXCMG method for Example 4.3 with $tol = 10^{-9}$ and $\tau = h$.

$N_x \times N_y \times N_t$	Its	$\|U - u\|_\infty$	Order	$\|W - U\|_2$	Order
$32 \times 32 \times 32$	4	3.80e-05		5.76e-05	
$64 \times 64 \times 64$	11	9.24e-06	2.0381	7.45e-06	2.9515
$128 \times 128 \times 128$	24	2.30e-06	2.0066	9.48e-07	2.9743
$256 \times 256 \times 256$	25	5.79e-07	1.9910	1.20e-07	2.9801
$512 \times 512 \times 512$	1	1.80e-07	1.6851	3.50e-09	5.0997
2322.62 s	3.37 WU				

Example 4.4 ([8]). Consider Eqs. (1.1)-(1.2) with $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = (y, -x)$, $c(\mathbf{x}) = 0$ and the source term

$$f(\mathbf{x}, t) = e^{-(x^2+3y^2+2t)}\big(6 - 4xy - 4(x^2 + 9y^2)\big),$$

we get the following linear parabolic problems:

$$\partial_t u = \Delta u - y u_x + x u_y + f, \quad 0 \le x, y \le 1, \quad 0 \le t \le 1$$

with exact solution

$$u(x, y, t) = e^{-(x^2+3y^2+2t)}.$$

As shown in Table 4, it is clear that the numerical solution U reaches second-order convergence with only one iteration on the finest mesh, thanks to the initial value W, which provides an exceptionally accurate approximation. This plays a key role in the efficient solution of large linear systems. Also, this reaffirms the applicability of our proposed method for handling the convection-diffusion equation with variable coefficients.

Table 4: Convergence of the STEXCMG method for Example 4.4 with $tol = 10^{-9}$ and $\tau = h$.

$N_x \times N_y \times N_t$	Its	$\|U-u\|_\infty$	Order	$\|W-U\|_2$	Order
$32 \times 32 \times 32$	5	1.26e-04		5.44e-05	
$64 \times 64 \times 64$	12	3.14e-05	2.0112	6.80e-06	2.9993
$128 \times 128 \times 128$	24	7.83e-06	2.0025	8.43e-07	3.0117
$256 \times 256 \times 256$	14	1.95e-06	2.0041	1.05e-07	3.0041
$512 \times 512 \times 512$	1	4.92e-07	1.9898	1.22e-08	3.1107
1863.21 s	3.15 WU				

Example 4.5 ([23]). Consider Eqs. (1.1)-(1.2) with $a(\mathbf{x}) = 10^{-5}\sin(\pi xy)$, $b(\mathbf{x}) = c(\mathbf{x}) = 0$ and the source term

$$f(\mathbf{x},t) = e^{-t}x(1-x)\left[2\times 10^{-5}\sin(\pi xy) - y(1-y) - \pi\times 10^{-5}\cos(\pi xy)x(1-2y)\right] + e^{-t}y(1-y)\left[2\times 10^{-5}\sin(\pi xy) - \pi\times 10^{-5}\cos(\pi xy)y(1-2x)\right],$$

defined as a homogeneous heat equation given as follows:

$$\partial_t u = \nabla\cdot\left(10^{-5}\sin(\pi xy)\nabla u\right) + f, \quad 0\le x,y\le 1, \quad 0\le t\le 1$$

with exact solution

$$u(x,y,t) = e^{-t}x(1-x)y(1-y).$$

One more time, we use 5 embedded space-time grids with the finest grid $512\times512\times512$, the numerical simulations obtained by the STEXCMG method are enumerated in Table 5. From Table 5, we observe that only a few iterations are needed to process numerical solution U with second-order convergence because the initial value W already is an extremely good approximation of the numerical solution. This means that our proposed method is also suitable for the diffusion equation with variable coefficients.

Table 5: Convergence of the STEXCMG method for Example 4.5 with $tol = 10^{-9}$ and $\tau = h$.

$N_x \times N_y \times N_t$	Its	$\|U-u\|_\infty$	Order	$\|W-U\|_2$	Order
$32 \times 32 \times 32$	1	1.61e-06		4.39e-06	
$64 \times 64 \times 64$	1	4.02e-07	2.0000	2.84e-07	3.9492
$128 \times 128 \times 128$	2	1.01e-07	2.0000	1.83e-08	3.9549
$256 \times 256 \times 256$	4	2.51e-08	2.0000	1.22e-09	3.9060
$512 \times 512 \times 512$	1	6.28e-09	1.9998	6.03e-11	4.3417
1354.05 s	1.55 WU				

Example 4.6 ([17]). Finally, consider the case when $a(\mathbf{x}) = 1$, $b(\mathbf{x}) = (-\sin(x)\sin(y), -\cos(x)\cos(y))$, $c(\mathbf{x}) = 1$ and $f(\mathbf{x},t) = 0$. Then Eqs. (1.1)-(1.2) takes the following form:

$$\partial_t u = \Delta u + \sin(x)\sin(y)u_x + \cos(x)\cos(y)u_y + u, \quad 0\le x,y\le 1, \quad 0\le t\le 1$$

Table 6: Convergence of the STEXCMG method for Example 4.6 with $tol = 10^{-10}$ and $\tau = h$.

$N_x \times N_y \times N_t$	Its	$\|U-u\|_\infty$	Order	$\|W-U\|_2$	Order
$32 \times 32 \times 32$	5	2.31e-06		5.91e-06	
$64 \times 64 \times 64$	12	5.76e-07	2.0057	7.65e-07	2.9506
$128 \times 128 \times 128$	24	1.44e-07	1.9989	9.74e-08	2.9739
$256 \times 256 \times 256$	15	3.64e-08	1.9852	1.23e-08	2.9836
$512 \times 512 \times 512$	1	1.09e-08	1.7407	1.34e-09	3.1962
2054.64 s	3.27 WU				

with exact solution

$$u(\mathbf{x}, t) = e^{(-t)} \sin(x) \cos(y).$$

Last, the results in Table 6 agree with our analysis. Table 6 shows that the numerical solution U obtained by the STEXCMG with only one iteration on the finest $512 \times 512 \times 512$ grid has the full second-order accuracy in the L^∞-norm, and the initial guess W is a third-order approximation to the numerical solution. This means that our proposed method is also suitable for the AaO system from the CN temporal scheme.

4.2. Computational efficiency

Due to the high computational cost of SSOR-GMRES on $512 \times 512 \times 512$, an efficiency comparison is made between our method and SSOR-GMRES for Examples 4.3-4.6 on $256 \times 256 \times 256$. The number of iterations and CPU time are provided in Table 7. We observe that the STEXCMG method exhibits a considerably higher speed compared to the SSOR-GMRES method across all four examples. In detail, our method is more than 30 times faster than the SSOR-GMRES method for Example 4.4, while it is about 19 times, for Example 4.6 on a grid of $256 \times 256 \times 256$. These results indicate that the STEXCMG method, when combined with the SSOR-GMRES smoother, is capable of generating highly effective initial guesses for the preconditioned GMRES iteration, thereby achieving a substantial reduction in both the iteration count and the overall computational time.

Table 7: Comparison of the two iterative solvers on $256 \times 256 \times 256$.

Examples	tol	SSOR-GMRES		STEXCMG	
		Iters	CPU time(s)	Iters	CPU time(s)
Example 4.3	10^{-10}	311	30898.35	15	1775.84
Example 4.4	10^{-9}	251	25031.01	7	825.42
Example 4.5	10^{-9}	12	1089.88	4	390.77
Example 4.6	10^{-11}	350	34466.30	15	1781.22

5. Conclusions

In this paper, we introduce the STEXCMG method as an efficient solver for the AaO system arising from linear parabolic partial differential equations. Namely, we provide an effective initial guess using Richardson extrapolation and Lagrange interpolation, enabling the SSOR-preconditioned GMRES to handle a wide range of nonsymmetric AaO systems arising from discretizing the underlying equation. The method's performance and convergence are analyzed through various test problems, encompassing both one- and two-dimensional linear equations in Section 4. Furthermore, this novel space-time multigrid algorithm is not limited to the heat equation but is broadly applicable to general parabolic problems. In view of the advantages of high-order compact schemes and extrapolation, extensions of our algorithm to hyperbolic equations will be investigated in our future research work.

Acknowledgments

H.L. Hu was supported by the National Natural Science Foundation of China (Grant No. 12071128) and by the Natural Science Foundation of Hunan Province (Grant No. 2021JJ30434). We are grateful for resources from the High Performance Computing Center of Central South University.

References

[1] G. Akrivis, C. Makridakis, and R. Nochetto, *A posteriori error estimates for the Crank-Nicolson method for parabolic equations*, Math. Comput. 75(254) (2006), 511–531.
[2] J. Amoah-Mensah, F. O. Boateng, and K. Bonsu, *Numerical solution to parabolic PDE using implicit finite difference approach*, Math. Theory Model. 6(8) (2016), 74–84.
[3] P. Benedusi, M. L. Minion, and R. Krause, *An experimental comparison of a space-time multigrid method with PFASST for a reaction-diffusion problem*, Comput. Math. Appl. 99 (2021), 162–170.
[4] C. M. Chen, H. L. Hu, Z. Q. Xie, and C. L. Li, *Analysis of extrapolation cascadic multigrid method (EXCMG)*, Sci. China Ser. A Math. 51(8) (2008), 1349–1360.
[5] C. M. Chen and Q. Lin, *Extrapolation of finite element approximation in a rectangular domain*, J. Comput. Math. 7 (1989), 227–233.
[6] C. M. Chen, Z. C. Shi, and H. L. Hu, *On extrapolation cascadic multigrid method*, J. Comput. Math. 29(6) (2011), 684–697.
[7] J. Crank and P. Nicolson, *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type*, Adv. Comput. Math. 6(1) (1996), 207–226.
[8] M. C. Ding, X. F. Cai, W. Guo, and J. M. Qiu, *A semi-Lagrangian discontinuous Galerkin (DG) – local DG method for solving convection-diffusion equations*, J. Comput. Phys. 409 (2020), Paper No. 109295.
[9] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput. 36(6) (2014), C635–C661.

[10] S. R. Franco, F. J. Gaspar, M. A. V. Pinto, and C. Rodrigo, *Multigrid method based on a space-time approach with standard coarsening for parabolic problems*, Appl. Math. Comput. 317 (2018), 25–34.
[11] M. J. Gander, *50 years of time parallel time integration*, in: Multiple Shooting and Time Domain Decomposition Methods, Springer, (2015), 69–113.
[12] M. J. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Comput. Vis. Sci. 19(3-4) (2018), 59–74.
[13] M. J. Gander, J. Liu, S. L. Wu, X. Q. Yue, and T. Zhou, *ParaDiag: Parallel-in-time algorithms based on the diagonalization technique*, arXiv:2005.09158v4, (2021).
[14] M. J. Gander and M. Neumuller, *Analysis of a new space-time parallel multigrid algorithm for parabolic problems*, SIAM J. Sci. Comput. 38(4) (2016), A2173–A2208.
[15] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration metho*, SIAM J. Sci. Comput. 29(2) (2007), 556–578.
[16] A. Goddard and A. Wathen, *A note on parallel preconditioning for all-at-once evolutionary PDEs*, Electron. Trans. Numer. Anal. 51 (2019), 135–150.
[17] A. Golbabai and M. Javidi, *A variational iteration method for solving parabolic partial differential equations*, Comput. Math. Appl. 54 (2007), 987–992.
[18] W. Hackbusch, *Parabolic multi-grid methods*, Computing Methods in Applied Sciences and Engineering, VI, (1984), 189–197.
[19] J. Hahne, B. S. Southworth, and S. Friedhoff, *Asynchronous truncated multigrid-reduction-in-time*, SIAM J. Sci. Comput. 45(3) (2023), S281–S306.
[20] S. Hon and S. Serra-Capizzano, *A block Toeplitz preconditioner for all-at-once systems from linear wave equations*, Electron. Trans. Numer. Anal. 58 (2023), 177–195.
[21] G. Horton and S. Vandewalle, *A space-time multigrid method for parabolic partial differential equations*, SIAM J. Sci. Comput. 16(4) (1995), 848–864.
[22] Z. Hu and G. Li, *An efficient nonlinear multigrid solver for the simulation of rarefied gas cavity flow*, Commun. Comput. Phys. 34 (2023), 357–391.
[23] X. L. Lin and M. Ng, *An all-at-once preconditioner for evolutionary partial differential equations*, SIAM J. Sci. Comput. 43(4) (2021), A2766–A2784.
[24] J. L. Lions, Y. Maday, and G. Turinici, *A parareal in time discretization of PDEs*, C.R. Acad. Sci. Paris, Serie I 332 (2021), 661–668.
[25] G. R. Liu and S. S. Quek, *The Finite Element Method: A Practical Course*, Butterworth-Heinemann, (2013).
[26] C. Lubich and A. Ostermann, *Multi-grid dynamic iteration for parabolic equation*, BIT 27 (1987), 216–234.
[27] N. Margenberg and P. Munch, *Space-time multigrid method for space-time finite element discretizations of parabolic and hyperbolic PDEs*, arXiv:2408.04372, (2024).
[28] E. McDonald, J. Pestana, and A. Wathen, *Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations*, SIAM J. Sci. Comput. 40(2) (2018), A1012–A1033.
[29] Y. Notay, *Rigorous convergence proof of space-time multigrid with coarsening in space*, Numer. Algorithms 89 (2021), 675–699.
[30] K. J. Pan, D. D. He, and H. L. Hu, *An extrapolation cascadic multigrid method combined with a fourth-order compact scheme for 3D Poisson equation*, J. Sci. Comput. 70(3) (2017), 1180–1203.
[31] K. J. Pan, D. D. He, H. L. Hu, and Z. Y. Ren, *A new extrapolation cascadic multigrid method for three dimensional elliptic boundary value problems*, J. Comput. Phys. 344

(2017), 499–515.
[32] V. V. SHAIDUROV, *Some estimates of the rate of convergence for the cascadic conjugate-gradient method*, Comput. Math. Appl. 31(1) (1996), 161–171.
[33] G. STAFF AND E. RØNQUIST, *Stability of the parareal algorithm*, Lect. Notes Comput. Sci. Eng. 40, Springer (2005), 449–456.
[34] L. L. SUN, H. BI, AND Y. YANG, *A multigrid discretization of discontinuous Galerkin method for the Stokes eigenvalue problem*, Commun. Comput. Phys. 34 (2023), 1391–1419.
[35] R. TIELEN, M. MÖLLER, AND C. VUIK, *Combining p-multigrid and multigrid reduction in time methods to obtain a scalable solver for isogeometric analysis*, SN Appl. Sci. 4 (2022), Paper No. 163.
[36] A. WATHEN, *Some observations on preconditioning for non-self-adjoint and time-dependent problems*, Comput. Math. Appl. 116 (2022), 176–180.
[37] T. WEINZIERL AND T. KÖPPL, *A geometric space-time multigrid algorithm for the heat equation*, Numer. Math. Theory Methods Appl. 5(1) (2012), 110–130.
[38] H. WU AND W. ZHENG, *Uniform convergence of multigrid V-cycle on adaptively refined finite element meshes for elliptic problems with discontinuous coefficients*, Commun. Math. Res. 39 (2023), 437–475.
[39] Y. L. ZHAO, X. M. GU, AND C. W. OOSTERLEE, *A parallel preconditioner for the all-at-once linear system from evolutionary PDEs with Crank-Nicolson discretization*, arXiv:2401.16113, (2024).