Numerical Algorithms (2023) 94:29–72
https://doi.org/10.1007/s11075-022-01492-8

ORIGINAL PAPER

Interpretation of parareal as a two-level additive Schwarz in time preconditioner and its acceleration with GMRES

Van-Thanh Nguyen[1,2] · Laura Grigori[1,2]

Received: 20 March 2022 / Accepted: 20 December 2022 / Published online: 1 March 2023

Abstract
We describe an interpretation of parareal as a two-level additive Schwarz preconditioner in the time domain. We show that this two-level preconditioner in time is equivalent to parareal and to multigrid reduction in time (MGRIT) with F-relaxation. We also discuss the case when additional fine or coarse propagation steps are applied in the preconditioner. This leads to procedures equivalent to MGRIT with FCF-relaxation and to MGRIT with F(CF)^2-relaxation or overlapping parareal. Numerical results show that these variants have faster convergence in some cases. In addition, we also apply a Krylov subspace method, namely GMRES (generalized minimal residual), to accelerate the parareal algorithm. Better convergence is obtained, especially for the advection-reaction-diffusion equation in the case when advection and reaction coefficients are large.

Keywords Parareal · Two-level additive Schwarz in time preconditioner · MGRIT with F-relaxation · FCF-relaxation · F(CF)^2-relaxation · GMRES

1 Introduction

In this paper, we focus on parareal, an algorithm introduced by J.L. Lions et al. [1] in 2001, which allows to exploit parallelism in time for initial value problems. Over

Laura Grigori contributed equally to this work.

✉ Van-Thanh Nguyen
van-thanh.nguyen@inria.fr

Laura Grigori
laura.grigori@inria.fr

[1] Inria, Project-Team ALPINES, 2 Rue Simone IFF, Paris, 75012, France

[2] CNRS, Laboratoire Jacques-Louis Lions (LJLL), Sorbonne Université and Université de Paris,
4 Place Jussieu, Paris, 75005, France

the last two decades, this algorithm has been studied for a range of applications, going from molecular dynamics simulations [2], unsteady hydrodynamic simulations [3], kinetic neutron diffusion equation [4, 5], the Korteveg-deVries-Burgers' equations [6], Hamiltonian systems [7, 8], to financial mathematics as the Black-Scholes equations [9–11]. Its stability and convergence are studied in a series of papers, e.g., [12–15]. Given a time-dependent problem, parareal allows parallel in time integration by relying on a combination between a fine propagator, which gives a very accurate approximate of the solution, and a coarse propagator, which is less expensive and gives a coarse approximate of the solution. For this, the time domain is decomposed into a number of uniform time subdomains. From an initial solution obtained by sequentially using the coarse propagator, parareal iteratively corrects it by the difference between the fine solution obtained in parallel using the fine propagator and the coarse solution obtained from the previous iteration.

Several different interpretations of parareal exist in the literature. A derivation of the parareal algorithm as a multiple shooting method is given in [13]. An investigation of the usage of spectral deferred corrections in the framework of parareal is given in [16, 17]. Coupling parareal in time with Schwarz waveform relaxation methods [18, 19] to exploit parallelism in both time and space are promising directions of research as well. Parareal can also be interpreted as a multigrid method in time, referred to as MGRIT with F-relaxation [20, 21]. Following this interpretation, several different variants have been investigated, as MGRIT with FCF-relaxation, MGRIT with F(CF)^2-relaxation, e.g., [21, 22], where F refers to the F-relaxation and C refers to the C-relaxation.

Given that parareal relies on a decomposition of the time domain into subdomains, in this paper, we study the connection between parareal and domain decomposition methods. Traditionally domain decomposition methods are used for solving a linear system of equations $\tilde{A}\tilde{u} = \tilde{f}$, $\tilde{A} \in \mathbb{R}^{n\times n}$, arising from the discretization of a PDE by using, for example, the finite element method, and they rely on a decomposition of the space domain into subdomains. We consider here the case in which this linear system is solved by using an iterative method as a Krylov subspace method, preconditioned by $\tilde{M}^{-1}$,

$$\tilde{M}^{-1}\tilde{A}\tilde{u} = \tilde{M}^{-1}\tilde{f},$$

where $\tilde{M}^{-1}$ is a domain decomposition method. One-level domain decomposition preconditioners such as additive and multiplicative Schwarz preconditioners are well-known in the literature for domain decomposition in space, see, e.g., [23]. However, their convergence rate deteriorates when the number of subdomains becomes large because of a lack of global information coupling the subdomains. In order to obtain a scalable domain decomposition algorithm which depends weakly on the number of subdomains, a coarse space can be used to couple global information of all subdomains. This leads to the idea of two-level domain decomposition preconditioners. Given a spatial decomposition of the degrees of freedom of $\tilde{A}$ into $\tilde{N}$ subdomains, the restriction of $\tilde{A}$ to a spatial subdomain i, for $i = 1, \ldots, \tilde{N}$, is referred to as $\tilde{A}_i$ and is obtained by defining a restriction matrix $\tilde{R}_i$ together with a prolongation matrix $\tilde{R}_i^T$, such that $\tilde{A}_i = \tilde{R}_i\tilde{A}\tilde{R}_i^T$. By defining the coarse matrix $\tilde{A}_0$ and corresponding restriction and prolongation matrices $\tilde{R}_0$, $\tilde{R}_0^T$, the two-level additive Schwarz preconditioner is defined as follows,

$$\tilde{M}_{AS2}^{-1} = \tilde{R}_0^T \tilde{A}_0^{-1} \tilde{R}_0 + \sum_{i=1}^{\tilde{N}} \tilde{R}_i^T \tilde{A}_i^{-1} \tilde{R}_i,$$

and the two-level multiplicative Schwarz preconditioner is analogously defined as follows

$$\tilde{M}_{MS2}^{-1} = \left[\mathcal{I} - (\mathcal{I} - \tilde{R}_0^T \tilde{A}_0^{-1} \tilde{R}_0 \tilde{A}) \prod_{i=1}^{\tilde{N}} (\mathcal{I} - \tilde{R}_i^T \tilde{A}_i^{-1} \tilde{R}_i \tilde{A}) \right] \tilde{A}^{-1}.$$

To show the equivalence between parareal and two-level domain decomposition methods, we consider the linear time-dependent problem,

$$\frac{du}{dt} = f(u) \ \ , \ u(0) = \mathrm{u}_0, \ \ u(t) \in \mathbb{R}^d, t \ \text{ in } (0, T), \tag{1}$$

and an algebraic framework in which the solution to (1) can be obtained by solving with a residual correction scheme the linear system of equations,

$$AU_F = f, \tag{2}$$

where the time domain $(0, T)$ was decomposed into N time subdomains, $A \in \mathbb{R}^{(N+1)d \times (N+1)d}$ is bidiagonal and denotes the time-stepping coefficient matrix with the form,

$$A := \begin{bmatrix} \mathcal{I} & & & \\ -\phi & \mathcal{I} & & \\ & \ddots & \ddots & \\ & & -\phi & \mathcal{I} \end{bmatrix}.$$

In this equation, $\mathcal{I} \in \mathbb{R}^{d \times d}$ is the identity matrix, $\phi \in \mathbb{R}^{d \times d}$ denotes an arbitrary stable discretization method in space and time, $U_F := [u_0, \ldots, u_N]^T$ denotes the solution at fine time steps, and $f := [\mathrm{u}_0, 0 \ldots, 0]^T$ is the right-hand side. The matrix A includes all the time steps for the whole time domain. If N and d become large, (2) results in a very large and sparse system. This is the case where domain decomposition type methods show their advantages. We consider the problem on a uniform grid, the time steps and space steps do not change from one to the next so the discretization matrix for each time step, namely ϕ, does not change . We show that parareal is equivalent to using the preconditioned stationary iteration which computes a new approximate solution U_F^{k+1} from U_F^k,

$$U_F^{k+1} = U_F^k + M_{SC}^{-1}(f - AU_F^k),$$

where M_{SC}^{-1} is a two-level additive Schwarz in time preconditioner defined as follows,

$$M_{SC}^{-1} = (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \left(\sum_{i=1}^{\hat{N}} R_i^T A_i^{-1} R_i \right). \tag{3}$$

Remark 1 The preconditioner M_{SC}^{-1} in (3) is different from the so-called hybrid preconditioner where subdomain preconditioning is applied additively, but the coarse solve is applied multiplicatively in the second stage, i.e.,

$$M_{hybrid}^{-1} = R_0^T A_0^{-1} R_0 + (\mathbb{I} - R_0^T A_0^{-1} R_0) \left(\sum_{i=1}^{\hat{N}} R_i^T A_i^{-1} R_i \right).$$

A symmetrized version of this preconditioner appears as P_{hy1} in the standard reference of Toselli and Widlund (2005) for domain decomposition methods, [24].

We give in Section 3 the exact definitions of the subdomain matrices A_i, for $i = 1, \ldots, \hat{N}$, the coarse time correction matrix A_0, as well as the restriction and prolongation matrices R_i, R_i^T, for $i = 0, \ldots, \hat{N}$, where $\hat{N}$ is the number of subdomain matrices of A. The matrix $\mathbb{I} \in \mathbb{R}^{(N+1)d \times (N+1)d}$ is the identity matrix. The first term denotes an additive Schwarz preconditioner in time, which is computed in parallel by using the fine propagators, followed by a coarse correction in time, based on a coarse propagator, which is computed sequentially and transfers the information globally between the different time subdomains.

Furthermore, we show that this two-level additive Schwarz in time preconditioner has the same error propagation as MGRIT with F-relaxation at coarse time points, discussed in [20, 21, 25]. As expected, this shows that the three algorithms parareal, MGRIT with F-relaxation, and two-level additive Schwarz in time preconditioner from (3) are equivalent. We also discuss that applying additional fine or coarse propagation steps in the two-level additive Schwarz in time preconditioner is equivalent to MGRIT with FCF-relaxation and MGRIT with F(CF)^2-relaxation or overlapping parareal, discussed in [26]. Faster convergence can be achieved in some cases, but the trade-off is also important to consider. To improve the convergence, a variant of two-level domain decomposition method, referred to as SCS^2 two-level additive Schwarz in time preconditioner, provides a good alternative, since it relies on increasing the number of additive Schwarz in time steps, while keeping only one coarse correction step, which is performed in sequential. Note that the notations S and C used here in the context of two-level additive Schwarz in time preconditioner correspond to the use of fine and coarse propagators. They are different from F-relaxation and C-relaxation used in MGRIT. Specifically, S and C propagation steps in the two-level additive Schwarz in time preconditioner start from the same coarse time points and propagate to obtain the approximate solution at the end of each time subdomain. While F-relaxation propagates to obtain the approximate solution at fine time points based on the coarse time points, and C-relaxation propagates to obtain the approximate solution at coarse time points based on the previous fine time points, for more details, see [25]. We also explore the usage of Krylov subspace methods for solving the system (2). This gives promising numerical results, especially for solving the advection-reaction-diffusion equation with large advection and reaction terms.

The paper is organized as follows. Section 2 recalls parareal algorithm and its formulation as a residual correction scheme. Section 3 introduces an interpretation of parareal as a two-level additive Schwarz in time preconditioner. Section 4 discusses several variants of this two-level additive Schwarz in time preconditioner and gives their convergence analysis. Furthermore, theoretical convergence bounds are given in Section 5. Several numerical experiments are presented in Section 6, where we consider the Dahlquist problem, the heat equation, and the advection-reaction-diffusion equation. Conclusions and perspectives are given in Section 7.

2 Parareal algorithm

In this section, we describe the parareal algorithm by following its presentation from, e.g., [13]. For the simplicity of the exposition, we consider the scalar linear time-dependent problem,

$$\frac{du}{dt} = f(u) \ , \ u(0) = \mathrm{u}_0, \quad u(t) \in \mathbb{R}^d, t \ \text{ in } (0, T), \tag{4}$$

The time interval $[0, T]$ is decomposed into N_C uniform time subdomains $[T_n, T_{n+1}]$ with $n = 0, \ldots, N_C - 1$. Parareal uses two solvers, a fine solver $\mathcal{F}(T_{n+1}, T_n, U_n)$, which gives a very good approximate, and a coarse solver $\mathcal{G}(T_{n+1}, T_n, U_n)$, which gives a coarse approximate of the solution at time T_{n+1} starting from the initial solution U_n at time T_n. The initial approximate U_n^0 at coarse time points is obtained typically by using sequentially the coarse solver,

$$U_{n+1}^0 = \mathcal{G}(T_{n+1}, T_n, U_n^0), \ U_0^0 = \mathrm{u}_0.$$

From this initial solution in time, parareal iteratively computes a new approximate of the solution of (4) until some convergence criterion is met. At each iteration $k + 1$, $k \geq 0$, a new approximate is computed as follows,

$$U_{n+1}^{k+1} = \mathcal{G}(T_{n+1}, T_n, U_n^{k+1}) + \mathcal{F}(T_{n+1}, T_n, U_n^k) - \mathcal{G}(T_{n+1}, T_n, U_n^k). \tag{5}$$

The coarse and the fine solvers can be chosen in various ways. Very often a higher order approximation is used for the fine solver and a lower order approximation is used for the coarse solver. The coarse solver can also solve a different problem, which is simpler to solve than the original one, as long as it gives an acceptable approximate of the solution. However, the coarse solver plays an important role in the convergence of the parareal algorithm. It should be chosen in such a way that it is cheap but accurate enough compared to the fine one, otherwise parareal algorithm can converge slowly. One simple approach is to choose the same discretizations in both time and space for both coarse and fine solvers, but with larger time step Δt for the coarse solver and smaller δt for the fine solver. Furthermore, one can also use a coarsened spatial mesh for the coarse solver, see [27].

2.1 Parareal execution from an algebraic point of view

Consider the time-dependent problem from (4) for which the time interval $[0, T]$ is divided into N uniform time slices $[t_n, t_{n+1}]$ with length δt, for $n = 0, \ldots, N - 1$. On the other hand, $[0, T]$ is also partitioned into N_C uniform coarse time intervals $[T_l, T_{l+1}]$ with length ΔT, for $l = 0, \ldots, N_C - 1$. We denote by ϕ a stable discretization method in time such as forward Euler, backward Euler, Runge-Kutta or higher order methods, and by $\phi_{\Delta T}$ the coarse solver for which the same methods are used but with larger time step, or lower order methods or spatial coarsening, in particular $\phi_{\Delta T}$ approximates the fine solver ϕ^m. Let δt be the fine time step and $\Delta t = \Delta T = m\delta t$ be the coarse time step (we use one coarse time step for the coarse solver on each coarse time interval), in which m denotes the number of fine time steps on each coarse time interval. We note that the error propagation and convergence analysis in Sections 3, 4, and 5 are based on the assumption that ϕ and $\phi_{\Delta T}$ can be diagonalized by the same set of eigenvectors, in cases when when ϕ and $\phi_{\Delta T}$ have the same spatial discretization as stated in [25]. Furthermore, the analysis of spatial discretization can also be found in [28]. Without loss of generality, we consider in this work the same discretization methods in both time and space for both coarse and fine solvers, namely the backward Euler in time and centered finite difference method in space. However, discretizations as forward Euler, Runge-Kutta or higher order methods can also be used in the same framework, we illustrate this by using Runge-Kutta 4 for the fine solver in Section 6.4. In this paper, we focus on the linear constant-coefficient partial differential equations, in particular the heat equation and the advection-reaction-diffusion equation. By sequentially applying ϕ, the linear system of equations obtained has the form:

$$AU_F := \begin{bmatrix} \mathcal{I} & & & \\ -\phi & \mathcal{I} & & \\ & \ddots & \ddots & \\ & & -\phi & \mathcal{I} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} \mathrm{u}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} =: f, \tag{6}$$

where $A \in \mathbb{R}^{(N+1)d\times(N+1)d}$ denotes the time-stepping coefficient matrix, $\mathcal{I} \in \mathbb{R}^{d\times d}$ denotes the identity matrix, and $\phi \in \mathbb{R}^{d\times d}$ denotes the discretization matrix. This system of equations can be solved by using a direct method in which the solutions $u_i, i = 0, \ldots, N$ at different time steps are obtained sequentially. This results in a complexity of N time steps, each time step being solved by using ϕ. But instead of just using ϕ, parareal combines the use of both coarse and fine solvers to result in a faster algorithm in which the fine solvers are performed in parallel.

We describe parareal by considering a simple two-level temporal mesh for which $m = 2$, as displayed in Fig. 1. With this choice, the fine nodes are defined at

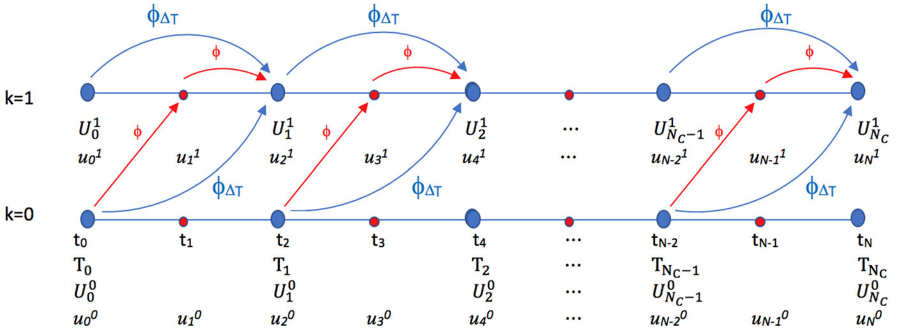

Fig. 1 Two-level temporal mesh and parareal execution

all time points $\{t_0, t_1, t_2, \ldots, t_N\}$, while the coarse nodes are defined at even time points $\{t_0, t_2, \ldots, t_N\}$. At the initial step $k = 0$, the initial approximate of the coarse solution is obtained by applying $\phi_{\Delta T}$ sequentially and the fine solution is obtained by interpolating. Let $\mathcal{F}(T_{n+1}, T_n, U_n^k) := \phi^2 U_n^k$ be the fine propagator and $\mathcal{G}(T_{n+1}, T_n, U_n^k) := \phi_{\Delta T} U_n^k$ be the coarse propagator, parareal iteration from (5) becomes,

$$U_{n+1}^{k+1} = \phi_{\Delta T} U_n^{k+1} + \phi^2 U_n^k - \phi_{\Delta T} U_n^k,$$

where U_n^k corresponds to u_{2n}^k which denotes the parareal solution at coarse time point $t_{2n},\ n = 0, \ldots, N/2$ and iteration k. In detail, parareal computes the approximate solutions at fine time points as follows,

$$u_i^{k+1} = \begin{cases} \mathrm{u}_0, & \text{if } i = 0, \\ \phi u_{i-1}^k, & \text{for } i = 1, 3, \ldots, N-1, \\ \phi_{\Delta T} u_{i-2}^{k+1} + \phi u_{i-1}^{k+1} - \phi_{\Delta T} u_{i-2}^k, & \text{for } i = 2, 4, \ldots, N. \end{cases} \tag{7}$$

As it can be seen from Fig. 1, the fine approximate solutions can be computed in parallel based on the coarse approximate solutions from the previous iterations. Generally for arbitrary $m \geq 2$, we have similarly,

$$U_{n+1}^{k+1} = \phi_{\Delta T} U_n^{k+1} + \phi^m U_n^k - \phi_{\Delta T} U_n^k. \tag{8}$$

2.2 Expression of the standard residual correction scheme

As presented in, e.g., [26], parareal algorithm can be seen as a preconditioned residual correction scheme of a reduced system representing only the coarse time solutions, which is obtained from the original system of (6). For this, a coarse matrix A_C represents the time steps of the coarse level (here we keep every second time point on each time interval), U_C represents the unknown solutions and f_C the right-hand side at coarse time points,

$$A_C U_C := \begin{bmatrix} \mathcal{I} & & & & \\ -\phi^2 & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & -\phi^2 & \mathcal{I} & \\ & & & -\phi^2 & \mathcal{I} \end{bmatrix} \begin{bmatrix} u_0 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_N \end{bmatrix} = \begin{bmatrix} \mathrm{u}_0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} =: f_C. \tag{9}$$

This reduced system of (9) produces exactly the same solutions as the original system (6) at coarse time points. A preconditioner $\tilde{M}$ which approximates the coarse matrix A is obtained by approximating each fine time integration propagator ϕ^2 by one coarse integration propagator $\phi_{\Delta T}$,

$$\tilde{M} := \begin{bmatrix} \mathcal{I} & & & & \\ -\phi_{\Delta T} & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & -\phi_{\Delta T} & \mathcal{I} & \\ & & & -\phi_{\Delta T} & \mathcal{I} \end{bmatrix}.$$

By using the preconditioned stationary iteration at the coarse level, we obtain at iteration k,

$$U_C^{k+1} = U_C^k + \tilde{M}^{-1}(f_C - A_C U_C^k), \tag{10}$$

which can be written as follows,

$$\tilde{M}(U_C^{k+1} - U_C^k) = f_C - A_C U_C^k, \tag{11}$$

or explicitly written as follows,

$$\begin{bmatrix} \mathcal{I} & & & & \\ -\phi_{\Delta T} & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & -\phi_{\Delta T} & \mathcal{I} & \\ & & & -\phi_{\Delta T} & \mathcal{I} \end{bmatrix} \begin{bmatrix} u_0^{k+1} - u_0^k \\ u_2^{k+1} - u_2^k \\ \vdots \\ u_{N-2}^{k+1} - u_{N-2}^k \\ u_N^{k+1} - u_N^k \end{bmatrix} = \begin{bmatrix} \mathrm{u}_0 - u_0^k \\ \phi^2 u_0^k - u_2^k \\ \vdots \\ \phi^2 u_{N-4}^k - u_{N-2}^k \\ \phi^2 u_{N-2}^k - u_N^k \end{bmatrix}. \tag{12}$$

It can be easily seen that the solutions u_{2i}^{k+1} for $i = 0, \ldots, N/2$ obtained by solving (12) are the same as the solutions obtained by parareal in (7).

We consider now solving the system $AU_F = f$ from (6) at the fine level. We introduce a matrix M_{SC} and we show that parareal algorithm is equivalent to solving $AU_F = f$ by using a stationary iteration preconditioned by M_{SC}^{-1}. The preconditioned stationary iteration for solving $AU_F = f$ at the fine level becomes,

$$U_F^{k+1} = U_F^k + M_{SC}^{-1}(f - AU_F^k), \tag{13}$$

or equivalently,

$$M_{SC}(U_F^{k+1} - U_F^k) = f - AU_F^k, \tag{14}$$

Note that (14) acts at the fine level, so M_{SC} is different from $\tilde{M}$ in (11). In other words, M_{SC} has to deal with both unknowns at coarse and fine time points. The matrix M_{SC} is defined in the following lemma.

Lemma 1 *Let $\mathcal{F}(T_{n+1}, T_n, u_n^k) := \phi^m u_n^k$ and $\mathcal{G}(T_{n+1}, T_n, u_n^k) := \phi_{\Delta T} u_n^k$ denote the fine and the coarse solvers, respectively. For $m \geq 2$, (14) is equivalent to parareal algorithm with M_{SC} defined as follows,*

$$M_{SC} := \begin{bmatrix} \left.\begin{matrix} \mathcal{I} & & & & \\ & \mathcal{I} & & & \\ & -\phi & \mathcal{I} & & \\ & & & \ddots & \\ -\phi_{\Delta T} & & & -\phi & \mathcal{I} \end{matrix}\right\} md \times md & & & & \\ & \ddots & & & \\ & & \mathcal{I} & & \\ & & -\phi & \mathcal{I} & \\ & & & \ddots & \\ & -\phi_{\Delta T} & & -\phi & \mathcal{I} \end{bmatrix}. \tag{15}$$

Proof For $m = 2$, (14) becomes,

$$\begin{bmatrix} \mathcal{I} & & & & & \\ & \mathcal{I} & & & & \\ -\phi_{\Delta T} & -\phi & \mathcal{I} & & & \\ & & & \ddots & & \\ & & & & \mathcal{I} & \\ & & & -\phi_{\Delta T} & -\phi & \mathcal{I} \end{bmatrix} \begin{bmatrix} u_0^{k+1} - u_0^k \\ u_1^{k+1} - u_1^k \\ u_2^{k+1} - u_2^k \\ \vdots \\ u_{N-1}^{k+1} - u_{N-1}^k \\ u_N^{k+1} - u_N^k \end{bmatrix} = \begin{bmatrix} \mathrm{u}_0 - u_0^k \\ \phi u_0^k - u_1^k \\ \phi u_1^k - u_2^k \\ \vdots \\ \phi u_{N-2}^k - u_{N-1}^k \\ \phi u_{N-1}^k - u_N^k \end{bmatrix}. \qquad (16)$$

Simplifying (16) gives

$$u_i^{k+1} = \begin{cases} \mathrm{u}_0, & \text{if } i = 0, \\ \phi u_{i-1}^k, & \text{for } i = 1, 3, \ldots, N-1, \\ \phi_{\Delta T} u_{i-2}^{k+1} + \phi u_{i-1}^{k+1} - \phi_{\Delta T} u_{i-2}^k, & \text{for } i = 2, 4, \ldots, N. \end{cases}$$

Generalize for $m > 2$, for $j = 0, m, 2m, \ldots, N-m$ and by induction, we have,

$$\begin{aligned} u_0^{k+1} &= \mathrm{u}_0, \\ u_{j+1}^{k+1} &= \phi u_j^k, \\ u_{j+2}^{k+1} &= \phi u_{j+1}^{k+1}, \\ &\vdots \\ u_{j+m-1}^{k+1} &= \phi u_{j+m-2}^{k+1}, \\ u_{j+m}^{k+1} &= \phi_{\Delta T} u_j^{k+1} + \phi u_{j+m-1}^{k+1} - \phi_{\Delta T} u_j^k = \phi_{\Delta T} u_j^{k+1} + \phi^m u_j^k - \phi_{\Delta T} u_j^k, \end{aligned}$$

which is identical to (8) and concludes the proof. □

Hence, instead of solving the system (6) by using a direct method, parareal algorithm uses the stationary iteration defined in (13) preconditioned by M_{SC} as defined in (15). In addition, Krylov subspace methods as GMRES can also be used to accelerate the convergence of parareal. In the numerical experiments, Section 6.3 we present results obtained by using GMRES for solving the preconditioned linear system,

$$M_{SC}^{-1} A U_F = M_{SC}^{-1} f.$$

It will be seen that GMRES improves slightly the convergence of parareal and it allows to solve problems for which parareal has difficulty to converge, as in the case when the advection and reaction coefficients are large compared to the diffusion term for the advection-reaction-diffusion problem. However, in general, it does not improve drastically the convergence of parareal for our test problems, and this was also observed in previous works as [29] which studied the acceleration of waveform relaxation methods.

3 Interpretation of parareal as a two-level additive Schwarz in time preconditioner

In this section, we present an interpretation of parareal as a two-level domain decomposition method. For this, we show that the inverse of the preconditioner M_{SC} from (15) can be expressed as a first level additive Schwarz preconditioner that relies on using the fine propagator ϕ^m in each time subdomain, followed by a coarse time correction based on using the coarse propagator $\phi_{\Delta T}$.

We introduce first some notations. Let $A \in \mathbb{R}^{(N+1)d\times(N+1)d}$ be the time-stepping matrix as defined in Section 2.1. The matrices $\mathbb{I} \in \mathbb{R}^{(N+1)d\times(N+1)d}$ and $\mathcal{I} \in \mathbb{R}^{d\times d}$ are identity matrices. The matrix A is decomposed into $N_C + 1$ non-overlapping subdomains $\{\Omega_i\}_{1\le i\le N_C+1}$, where $N_C = N/m$ denotes the number of coarse time intervals. The matrix A is a block matrix, the blocks being defined as $\{A_{ij}\}_{1\le i,j\le N+1} \in \mathbb{R}^{d\times d}$. As displayed in (6), A_{ij} can be the ϕ matrix, the identity or the zero matrix. Let $\mathfrak{N} = \{1,\ldots,N+1\}$ be the set of indices of A, which corresponds to the fine time steps $\{t_0,\ldots,t_N\}$. Let $\mathfrak{N}_i$, $i \in \{1,\ldots,N_C+1\}$ be the subset of $\mathfrak{N}$ such that $\mathfrak{N}_i$ represents the subset of indices of subdomain i, we define $\mathfrak{N}_i$ as follows,

$$\mathfrak{N}_i = \begin{cases} \{1\}, & \text{if } i = 1, \\ \{m(i-2)+2,\ldots,m(i-1)+1\}, & \text{for } i = 2,\ldots,N_C+1, \end{cases} \tag{17}$$

the restriction matrix R_i is defined as follows,

$$R_i = \begin{cases} \mathcal{I}, & \text{if } i = 1, \\ \mathbb{I}(\mathfrak{N}_i, :), & \text{for } i = 2,\ldots,N_C+1, \end{cases} \tag{18}$$

where $\mathbb{I}(\mathfrak{N}_i, :)$ denotes the submatrix of $\mathbb{I}$ formed by the rows whose indices belong to $\mathfrak{N}_i$. The prolongation matrix R_i^T is the transpose of R_i. The subdomain matrices $\{A_i\}_{1\le i\le N_C+1}$ are defined as follows,

$$A_i = \begin{cases} \mathcal{I}, & \text{if } i = 1, \\ R_i A R_i^T = \begin{bmatrix} \mathcal{I} & & & & \\ -\phi & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & -\phi & \mathcal{I} & \\ & & & -\phi & \mathcal{I} \end{bmatrix}, & \text{for } i = 2,\ldots,N_C+1. \end{cases}$$

For $i \geq 2$, $A_i = R_i A R_i^T$ is an $md \times md$ block matrix. The inverse of A_i can be computed as follows,

$$A_i^{-1} = \begin{cases} \mathcal{I}, & \text{if } i = 1, \\ \begin{bmatrix} \mathcal{I} & & & & & & \\ \phi & \mathcal{I} & & & & & \\ \phi^2 & \phi & \mathcal{I} & & & & \\ \phi^3 & \phi^2 & \phi & \mathcal{I} & & & \\ \vdots & \vdots & & \ddots & \ddots & & \\ \phi^{m-2} & \phi^{m-3} & & \dots & \phi & \mathcal{I} & \\ \phi^{m-1} & \phi^{m-2} & & \dots & & \phi & \mathcal{I} \end{bmatrix}, & \text{for } i = 2, \dots, N_C + 1. \end{cases}$$

The first level additive Schwarz in time preconditioner is $\sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i$ as presented in Section 1. The second level coarse time correction is defined as following. Let $\mathfrak{N}_0 = \{1 + im\}_{0 \leq i \leq N_C}$ be the set of indices corresponding to coarse time points and $A_0 \in \mathbb{R}^{(N_C+1)d \times (N_C+1)d}$ be the coarse matrix that solves the reduced system from (6) at every coarse time point by using the coarse integration propagator $\phi_{\Delta T}$,

$$A_0 = \begin{bmatrix} \mathcal{I} & & & & \\ -\phi_{\Delta T} & \mathcal{I} & & & \\ & \ddots & \ddots & & \\ & & -\phi_{\Delta T} & \mathcal{I} & \\ & & & -\phi_{\Delta T} & \mathcal{I} \end{bmatrix}. \tag{19}$$

The coarse problem at coarse time points in the time domain is obtained by using a restriction matrix $R_0 \in \mathbb{R}^{(N_C+1)d \times (N+1)d}$, defined such that the entries of R_0 are identities at positions corresponding to the coarse time points and 0 elsewhere. In particular, R_0 is defined as follows,

$$R_0 = \mathbb{I}(\mathfrak{N}_0, :), \tag{20}$$

in which $\mathfrak{N}_0 = \{1, 1+m, 1+2m, \dots, 1+N_C m\}$ and the prolongation matrix for the coarse problem is the transpose of R_0. The inverse of A_0 can be computed as follows,

$$A_0^{-1} = \begin{bmatrix} \mathcal{I} & & & & & & \\ \phi_{\Delta T} & \mathcal{I} & & & & & \\ \phi_{\Delta T}^2 & \phi_{\Delta T} & \mathcal{I} & & & & \\ \phi_{\Delta T}^3 & \phi_{\Delta T}^2 & \phi_{\Delta T} & \mathcal{I} & & & \\ \vdots & \vdots & & \ddots & \ddots & & \\ \phi_{\Delta T}^{N_C-1} & \phi_{\Delta T}^{N_C-2} & & \dots & \phi_{\Delta T} & \mathcal{I} & \\ \phi_{\Delta T}^{N_C} & \phi_{\Delta T}^{N_C-1} & & \dots & & \phi_{\Delta T} & \mathcal{I} \end{bmatrix}.$$

Lemma 2 *The matrix M_{SC} defined in (15) can be factored as follows,*

$$M_{SC} = \left(\sum_{i=1}^{N_C+1} R_i^T A_i R_i \right) (R_0^T A_0 R_0 + \mathbb{I} - R_0^T R_0),$$

and the additive Schwarz in time preconditioner M_{SC}^{-1} is formed by the product of the additive Schwarz term $\sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i$ and the coarse time correction term $R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0$,

$$M_{SC}^{-1} = (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \left(\sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i \right). \tag{21}$$

Proof We have

$$\left(\sum_{i=1}^{N_C+1} R_i^T A_i R_i \right) (R_0^T A_0 R_0 + \mathbb{I} - R_0^T R_0)$$

$$= \begin{bmatrix} \left.\begin{matrix} \mathcal{I} & & & \\ & \mathcal{I} & & \\ & -\phi\ \mathcal{I} & & \\ & & \ddots & \\ & & -\phi & \mathcal{I} \end{matrix}\right\} md \times md & & & \\ & \ddots & & \\ & & \begin{matrix} \mathcal{I} & \\ -\phi & \mathcal{I} \\ & \ddots \\ & -\phi\ \mathcal{I} \end{matrix} & \end{bmatrix} \begin{bmatrix} \left.\begin{matrix} \mathcal{I} & & & \\ & \mathcal{I} & & \\ & & \mathcal{I} & \\ & & & \ddots \\ -\phi_{\Delta T} & & & \mathcal{I} \end{matrix}\right\} md \times md & & \\ & \ddots & \\ & & \begin{matrix} \mathcal{I} & & \\ & \mathcal{I} & \\ & & \ddots \\ -\phi_{\Delta T} & & \mathcal{I} \end{matrix} \end{bmatrix}$$

$$= \begin{bmatrix} \left.\begin{matrix} \mathcal{I} & & & \\ & \mathcal{I} & & \\ & -\phi\ \mathcal{I} & & \\ & & \ddots & \\ -\phi_{\Delta T} & & -\phi & \mathcal{I} \end{matrix}\right\} md \times md & & \\ & \ddots & \\ & & \begin{matrix} & \mathcal{I} & \\ & -\phi\ \mathcal{I} & \\ & & \ddots \\ -\phi_{\Delta T} & & -\phi\ \mathcal{I} \end{matrix} \end{bmatrix} = M_{SC}.$$

We observe that the matrix $(R_0^T A_0 R_0 + \mathbb{I} - R_0^T R_0)$ can be permuted to a matrix whose first diagonal block is A_0 followed by an identity matrix. Additionally the term $\sum_{i=1}^{N_C+1} R_i^T A_i R_i$ is a block diagonal matrix. Thus, we obtain,

$$\begin{aligned} M_{SC}^{-1} &= (R_0^T A_0 R_0 + \mathbb{I} - R_0^T R_0)^{-1} \left(\sum_{i=1}^{N_C+1} R_i^T A_i R_i \right)^{-1} \\ &= (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \left(\sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i \right). \end{aligned}$$

□

The preconditioner M_{SC}^{-1} is applied to a vector at each iteration of the residual correction scheme (13). The inverses A_i^{-1} and A_0^{-1} are never formed explicitly, they are applied to a vector by using a backward solve. We refer to this preconditioner as the SC two-level additive Schwarz in time preconditioner. It is formed by the additive Schwarz preconditioner $\sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i$, which corresponds to the use of the fine propagators computed in parallel, followed by a coarse time correction $R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0$, which corresponds to the use of the coarse propagator computed sequentially.

Corollary 1 *Solving* (6) *by using parareal is equivalent to using the residual correction scheme from* (13) *at the fine level, preconditioned by the SC two-level additive Schwarz in time preconditioner. Each iteration becomes:*

$$U_F^{k+1} = U_F^k + (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^k). \tag{22}$$

Proof The proof is done by combining Lemma 1 and 2. □

We illustrate these results by considering the simple linear time-dependent problem (4) with $m = 2$, as it can be seen in Fig. 2. After discretization, the linear system from (6) needs to be solved. We first decompose the whole time domain into non-overlapping subdomains with indices $\mathfrak{N}_i$ given by (17), with the restriction matrices $R_1 \in \mathbb{R}^{d \times (N+1)d}$, $R_i \in \mathbb{R}^{md \times (N+1)d}$, and the prolongation matrices $R_1^T \in \mathbb{R}^{(N+1)d \times d}$, $R_i^T \in \mathbb{R}^{(N+1)d \times md}$ for $i = 2, \ldots, N_C + 1$ satisfy (18) such that

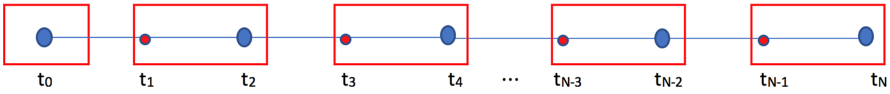

Fig. 2 Non-overlapping time subdomains with $m = 2$. The fine nodes are defined at all time points $\{t_0, t_1, t_2, \ldots, t_N\}$ and the coarse nodes are defined at even time points $\{t_0, t_2, t_4, \ldots, t_N\}$. The first time subdomain is always defined at $\{t_0\}$, while following time subdomains are defined at $\{t_n, t_{n+1}\}$ for $n = 1, \ldots, N - 1$

their entries are $\mathcal{I}$ at positions corresponding to the ith subdomain and 0 elsewhere, specifically,

$$\begin{aligned} R_1 &= \begin{bmatrix} \mathcal{I} \; 0 \; 0 \; 0 \ldots 0 \; 0 \end{bmatrix}, R_2 = \begin{bmatrix} 0 & \mathcal{I} & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \mathcal{I} & 0 & 0 & \ldots & 0 \end{bmatrix}, \ldots, \\ R_{N_C+1} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & \mathcal{I} & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & \mathcal{I} \end{bmatrix}. \end{aligned}$$

The subdomain matrices $A_i = R_i A R_i^T$, for $i = 1, \ldots, N_C + 1$, become,

$$A_i = \begin{cases} \mathcal{I}, & \text{for } i = 1, \\ \begin{bmatrix} \mathcal{I} & 0 \\ -\phi & \mathcal{I} \end{bmatrix}, & \text{for } i = 2, \ldots, N_C + 1. \end{cases}$$

Let $\mathfrak{N}_0 = \{1, 3, 5, \ldots, N + 1\}$ be the set of indices corresponding to coarse time points $\{t_0, t_2, \ldots t_N\}$ as displayed in Fig. 3. Let $A_0 \in \mathbb{R}^{(N_C+1)d\times(N_C+1)d}$ be the coarse matrix as defined in (19) and the restriction matrix $R_0 \in \mathbb{R}^{(N_C+1)d\times(N+1)d}$ satisfies (20), namely,

$$R_0 = \begin{bmatrix} \mathcal{I} & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \mathcal{I} & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \mathcal{I} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathcal{I} \end{bmatrix}.$$

The matrix M_{SC} becomes,

$$\begin{aligned} M_{SC} &= \sum_{i=1}^{N_C+1} R_i^T A_i R_i (R_0^T A_0 R_0 + \mathbb{I} - R_0^T R_0) \\ &= \begin{bmatrix} \mathcal{I} &&&&&&& \\ 0 & \mathcal{I} &&&&&& \\ & -\phi & \mathcal{I} &&&&& \\ && 0 & \mathcal{I} &&&& \\ &&& -\phi & \mathcal{I} &&& \\ &&&& \ddots & \ddots && \\ &&&&& 0 & \mathcal{I} & \\ &&&&&& -\phi & \mathcal{I} \end{bmatrix} \begin{bmatrix} \mathcal{I} &&&&&&&& \\ 0 & \mathcal{I} &&&&&&& \\ -\phi_{\Delta T} & 0 & \mathcal{I} &&&&&& \\ && 0 & \mathcal{I} &&&&& \\ && -\phi_{\Delta T} & 0 & \mathcal{I} &&&& \\ &&&& \ddots && \ddots && \\ &&&&&& 0 & 0 & \mathcal{I} \\ &&&&&& -\phi_{\Delta T} & 0 & \mathcal{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{I} &&&&&&& \\ 0 & \mathcal{I} &&&&&& \\ -\phi_{\Delta T} & -\phi & \mathcal{I} &&&&& \\ && 0 & \mathcal{I} &&&& \\ && -\phi_{\Delta T} & -\phi & \mathcal{I} &&& \\ &&&& \ddots && \ddots & \\ &&&&& 0 & 0 & \mathcal{I} \\ &&&&& -\phi_{\Delta T} & -\phi & \mathcal{I} \end{bmatrix}. \end{aligned} \tag{23}$$

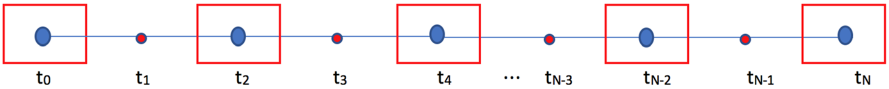

Fig. 3 Coarse time correction defined at even time points $\{t_0, t_2, t_4, \ldots, t_N\}$

It can be seen that M_{SC} from (23) is the same as the matrix M_{SC} defined in Lemma 1 in case $m = 2$, the preconditioner M_{SC}^{-1} is computed following Lemma 2,

$$M_{SC}^{-1} = (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i,$$

and then Corollary 1 gives the residual correction scheme of the problem (4) at the fine level (22) with SC two-level additive Schwarz in time preconditioner M_{SC}^{-1} which is equivalent to parareal.

It was shown in a series of papers, e.g., [13, 21], that MGRIT with F-relaxation is equivalent to parareal algorithm. We show now that MGRIT with F-relaxation is also equivalent to SC two-level additive Schwarz in time preconditioner by computing the error propagation matrix at coarse time points. The error propagation of (22) is governed by

$$e^{k+1} = (\mathbb{I} - M_{SC}^{-1} A) e^k, \tag{24}$$

where $e^k := U_F - U_F^k$, and U_F, U_F^k denote the exact solution and the approximate solution, respectively. The iteration matrix has the form,

$$\mathbb{I} - M_{SC}^{-1} A = \mathbb{I} - (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{i=1}^{N_C+1} R_i^T A_i^{-1} R_i A.$$

Note that we consider M_{SC}^{-1} as a two-level additive Schwarz preconditioner in the time domain and the matrix A is not symmetric. Hence, we cannot exploit the theory of Schwarz-type algorithms for symmetric positive definite matrices for which the preconditioned system $M_{SC}^{-1} A$ can be expressed as sums of orthogonal projection matrices P_i, for $i = 1, 2, \ldots, N_C + 1$, for further details, see [23]. Instead we study the error propagation matrix produced in the residual correction scheme (22). The following lemma shows that the error propagation matrix produces exactly the same error after one iteration at coarse time points as MGRIT with F-relaxation, for which the error is given in [25, Lemma 3.1].

Remark 2 The error propagation matrix $\mathbb{I} - M_{SC}^{-1} A$ in (24) describes the propagation of errors of (22) at both coarse and fine levels. In the following sections, e.g., Lemma 3, 4, 5, and 6, for the convenience of comparison with parareal and the variants, we only consider the error propagation matrices at the coarse level. We also remark that ϕ and $\phi_{\Delta T}$ commute due to the assumption that they can be diagonalized by the same set of eigenvectors.

Lemma 3 *Let U_F be the exact solution of* (6)*, U_F^k be an approximate solution from* (13)*, $e^k := U_F - U_F^k$ and denote by e_j^k the error at iteration k and time t_j with*

$j = 1, 2, \ldots, N$. *The error at coarse time points generated at iteration* $k+1$ *of* (13) *with SC two-level additive Schwarz in time preconditioner defined in* (21) *satisfies:*

$$\begin{aligned} e_0^{k+1} &= 0, \\ e_{hm}^{k+1} &= \sum_{r=0}^{h-1} \phi_{\Delta T}^{h-1-r}(\phi^m - \phi_{\Delta T}) e_{rm}^k, \quad h = 1, 2, \ldots, N_C. \end{aligned} \tag{25}$$

Proof We denote by $e_{C,SC}^k$ and $e_{C,SC}^{k+1}$ the errors at coarse time points at iteration k and $k+1$ respectively and by $E_{SC} := \mathbb{I} - M_{SC}^{-1} A$, the error propagation matrix at coarse time points for SC two-level additive Schwarz in time preconditioner. The error propagation from (24) at coarse time points yields,

$$e_{C,SC}^{k+1} = E_{SC} e_{C,SC}^k, \tag{26}$$

note that ϕ and $\phi_{\Delta T}$ commute, so (26) can also be written as follows,

$$\begin{bmatrix} e_0^{k+1} \\ e_m^{k+1} \\ e_{2m}^{k+1} \\ e_{3m}^{k+1} \\ \vdots \\ e_{N_C m}^{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ \phi^m - \phi_{\Delta T} & 0 & 0 & \ldots & 0 & 0 \\ \phi_{\Delta T}(\phi^m - \phi_{\Delta T}) & \phi^m - \phi_{\Delta T} & 0 & \ldots & 0 & 0 \\ \phi_{\Delta T}^2(\phi^m - \phi_{\Delta T}) & \phi_{\Delta T}(\phi^m - \phi_{\Delta T}) & \phi^m - \phi_{\Delta T} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_{\Delta T}^{N_C-1}(\phi^m - \phi_{\Delta T}) & \phi_{\Delta T}^{N_C-2}(\phi^m - \phi_{\Delta T}) & \phi_{\Delta T}^{N_C-3}(\phi^m - \phi_{\Delta T}) & \ldots & \phi^m - \phi_{\Delta T} & 0 \end{bmatrix} \begin{bmatrix} e_0^k \\ e_m^k \\ e_{2m}^k \\ e_{3m}^k \\ \vdots \\ e_{N_C m}^k \end{bmatrix}.$$

Equation (25) follows. □

4 Variants of SC two-level additive Schwarz in time preconditioner and convergence analysis

In this section, we study several variants of SC two-level additive Schwarz in time preconditioner and discuss their equivalence with MGRIT with FCF-relaxation, MGRIT with F(CF)^2-relaxation, or overlapping parareal. In addition, we derive a method, referred to as SCS^2 two-level additive Schwarz in time preconditioner, and discuss its suitability for exploiting parallel computing.

We first describe the SCS variant of SC two-level additive Schwarz in time preconditioner. It is obtained by first applying SC two-level additive Schwarz in time preconditioner, that is one fine solve followed by one coarse solve, and then adding one more fine solve. In detail, one iteration of the residual correction scheme is performed as follows:

$$\begin{aligned} U_F^{k+\frac{1}{2}} &= U_F^k + (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^k), \\ U_F^{k+1} &= U_F^{k+\frac{1}{2}} + \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^{k+\frac{1}{2}}). \end{aligned}$$

The error propagation matrix is defined as follows,

$$\left[\mathbb{I}-\sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right]\left[\mathbb{I}-(R_0^T A_0^{-1} R_0+\mathbb{I}-R_0^T R_0)\sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right].$$

The following lemma gives the error propagation of the SCS variant of SC two-level additive Schwarz in time preconditioner. It can be seen that the error propagation matrix produces exactly the same error at coarse time points after one iteration as MGRIT with FCF-relaxation. The result for MGRIT with FCF-relaxation is described in [25, Lemma 3.2].

Lemma 4 *Let U_F be the exact solution of* (4)*, U_F^k be an approximate solution from* (13)*, $e^k := U_F - U_F^k$ and denote by e_j^k the error at iteration k and time t_j with $j = 1, 2, \ldots, N$. The error at coarse time points generated at iteration $k + 1$ of the residual correction scheme from* (13) *preconditioned by SCS two-level additive Schwarz in time preconditioner satisfies:*

$$\begin{aligned} e_0^{k+1} &= 0,\\ e_m^{k+1} &= 0,\\ e_{hm}^{k+1} &= \sum_{r=0}^{h-2} \phi_{\Delta T}^{h-2-r}(\phi^m-\phi_{\Delta T})\phi^m e_{rm}^k, \quad h = 2, 3, \ldots, N_C. \end{aligned} \tag{27}$$

Proof We denote by $e_{C,SCS}^k$ and $e_{C,SCS}^{k+1}$ the errors at coarse time points at iteration k and $k+1$ respectively and by E_{SCS} the error propagation matrix at coarse time points for SCS two-level additive Schwarz in time preconditioner. We have the relation,

$$e_{C,SCS}^{k+1} = E_{SCS} e_{C,SCS}^k, \tag{28}$$

in which $E_{SCS} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 & 0\\ 0 & 0 & 0 & 0 & \ldots & 0 & 0\\ (\phi^m-\phi_{\Delta T})\phi^m & 0 & 0 & 0 & \ldots & 0 & 0\\ \phi_{\Delta T}(\phi^m-\phi_{\Delta T})\phi^m & (\phi^m-\phi_{\Delta T})\phi^m & 0 & 0 & \ldots & 0 & 0\\ \phi_{\Delta T}^2(\phi^m-\phi_{\Delta T})\phi^m & \phi_{\Delta T}(\phi^m-\phi_{\Delta T})\phi^m & (\phi^m-\phi_{\Delta T})\phi^m & 0 & \ldots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ \phi_{\Delta T}^{N_C-2}(\phi^m-\phi_{\Delta T})\phi^m & \phi_{\Delta T}^{N_C-3}(\phi^m-\phi_{\Delta T})\phi^m & \phi_{\Delta T}^{N_C-4}(\phi^m-\phi_{\Delta T})\phi^m & \ldots & (\phi^m-\phi_{\Delta T})\phi^m & 0 & 0 \end{bmatrix}.$$

The relations in (27) follow. □

The SCS^2 variant of SC two-level additive Schwarz in time preconditioner is obtained by adding one more fine solve based on additive Schwarz as follows:

$$\begin{aligned}
U_F^{k+\frac{1}{3}} &= U_F^k + (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - AU_F^k),\\
U_F^{k+\frac{1}{2}} &= U_F^{k+\frac{1}{3}} + \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - AU_F^{k+\frac{1}{3}}),\\
U_F^{k+1} &= U_F^{k+\frac{1}{2}} + \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - AU_F^{k+\frac{1}{2}}).
\end{aligned}$$

The error propagation matrix is defined as follows,

$$\left[\mathbb{I} - \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right]^2 \left[\mathbb{I} - (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right].$$

Lemma 5 *Let U_F be the exact solution of* (4)*, U_F^k be an approximate solution from* (13)*, $e^k := U_F - U_F^k$ and denote by e_j^k the error at iteration k and time t_j with $j = 1, 2, \dots, N$. The error at coarse time points generated at iteration $k+1$ of the residual correction scheme from* (13) *with SCS^2 two-level additive Schwarz in time preconditioner satisfies:*

$$\begin{aligned}
e_0^{k+1} &= 0,\\
e_m^{k+1} &= 0,\\
e_{2m}^{k+1} &= 0,\\
e_{hm}^{k+1} &= \sum_{r=0}^{h-3} \phi_{\Delta T}^{h-3-r} (\phi^m - \phi_{\Delta T}) \phi^{2m} e_{rm}^k, \quad h = 3, 4, \dots, N_C.
\end{aligned} \tag{29}$$

Proof Let e_{C,SCS^2}^k and e_{C,SCS^2}^{k+1} be the errors at coarse time points at iteration k and $k+1$ respectively and let E_{SCS^2} be the error propagation matrix at coarse time points for SCS^2 two-level additive Schwarz in time preconditioner. We have the relation,

$$e_{C,SCS^2}^{k+1} = E_{SCS^2} e_{C,SCS^2}^k, \tag{30}$$

in which $E_{SCS^2} =$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
(\phi^m - \phi_{\Delta T})\phi^{2m} & 0 & 0 & 0 & \dots & 0 & 0 & 0\\
\phi_{\Delta T}(\phi^m - \phi_{\Delta T})\phi^{2m} & (\phi^m - \phi_{\Delta T})\phi^{2m} & 0 & 0 & \dots & 0 & 0 & 0\\
\phi_{\Delta T}^2(\phi^m - \phi_{\Delta T})\phi^{2m} & \phi_{\Delta T}(\phi^m - \phi_{\Delta T})\phi^{2m} & (\phi^m - \phi_{\Delta T})\phi^{2m} & 0 & \dots & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots\\
\phi_{\Delta T}^{N_C-3}(\phi^m - \phi_{\Delta T})\phi^{2m} & \phi_{\Delta T}^{N_C-4}(\phi^m - \phi_{\Delta T})\phi^{2m} & \phi_{\Delta T}^{N_C-5}(\phi^m - \phi_{\Delta T})\phi^{2m} & \dots & (\phi^m - \phi_{\Delta T})\phi^{2m} & 0 & 0 & 0
\end{bmatrix}.$$

Equation (29) follows. □

A variant known in the literature as MGRIT with F(CF)$^\nu$-relaxation or overlapping parareal has been shown to converge at most after $k = [N/(\nu+1)]$ iterations [26, Theorem 5]. For the case $\nu = 2$, in the framework of domain decomposition, this variant is referred to as S(CS)^2 two-level additive Schwarz in time preconditioner and it is obtained as follows:

$$\begin{aligned}
U_F^{k+\frac{1}{3}} &= U_F^k + (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^k),\\
U_F^{k+\frac{1}{2}} &= U_F^{k+\frac{1}{3}} + (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^{k+\frac{1}{3}}),\\
U_F^{k+1} &= U_F^{k+\frac{1}{2}} + \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j (f - A U_F^{k+\frac{1}{2}}).
\end{aligned}$$

The error propagation matrix is defined as follows,

$$\left[\mathbb{I} - \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right]\left[\mathbb{I} - (R_0^T A_0^{-1} R_0 + \mathbb{I} - R_0^T R_0) \sum_{j=1}^{N_C+1} R_j^T A_j^{-1} R_j A\right]^2.$$

For completeness, we give in the following lemma the error of this variant.

Lemma 6 *Let U_F be the exact solution of* (4)*, U_F^k be an approximate solution from* (13)*, $e^k := U_F - U_F^k$ and denote by e_j^k the error at iteration k and time t_j with $j = 1, 2, \ldots, N$. The error at coarse time points generated at iteration $k+1$ of* (13) *with S(CS)^2 two-level additive Schwarz in time preconditioner satisfies:*

$$\begin{aligned}
e_0^{k+1} &= 0,\\
e_m^{k+1} &= 0,\\
e_{2m}^{k+1} &= 0,\\
e_{hm}^{k+1} &= \sum_{r=0}^{h-3} (h-2-r)\phi_{\Delta T}^{h-3-r} (\phi^m - \phi_{\Delta T})^2 \phi^m e_{rm}^k, \quad h = 3, 4, \ldots, N_C. \qquad (31)
\end{aligned}$$

Proof We denote by $e_{C,S(CS)^2}^k$ and $e_{C,S(CS)^2}^{k+1}$ the errors at coarse time points at iteration k and $k+1$ respectively and by $E_{S(CS)^2}$ the error propagation matrix at coarse time points for S(CS)^2 two-level additive Schwarz in time preconditioner. We obtain the relation,

$$e_{C,S(CS)^2}^{k+1} = E_{S(CS)^2} e_{C,S(CS)^2}^k, \qquad (32)$$

in which $E_{S(CS)^2} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \Phi & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 2\phi_{\Delta T}\Phi & \Phi & 0 & 0 & \dots & 0 & 0 & 0 \\ 3\phi_{\Delta T}^2\Phi & \phi_{\Delta T}\Phi & \Phi & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ (N_C-2)\phi_{\Delta T}^{N_C-3}\Phi & (N_C-3)\phi_{\Delta T}^{N_C-4}\Phi & (N_C-4)\phi_{\Delta T}^{N_C-5}\Phi & \dots & \Phi & 0 & 0 & 0 \end{bmatrix}$$

from which (31) follows, where $\Phi := (\phi^m - \phi_{\Delta T})^2\phi^m$. □

These different variants have different costs in a parallel environment. Given that the fine solve phase based on additive Schwarz is done in parallel and the coarse solve phase has to be done in sequential, the coarse solve is the major limiting factor. For that reason, the advantage becomes more noticeable when they use more fine solve phases based on additive Schwarz and one coarse solve phase which is done in sequential. The impact of additional fine or coarse solve phases in the preconditioner to the error convergence as well as the computational costs will be discussed in more detail in the next section.

5 Convergence estimate

In this section, we estimate the convergence of SC two-level additive Schwarz in time preconditioner and its variants by computing the norms of the error propagation matrices. The convergence is estimated based on an eigenvalue analysis for which the coarse and the fine propagators must have the same eigenvectors. As the assumption in Section 2 that ϕ and $\phi_{\Delta T}$ have the same eigenvectors, there exists a unitary matrix X, e.g., $X^*X = XX^* = \mathcal{I}$ such that ϕ and $\phi_{\Delta T}$ can be diagonalized as follows,

$$\begin{aligned} \Lambda &= X^*\phi X = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_d), \\ \Lambda_{\Delta T} &= X^*\phi_{\Delta T} X = \mathrm{diag}(\mu_1, \mu_2, \dots, \mu_d), \end{aligned}$$

with $|\lambda_i| < 1$ and $|\mu_i| < 1$ for $i = 1, 2, \dots, d$ since ϕ and $\phi_{\Delta T}$ are stable time-stepping methods.

The matrix E_{SC} defined in (26) is the error propagation matrix corresponding to SC two-level additive Schwarz in time preconditioner, each element of this matrix is a block matrix of dimension $d \times d$. The error propagation matrices E_{SCS}, E_{SCS^2}, $E_{S(CS)^2}$ corresponding to the variants of SC two-level additive Schwarz in time preconditioner are defined in (28), (30), and (32). Let $E_{SC} = YBY^*$, where $Y \in \mathbb{R}^{N_C d \times N_C d}$ is a block diagonal matrix, $Y = BlockDiag(X, X, \ldots, X)$ and B has the form

$$B = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ \Lambda^m - \Lambda_{\Delta T} & 0 & 0 & \ldots & 0 & 0 \\ \Lambda_{\Delta T}(\Lambda^m - \Lambda_{\Delta T}) & \Lambda^m - \Lambda_{\Delta T} & 0 & \ldots & 0 & 0 \\ \Lambda_{\Delta T}^2(\Lambda^m - \Lambda_{\Delta T}) & \Lambda_{\Delta T}(\Lambda^m - \Lambda_{\Delta T}) & \Lambda^m - \Lambda_{\Delta T} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \Lambda_{\Delta T}^{N_C-1}(\Lambda^m - \Lambda_{\Delta T}) & \Lambda_{\Delta T}^{N_C-2}(\Lambda^m - \Lambda_{\Delta T}) & \Lambda_{\Delta T}^{N_C-3}(\Lambda^m - \Lambda_{\Delta T}) & \ldots & \Lambda^m - \Lambda_{\Delta T} & 0 \end{bmatrix}.$$

We then have,

$$||B||_1 = ||B||_\infty = \max_{1 \le j \le d} \sum_{i=0}^{N_C-1} |\mu_j^i(\lambda_j^m - \mu_j)|. \tag{33}$$

On the other hand, we also have,

$$\begin{aligned} ||E_{SC}||_2 = ||YBY^*||_2 = ||B||_2 &\le \sqrt{||B||_1||B||_\infty} \\ &= \max_{1 \le j \le d} \sum_{i=0}^{N_C-1} |\mu_j^i(\lambda_j^m - \mu_j)| \\ &\le \max_{1 \le j \le d} \sum_{i=0}^{N_C-1} |\mu_j^i||\lambda_j^m - \mu_j| \\ &= \max_{1 \le j \le d} \left\{ |\lambda_j^m - \mu_j| \sum_{i=0}^{N_C-1} |\mu_j^i| \right\} \\ &= \max_{1 \le j \le d} \left\{ |\lambda_j^m - \mu_j| \frac{1 - |\mu_j|^{N_C}}{1 - |\mu_j|} \right\}. \end{aligned} \tag{34}$$

Similarly we have,

$$||E_{SCS}||_2 \le \max_{1 \le j \le d} \left\{ |\lambda_j^m - \mu_j| \frac{1 - |\mu_j|^{N_C-1}}{1 - |\mu_j|} |\lambda_j|^m \right\}, \tag{35}$$

$$||E_{SCS^2}||_2 \le \max_{1 \le j \le d} \left\{ |\lambda_j^m - \mu_j| \frac{1 - |\mu_j|^{N_C-2}}{1 - |\mu_j|} |\lambda_j|^{2m} \right\}, \tag{36}$$

$$||E_{S(CS)^2}||_2 \le \max_{1 \le j \le d} \mathcal{C}_j, \tag{37}$$

in which

$$\mathcal{C}_j = (\lambda_j^m - \mu_j)^2 \frac{1 - (N_C - 1)|\mu_j|^{N_C-2} + (N_C - 2)|\mu_j|^{N_C-1}}{(1 - |\mu_j|)^2} |\lambda_j|^m. \tag{38}$$

The 2-norm of the errors is estimated in the following theorem.

Theorem 1 *Let ϕ and $\phi_{\Delta T}$ be simultaneously diagonalizable by the same unitary matrix X and be stable time-stepping methods with eigenvalues λ_i and μ_i respectively , e.g., $|\lambda_i| < 1$ and $|\mu_i| < 1$ for $i = 1, \ldots, d$. The error at coarse time points generated at iteration $k+1$ of (13) satisfies:*

$$||e^{k+1}_{C,SC}||_2 \leq \max_{1\leq j\leq d} \left\{ |\lambda_j^m - \mu_j| \frac{1-|\mu_j|^{N_C}}{1-|\mu_j|} \right\} ||e^k_{C,SC}||_2, \tag{39}$$

$$||e^{k+1}_{C,SCS}||_2 \leq \max_{1\leq j\leq d} \left\{ |\lambda_j^m - \mu_j| \frac{1-|\mu_j|^{N_C-1}}{1-|\mu_j|} |\lambda_j|^m \right\} ||e^k_{C,SCS}||_2, \tag{40}$$

$$||e^{k+1}_{C,SCS^2}||_2 \leq \max_{1\leq j\leq d} \left\{ |\lambda_j^m - \mu_j| \frac{1-|\mu_j|^{N_C-2}}{1-|\mu_j|} |\lambda_j|^{2m} \right\} ||e^k_{C,SCS^2}||_2, \tag{41}$$

$$||e^{k+1}_{C,S(CS)^2}||_2 \leq \max_{1\leq j\leq d} \mathcal{C}_j ||e^k_{C,S(CS)^2}||_2, \tag{42}$$

where $\mathcal{C}_j$ is defined in (38).

Proof Combining (33), (34), (35), (36), and (37) leads to the desired results. □

The convergence bounds for SC from (39) and SCS from (40) are already given in the context of MGRIT with F-relaxation and with FCF-relaxation, see [25], in which the authors estimate the convergence by using the eigenvector expansion of the error to compute the error norm for each eigenmode. In this paper, we estimate the convergence of SC two-level additive Schwarz in time preconditioner and its variants by computing directly the norms of the error propagation matrices generated at iteration $k+1$ of the residual correction scheme from (13). The theoretical convergence bounds we obtained for SC and SCS two-level additive Schwarz in time preconditioner are exactly the same with those for MGRIT with F-relaxation and with FCF-relaxation. This once again confirms the equivalence between parareal, MGRIT with F-relaxation, and SC two-level additive Schwarz in time preconditioner.

As the work presented in [25], those estimates have a removable singularity that is when $|\mu_j|$ tends to 1. They are also shown to be bounded independently of N_C in many applications. Furthermore, the nominator $1-|\mu_j|^{N_C}$ can be replaced by 1 since the estimates hold for all N_C.

As mentioned in the end of the previous section, these variants have different computational costs for implementation. To make this clear, we follow our setting in Section 2 to recall the speedup of parareal algorithm from [30] as follows,

$$\mathcal{S}(N_C) = \frac{N_C m \tau_f}{N_C \tau_C + K(N_C \tau_C + m\tau_f)}, \tag{43}$$

in which the numerator describes the runtime for the fine propagator over N_C coarse time intervals while the denominator shows the runtime of parareal algorithm with N_C processors and K iterations, and τ_C, τ_f denote the computational cost of one step of the coarse and fine propagator. Depending on the number of coarse and fine propagator phases in the preconditioner, we then have different speedup of the variants, more precisely,

$$\mathcal{S}_{SCS}(N_C) = \frac{N_C m \tau_f}{N_C \tau_C + K(N_C \tau_C + 2m\tau_f)}, \tag{44}$$

$$\mathcal{S}_{SCS^2}(N_C) = \frac{N_C m \tau_f}{N_C \tau_C + K(N_C \tau_C + 3m\tau_f)}, \tag{45}$$

$$\mathcal{S}_{S(CS)^2}(N_C) = \frac{N_C m \tau_f}{N_C \tau_C + K(2N_C \tau_C + 3m\tau_f)}. \tag{46}$$

It is obvious that the speedup becomes less efficient as the number of coarse or fine propagator phases increases. However, those fine propagator phases are totally performed in parallel, this is a very important characteristic that we can exploit. By adding one or two additional fine propagation steps in the preconditioner, the convergence of parareal from (39) can be reduced by a factor of $|\lambda_j|^m$ or $|\lambda_j|^{2m}$ as it can be seen in (40) and (41), especially in the case when the eigenvalues are very small and the number of fine time step per time slice m is very large.

We provide now the estimation of the computational cost of parareal with GMRES acceleration. In this case we consider only scalar and 1D problems and we denote $\tau_C d$, $\tau_f d$ the computational cost of one step of the coarse and fine propagators to account for the linear cost in d, where d denotes the spatial dimension of ϕ. The operation count is presented in Table 1 and the total cost can be computed as,

$$\mathcal{P}_{\text{Parareal+GMRES}}(N_C, K) = N_C \tau_C d + Kd(N_C \tau_C + m\tau_f) + \mathcal{O}(K^2(N_C + md) + Kmd),$$

where we have subsumed the least squares solve cost into the $\mathcal{O}(K^2(N_C + md))$ term since $(N_C + md) \gg 1$. The normalized cost over sequential time-stepping can thus be computed as follows,

$$\mathbb{S}_{\text{Parareal+GMRES}}(N_C, K) = \frac{N_C \tau_C + K(N_C \tau_C + m\tau_f) + \mathcal{O}(K^2(N_C/d + m) + Km)}{N_C m \tau_f}, \tag{47}$$

Table 1 Operation count for parareal with GMRES acceleration

Operation	Cost
Initial coarse propagation	$N_C \tau_C d$
kth GMRES step (for $k = 1, \dots, K$)	
1. Multiplication by A	$\mathcal{O}(md)$
2. Parareal preconditioner application	$N_C \tau_C d + m\tau_f d$
3. Orthogonalization	$\mathcal{O}(k(N_C + md))$
4. Residual estimation	$\mathcal{O}(k)$
Total cost after K iterations	$K(N_C \tau_C d + m\tau_f d) + \mathcal{O}(K^2(N_C + md))$
Least squares solve	$\mathcal{O}(K^2)$
Solution reconstruction	$\mathcal{O}(Kmd)$

where we cancelled the d in both the numerator and denominator. Formula (47) will be used for the plots in Sections 6.3 and 6.4.

6 Numerical results

In this section, we first discuss results that show the equivalence between parareal and SC two-level additive Schwarz in time preconditioner for three different problems, Dahlquist problem, heat equation, and advection-reaction-diffusion equation. Numerical experiments investigate the behavior of the convergence rates on short and long time intervals when N_C and m vary. We then discuss the convergence of different variants of two-level domain decomposition preconditioners in time. A comparison between parareal or SC two-level additive Schwarz in time preconditioner and parareal with GMRES acceleration is also conducted. The three linear test cases considered here are the Dahlquist problem with $a_0 = -1, \mathrm{u}_0 = 1$,

$$\frac{du}{dt} = a_0 u, \qquad u(0) = \mathrm{u}_0, \qquad t \in [0, T], \tag{48}$$

the heat equation with $a^* = 3, L = 1, \Delta x = 0.1$, the exact solution $u_{\text{exact}} = x(L - x)^2 \exp(-2t)$,

$$\begin{cases} \frac{\partial u}{\partial t} & = a^* \frac{\partial^2 u}{\partial x^2} + f \quad in \ \ (0, L) \times (0, T), \\ u(x, 0) & = u_0(x) \qquad\quad x \in \ \ (0, L), \\ u(0, t) & = u(L, t) = 0 \ \ t \in (0, T), \end{cases} \tag{49}$$

and the advection-reaction-diffusion equation with $a = 1, b = 1, c = 1, L = 1, \Delta x = 0.1$, the exact solution $u_{\text{exact}} = \sin(2\pi x) \exp(-2t)$,

$$\begin{cases} \frac{\partial u}{\partial t} & = a \frac{\partial^2 u}{\partial x^2} - b \frac{\partial u}{\partial x} + cu + f \ \ in \ \ (0, L) \times (0, T), \\ u(x, 0) & = u_0(x) \qquad\qquad\qquad\quad x \in \ \ (0, L), \\ u(0, t) & = u(L, t) = 0 \qquad\qquad\quad t \in (0, T), \end{cases} \tag{50}$$

in which the unknowns $u(x, t)$ in (49) and (50) are considered in $(0, L) \times (0, T) \subset \mathbb{R}^d \times \mathbb{R}$, where d is the space dimension. The source term is denoted by f and is chosen to obtain the desired exact solution. For simplicity, we consider the Dirichlet boundary condition; however, the periodic boundary condition is also used in Section 6.4. Note that the same discretization methods are used for both coarse and fine solvers, namely centered finite difference in space and backward Euler in time except the end of Section 6.4 in which Runge-Kutta 4 is used for the fine solver.

6.1 Equivalence between parareal and SC two-level additive Schwarz in time preconditioner

In order to study the short time interval behavior, we use $N_C = 20, T = 1$, while for the long time interval behavior, we use $N_C = 100, T = 100$. With time steps $\Delta t = T/N_C, \delta t = \Delta t/m$, for $m = 20$, $d = 1$ for Dahlquist problem, $d = 10$ for the heat and advection-reaction-diffusion equations, the 2-norm (spectral norm) of the error between the approximate solution and the fine sequential solution (obtained by sequentially using the fine solver) is displayed in Fig. 4 for the three test cases. We

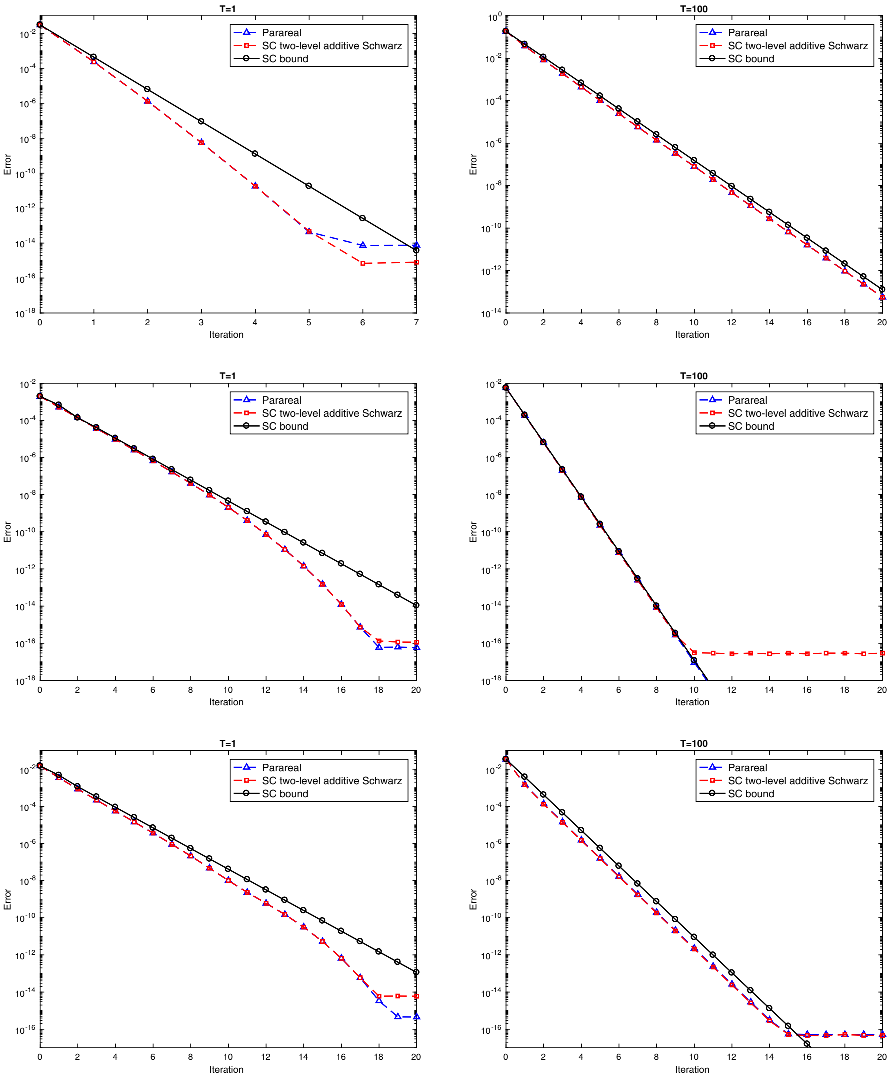

Fig. 4 Error between approximate solution and fine sequential solution with $m = 20$ for Dahlquist problem (top), heat equation (middle), and advection-reaction-diffusion equation (bottom), $T = 1$, $N_C = 20$ (first column), and $T = 100$, $N_C = 100$ (second column)

observe that the convergence curves of parareal and SC two-level additive Schwarz in time preconditioner are almost the same, except for the last iterations, when this may happen because of round-off errors. The bound for SC two-level additive Schwarz in time preconditioner derived in (39) is sharp, in particular for long time intervals.

For the Dahlquist problem, the errors of parareal and SC two-level additive Schwarz in time preconditioner are in superlinear convergence regime on short time intervals and in linear convergence regime on long time intervals. This behavior is also outlined in [13]. In particular on short time intervals, they reach 10^{-13} after 5 iterations while with the same number of iterations, the attained error is 10^{-4} on long time interval.

For the heat equation, a convergence to 10^{-16} is observed for short time interval after 18 iterations. For long time interval, both parareal and SC two-level additive Schwarz in time preconditioner converge to an error of 10^{-17} after 10 iterations.

For the advection-reaction-diffusion equation, in particular for short time interval, both parareal and SC two-level additive Schwarz in time preconditioner converge to an error of 10^{-14} after 18 iterations. For long time interval, both parareal and SC two-level additive Schwarz in time preconditioner converge to an error of 10^{-16} after 15 iterations.

In addition to this section, numerical tests are performed for the case when $m \gg N_C$, specifically, we set $N_C = 20$ and $m = 500$. The results are displayed in Fig. 5. We observe that the results are almost the same with the case when $m = 20$ in Fig. 4 for short time interval. On long time interval, a linear convergence with the same rate is obtained for both parareal and SC two-level additive Schwarz preconditioner.

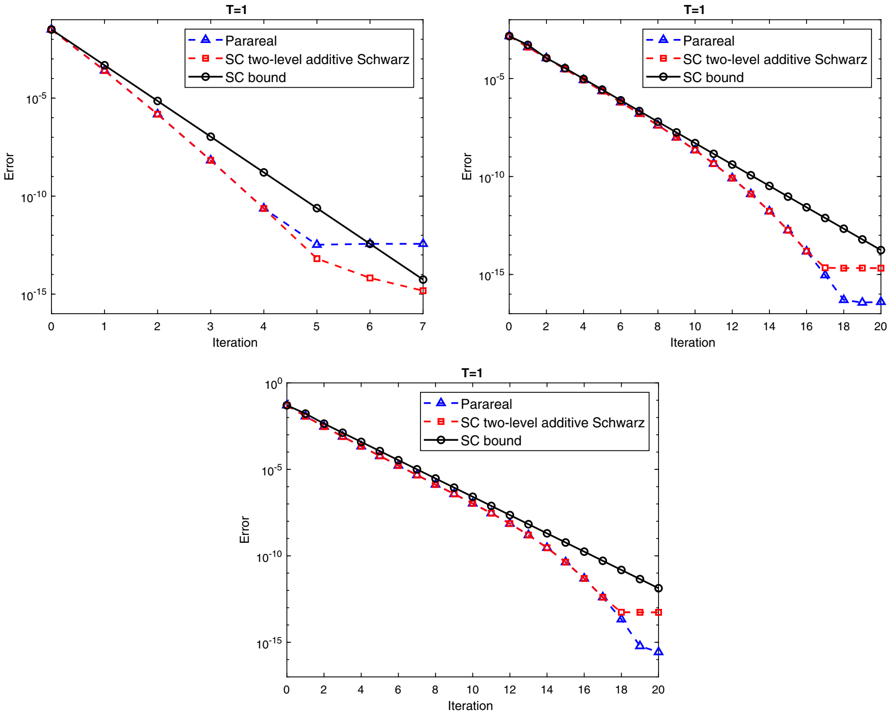

Fig. 5 Error between approximate solution and fine sequential solution with $m = 500$ for Dahlquist problem (left), heat equation (right), and advection-reaction-diffusion equation (bottom), $T = 1$, $N_C = 20$

6.2 Comparison between variants of SC two-level additive Schwarz in time preconditioner

Numerical experiments are performed to study the convergence of several variants of the SC two-level additive Schwarz in time preconditioner that use additional coarse or fine propagation steps. Similarly to the previous section, $d = 1$ for Dahlquist problem, $d = 10$ for the heat and advection-reaction-diffusion equations, the short time interval behavior uses $N_C = 20$ and $T = 1$, and the long time interval behavior uses $N_C = 100$ and $T = 100$. Figures 6, 8, and 10 display the error, in 2-norm, between the approximate solution and the fine sequential solution, with time steps $\Delta t = T/N_C$, $\delta t = \Delta t/m$, and $m \in \{2, 20\}$, for the Dahlquist problem, heat equation, and advection-reaction-diffusion equation, respectively.

For the Dahlquist problem (Fig. 6), on short time interval, the improvement of SCS, SCS^2 is not very important compared to SC two-level additive Schwarz in time preconditioner except the S(CS)^2 which converges faster than the others. However, on long time interval, the improvement becomes more important. In particular for $m = 2$, SC two-level additive Schwarz in time preconditioner converges to an error of 10^{-10} after 9 iterations, while SCS and SCS^2 converge in 7 and 5 iterations respectively, and S(CS)^2 converges in just 4 iterations to an error of 10^{-10}. For $m = 20$,

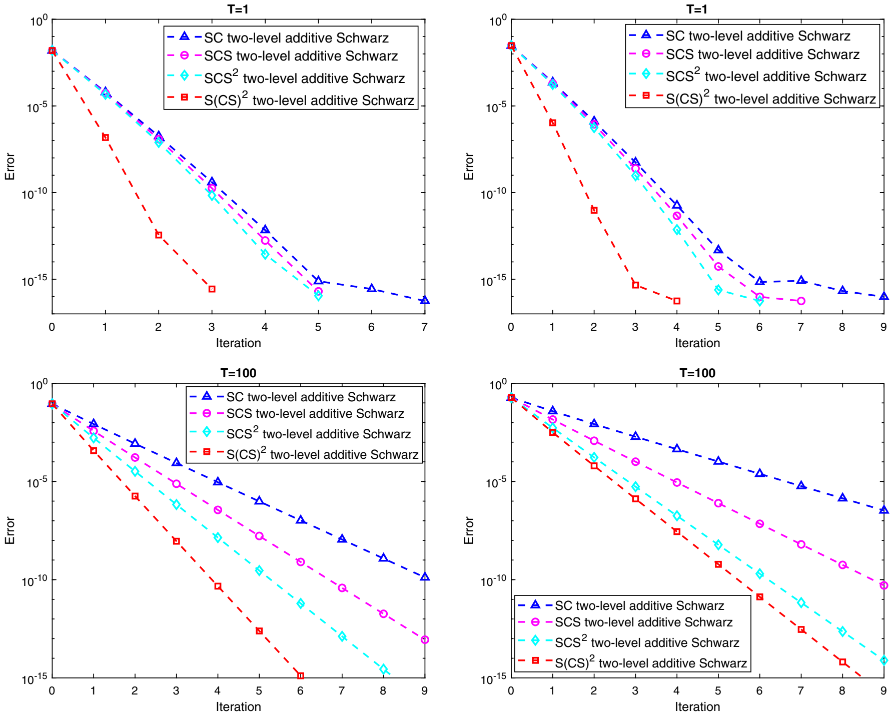

Fig. 6 Error between approximate solution and fine sequential solution for Dahlquist problem with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

after 9 iterations, SC two-level additive Schwarz in time preconditioner converges to an error of 10^{-6}, while SCS, S(CS)^2, and SCS^2 converge to much smaller errors, 10^{-10} and 10^{-14}, respectively.

Additionally, in this part, we present in Fig. 7 a comparison of the different methods in terms of their computational costs for the Dahlquist problem (when communication costs are neglected). For this purpose, the convergence is presented as a function of computational cost in which the x-axis corresponds to the computational cost normalized by the cost of sequential time-stepping, i.e., the inverse of the formulas (43), (44), (45), and (46). We choose $\tau_C = \tau_f = 8$ since the same integrator is used and we solve a tridiagonal system of dimension d with the computational cost $8d$. For short time interval, SC converges to the error of 10^{-15} with the cheapest cost compared to the others, while SCS^2 is the most expensive method. On the contrary, on long time interval $T = 100$, $N_C = 100$, SCS^2 converges with the lowest cost and SC is the most expensive method. Specifically, when $m = 20$, SCS^2 converges to the error of 10^{-14} with a cheaper cost compared to the cost of sequential time-stepping.

For the heat problem (Fig. 8), on short time interval, SCS, SCS^2, and S(CS)^2 converge faster than SC two-level additive Schwarz in time preconditioner. In particular, for $m = 2$, SC converges to an error around 10^{-16} after 13 iterations, while SCS,

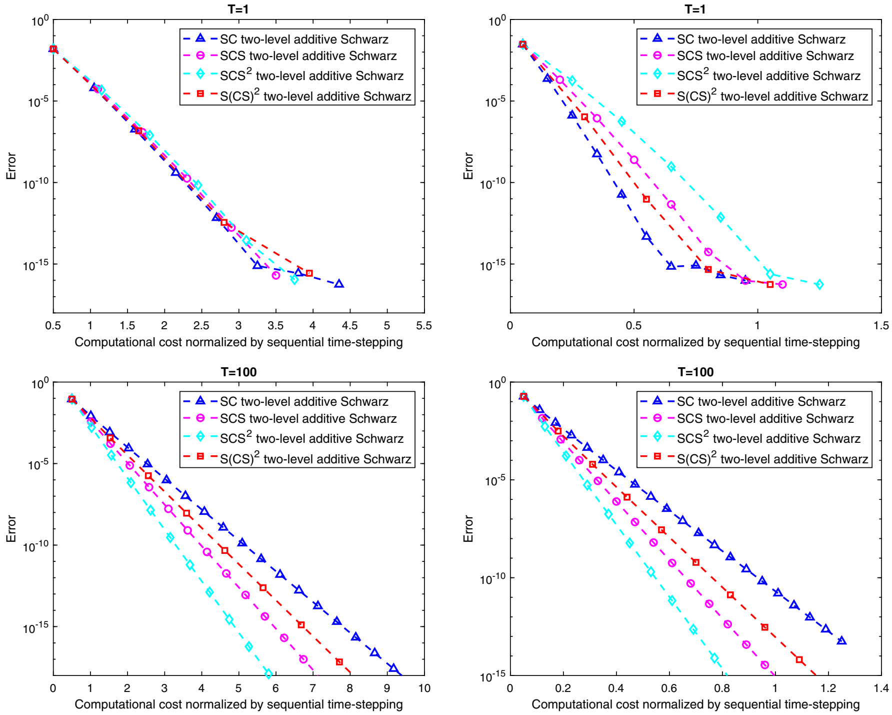

Fig. 7 Computational cost comparison of the error between approximate solution and fine sequential solution for Dahlquist problem with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

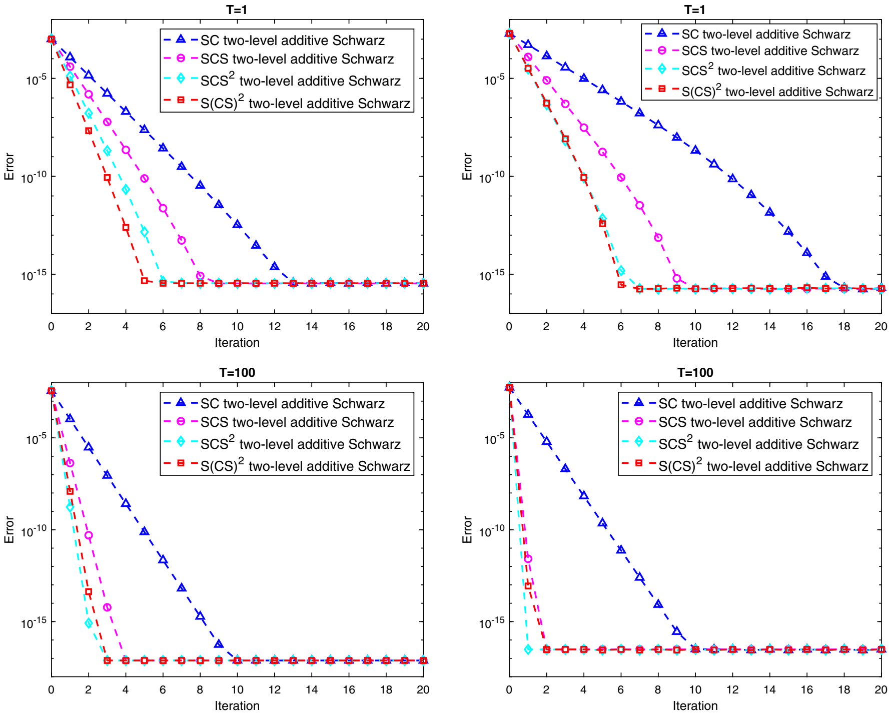

Fig. 8 Error between approximate solution and fine sequential solution for heat equation with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

SCS^2, and S(CS)^2 need 9, 6, and 5 iterations, respectively. For $m = 20$, the improvement becomes more important, specifically it takes 18 iterations for SC two-level additive Schwarz in time preconditioner to converge to an error of 10^{-16}, while SCS reaches this error in 10 iterations, and both SCS^2 and S(CS)^2 require only 7 iterations. We also observe that SCS^2 has a convergence rate close to S(CS)^2. On long time interval, both SCS and SCS^2 have a convergence rate close to the one of S(CS)^2, and SCS^2 converges faster than the other variants. In particular, for $m = 2$, SC two-level additive Schwarz in time preconditioner converges to an error of 10^{-17} after 10 iterations, while it takes 4 iterations for SCS and 3 iterations for both SCS^2 and S(CS)^2 to converge to the same error. For $m = 20$, SCS^2 converges to an error around 10^{-17} after one iteration, SCS and S(CS)^2 converge to the same error in 2 iterations, while SC two-level additive Schwarz in time preconditioner requires 10 iterations.

Similarly to the previous test case, a comparison of the different methods in terms of their computational costs for the heat problem is displayed in Fig. 9. Specifically, SCS^2 always converges with the lowest cost compared to the others. SCS and S(CS)^2 are slightly higher and SC is the most expensive method.

For the advection-reaction-diffusion problem (Fig. 10), the convergence behavior is similar to the heat equation for $m = 2$. For short time interval and $m = 20$, SC two-level additive Schwarz in time preconditioner converges to an error of 10^{-14} in

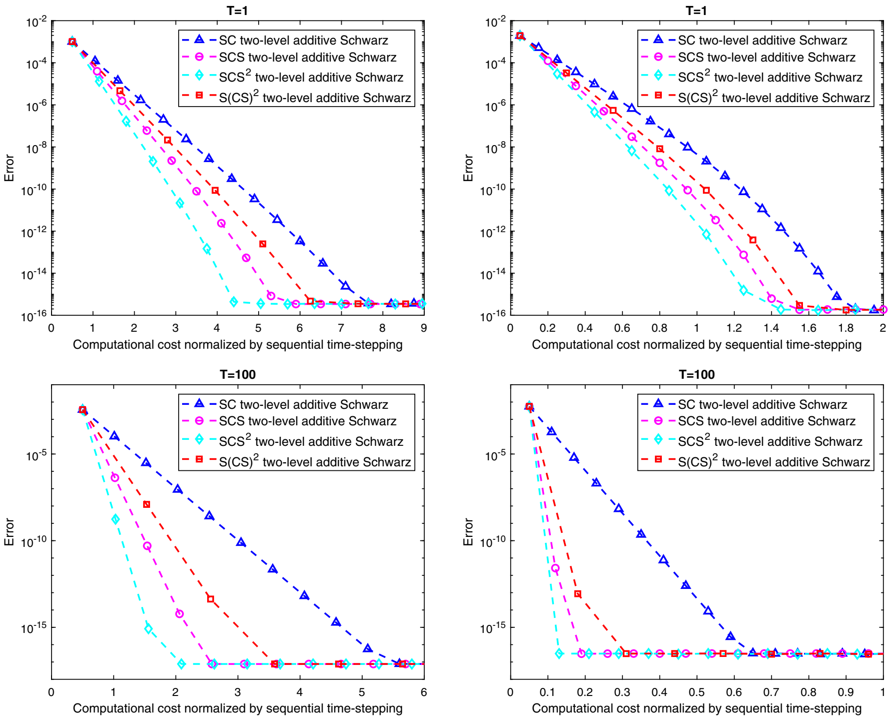

Fig. 9 Computational cost comparison of the error between approximate solution and fine sequential solution for heat equation with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

18 iterations, while SCS, SCS^2, and S(CS)^2 converge to the same error in 9, 7, and 6 iterations, respectively. On long time interval and $m = 20$, it takes 15 iterations for SC two-level additive Schwarz in time preconditioner to converge to an error of 10^{-16}, while SCS, S(CS)^2, and SCS^2 reach the same error in 4, 3, and 2 iterations, respectively.

As the previous test case, a comparison of the different methods in terms of their computational costs for the advection-reaction-diffusion problem is displayed in Fig. 11. We observe that SCS^2 always converges with the cheapest cost compared to the others, except for the short time interval with $m = 20$ in which SCS is slightly cheaper. S(CS)^2 is slightly higher compared to SCS^2 and SCS while SC is the most expensive method.

In summary, the SC two-level additive Schwarz in time preconditioner with no additional coarse or fine propagation steps has a slower convergence than the other variants for all our test cases. This indicates that the usage of additional coarse or fine propagation steps leads to a more efficient preconditioner. The S(CS)^2 variant, corresponding to overlapping parareal or MGRIT with F(CF)^2-relaxation, converges faster than the other variants in case of short time interval simulation. The SCS^2 variant converges faster than SCS for all our test cases. It is close to the convergence rate of S(CS)^2 for short time interval simulation, and it is even faster than S(CS)^2 for the

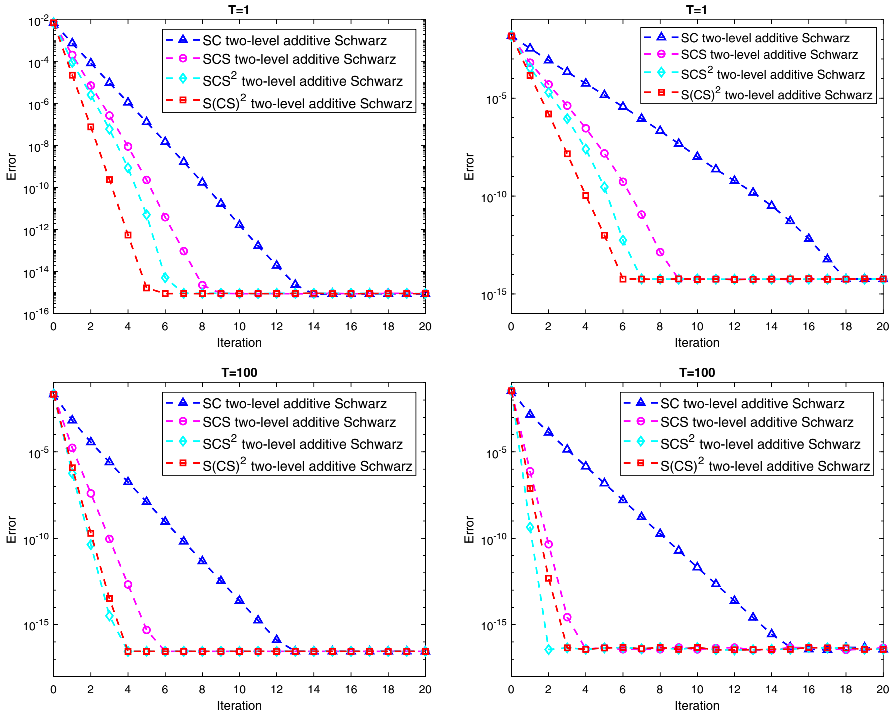

Fig. 10 Error between approximate solution and fine sequential solution for advection-reaction-diffusion equation with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

heat (49) and the advection-reaction-diffusion (50) on long time interval. It is efficient when m increases, for example, for $m = 20$, it reaches an error of 10^{-17} after one iteration. For the computational cost comparison, SCS^2 becomes the best candidate since it converges with the cheapest cost for almost cases of the three problems, especially on long time intervals.

Furthermore, in this part, we perform numerical experiments for the case when $m \gg N_C$, specifically, $N_C = 20$ and $m = 500$. The results are displayed in Fig. 12. For the Dahlquist test, S(CS)^2 reaches the error 10^{-15} after 3 iterations while SC, SCS, and SCS^2 converge to the errors of 10^{-13}, 10^{-14}, and10^{-15} after 5 iterations, respectively. For the heat problem, S(CS)^2 and SCS^2 converge nearly with the same rate to the error of 10^{-15} after 6 iterations while SC and SCS reach the same error after 17 and 9 iterations. For the advection-reaction-diffusion equation, we observe that S(CS)^2, SCS^2, SCS, and SC converge to the error of 10^{-15} after 6, 7, 9, and 18 iterations, respectively. In summary, the behavior of all methods in this case is quite similar with the case when $m = 20$ in which S(CS)^2 and SCS^2 have more advantage compared to SCS and SC.

A comparison of the different methods in terms of their computational costs for the three problems on short time interval is displayed in Fig. 13. For Dahlquist test, SC is the fastest, S(CS)^2, SCS are slightly slower and SCS^2 is the most expensive. For

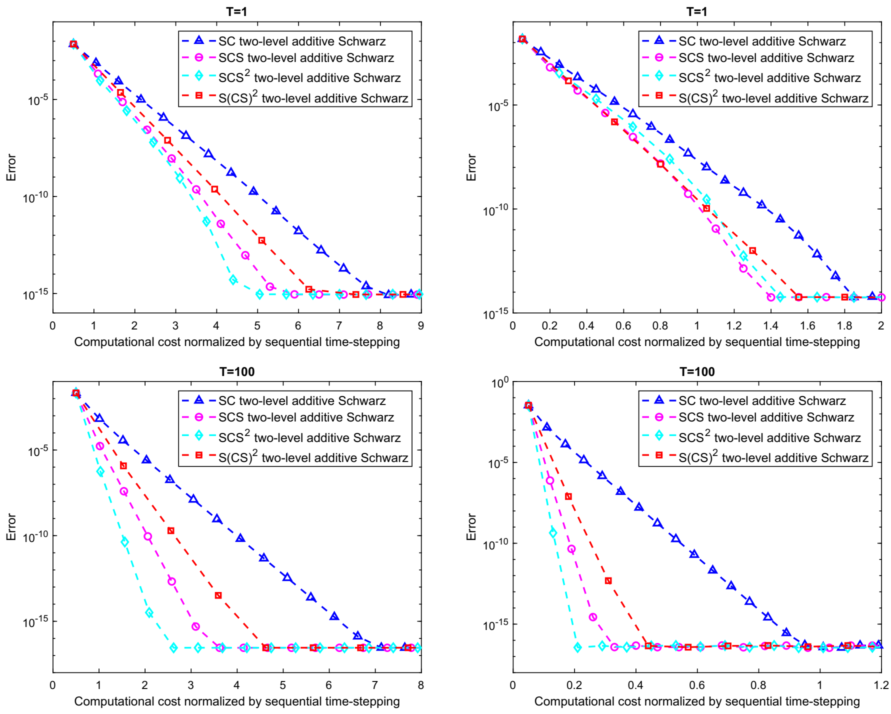

Fig. 11 Computational cost comparison of the error between approximate solution and fine sequential solution for advection-reaction-diffusion equation with $m = 2$ (first column) and $m = 20$ (second column), $T = 1$, $N_C = 20$ (first row), and $T = 100$, $N_C = 100$ (second row)

the heat and the advection-reaction-diffusion problems, the behaviors are the same and all methods converge nearly with the same computational cost, the difference is not very significant. Especially for the heat problem (Fig. 12), SCS^2 and S(CS)^2 converge after 6 iterations while SC converges after 17 iterations to the error of 10^{-15} nearly with the same computational cost. For long time interval $T = 100$, $N_C = 20$ (Fig. 14), SCS^2 converges with the lowest computational cost, SCS and S(CS)^2 are slightly slower, and SC is the most expensive for Dahlquist test. For the heat and the advection-reaction-diffusion problems, the behaviors are similar. Specifically, SCS converges with the cheapest computational cost, SCS^2 and S(CS)^2 converge with the same computational costs, and SC is the most expensive method.

6.3 Parareal with GMRES acceleration

In this section, we discuss the results obtained by parareal with GMRES acceleration. The tolerance for GMRES is set to 10^{-16}. For Dahlquist problem, the 2-norm of the error between the approximate solution and the fine sequential solution and of the relative residual are displayed in Fig. 15 for $N_C = 20, T = 1$, and for $N_C = 100, T = 100$. In both tests, $\Delta t = T/N_C$, $\delta t = \Delta t/m$, $m \in \{5, 20\}$, $d = 1$ for Dahlquist problem, $d = 10$ for the heat and advection-reaction-diffusion equations.

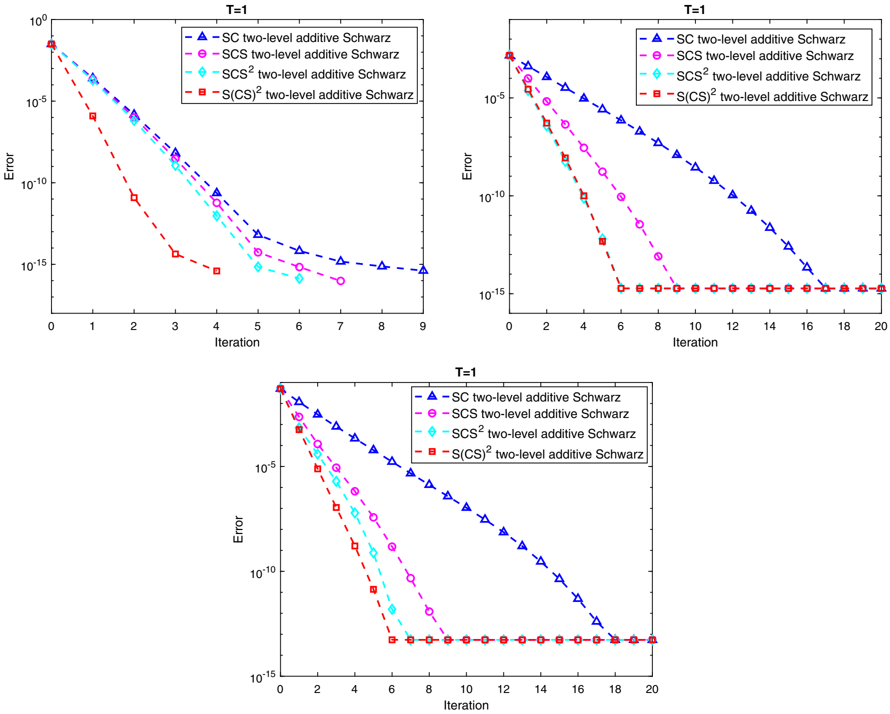

Fig. 12 Error between approximate solution and fine sequential solution with $m = 500$ for Dahlquist problem (left), heat equation (right), and advection-reaction-diffusion equation (bottom), $T = 1$, $N_C = 20$

On short time interval, we observe that GMRES slightly improves the convergence of parareal. On long time interval, the improvement becomes more noticeable. Specifically, for $m = 20$, parareal with GMRES acceleration converges to an error of 10^{-15} while parareal only converges to an error of 10^{-11}, after 16 iterations. For the relative residual, parareal with GMRES acceleration converges to 10^{-15} after 20 iterations, while parareal converges to 10^{-11} after the same numbers of iterations. Since the convergence behavior of the error and the relative residual are similar for the heat equation and the advection-reaction-diffusion equation, we present only the convergence results for the latter equation. They are displayed in Fig. 16 for short and long time intervals. On short time interval, we observe that the convergence rate of parareal with GMRES acceleration is slightly improved for both the error and the relative residual. Parareal with GMRES acceleration allows to reach the same error as parareal, while requiring 2 iterations less. For example for $m = 20$, parareal with GMRES acceleration converges to an error of 10^{-14} after 16 iterations, while parareal converges to the same error after 18 iterations. On long time interval, the improvement is even less important.

It can be seen that GMRES improves slightly the convergence of parareal for the three test cases as mentioned in the end of Section 2.2. In addition to this part, a comparison of the two methods in terms of their computational costs for Dahlquist

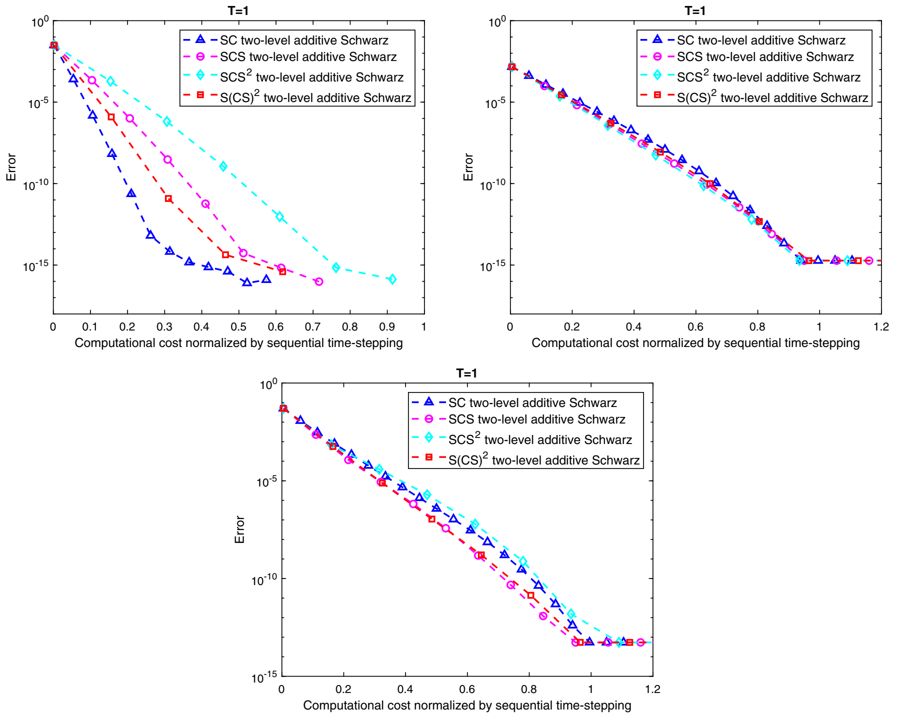

Fig. 13 Computational cost comparison of the error between approximate solution and fine sequential solution with $m = 500$ for Dahlquist problem (left), heat equation (right), and advection-reaction-diffusion equation (bottom), $T = 1$, $N_C = 20$

problem and advection-reaction-diffusion equation is displayed in Fig. 17, on short time intervals. Specifically, for Dahlquist problem, the difference is not very large at the beginning. It becomes more noticeable when the costs increase, and computational cost of plain parareal is 0.7 time less than the one of parareal with GMRES acceleration, to obtain the error of 10^{-15}. For advection-reaction-diffusion equation, the same behaviors are observed for the two curves and the computational cost of plain parareal is 0.5 time less than the one of parareal with GMRES acceleration, to obtain the error of 10^{-14}.

6.4 Impact of GMRES acceleration for the advection-reaction-diffusion equation with different coefficients

We study in this section the convergence of parareal with GMRES acceleration for the advection-reaction-diffusion equation with different coefficients than at the beginning of Section 6. We consider the following setting: $L = 1, T = 1, a = 0.01, b = 0.5, c = 100, N_C = 20, \Delta x \in \{0.2, 0.05\}, \Delta t = T/N_C, \delta t = \Delta t/m, m = 2$. The exact solution is $u_{exact} = \sin(2\pi x)\exp(-2t)$. The 2-norm of the error between the approximate solution and the fine sequential solution and of the relative residual

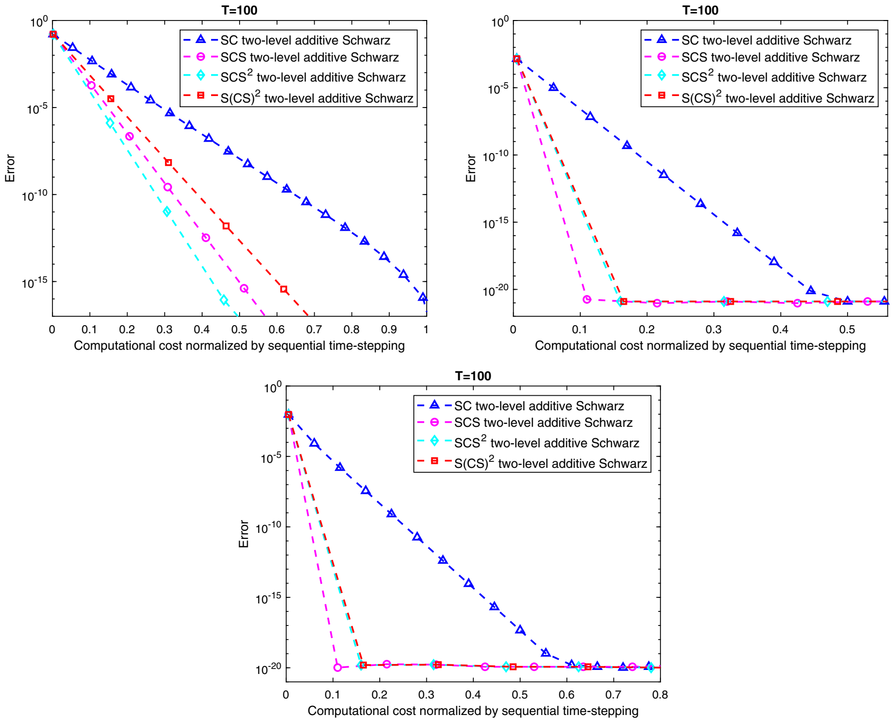

Fig. 14 Computational cost comparison of the error between approximate solution and fine sequential solution with $m = 500$ for Dahlquist problem (left), heat equation (right), and advection-reaction-diffusion equation (bottom), $T = 100$, $N_C = 20$

are displayed in Fig. 18. We observe that both parareal and parareal with GMRES acceleration converge within 20 iterations. However, the error and relative residual of parareal seem to stagnate ($\Delta x \in \{0.2, 0.05\}$) and even increase ($\Delta x = 0.05$), while those of parareal with GMRES acceleration always decrease. Specifically, for $\Delta x = 0.2$, parareal converges slowly within the first 5 iterations, then stagnates, and continues to converge after 16 iterations. Hence, GMRES acceleration provides a more robust approach on short time interval $T = 1$. However, on long time interval $T = 100$, both methods converge with the same rate. As the previous section, a comparison of the two methods in terms of their computational costs is displayed in Fig. 19. For $\Delta x = 0.2$, the computational cost of parareal with GMRES acceleration is even less than the computational cost of the plain parareal at the beginning. However, to achieve the error of 10^{-15}, the computational cost of parareal with GMRES acceleration is 1.7 times higher than the one of plain parareal. For $\Delta x = 0.05$, while parareal stagnates and even blows up at the beginning, parareal with GMRES acceleration still converges. Particularly, with almost the same computational cost, parareal with GMRES acceleration reaches an error less than 10^{-4} while the plain parareal only reaches an error of 10^{-3}. Moreover, with a 1.2 times higher computational cost, parareal with GMRES acceleration achieves an error of 10^{-15}, while the plain paraeal only reaches an error of 10^{-13}.

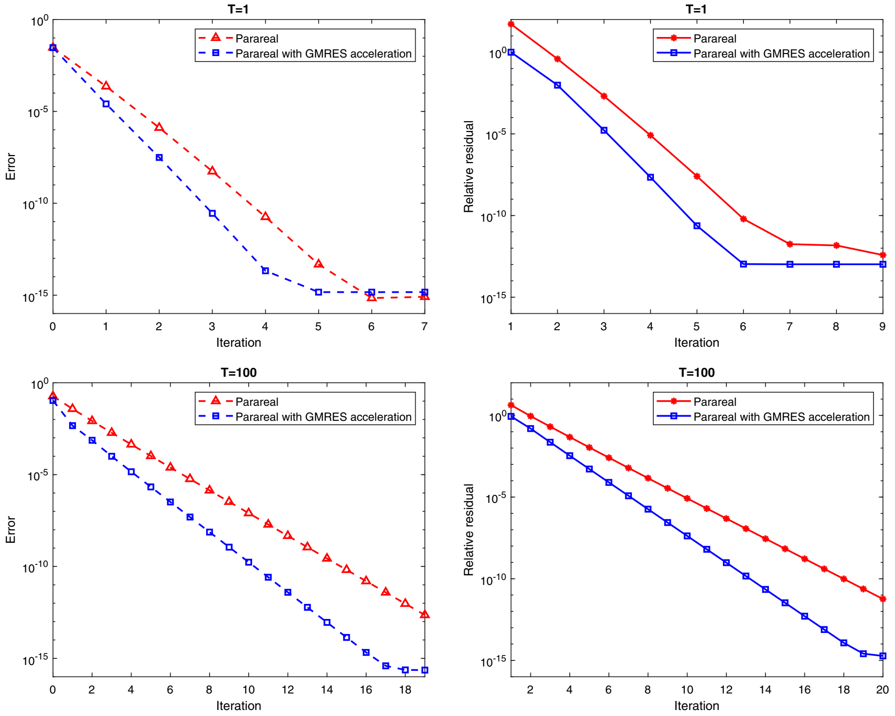

Fig. 15 Computational cost comparison of the error between approximate solution and fine sequential solution (first column) and relative residual (second column) in 2-norm for Dahlquist problem, $T = 1$, $N_C = 20$ (first row), $T = 100$, $N_C = 100$ (second row) with $m = 20$ in both cases

In this section, we also present numerical experiments for the advection-reaction-diffusion equation in two cases, diffusion dominated and advection dominated. For both cases, we consider the advection-reaction-diffusion (50) with the periodic boundary condition

$$\begin{cases} u(0,t) = u(L-\Delta x, t), \\ u(L,t) = u(\Delta x, t), \end{cases}$$

with $L = 1, T = 1, N_C = 20, \Delta x = 0.1, \Delta t = T/N_C, \delta t = \Delta t/m, m = 5$ and the exact solution $u_{exact} = \sin(2\pi(x - bt)) \exp(-2t)$. For the advective case, we consider $a = 0.0005, b = 1, c = 1$, and for the diffusive case, we consider $a = 1, b = 0.0005, c = 1$. The 2-norm of the error between the approximate solution and the fine sequential solution and of the relative residual are displayed in Fig. 20. We observe that parareal with GMRES acceleration always converges faster than the plain parareal in both cases. In particular for the advective case, parareal converges to the error of 10^{-14} after 17 iterations while parareal with GMRES acceleration reaches the same error after 15 iterations. For the diffusive case, we observe that both parareal and parareal with GMRES acceleration converge with a slower rate than the advective case. In particular, parareal with GMRES acceleration needs 16 iterations to converge to the error of 10^{-14} while parareal needs 18 iterations to reach the same error. It can be seen that GMRES again improves slightly the convergence of parareal as the

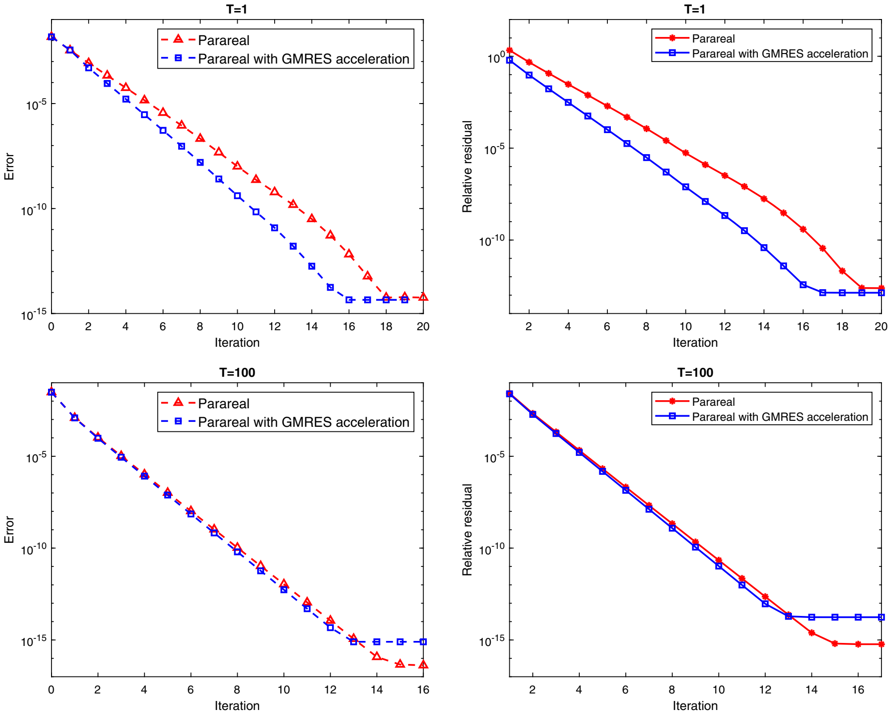

Fig. 16 Error between approximate solution and fine sequential solution (first column) and relative residual (second column) in 2-norm for advection-reaction-diffusion equation, $T = 1$, $N_C = 20$, $m = 20$ (first row), and $T = 100$, $N_C = 100$, $m = 5$ (second row)

numerical results in Section 6.3. A comparison of the two methods in terms of their computational costs is also displayed in Fig. 21. We observe that the computational costs in the diffusive case are lower than the ones in the advective case. For the

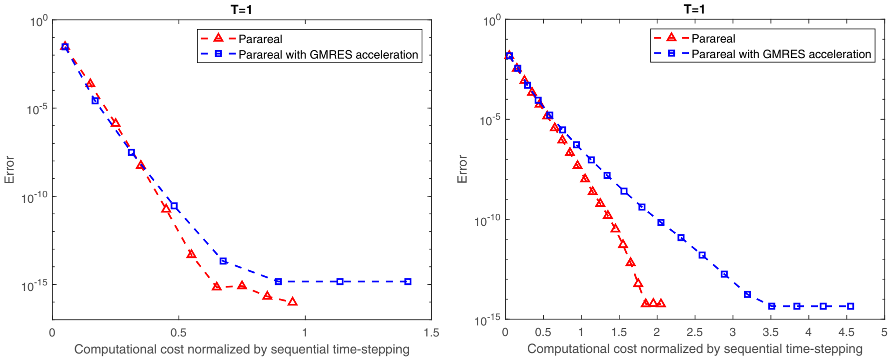

Fig. 17 Computational cost comparison of the error between approximate solution and fine sequential solution in 2-norm for Dahlquist problem (left), advection-reaction-diffusion equation (right) , $T = 1$, $N_C = 20$ with $m = 20$

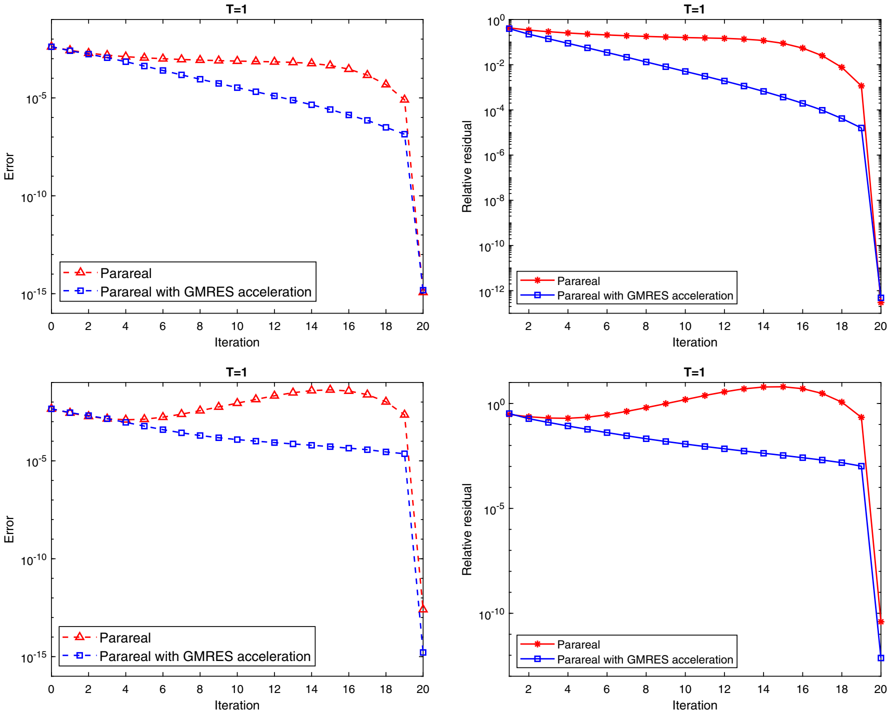

Fig. 18 Error between approximate solution and fine sequential solution (first column) and relative residual (second column) in 2-norm for advection-reaction-diffusion equation with the Dirichlet boundary condition, $T = 1$, $N_C = 20$, $m = 2$, $\Delta x = 0.2$ (first row), and $\Delta x = 0.05$ (second row), with backward Euler for both propagators

advective case, the difference is not significant at the beginning. However, to achieve the error of 10^{-14}, the computational cost of parareal with GMRES acceleration is slightly higher. For the diffusive case, the difference is slightly larger with a small

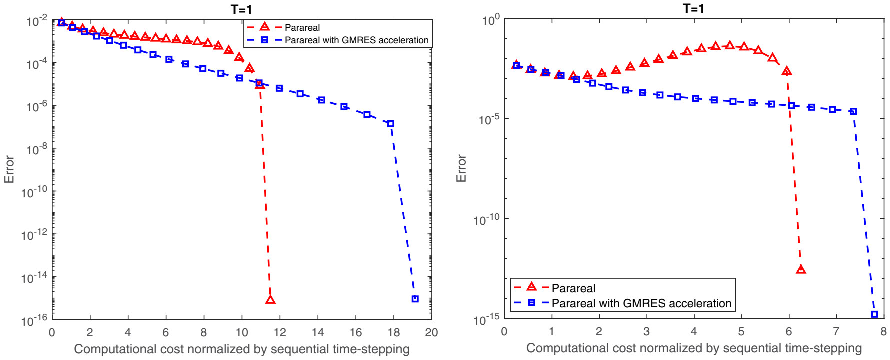

Fig. 19 Computational cost comparison of the error between approximate solution and fine sequential solution in 2-norm for advection-reaction-diffusion equation , $T = 1$, $N_C = 20$, $m = 2$, with $\Delta x = 0.2$ (left) and $\Delta x = 0.05$ (right)

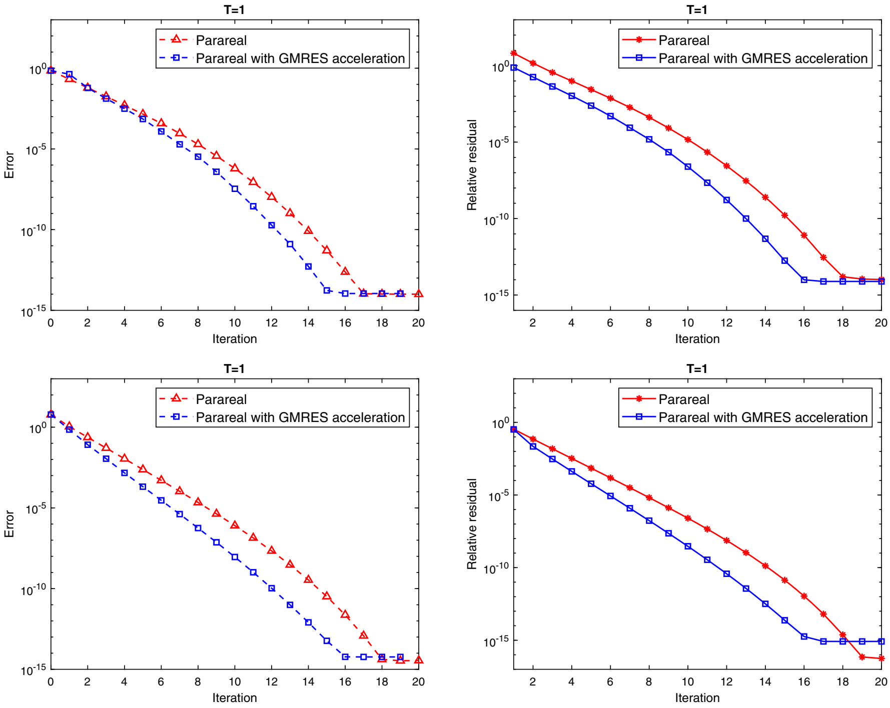

Fig. 20 Error between approximate solution and fine sequential solution (first column) and relative residual (second column) in 2-norm for advection-reaction-diffusion equation with the periodic boundary condition, $T = 1$, $N_C = 20$, $m = 5$ for advective case (first row), and for diffusive case (second row), with backward Euler for both propagators

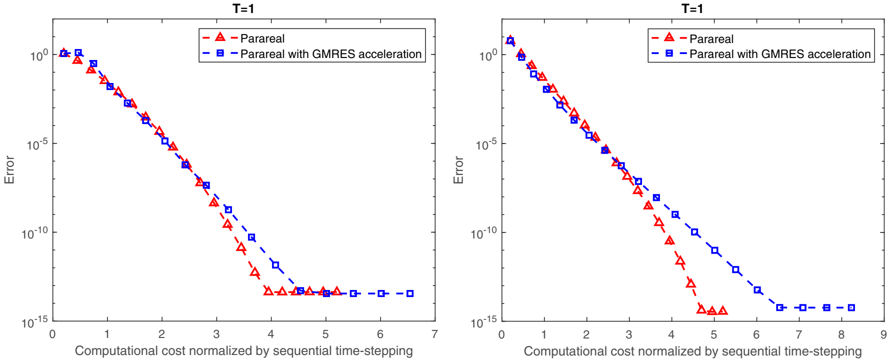

Fig. 21 Computational cost comparison of the error between approximate solution and fine sequential solution in 2-norm for advection-reaction-diffusion equation with the periodic boundary condition, $T = 1$, $N_C = 20$, $m = 5$ for advective case (left) and for diffusive case (right), with backward Euler for both propagators

advantage for parareal with GMRES acceleration at the beginning. Nevertheless, as the advective case, the computational cost of parareal with GMRES acceleration is 1.3 times higher to reach the error of 10^{-14}.

Additionally in the end of this section, we show the convergence behaviors of parareal and parareal with GMRES acceleration with a different method for the fine propagator. In particular, we keep using backward Euler in time for the coarse propagator but Runge-Kutta 4 for the fine propagator. For the discretization in space, we keep the same centered finite difference method for both coarse and fine propagators. Following the same setting with the periodic boundary condition, the convergence results are displayed in Fig. 22. Specifically, for the advective case, a convergence to the error of 10^{-13} is obtained for parareal after 15 iterations while parareal with GMRES acceleration reaches the same error after 12 iterations. For the diffusive case, a slower convergence rate than the advective case is observed for both parareal and parareal with GMRES acceleration. Specifically, it takes 15 iterations for parareal while parareal with GMRES acceleration needs 13 iterations to converge to the error of 10^{-14}. We also observe that parareal with GMRES acceleration slightly improves the convergence in both cases as the previous results in Fig. 20. Moreover, with the more accurate discretization for the fine propagator Runge-Kutta 4, the convergence

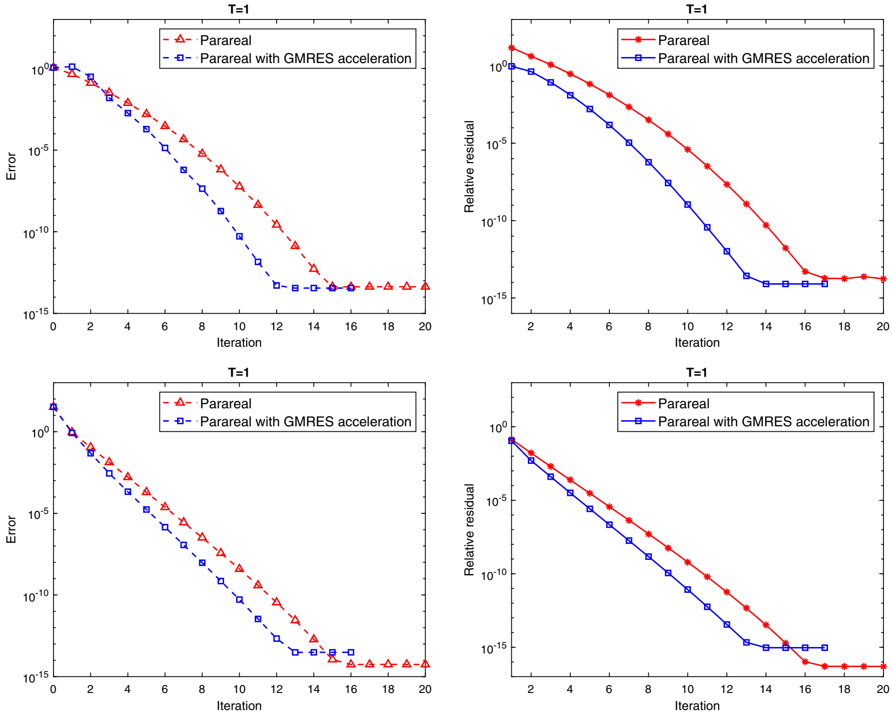

Fig. 22 Error between approximate solution and fine sequential solution (first column) and relative residual (second column) in 2-norm for advection-reaction-diffusion equation with the periodic boundary condition, $T = 1, N_C = 20, m = 5$ for advective case (first row), and for diffusive case (second row), with backward Euler for the coarse propagator and Runge-Kutta 4 for the fine propagator

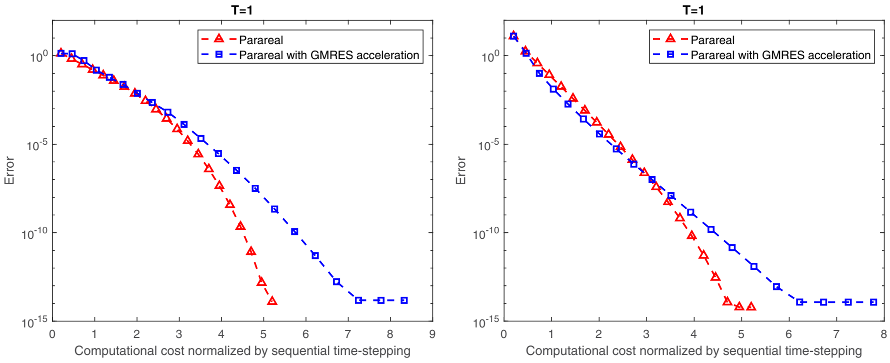

Fig. 23 Computational cost comparison of the error between approximate solution and fine sequential solution in 2-norm for advection-reaction-diffusion equation with the periodic boundary condition, $T = 1, N_C = 20, m = 5$ for advective case (left) and for diffusive case (right), with backward Euler for the coarse propagator and Runge-Kutta 4 for the fine propagator

curves are slightly faster than the ones using backward Euler in Fig. 20. Specifically, in the diffusive case, parareal with GMRES acceleration converges to the error of 10^{-14} after 13 iterations while it needs 16 iterations to reach the same error in case of using backward Euler for the fine propagator as it can be seen in Fig. 20. We also give in Fig. 23 a comparison of the two methods in terms of their computational costs. It can be seen that the computational costs of both plain parareal and parareal with GMRES acceleration are slightly higher than the ones in the previous test case in Fig. 21 for the avective case. Specifically, the difference is not very significant at the beginning, however to reach an error of 10^{-14}, the computational cost of parareal with GMRES acceleration is 1.4 times higher than the one of plain parareal. For the diffusive case, the same observation is obtained as the previous test case in figure 21.

We observe from numerical experiments in this section that GMRES acceleration is helpful at the beginning when the overhead is cheap, and becomes less helpful as the orthogonalization cost increases. One potential choice is to exploit a restart strategy after a few iterations.

7 Conclusions and perspectives

In this paper, we propose an interpretation of parareal algorithm based on a domain decomposition strategy that we refer to as SC two-level additive Schwarz in time preconditioner. This preconditioner in time is equivalent to MGRIT with F-relaxation. We study variants of this preconditioner and show that additional fine or coarse propagation steps lead to MGRIT with FCF-relaxation, MGRIT with F(CF)^2-relaxation, or overlapping parareal. We also find that SCS^2 two-level additive Schwarz in time preconditioner converges faster than MGRIT with F(CF)^2-relaxation or overlapping parareal on long time interval and with a large number of subdomains. The efficiency of the variants as well as their computational costs has been shown in numerical experiments, especially on long time intervals. Theoretical convergence bounds are verified and numerical results show that they are sharp especially for long time intervals. We also propose using Krylov subspace method, especially GMRES, to accelerate the parareal algorithm. We find that for a specific case of the advection-reaction-diffusion equation in which the advection and reaction coefficients are large compared to the diffusion term, the error of parareal stagnates or even increases for the first iterations, while GMRES provides a faster decrease of the error. This phenomena as well as the convergence analysis of parareal with GMRES acceleration will be studied in our future work.

Acknowledgements We would like to thank Martin J. Gander for his valuable discussions and the reviewers for constructive comments that helped us improve the manuscript.

Author contribution - Describe an interpretation of parareal as a two-level additive Schwarz preconditioner in the time domain so-called SC two-level additive Schwarz in time preconditioner.

- Show the equivalence between the three methods: SC two-level additive Schwarz in time preconditioner, MGRIT with F-relaxation and parareal.
- Introduce some variants as SCS, S(CS)^2 two-level additive Schwarz in time preconditioner and show their equivalence to MGRIT with FCF-relaxation, and to MGRIT with F(CF)^2-relaxation or overlapping parareal.
- Propose a variant referred to as SCS^2 two-level additive Schwarz in time preconditioner which converges faster and efficiently exploits parallel computing.
- Present the convergence analysis and convergence estimate of SC two-level additive Schwarz in time preconditioner and its variants.
- Present the computational cost analysis of parareal with GMRES acceleration.
- Conduct numerical experiments to show the equivalence between parareal and SC two-level additive Schwarz in time preconditioner, the comparison between SC two-level additive Schwarz in time preconditioner and its variants, the acceleration of parareal with GMRES and the computational cost comparison.

Funding Funding was provided by the French National Research Agency (ANR) Contract ANR-15-CE23-0019 (project CINE-PARA).

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication The authors approved it for publication.

Conflict of interest The authors declare no competing interests.

References

1. Lions, J.-L., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. Comptes Rendus de l'Académie des Sci.- Series I - Math. **332**, 661–668 (2001). https://doi.org/10.1016/S0764-4442(00)01793-6
2. Baffico, L., Bernard, S., Maday, Y., Turinici, G., Zérah, G.: Parallel-in-time molecular-dynamics simulations. Phys. Rev. E **66**, 057701 (2002). https://doi.org/10.1103/PhysRevE.66.057701
3. Eghbal, A., Gerber, A.G., Aubanel, E.: Acceleration of unsteady hydrodynamic simulations using the parareal algorithm. J. Comput. Sci. **19**, 57–76 (2016). https://doi.org/10.1016/j.jocs.2016.12.006
4. Baudron, A.M., Lautard, J.J., Maday, Y., Mula, O.: The parareal in time algorithm applied to the kinetic neutron diffusion equation. In: Domain decomposition methods in science and engineering XXI. Lecture notes in computational science and engineering, pp 437–445. Springer (2014). https://doi.org/10.1007/978-3-319-05789-7_41
5. Baudron, A.M., Lautard, J.J., Maday, Y., Riahi, M.K., Salomon, J.: Parareal in time 3D numerical solver for the LWR benchmark neutron diffusion transient model. J. Comput. Phys. **279**(0), 67–79 (2014). https://doi.org/10.1016/j.jcp.2014.08.037
6. Kaber, S.M., Maday, Y.: Parareal in time approximation of the Korteveg-deVries-Burgers' equations. PAMM **7**, 1026403–1026404 (2007). https://doi.org/10.1002/pamm.20070057
7. Dai, X., Le Bris, C., Legoll, F., Maday, Y.: Symmetric parareal algorithms for Hamiltonian systems. ESAIM: Math. Model. Numerical Anal. **47**, 717–742 (2013). https://doi.org/10.1051/m2an/2012046
8. Gander, M.J., Hairer, E.: Analysis for parareal algorithms applied to Hamiltonian differential equations. J. Comput. Appl. Math. **259 Part A**(0), 2–13 (2014). Proceedings of the Sixteenth International Congress on Computational and Applied Mathematics (ICCAM-2012), Ghent, Belgium 9-13 July, 2012, https://doi.org/10.1016/j.cam.2013.01.011
9. Bal, G., Maday, Y.: A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put. In: Pavarino, L., Toselli, A. (eds.) Recent Developments in Domain Decomposition Methods. Lecture notes in computational science and engineering, vol. 23, pp. 189–202. Springer (2002). https://doi.org/10.1007/978-3-642-56118-4_12
10. Pagès, G., Pironneau, O., Sall, G.: The parareal algorithm for american options. SIAM J. Financial Math. **9**(3), 966–993 (2018). https://doi.org/10.1137/17M1138832
11. Magoulès, F., Gbikpi-Benissan, G., Zou, Q.: Asynchronous iterations of parareal algorithm for option pricing models. Mathematics, vol. 6(4). https://doi.org/10.3390/math6040045 (2018)
12. Bal, G.: On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: Barth, T.J., Griebel, M., Keyes, D.E., Nieminen, R.M., Roose, D., Schlick, T., Kornhuber, R., Hoppe, R., Périaux, J., Pironneau, O., Widlund, O., Xu, J. (eds.) Domain Decomposition Methods in Science and Engineering, pp. 425–432. Springer (2005)
13. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007). https://doi.org/10.1137/05064607X
14. Staff, G.A., RØnquist, E.M.: Stability of the parareal algorithm. In: Barth, T.J., Griebel, M., Keyes, D.E., Nieminen, R.M., Roose, D., Schlick, T., Kornhuber, R., Hoppe, R., Périaux, J., Pironneau, O., Widlund, O., Xu, J. (eds.) Domain Decomposition Methods in Science and Engineering, pp. 449–456. Springer (2005)
15. Gander, M.J., Hairer, E.: Nonlinear convergence analysis for the parareal algorithm. In: Langer, U., Discacciati, M., Keyes, D.E., Widlund, O.B., Zulehner, W. (eds.) Domain Decomposition Methods in Science and Engineering XVII, pp 45–56. Springer (2008)
16. Minion, M.L., Williams, S.A.: Parareal and spectral deferred corrections. In: AIP conference proceedings, vol. 1048, pp. 388 (2008). https://doi.org/10.1063/1.2990941
17. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci. **5**(2), 265–301 (2010). https://doi.org/10.2140/camcos.2010.5.265
18. Gander, M.J., Jiang, Y.L., Li, R.J.: Parareal Schwarz waveform relaxation methods. In: Bank, R., Holst, M., Widlund, O., Xu, J. (eds.) Domain Decomposition Methods in Science and Engineering XX. Lecture notes in computational science and engineering, vol. 91, pp. 451–458. Springer (2013). https://doi.org/10.1007/978-3-642-35275-1_53
19. Gander, M., Jiang, Y., Song, B.: A superlinear convergence estimate for the parareal schwarz waveform relaxation algorithm. SIAM J. Sci. Comput. **41**(2), 1148–1169 (2019). https://doi.org/10.1137/18M1177226

20. Friedhoff, S., Falgout, R.D., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: A multigrid-in-time algorithm for solving evolution equations in parallel. In: Presented At: sixteenth copper mountain conference on multigrid methods, copper mountain, CO, United States, 17-22 Mar, 2013 (2013). http://www.osti.gov/scitech/servlets/purl/1073108
21. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM J. Sci. Comput. **36**, 635–661 (2014). https://doi.org/10.1137/130944230
22. Hessenthaler, A., Nordsletten, D., Rḧrle, O., Schroder, J.B., Falgout, R.D.: Convergence of the multigrid reduction in time algorithm for the linear elasticity equations. Numerical Linear Algebra Appl. **25**(3), 2155 (2018). https://doi.org/10.1002/nla.2155
23. Chan, T.F., Mathew, T.P.: Domain decomposition algorithms. Acta. Numerica. **3**, 61–143 (1994). https://doi.org/10.1017/S0962492900002427
24. Toselli, A., Widlund, O.: Domain Decomposition Methods – Algorithms and Theory. Springer Series in Computational Mathematics. **34** (2005). https://doi.org/10.1007/b137868
25. Dobrev, V., Kolev, T., Petersson, N., Schroder, J.: Two-level convergence theory for parallel time integration with multigrid. SIAM J. Scientific Comput. **39**, 501–527 (2017). https://doi.org/10.1137/16M1074096
26. Gander, M.J., Kwok, F., Zhang, H.: Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. Comput. Visualization Sci. https://doi.org/10.1007/s00791-018-0297-y (2018)
27. Ruprecht, D.: Convergence of Parareal with spatial coarsening. PAMM **14**(1), 1031–1034 (2014). https://doi.org/10.1002/pamm.201410490
28. Hirsch, C.: Numerical computation of internal and external flows: the fundamentals of computational fluid dynamics. Elsevier. https://www.bibsonomy.org/bibtex/2dbbc53f6feab1cd458e1c4f2a5c7e11c/tobydriscoll (2007)
29. Nevanlinna, O.: Linear acceleration of picard-lindelöf iteration. Numer. Math. **57**, 147–156 (1990). https://doi.org/10.1007/BF01386404
30. Minion, M.: A hybrid parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci., vol. 5. https://doi.org/10.2140/camcos.2010.5.265 (2010)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.