



Highly parallel space-time domain decomposition methods for parabolic problems

Shishun Li^{1,2} · Xinping Shao³ · Xiao-Chuan Cai⁴

Received: 12 October 2018 / Accepted: 25 February 2019
© China Computer Federation (CCF) 2019

Abstract

In the past few years, the number of processor cores of top ranked supercomputers has increased drastically. It is challenging to design efficient parallel algorithms that offer such a high degree of parallelism, especially for certain time-dependent problems because of the sequential nature of “time”. To increase the degree of parallelization, some parallel-in-time algorithms have been developed. In this paper, we give an overview of some recently introduced parallel-in-time methods, and present in detail the class of space-time Schwarz methods, including the standard and the restricted versions, for solving parabolic partial differentialequations. Some numerical experiments carried out on a parallel computer with a large number of processor cores for three-dimensional problems are given to show the parallel scalability of the methods. In the end of the paper, we provide a comparison of the parallel-in-time algorithms with a traditional algorithm that is parallelized only in space.

Keywords Parallel space-time method · Additive Schwarz method · Parabolic problem · Implicit method · Parallel scalability

Mathematics Subject Classification 65N55 · 65M55 · 65M60

The work was partly supported by the National Key R&D Program of China 2016YFB0200601, and the Shenzhen basic research grant JCYJ20160331193229720, JCYJ20170307165328836, JCYJ20170818153840322, and NSFC 11701133, 61531166003, 11726636.

✉ Xiao-Chuan Cai
cai@cs.colorado.edu

Shishun Li
lss6@sina.com

Xinping Shao
shaoxinping1983@163.com

¹ School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454003, China

² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, Guangdong, China

³ School of Science, Hangzhou Dianzi University, Hangzhou 310018, China

⁴ Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA

1 Introduction

Generally speaking, time is sequential in nature. Traditional parallel algorithms for solving time-dependent problems restrict the parallelization in the spatial dimension, and are purely sequential in the temporal dimension. In such approaches, the solution at the current time step can not be obtained without knowing the solution at previous time steps. For applications with sufficient amount of degrees of freedom in the spatial dimension, parallelization in space is often enough for the efficient use of the parallel machine. However, for applications with limited degrees of freedom in the spatial dimension and many time steps are required to resolve the dynamics of the time-dependent solution, parallelization in both space and time is a better option. The oldest time parallel algorithm was present in 1964 by Nievergelt (1964). In the recent years, parallel algorithms with space-time concurrency have received more and more attention and we briefly review some of the approaches below.

Waveform relaxation (WR) is an important time parallel algorithm which was initially presented for solving problems in large-scale circuit simulations (Lelarsmee et al. 1982). The word *waveform* denotes signals traveling along

the circuit, and *relaxation* is related to the algebraic relaxation scheme, such as Gauss–Seidel relaxation and Jacobi relaxation. The basic idea of WR is decomposing the system into decoupled subsystems by applying relaxation directly to the system of linear or nonlinear algebraic-differential equations, and the decoupled subsystems can be solved in parallel on the whole time interval. As a special iterative method, WR has been successfully applied to many problems in circuit simulations (Beyene 2008; Jiang et al. 2000; Ruehli and Johnson 1999) and to solving ordinary differential equations (ODEs) (Burrage 1995). Further, this method has been extended to solving parabolic PDEs (Bjorhus and Stuart 1997; Jeltsch and Pohl 1995), in which the PDE is first transformed into ODEs by a semi-discretization in space, then the system of ODEs is solved by WR. Actually, WR is closely related to the well-known simple Picard-Lindelöf iteration. We refer to Mikkala and Nevanlinna (1987), Nevanlinna (1990) for the convergence analysis and Belen et al. (1993) for the stability analysis. One drawback of WR is its slow convergence for some problems. By combining WR and the overlapping domain decomposition (DD) method (Smith et al. 1996; Toselli and Widlund 2005), the Schwarz WR (SWR) method was developed for linear and semilinear parabolic equations in Gander and Stuar (1998), Giladi and Keller (2002), Simoens and Vandewalle (2000), Tran (2014). It shows that the convergence of SWR is faster than the classical WR Gander (1999), Gander and Stuar (1998), Giladi and Keller (2002). By introducing optimal transmission operators, such as the Robin or Ventcell type operators, one can obtain some optimized SWR (oSWR) methods (Bennequin et al. 2009; Gander 2006; Wu and Xu 2017) that are faster than SWR, however, it is sometimes expensive to find the optimization parameters in the optimal transmission operators. Some techniques about how to choose the optimization parameters are presented in Al-Khaleel et al. (2014), Halpern et al. (2012).

The parareal algorithm was first presented in Lions et al. (2001) for solving evolution problems. The name *parareal* is derived from parallel real time computations of evolution problems. The basic idea of the algorithm is to decompose the global problem in the temporal dimension into a series of independent problems on smaller time intervals, which are then solved in parallel. The parareal algorithm can be described in terms of two computation processes, i.e., the propagator \mathcal{G} on the coarse mesh with time step size δT and the propagator \mathcal{F} on the fine mesh with time step size δt , respectively. And these two time propagators are combined by a predictor-corrector scheme. More precisely, the fine time propagator \mathcal{F} solves the independent problems on time subdomains in parallel using certain sequential-in-time methods, such as backward-Euler, Crank-Nicolson, Runge–Kutta and other high-order methods. Since $\delta t \ll \delta T$, the solution computed by propagator \mathcal{F} is more accurate, but

the cost is high. On the contrary, the coarse time propagator \mathcal{G} solves the problem by a simple, often less accurate, fast sequential time integrator. The solution computed by \mathcal{G} provides a correction for \mathcal{F} and therefore the overall algorithm converges. The parareal algorithm has received a lot of attention over the past few years (Bal 2005; Dai and Maday 2013; Gander and Vandewalle 2007; Wu and Zhou 2015), and it has been successfully applied to a variety of problems, such as parabolic problems (Gander and Vandewalle 2007; Ruprecht et al. 2016), hyperbolic problems (Dai and Maday 2013), optimal control problems (Du et al. 2013; Maday et al. 2007; Mathew et al. 2010), Navier–Stokes equations (Trindade and Pereira 2006) and non-differential equations (Bal and Maday 2002). The parallel efficiency of the parareal algorithm is far from ideal, it is proportional to $1/k$, here k denotes the number of iterations (Bal 2005; Farhat and Chandesris 2003). To improve the parallel efficiency, a hybrid algorithm has recently been proposed based on the parareal algorithm and the spectral deferred corrections algorithm. It shows that the parallel efficiency of this algorithm can be much closer to 1 (Minion 2010). Several variants of the parareal algorithm are also available, for example, PITA (parallel implicit time integrator) (Farhat and Chandesris 2003) and PFASST (parallel full approximation scheme in space and time) (Bolten et al. 2017; Minion et al. 2015).

The space-time multigrid method (Hackbusch 1984) is an interesting method that uses the elliptic multigrid method with a standard spatial smoother to solve parabolic PDEs on many time-levels simultaneously. This *parabolic multigrid method* is further studied in Horton (1992). A Fourier mode analysis shows that the success of this method requires that the meshes have a large aspect ratio τ/h^2 (τ and h denote the time step size and the mesh size, respectively) (Vandewalle and Horton 1995). Instead of the spatial smoother used in the parabolic multigrid method, the multigrid WR method with a zebra Gauss–Seidel smoother was proposed in Lubich and Ostermann (1987). This method was carefully studied for solving incompressible Navier–Stokes equations (Oosterlee and Wesseling 1993) and a chemical reaction-diffusion problem (Vandewalle and Piessens 1991). A variant of the multigrid WR algorithm with a line relaxation smoother was introduced in (Horton et al. 1995), which has a fast convergence rate but the computational complexity is increased by a logarithmic factor. By employing a parallel pointwise smoother instead of the line-wise smoother, Horton and Vandewalle (1995) presented a space-time multigrid that solves the entire space-time problem simultaneously. For some applications, the fully discrete parabolic problem may become strongly anisotropic. To improve the robustness of the method one can change the coarsening strategy, i.e., using coarsening only in the direction where point smoothing is successful. Another approach presented in Gander and Neumüller (2016) is to use a block Jacobi smoother instead

of the point Jacobi smoother. A local Fourier mode analysis shows that this new space-time multigrid method with block Jacobi smoother converges well, and numerical experiments demonstrate that it has excellent strong and weak scalability properties on large scale parallel computers.

Recently, a nonintrusive, optimal-scaling time parallel method based on multigrid reduction (MGR) was proposed Dobrev et al. (2017), Falgout et al. (2014), Falgout et al. (2017). The main advantage of this method is that the software is built on top of some existing sequential time-stepping routines, and the numerical comparison with the traditional time-stepping algorithm shows that this method is able to achieve a significant speedup. More details on numerical comparisons between the space-time multigrid with point-wise relaxation (Horton and Vandewalle 1995), space-time multigrid with block relaxation (Gander and Neumüller 2016), space-time concurrent multigrid waveform relaxation with cyclic reduction (Horton et al. 1995) and multigrid-reduction-in-time (Falgout et al. 2014) can be found in Falgout et al. (2017). Some other time parallel methods, such as the direct time parallel methods, can be found in a recent review paper (Gander 2015).

Note that all the methods mentioned above are developed by modifying some existing, standard time integration schemes (parallel in space, but sequential in time). The modifications introduce the desired extra parallelism, but also alter the discretization error and/or the stability conditions. Recently, several implicit space-time DD preconditioners were presented for parabolic PDEs (Cong et al. 2014), including some one- and two-level overlapping Schwarz preconditioned GMRES methods. Different from other space-time methods, these methods do not alter the accuracy or stability condition of the original, non-time-parallel schemes since they are *preconditioners*. The optimal convergence analysis of two-level and multilevel space-time additive Schwarz methods for solving parabolic PDEs was established in Li and Cai (2015), Li et al. (2018). The numerical experiments carried out on a parallel supercomputer show that these methods exhibit excellent strong and weak scalabilities when the number of processors is large. Some of the techniques have also been used to solve the unsteady inverse source problems (Deng et al. 2015, 2016) and flow control and heat transfer problems (Yang and Cai 2017). In Badia and Olmand (2017), a space-time BDDC (balancing DD by constraints) preconditioned GMRES method was introduced for solving a set of linear and nonlinear parabolic problems, and an excellent weak scalability result is achieved on a supercomputer with thousands of processors.

In this paper, we focus mainly on the space-time Schwarz methods including both additive and restricted additive Schwarz. We investigate the parallel performance of these methods and show their superiority compared to a traditional

time-stepping method. This rest of the paper is organized as follows. In Sect. 2, we introduce a model parabolic boundary value problem and describe the space-time Schwarz algorithms. In Sect. 3, some numerical experiments are reported to illustrate the performance of the algorithms. Finally, some conclusions are given in the last section.

2 A model problem and the space-time additive Schwarz algorithms

We consider a model parabolic equation:

$$\begin{cases} u_t - \Delta u = f, & \text{in } \Omega \times (0, T], \\ u(x, t) = 0, & \text{on } \partial\Omega \times (0, T], \\ u(x, 0) = u_0(x), & \text{in } \Omega, \end{cases} \quad (1)$$

where $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) is a bounded, open polygonal (or polyhedra) domain and $f(x, t) \in L^2(\Omega \times [0, T])$. Let $0 = t^0 < t^1 < \dots < t^n = T$ and $\tau \equiv \Delta t^k = t^k - t^{k-1}$. Suppose u^k is the solution at time t^k . Traditional time-stepping methods solve (1) time step by time step. Consider the backward Euler scheme for the time discretization, the variational form of (1) at time t^k is to find $u^k \in H_0^1(\Omega)$, $k \geq 1$, such that

$$(u^k, v) + \tau A(u^k, v) - (u^{k-1}, v) = \tau(f, v) \quad \forall v \in H_0^1(\Omega), \quad (2)$$

where the bilinear forms $A(u, v) = \int_{\Omega} \nabla u \cdot \nabla v dx$ and $(u, v) = \int_{\Omega} u \cdot v dx$.

Different from the traditional time-stepping methods, the idea of the space-time method is to group equations for s ($s \leq n$) time steps into a single system which is then solved by a Schwarz preconditioned Krylov subspace method in parallel

$$\begin{cases} (u^1, v^1) + \tau B(u^1, v^1) - (u^0, v^1) = \tau(f^1, v^1) \\ (u^2, v^2) + \tau B(u^2, v^2) - (u^1, v^2) = \tau(f^2, v^2) \\ \vdots \\ (u^s, v^s) + \tau B(u^s, v^s) - (u^{s-1}, v^s) = \tau(f^s, v^s), \end{cases} \quad (3)$$

where s is often called the window size. By using the Galerkin finite element method for the spatial discretization, we obtain a fully discretized linear system of equations:

$$\mathbf{A}u^* = \mathbf{f}, \quad (4)$$

where \mathbf{f} is the vector associated with the right-hand side of (3) and

$$\mathbf{A} = \begin{pmatrix} \tau A + M & & & & \\ -M & \tau A + M & & & \\ & & \ddots & \ddots & \\ & & & \ddots & \\ & & & & -M & \tau A + M \end{pmatrix},$$

here \mathbf{M} and \mathbf{A} denote the mass matrix associated with the bilinear form (\cdot, \cdot) and the stiffness matrix associated with the bilinear form $A(\cdot, \cdot)$, respectively.

To present the space-time Schwarz algorithms, we first introduce a space-time partition (Li et al. 2018). Let $\mathcal{T}_l = \{K_i^l\}_{i=1}^{N^l}$ ($l = 0, 1, \dots, L$) be a shape regular family of nested conforming meshes covering Ω , here L is the number of levels and N^l denotes the number of elements on level l . Let $V_l = V_{h_l}$ be the space of continuous piecewise linear functions associated with the partition \mathcal{T}_l . Set $h_l = \max_i \text{diam}(K_i^l)$. For simplicity, we set $V_H = V_0$ with mesh size $H = h_0$ and $V_h = V_L$ with mesh size $h = h_L$. Let $\{\Omega_{ij}^0\}_{i=1}^{N_i}$ be a set of non-overlapping spatial subdomains such that $\Omega = \bigcup_{i=1}^{N_i} \Omega_{ij}^0$, and $\{I_{ij}\}_{j=1}^{M_i}$ be a non-overlapping partition of $I = [0, T] = [0, t_i^{s_l}]$ such that $I = \bigcup_{j=1}^{M_i} I_{ij}$, where s_l denotes the window size of the time level l . Denote the overlapping subdomains corresponding to $\{\Omega_{ij}^0\}$ and $\{I_{ij}\}$ by $\{\Omega_{ij}^\delta\}$ (with size $H_l = O(h_{l-1})$) and $\{I_{ij}^\delta\}$ (with size $H_l^\tau = O(\tau_{l-1})$), respectively. The corresponding overlaps are denoted by $\delta_l > 0$ and $\delta_l^\tau \geq 0$. In this paper, we only consider the case $\delta_l^\tau = 0$, i.e., there is no overlap in the temporal direction. s_{ij} denotes the number of time steps in I_{ij} . For convenience, a three-level space-time mesh is presented in Fig. 1.

For each Ω_{ij}^δ , the finite element space is defined as $V_{l,\delta}^i = V_l \cap H^1(\Omega_{ij}^\delta)$. Then, we can define the space-time finite element subspaces on subsomain $\Omega_{ij}^\delta \times I_{ij}$ as

$$(V_{l,\delta}^i)^{s_{ij}} = \bigcup_{t_i^\delta \in I_{ij}} V_{l,\delta}^{ik}$$

where $V_{l,\delta}^{ik} = 0 \times \dots \times 0 \times \underbrace{V_{l,\delta}^i}_k \times 0 \times \dots \times 0$, and the 0's denote s_l

the zero spaces at the other time steps and $k = 1, 2, \dots, s_l$. Let $\mathbf{R}_0^T : (V_H)^{s_0} \rightarrow (V_h)^s$ and $(\mathbf{R}_{ij}^{l,\delta})^T : (V_{l,\delta}^i)^{s_{ij}} \rightarrow (V_l)^s$ be the interpolation operators, then the whole space-time finite

element space $(V_h)^s$ can be decomposed into the sum of the coarse space $(V_H)^{s_0}$ and the subspaces as

$$(V_h)^s = \mathbf{R}_0^T (V_H)^{s_0} + \bigcup_{l=1}^L \bigcup_{j=1}^{M_l} \bigcup_{i=1}^{N_l} (\mathbf{R}_{ij}^{l,\delta})^T (V_{l,\delta}^i)^{s_{ij}},$$

where s_0 denotes the window size of the coarsest time level.

By restricting and extending the global matrix \mathbf{A} , we obtain a coarse matrix $\mathbf{A}_0 = \mathbf{R}_0 \mathbf{A} \mathbf{R}_0^T$ and all the subspace matrices $\mathbf{A}_{ij}^{l,\delta} = \mathbf{R}_{ij}^{l,\delta} \mathbf{A} (\mathbf{R}_{ij}^{l,\delta})^T$. The l -level space-time additive Schwarz preconditioners for the matrix \mathbf{A} are defined by using the coarse and the submatrices

$$\begin{aligned} \mathbf{M}_0^{-1} &= \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \quad \text{and} \\ \mathbf{M}_{l,\delta}^{-1} &= \sum_{j=1}^{M_l} \sum_{i=1}^{N_l} (\mathbf{R}_{ij}^{l,\delta})^T (\mathbf{A}_{ij}^{l,\delta})^{-1} \mathbf{R}_{ij}^{l,\delta}, \quad l \geq 1 \end{aligned} \tag{5}$$

as $\mathbf{M}_{MAS}^{-1} = \mathbf{M}_0^{-1} + \sum_{l=1}^L \mathbf{M}_{l,\delta}^{-1}$. More precisely, the multilevel space-time additive Schwarz algorithm for solving (4) can be written as

$$\mathbf{P}_{MAS} \mathbf{u} = \mathbf{b}_{MAS}, \tag{6}$$

where $\mathbf{P}_{MAS} = \mathbf{M}_0^{-1} \mathbf{A} + \sum_{l=1}^L \mathbf{M}_{l,\delta}^{-1} \mathbf{A} = \mathbf{M}_{MAS}^{-1} \mathbf{A}$, and $\mathbf{b}_{MAS} = \mathbf{M}_0^{-1} \mathbf{A} \mathbf{u}^* + \sum_{l=1}^L \mathbf{M}_{l,\delta}^{-1} \mathbf{A} \mathbf{u}^* = \mathbf{M}_{MAS}^{-1} \mathbf{f}$.

Now we present the multilevel space-time additive Schwarz algorithm.

Algorithm 1 (Multilevel space-time additive Schwarz algorithm) Find the solution of (4) by solving (6) with GMRES, where the action of the preconditioner \mathbf{M}_{MAS}^{-1} on the global residual \mathbf{r} is carried out as follow:

```

z ← R_0^T A_0^{-1} R_0 r
For l = 1, 2, ..., L
    z ← z + sum_{j=1}^{M_l} sum_{i=1}^{N_l} (R_{ij}^{l,delta})^T (A_{ij}^{l,delta})^{-1} R_{ij}^{l,delta} r
End
    
```

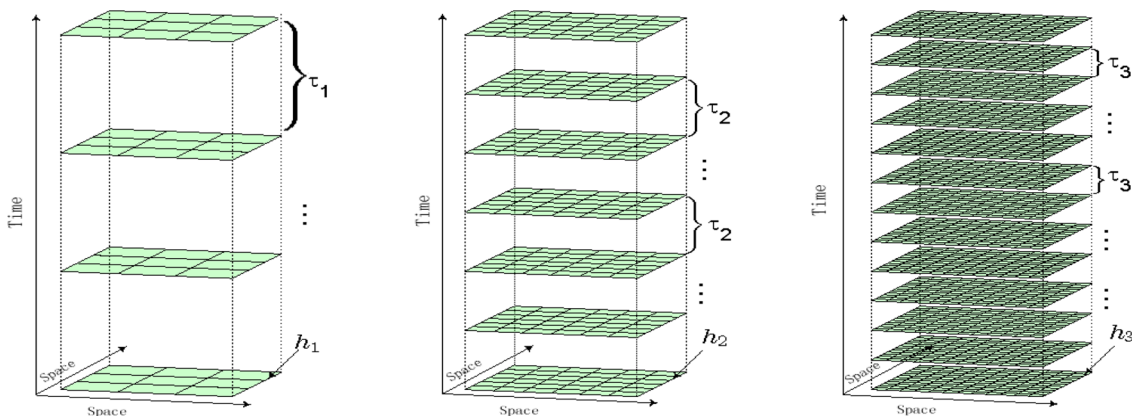


Fig. 1 A multilevel space-time mesh of $\Omega \times [0, T]$ consisting of three meshes. Space-time interpolations and restrictions are used to transfer functions from one mesh to another

The restricted additive Schwarz (RAS) preconditioning technique was first introduced for solving nonsingular linear systems (Cai and Sarkis 1999). It shows that RAS outperforms the classical AS since it requires fewer iterations, as well as less communication and CPU time (Smith et al. 1996). RAS has received great attention and is the default Schwarz preconditioner for solving linear systems in the popular PETSc software package (Balay et al. 2018). Next, we present a multilevel space-time restricted additive Schwarz algorithm.

Define the space-time finite element subspaces on subdomains $\Omega_{li}^0 \times I_{lj}$ as

$$(V_{l,0}^i)^{s_{ij}} = \bigcup_{I_{lj}^k} V_{l,0}^{ik},$$

$$\text{where } V_{l,0}^{ik} = 0 \times \cdots \times 0 \times \underbrace{V_{l,0}^i}_{s_l} \times 0 \times \cdots \times 0 \quad \text{and}$$

$V_{l,0}^i = V_l \cap H_0^1(\Omega_{li}^0)$. Let $(\mathbf{R}_{ij}^{l,0})^T : (V_{l,0}^i)^{s_{ij}} \rightarrow (V_l)^s$ be the interpolation operators. Define the l -level space-time restricted additive Schwarz preconditioners by

$$\mathbf{M}_{l,0}^{-1} = \sum_{j=1}^{M_l} \sum_{i=1}^{N_l} (\mathbf{R}_{ij}^{l,0})^T (\mathbf{A}_{ij}^{l,\delta})^{-1} \mathbf{R}_{ij}^{l,\delta}, \quad l \geq 1. \quad (7)$$

The multilevel space-time restricted additive algorithm for solving (4) can be written as

$$\mathbf{P}_{MRAS} \mathbf{u} = \mathbf{b}_{MRAS}. \quad (8)$$

Here $\mathbf{P}_{MRAS} = \mathbf{M}_0^{-1} \mathbf{A} + \sum_{l=1}^L \mathbf{M}_{l,0}^{-1} \mathbf{A} = \mathbf{M}_{MRAS}^{-1} \mathbf{A}$ and $\mathbf{b}_{MRAS} = \mathbf{M}_{MRAS}^{-1} \mathbf{f}$.

Algorithm 2 (Multilevel space-time restricted additive Schwarz algorithm) Find the solution of (4) by solving (8) with GMRES method, where the action of the preconditioner \mathbf{M}_{MRAS}^{-1} on the global residual \mathbf{r} is computed as follow:

```

 $\mathbf{z} \leftarrow \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \mathbf{r}$ 
For  $l = 1, 2, \dots, L$ 
     $\mathbf{z} \leftarrow \mathbf{z} + \sum_{j=1}^{M_l} \sum_{i=1}^{N_l} (\mathbf{R}_{ij}^{l,0})^T (\mathbf{A}_{ij}^{l,\delta})^{-1} \mathbf{R}_{ij}^{l,\delta} \mathbf{r}$ 
End
    
```

For elliptic problems, an optimal convergence theory was developed for the additive Schwarz methods (Toselli and Widlund 2005), but the theory does not apply for the restricted Schwarz methods. A convergence analysis at the algebraic level is presented in Frommer and Szyld (2001), Nabben and Szyld (2003). Recently an optimal space-time additive Schwarz theory is also developed (Li et al. 2018); under some reasonable conditions, the convergence rate of the two- and multilevel space-time additive Schwarz methods is bounded independently of the mesh sizes, the number

of subdomains and the window size. Let $\hat{\mathbf{A}}$ be the symmetric and positive part of \mathbf{A} , we define the $\hat{\mathbf{A}}$ -inner product by

$$(\mathbf{u}, \mathbf{v})_{\hat{\mathbf{A}}} = \mathbf{u}^T \hat{\mathbf{A}} \mathbf{v}.$$

The convergence rate of Algorithm 1 can be characterized by the two quantities

$$c_p = \inf_{\mathbf{u} \neq \mathbf{0}} \frac{(\mathbf{P}_{MAS} \mathbf{u}, \mathbf{u})_{\hat{\mathbf{A}}}}{(\mathbf{u}, \mathbf{u})_{\hat{\mathbf{A}}}} \quad \text{and} \quad C_p = \sup_{\mathbf{u} \neq \mathbf{0}} \frac{\|\mathbf{P}_{MAS} \mathbf{u}\|_{\hat{\mathbf{A}}}}{\|\mathbf{u}\|_{\hat{\mathbf{A}}}},$$

and the residual at the k th iteration is bounded as

$$\|\mathbf{r}_k\|_{\hat{\mathbf{A}}} \leq \left(1 - \frac{c_p^2}{C_p^2}\right)^{\frac{k}{2}} \|\mathbf{r}_0\|_{\hat{\mathbf{A}}},$$

where $\mathbf{r}_k = \mathbf{b} - \mathbf{P}_{MAS} \mathbf{u}^k$. Here $c_p = c(\max_{1 \leq l \leq L}(1 + h_{l-1}/\delta_l))^{-1}$ and $C_p = C(1 + \|\Theta\|_2)$ in Li et al. (2018), where C and c are constants independent of the mesh sizes, the number of subdomains, the window size and the number of levels. And the matrix $\Theta = \{\theta_{jq,ip}^{lk}\}_{l,k \leq L, j \leq M_l, q \leq M_k, i \leq N_l, p \leq N_k}$, $\theta_{jq,ip}^{lk}$ is the cosine of the angle between the subspaces $(V_l^i)^{s_{ij}}$ and $(V_k^p)^{s_{kq}}$, i.e., $\theta_{jq,ip}^{lk} = \cos((V_l^i)^{s_{ij}}, (V_k^p)^{s_{kq}})$.

3 Numerical results

In this section, we present some numerical results to illustrate the parallel performance of the space-time Schwarz algorithms. We implement these algorithms on the top of an open source package PETSc (Balay et al. 2018). All experiments are carried out on the Tianhe-2 supercomputer located at the National Supercomputer Center in Guangzhou. Only CPUs are used in the tests.

In the numerical experiment, we divide the space-time domain $[0, 1]^3 \times [0, 32]$ into $N_p = P_x \times P_y \times P_z \times P_t$ subdomains and assign each subdomain to one processor. All the subdomain problems are solved using ILU, and the overlap is set as 1. The problem is solved by a preconditioned GMRES(30) method with the stopping condition

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-6}.$$

Example 1 Consider a three dimensional convection diffusion equation

$$\begin{cases} u_t - \Delta u + \mathbf{b} \cdot \nabla u = f, & \text{in } \Omega \times (0, T], \\ u(x, y, z, 0) = \frac{1}{0.8} e^{-(x^2+y^2+z^2)/(0.8)}, & \text{in } \Omega, \end{cases}$$

Table 1 The averaged number of iterations and the total compute time for solving Example 1 on a $65 \times 65 \times 65 \times 2048$ space-time mesh using Algorithm 1

N_p		8	16	32	64	128	256	512	1024	2048	4096	8192
$s = 1$	GMRES	4.3	–	–	4.4	–	–	4.5	–	–	4.7	4.9
	Time(s)	293.3	–	–	58.1	–	–	42.0	–	–	64.0	67.5
$s = 2$	GMRES	4.5	–	–	4.7	–	–	4.8	–	–	5.0	5.2
	Time(s)	321.2	–	–	55.9	–	–	28.1	–	–	43.1	48.3
$s = 4$	GMRES	4.7	–	–	5.0	–	–	5.1	–	–	5.2	5.4
	Time(s)	348.6	–	–	56.5	–	–	22.7	–	–	28.7	35.8
$s = 32$	GMRES	*	8.0	8.0	8.0	8.0	8.5	8.5	8.5	9.7	9.5	9.5
	Time(s)	*	307.6	155.2	79.8	44.3	30.6	22.6	20.7	17.9	15.2	16.0
$s = 256$	GMRES	*	*	9.0	9.0	9.0	9.0	9.0	9.0	9.5	9.5	9.5
	Time(s)	*	*	324.4	163.4	86.2	46.5	30.0	21.0	15.4	10.4	11.8
$s = 512$	GMRES	*	*	*	9.5	9.5	9.5	9.5	9.5	9.5	10.0	10.0
	Time(s)	*	*	*	255.3	134.1	72.4	44.1	28.0	15.7	13.5	13.5
$s = 1024$	GMRES	*	*	*	*	10.5	10.5	10.5	10.5	10.5	10.5	11.0
	Time(s)	*	*	*	*	233.1	122.7	72.7	44.6	24.0	14.9	17.4
$s = 2048$	GMRES	*	*	*	*	*	12.0	12.0	12.0	12.0	12.0	12.0
	Time(s)	*	*	*	*	*	216.0	118.6	67.8	39.2	21.5	13.9

(corresponding to rows with $s > 1$) and the traditional time stepping method parallelized only in space with a two-level additive Schwarz method with $h_0 = \tau_0 = 1/32$ and $h_1 = \tau_1 = 1/64$ (corresponding to the row with $s = 1$)

Table 2 The averaged number of iterations and the total compute time for solving Example 1 on a $65 \times 65 \times 65 \times 2048$ space-time mesh using Algorithm 2

N_p		8	16	32	64	128	256	512	1024	2048	4096	8192
$s = 1$	GMRES	4.3	–	–	4.4	–	–	4.5	–	–	4.7	4.9
	Time(s)	292.9	–	–	59.3	–	–	41.6	–	–	62.2	66.6
$s = 2$	GMRES	4.5	–	–	4.7	–	–	4.8	–	–	5.0	5.2
	Time(s)	324.2	–	–	55.3	–	–	28.0	–	–	41.5	48.0
$s = 4$	GMRES	4.7	–	–	5.0	–	–	5.1	–	–	5.2	5.4
	Time(s)	348.5	–	–	56.0	–	–	22.9	–	–	28.7	35.3
$s = 32$	GMRES	*	8.0	8.0	8.0	8.0	8.5	8.5	8.5	9.7	9.5	9.5
	Time(s)	*	307.1	154.8	79.7	44.3	30.4	22.6	20.1	17.8	15.2	15.9
$s = 256$	GMRES	*	*	9.0	9.0	9.0	9.0	9.0	9.0	9.5	9.5	9.5
	Time(s)	*	*	322.9	162.6	86.2	46.3	29.9	20.8	15.6	10.6	11.6
$s = 512$	GMRES	*	*	*	9.5	9.5	9.5	9.5	9.5	9.5	10.0	10.0
	Time(s)	*	*	*	256.6	134.4	71.5	44.0	28.6	15.4	13.5	12.9
$s = 1024$	GMRES	*	*	*	*	10.5	10.5	10.5	10.5	10.5	10.5	11.0
	Time(s)	*	*	*	*	233.0	122.3	72.7	44.9	23.8	14.5	17.2
$s = 2048$	GMRES	*	*	*	*	*	12.0	12.0	12.0	12.0	12.0	12.0
	Time(s)	*	*	*	*	*	215.1	118.5	68.1	39.3	21.6	12.6

(corresponding to rows with $s > 1$) and the traditional time stepping method parallelized only in space with a two-level additive Schwarz method with $h_0 = \tau_0 = 1/32$ and $h_1 = \tau_1 = 1/64$ (corresponding to the row with $s = 1$)

where $\mathbf{b} = (1, 1, 1)$, $f = \frac{1-x-y-z}{8(t+0.2)^2} e^{-(x^2+y^2+z^2)/(4(t+0.2))}$ and the analytical solution is

$$u(x, y, z, t) = \frac{1}{4(t+0.2)} e^{-(x^2+y^2+z^2)/(4(t+0.2))}$$

with consistent Dirichlet boundary conditions.

We test this example to illustrate the strong scalability of the space-time methods with different s on a $65 \times 65 \times 65 \times 2048$ space-time mesh. The mesh size and the time step size on level l ($l = 0, 1$) are set as $h_0 = \tau_0 = 1/32$ and $h_1 = \tau_1 = 1/64$, respectively. To obtain better parallel performance, the spatial mesh is decomposed evenly such that each processor holds approximately the same number

of mesh points. And we omit some cases marked by “-” in Tables 1 and 2. Moreover, “*” indicates that the number of processors is too small to solve the problem.

The numerical results of Algorithm 1 are presented in Table 1 and Fig. 2. Note that the sequential time stepping algorithm is the special case of Algorithm 1 with $s = 1$. Table 1 shows that the average number of iterations is bounded independent of the number of processors for each window size s . For the case $s = 1$, the total compute time decreases quickly at the beginning and then increases a little with the number of processors. It means that the computation/communication ratio decreases as we increase the number of processors. The compute time curve on the left of Fig. 2 shows that the strong scalability of the classical time stepping algorithm is not good when the number of

processors is large. For the cases $s = 2, 4, 32, 256, 512, 1024$, the compute time curves behave similar to the case of $s = 1$, but they perform much better than the classical time stepping algorithm as we increase the number of processors. It is clear that the compute time curve is linear for the case $s = 2048$, i.e., we obtain linear speedup when solving all time steps of the problem at once. The crossover points (intersections with the curve corresponding to $s = 1$) in Fig. 2 demonstrate the benefit of the space-time algorithm compared to the time stepping algorithm. For examples, the crossover point is at about 64 processors for the cases $s = 2, 4$, and it is at larger number of processors with larger s . When the problem size is fixed, we observe that the time stepping algorithm is faster when the number of processors is small, but the space-time Schwarz algorithm performs much better once the number

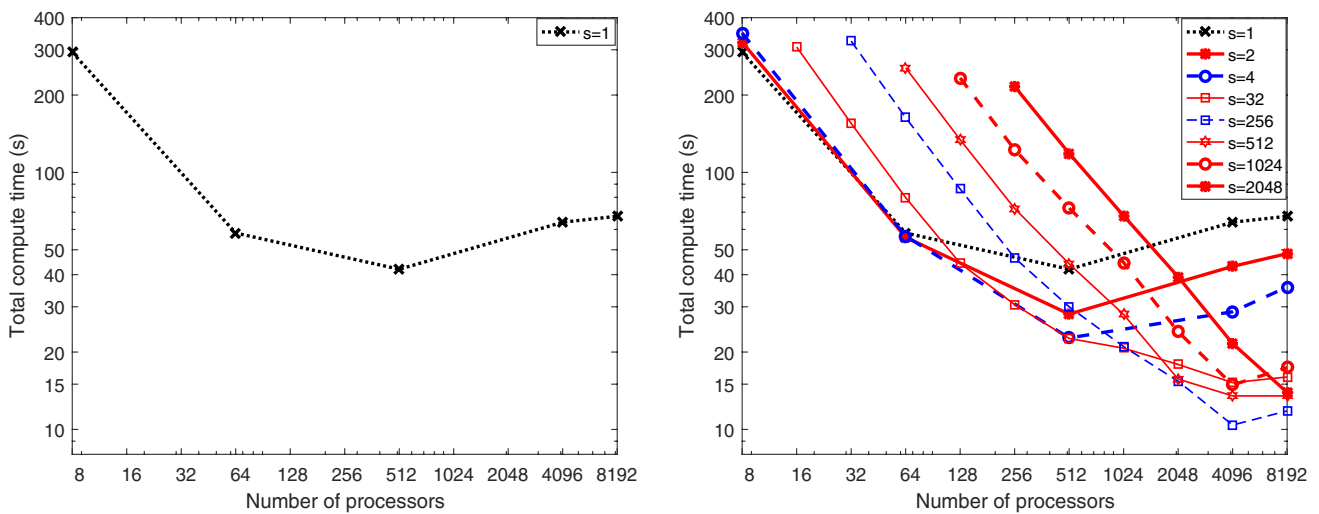


Fig. 2 The total compute time for solving Example 1 using sequential time-stepping (left) and Algorithm 1 with different window size s (right)

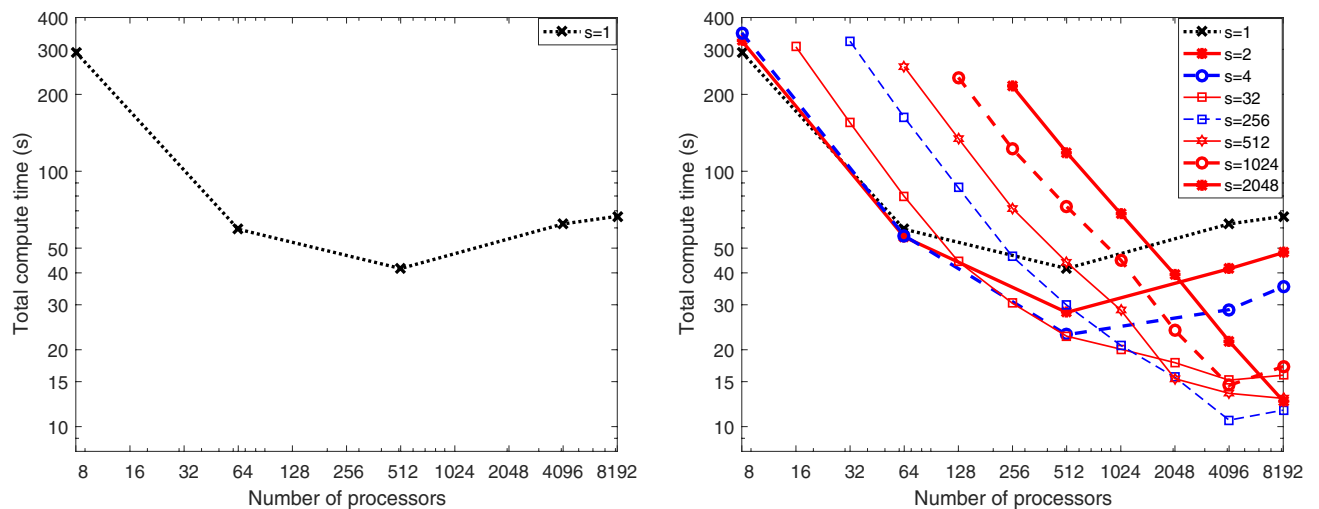


Fig. 3 The total compute time for solving Example 1 using sequential time-stepping (left) and Algorithm 2 with different window size s (right)

of processors goes beyond a certain number. We next study Algorithm 2. The numerical results in Table 2 and Fig. 3 show that the average number of iterations is almost the same as that of Algorithm 1, and the compute time is a little less than that required by Algorithm 1 for each s.

4 Concluding remarks

To exploit the full power of supercomputers, a new generation of methods with space-time concurrency are developed to solve time-dependent problems. In this paper, we gave an overview of several important time parallel methods including waveform relaxation, parareal and space-time multigrid. And we also introduced a class of implicit space-time domain decomposition methods for solving parabolic PDEs. More precisely, we extend the classical Schwarz methods for elliptic problems as space-time preconditioners applied to solving parabolic problems, and describe the optimal convergence theory for the space-time additive Schwarz algorithms by using the matrix-based formulation. The theory for the restricted version of the method is not available. The numerical experiments for three dimensional problems show both the additive and the restricted additive Schwarz methods have excellent parallel scalability and both outperform the classical time stepping method when the number of processors is large, but they require more memory as expected. Future work will focus on extending the methods for more realistic applications.

References

- Al-Khaleel, M.D., Gander, M.J., Ruehli, A.E.: Optimization of transmission conditions in waveform relaxation techniques for RC circuits. *SIAM J. Numer. Anal.* **52**, 1076–1101 (2014)
- Badia, S., Olmand, M.: Space-time balancing domain decomposition. *SIAM J. Sci. Comput.* **39**, C194–C213 (2017)
- Bal, G.: On the convergence and the stability of the parareal algorithm to solve partial differential equations, in *Domain Decomposition Methods in Science and Engineering*. In: R. Kornhuber, R. H. W. Hoppe, D. E. Keyes, J. Periaux, O. Pironneau, and J. Xu, (eds.), vol. 40, SIAM, Springer, Berlin, 425–432 (2005)
- Bal, G., Maday, Y.: A “parareal” time discretization for non-linear PDE’s with application to the pricing of an American put. In: *Recent developments in domain decomposition methods* (Zürich, 2001), 189–202, *Lect. Notes Comput. Sci. Eng.*, 23, Springer, Berlin (2002)
- Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Kaushik, D., Knepley, M. G., McInnes, L. C., Gropp, W. D., Rupp, K., Smith, B. F., Zampini, S., Zhang, H.: *PETSc Users Manual*, Technical report ANL 95/11-Revision 3.7. Argonne National Laboratory, Argonne, IL, (2018)
- Bellen, A., Jackiewicz, Z., Zennaro, M.: Time-point relaxation Runge–Kutta methods for ordinary differential equations. *J. Comput. Appl. Math.* **45**, 121–137 (1993)
- Bennequin, D., Gander, M.J., Halpern, L.: A homographic best approximation problem with application to optimized Schwarz waveform relaxation. *Math. Comp.* **78**, 185–223 (2009)
- Beyene, W.T.: Application of multilinear and waveform relaxation methods for efficient simulation of interconnect-dominated nonlinear networks. *IEEE Trans. Adv. Packag.* **31**, 637–648 (2008)
- Bjorhus, M., Stuart, A.M.: Waveform relaxation as a dynamical system. *Math. Comp.* **66**, 1101–1117 (1997)
- Bolten, M., Moser, D., Speck, R.: A multigrid perspective on the parallel full approximation scheme in space and time. *Numer. Linear Algebra Appl.* **24**, e2110 (2017)
- Burrage, K.: *Parallel and sequential methods for ordinary differential equations*. Oxford Science Publications, New York (1995)
- Cai, X.-C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.* **21**, 792–797 (1999)
- Cong, C., Cai, X.-C., Gustafson, K.: Implicit space-time domain decomposition methods for stochastic parabolic partial differential equations. *SIAM J. Sci. Comput.* **36**, C1–C24 (2014)
- Dai, X., Maday, Y.: Stable parareal in time method for first- and second-order hyperbolic systems. *SIAM J. Sci. Comput.* **35**, A52–A78 (2013)
- Deng, X., Cai, X.-C., Zou, J.: A parallel space-time domain decomposition method for unsteady source inversion problems. *Inverse Probl. Imaging* **9**, 1069–1091 (2015)
- Deng, X., Cai, X.-C., Zou, J.: Two-level space-time domain decomposition methods for three-dimensional unsteady inverse source problems. *J. Sci. Comput.* **67**, 860–882 (2016)
- Dobrev, V.A., Kolev, T.Z., Petersson, N.A., Schroder, J.B.: Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM J. Sci. Comput.* **39**, S501–S527 (2017)
- Du, X.H., Sarkis, M., Schaerer, C.F., Szyld, D.B.: Inexact and truncated parareal-in-time Krylov subspace methods for parabolic optimal control problems. *Electron. Trans. Numer. Anal.* **40**, 36–57 (2013)
- Falgout, R.D., Friedho, S., Kolev, T.Z.V., Maclachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. *SIAM J. Sci. Comput.* **36**, C635–C661 (2014)
- Falgout, R.D., Friedho, S., Kolev, T.Z.V., MacLachlan, S.P., Schroder, J.B., Vandewalle, S.: Multigrid methods with space-time concurrency. *Comput. Visual. Sci.* **18**, 123–143 (2017)
- Falgout, R.D., Manteuffel, T.A., O’Neill, B., Schroder, J.B.: Multigrid reduction in time for nonlinear parabolic problems: a case study. *SIAM J. Sci. Comput.* **39**, S298–S322 (2017)
- Farhat, C., Chandris, M.: Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *Internat. J. Numer. Methods Eng.* **58**, 1397–1434 (2003)
- Frommer, A., Szyld, D.B.: An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM J. Numer. Anal.* **39**, 463–479 (2001)
- Gander, M.J.: A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations. *Numer. Linear Algebra Appl.* **6**, 125–145 (1999)
- Gander, M.J.: Optimized schwarz methods. *SIAM J. Numer. Anal.* **44**, 699–731 (2006)
- Gander, M.J.: 50 years of time parallel time integration. Multiple shooting and time domain decomposition methods, 69–113, *Contrib. Math. Comput. Sci.* **9**, Springer, Cham (2015)
- Gander, M.J., Neumüller, M.: Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM J. Sci. Comput.* **38**, A2173–A2208 (2016)
- Gander, M.J., Stuart, A.M.: Space-time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.* **19**, 2014–2031 (1998)

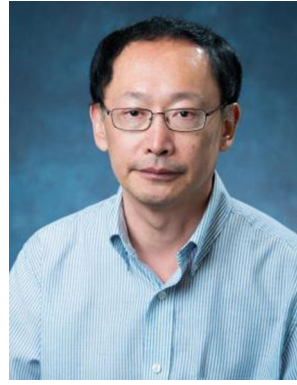
- Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.* **29**, 556–578 (2007)
- Giladi, E., Keller, H.B.: Space-time domain decomposition for parabolic problems. *Numer. Math.* **93**, 279–313 (2002)
- Hackbusch, W.: Parabolic multigrid methods. *Comput. Methods Appl. Sci. Eng. VI (Versailles, 1983)*, North-Holland, Amsterdam, 189–197 (1984)
- Halpern, L., Japhet, C., Szeftel, J.: Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems. *SIAM J. Numer. Anal.* **50**, 2588–2611 (2012)
- Horton, G.: The time-parallel multigrid method. *Comm. Appl. Numer. Methods.* **8**, 585–595 (1992)
- Horton, G., Vandewalle, S.: A space-time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.* **16**, 848–864 (1995)
- Horton, G., Vandewalle, S., Worley, P.: An algorithm with polylog parallel complexity for solving parabolic partial differential equations. *SIAM J. Sci. Comput.* **16**, 531–541 (1995)
- Jeltsch, R., Pohl, B.: Waveform relaxation with overlapping splittings. *SIAM J. Sci. Comput.* **16**, 40–49 (1995)
- Jiang, Y.-L., Chen, R.M.M., Wing, O.: Convergence analysis of waveform relaxation for nonlinear differential-algebraic equations of index one. *IEEE Trans. Circuits Syst. I. Fund. Theory Appl.* **47**, 1639–1645 (2000)
- Lelarmsee, E., Ruehli, A. E., Sangiovanni-Vincentelli, A. L.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. CAD of ICAS, CAD-1.* 131–145 (1982)
- Li, S., Cai, X.-C.: Convergence analysis of two-level space-time additive Schwarz method for parabolic equations. *SIAM J. Numer. Anal.* **53**, 2727–2751 (2015)
- Li, S., Shao, X., Cai, X.-C.: Multilevel space-time additive Schwarz methods for parabolic equations. *SIAM J. Sci. Comput.* **40**, A3012–A3037 (2018)
- Lions, J.-L., Maday, Y., Turinici, G.: Résolution d’EDP par un schéma en temps “pararéel”. *Comptes Rendus de l’Académie des Sciences. Série I. Mathématique* **332**, 661–668 (2001)
- Lubich, C., Ostermann, A.: Multigrid dynamic iteration for parabolic equations. *BIT* **27**, 216–234 (1987)
- Maday, Y., Salomon, J., Turinici, G.: Monotonic parareal control for quantum systems. *SIAM J. Numer. Anal.* **45**, 2468–2482 (2007)
- Mathew, T.P., Sarkis, M., Schaerer, C.E.: Analysis of block parareal preconditioners for parabolic optimal control problems. *SIAM J. Sci. Comput.* **32**, 1180–1200 (2010)
- Miekkala, U., Nevanlinna, O.: Convergence of dynamic iteration methods for initial value problem. *SIAM J. Sci. Stat. Comput.* **8**, 459–482 (1987)
- Minion, M.L.: A hybrid parareal spectral deferred corrections method. *Commun. Appl. Math. Comput. Sci.* **5**, 265–301 (2010)
- Minion, M.L., Speck, R., Bolten, M., Emmett, M., Ruprecht, D.: Interweaving PFASST and parallel multigrid. *SIAM J. Sci. Comput.* **37**, S244–S263 (2015)
- Nabben, R., Szyld, D.B.: Convergence theory of restricted multiplicative Schwarz methods. *SIAM J. Numer. Anal.* **40**, 2318–2336 (2003)
- Nevanlinna, O.: Linear acceleration of Picard–Lindelöf iteration. *Numer. Math.* **57**, 147–156 (1990)
- Nievergelt, J.: Parallel methods for integrating ordinary differential equations. *Comm. ACM* **7**, 731–733 (1964)
- Oosterlee, C.W., Wesseling, P.: Multigrid schemes for time-dependent incompressible Navier–Stokes equations. *Impact Comput. Sci. Engry* **5**, 153–175 (1993)
- Ruehli, A.E., Johnson, T.A.: Circuit analysis computing by waveform relaxation in encyclopedia of electrical and electronics engineering. Wiley, New York (1999)
- Ruprecht, D., Speck, R., Krause, R.: Parareal for diffusion problems with space- and time-dependent coefficients. Domain decomposition methods in science and engineering XXII, 371–378, *Lect. Notes Comput. Sci. Eng.* **104**, Springer, Cham (2016)
- Simoens, J., Vandewalle, S.: Waveform relaxation with fast direct methods as preconditioner. *SIAM J. Sci. Comput.* **21**, 1755–1773 (2000)
- Smith, B., Bjørstad, P., Gropp, W.: Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press, Cambridge (1996)
- Toselli, A., Widlund, O.B.: Domain decomposition methods—algorithms and theory. Springer, Berlin Heidelberg (2005)
- Tran, M.-B.: Parallel Schwarz waveform relaxation algorithm for an N -dimensional semilinear heat equation. *ESAIM Math. Model. Numer. Anal.* **48**, 795–813 (2014)
- Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of two-dimensional, unsteady, incompressible laminar flows. *Numer. Heat Trans. Part B Fund.* **50**, 25–40 (2006)
- Vandewalle, S., Horton, G.: Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing* **54**, 317–330 (1995)
- Vandewalle, S., Piessens, R.: Numerical experiments with nonlinear multigrid waveform relaxation on a parallel processor. *Appl. Numer. Math.* **8**, 149–161 (1991)
- Wu, S.-L., Xu, Y.: Convergence analysis of schwarz waveform relaxation with convolution transmission conditions. *SIAM J. Sci. Comput.* **39**, A890–A921 (2017)
- Wu, S.-L., Zhou, T.: Convergence analysis for three parareal solvers. *SIAM J. Sci. Comput.* **37**, A970–A992 (2015)
- Yang, H., Cai, X.-C.: Two-level space-time domain decomposition methods for flow control problems. *J. Sci. Comput.* **70**, 717–743 (2017)



Shishun Li was born in Henan, P. R. China. He received M.S. and PhD degrees in Mathematics from Zhejiang University, Hangzhou, in 2007 and 2011, respectively. He has been a member of the faculty of Henan Polytechnic University since July 2011. He was a postdoc during 2013–2014 in the Department of Computer Science, University of Colorado Boulder. His research interests are in domain decomposition methods, multigrid methods and parallel algorithms.



Xinping Shao was born in October, 1983. He received his PhD degree in the Department of Mathematics at Zhejiang University in 2010. He joined Hangzhou Dianzi University in 2010, and was appointed as Associate Professor in 2016. From 2016 to 2017, he was a visiting scholar in the Department of Computer Science, University of Colorado Boulder. His research interests include finite element methods and parallel computing.



Xiao-Chuan Cai is a professor in the Department of Computer Science, University of Colorado Boulder, and his research area is scientific computing.