Computing and Visualization in Science manuscript No.
(will be inserted by the editor)

Applications of Time Parallelization

Benjamin W. Ong · Jacob B. Schroder

Received: date / Accepted: date

Abstract This review article serves to summarize the many advances in time-parallel computations since the excellent review article by Gander, "50 years of Time Parallel Integration" [41]. We focus, when possible, on applications of time parallelism and the observed speedup and efficiency, highlighting the challenges and benefits of parallel time computations. The applications covered range from numerous PDE-based simulations (both hyperbolic and parabolic), to PDE-constrained optimization, powergrid simulations, and machine learning. The time-parallel methods covered range from various iterative schemes (multigrid, waveform, multiple shooting, domain decomposition) to direct time-parallel methods.

Keywords Time-Parallelization · Multigrid-in-Time · Waveform Relaxation · Parallel-in-Time

Mathematics Subject Classification (2010) 34-02 · 65-02 · 65-Y05

1 Introduction

Numerical simulations are increasingly important in the study of complex systems in engineering, life sciences, medicine, chemistry, physics, and even non-traditional fields such as social sciences. Computer models and simulations, often referred to as the "third pillar of science" [110], allow us to leverage modern supercomputers as virtual laboratories and experimental facilities. However, a brief glance at technological advancements in microprocessors (Figure 1) shows that future speedup for computational simulations will come through using increasing numbers of cores and not through faster clock speeds. Thus as spatial parallelism techniques saturate, parallelization in the time direction offers a promising avenue for leveraging modern supercomputers as they can work in tandem with existing spatial parallelism to provide a multiplicative increase in concurrency [41]. The need for time-parallel integration is being driven by this massively parallel nature of modern computer architectures.

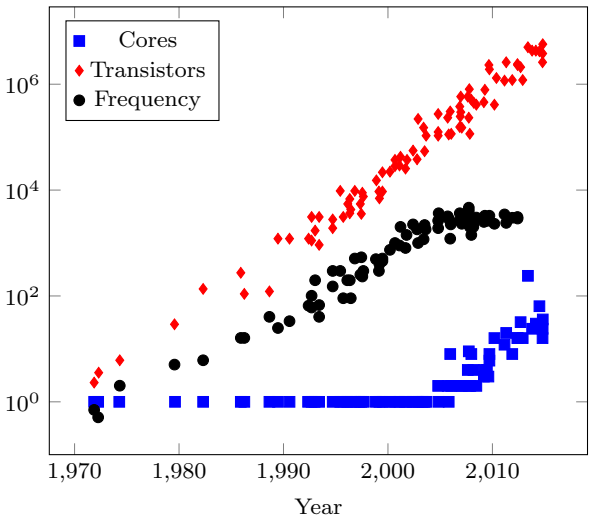

Fig. 1 Technological advances in micro-processors: Increased parallelism is expected in exascale systems, while clock frequencies are expected to remain stagnant.

Benjamin W. Ong
Michigan Technological University
Department of Mathematical Sciences
E-mail: ongbw@mtu.edu

Jacob B. Schroder
University of New Mexico
Department of Mathematics an Statistics
E-mail: jbschroder@unm.edu

Parallelization in the time direction is special because of the *causality principle*: solutions later in time are determined by solutions earlier in time. Algorithms trying to use the time direction for parallelization must account for this causality principle. Research on parallel-in-time integration started at least 50 years ago with the work of Nievergelt [102]. Since then, various approaches have been explored, with the review articles [41,54] and book [14] providing excellent introductions.

In this manuscript, we focus on applications of time parallelism, with an emphasis on recent articles published after Gander's review article [41]. The manuscript and discussion is organized loosely using the following four categories introduced by Gander [41]:

1. Section 2 focuses on methods based on *multiple shooting*;
2. Section 3 focuses on methods based on *waveform relaxation and domain decomposition*;
3. Section 4 focuses on methods based on *multigrid*; and
4. Section 5 focuses on *direct time-parallel methods*.

Each section begins with a brief overview of the category before an exposition of recent applications using the respective methods. Hybrid methods that span multiple categories are discussed after the appropriate background has been introduced. The reader should bear in mind that the classification of a parallel-in-time method into one of the above four categories is somewhat artificial in the sense that a parallel-in-time method can often be derived using the underlying mathematical tenets of a different category. For example, we discuss in section 4.3.1 that in some settings, a multigrid approach can be viewed as a method based on multiple-shooting.

2 Methods Based On Multiple Shooting

Fig. 2 Methods based on multiple shooting typically decompose a space-time domain by splitting the time domain into subintervals, solving related PDEs on the space-time blocks iteratively.

A first historical step that focuses entirely on parallelization in the time direction was by Nievergelt [102]. His revolutionary ideas were difficult to apply to higher dimensional problems, and a further essential ingredient was introduced in [16]. Consider an initial value problem (IVP) of the form

$$\partial_t u(t) = f(t, u(t)), \quad t \in (0, T], \quad u(0) = u^0. \tag{1}$$

One introduces intermediate targets: the time domain is split into subintervals $(T_{n-1}, T_n], n = 1, \ldots, N$ with $0 = T_0 < T_1 < \cdots < T_N = T$ (fig. 2), and the original IVP (1) is solved on subintervals

$$\begin{aligned} &\partial_t u_n(t) = f(t, u_n(t)), \quad t \in (T_n, T_{n+1}], \\ &\quad u_n(T_n) = U_n, \quad n = 0, 1, \ldots N-1. \end{aligned}$$

Observe that the system is consistent if

$$\begin{aligned} &U_0 = u_0(0) = u^0, \\ &U_1 = u_0(T_1) = u_0(T_1, U_0), \\ &\quad \vdots \\ &U_N = u_{N-1}(T_N) = u_{N-1}(T_N, U_{N-1}). \end{aligned} \tag{2}$$

Here, the parameters $U_n = u(T_n)$ are called the shooting parameters. System (2) can be posed as a root-finding problem for the shooting parameters,

$$F(\mathbf{U}) = \begin{pmatrix} U_0 - u^0 \\ U_1 - u_0(T_1, U_0) \\ \vdots \\ U_N - u_{N-1}(T_N, U_{N-1})) \end{pmatrix} = 0.$$

Newton's method can be applied to solve this root finding problem, giving rise to the recurrence relation,

$$U_{n+1}^{k+1} = u_n(t_{n+1}, U_n^k) + \frac{\partial u_n}{\partial U_n}(t_{n+1}, U_n^k)(U_n^{k+1} - U_n^k).$$

In [16], the authors showed that the algorithm converges locally quadratically, and global convergence is proved for dissipative systems. A more contemporary algorithm that has been widely studied is the parareal algorithm [91], which is based on an approximation of the derivative in the shooting method above. The idea is to (i) generate a rough approximation for the shooting parameters using a coarse (inaccurate) propagator, (ii) generate more accurate approximations given the approximate shooting parameters and a fine (accurate) propagator, and then (iii) iteratively apply the correction iteration. Specifically, let

1. $G(t_n, t_{n-1}, u_{n-1})$ be a rough approximation to $u(t_n)$ with initial condition $u(t_{n-1}) = u_{n-1}$,

2. $F(t_n, t_{n-1}, u_{n-1})$ be a more accurate approximation to the solution $u(t_n)$ with initial condition $u(t_{n-1}) = u_{n-1}$,

Starting with a coarse approximation U_n^0 at the time points $t_1, t_2, \ldots, t_N$, the parareal algorithm performs the correction at iteration k

$$U_{n+1}^{k+1} = F(t_{n+1}, t_n, U_n^k) + G(t_{n+1}, t_n, U_n^{k+1}) - G(t_{n+1}, t_n, U_n^k)), \quad \text{for } k = 0, 1, 2, \ldots$$

The parareal algorithm has been widely applied to solve a broad range of problems with varying success. Domain knowledge significantly influences the choice of the coarse and fine propagator. Generally, some speedup can be observed if *few* parareal iterations are required for convergence. We use the term convergence lightly in the following discussion, adopting the respective authors definition of when parareal can be terminated – the tolerance, and indeed the metric for convergence, may vary based on the application.[1] In earlier work which employed coarse and fine time grids, Bal and Maday [2] applied the parareal idea to solve the Black Scholes equation using a first-order time-splitting method. A speedup of 6× was observed when 50 processing cores were used. Later, Kaber and Maday [83] observed convergence of the parareal algorithm in 3 iterations for the Korteveg-deVries-Burgers equations. In their work, the coarse propagator solves a simplified equation that tracks the evolution of averaged solution quantities. Neglecting the time for the simulation of the coarse propagator, this translates to a theoretical speedup of 22× when 50 processing cores are used. Again, a time-splitting method is utilized, presumably with first-order splitting error. Mercerat et al. [98] applied the parareal algorithm to acoustic wave propagation, gaining a speedup of 5× with 50 processing cores. They explored a combination of Newmark schemes and time-DG methods [80] for the time advance. Schmitt et al. [115] show that a semi-Lagrangian discretization for the coarse propagator provides convergence benefits for parareal when applied to the 1D viscous Burgers equation, but does not report speedups. They conduct various experiments using second-order RK methods and first-order IMEX methods. Liu and Hu [93] applied the parareal algorithm to solve the Princeton Ocean Model and observed speedups of 5× with 201 processing cores because of stability restrictions imposed by the coarse and fine grid. The time advance involves a mode-splitting technique for computational efficiency. Parareal has also been applied to solve reduced plasma models (e.g., in the long wavelength limit) [113,112,111,4,5]. In this setting, parareal offers speedups of between 5–10× using $O(100)$ processors. The time advance in these implementations often uses a black-box stiff solver available via the VODPK package, which uses backward differentiation formulas of variable order. In [1], parareal is used to quickly plan robotic movements, with the goal of providing improved real-time commands for robotic movement. Explicit Euler integrators are used for the coarse and fine physics models. Bast et al. [3] solve for the steady-state of induction machines with a periodic parareal algorithm using an implicit Euler integrator and observe speedups of up to 28× using 80–128 processors. In a related work on dynamos, Clarke et al. [22] model time independent Roberts and time dependent Galloway-Proctor dynamos and see a speedup of roughly 300× with 1600 processors. Various combinations of first-, second-, and third-order IMEX methods are used as the fine solvers.

More recently, the asymptotic parareal method [66] was designed for nonlinear problems with a linearly oscillatory component. This work considers a nonlinear rotating shallow water equation model problem, with the goal to eventually solve problems such as weather, ocean, climate, and seismic models. A key innovation was to factor out the linearly oscillatory component (which is difficult for parareal to converge) when constructing new coarse time-grid equations. The coarse time-grid is then based on a time-averaged slow problem. The result is that parareal now converges quickly for the model problem, which it previously failed to effectively solve. This approach was further refined in [107], where the authors prove convergence for finite time-scale separation. In the numerical experiments, a second-order Strang splitting method is used for both the coarse and fine solves. Another related approach which formulates a new set of coarse time-grid equations, is the micro/macro parareal model [88]. In this work, the authors consider singularly perturbed ODEs, and the coarse (macroscopic) model is essentially the fine-scale (microscopic) model with the fast components removed. In the numerical experiments, an explicit Euler discretization is used. In [114], the micro/macro parareal model is applied to a climate model with discontinuous non-monotone coefficients, where the coarse-grid (macroscopic) model is a 0D approximation of the original fine-grid (microscopic) 1D model. The authors state that a theoretical performance gain of up to 10 is possible. The `odepack` library is used to solve the coarse macroscopic and fine microscopic models, where the library automatically switches between variable order multi-step schemes for integrating non-stiff prob-

[1] We will do the same regarding the convergence of the other iterative methods (multigrid, waveform, and domain decomposition).

lems, and variable order BDF formulas for integrating stiff problems.

Similarly, Blumers et al. [10] used a parareal type idea to solve the Navier-Stokes equations with stochastic microscopic properties. The coarse propagator used is an inexpensive continuum solver to capture macroscopic behavior. The fine solver is a Lagrangian integrator that resolves microscopic details. Good parallel efficiency is observed for each iteration – 65% efficiency with 1024 processing cores. Depending on the number of iterations required, some parallel speedup can be anticipated. If K parareal iterations are required for convergence, the overall parallel efficiency of the time integrator will be $\frac{65}{K}\%$.

3 Methods Based On Waveform Relaxation and Domain Decomposition

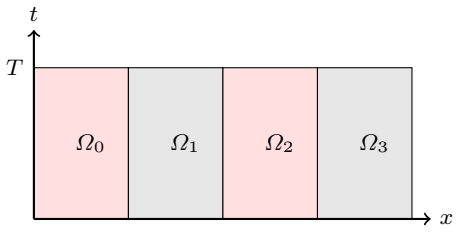

Fig. 3 Methods based on waveform relaxation decompose a space-time domain spatially to obtain a collection of coupled space-time subproblems.

Another class of parallel-in-time methods are iterative methods that arise from applying domain decomposition in a different way, the so-called waveform relaxation (WR) approach. WR methods were first introduced as a theoretical tool to study existence of a solution of an ODE [108]. Instead of solving $v' = f(t, v)$, Picard proposed to study

$$v^n(t) = v(0) + \int_0^t f\left(\tau, v^{n-1}(\tau)\right)\, d\tau, \tag{3}$$

for some initial guess v^0 and proved contraction to a fixed point, and thus existence and uniqueness of solutions to ODEs. A year later, Lindelöf [90] gave the famous super-linear convergence estimate for eq. (3). WR methods created an important engineering contribution by yielding a practical method for solving integrated circuit simulations [89]. WR methods were subsequently applied to solve parabolic PDEs [55,138,82, 75,51,56]. In this setting, the WR idea is to first decompose in space (Figure 3) to obtain a collection of coupled space-time subproblems. Each iteration then consists of exchanging interface information between the Ω_i over the entire time domain, and then solving the subproblems independently in parallel. There is extensive literature analyzing interface conditions for a wide range of PDEs, for example see [8,23,24,44]. In addition to the decomposition of the space-time domain, WR methods offer a further direction for parallelism-in-time: multiple waveform iterates can be simultaneously computed in a pipeline parallel fashion [55,138], provided one is willing to exchange coupling conditions at the interfaces more frequently. Implementations of the pipeline Schwarz WR was recently explored [103] and extended to Neumann-Neumann and Dirichlet-Neumann WR variants [105]. An adaptive WR variant was recently proposed and analyzed [84] when a theta-method is used for the time integration. It was shown that even for simple diffusion models, the speedup that can be obtained from using WR saturates at a relatively low number of time-parallel tasks: a speedup of 10× using 32 time-parallel tasks. The opportunity for exploring this area of time parallelization is vast – much of the research focus has been on developing and analyzing spatial coupling conditions in multiple dimensions, e.g., solutions to Schrödinger's equation [95] and the viscous shallow water equations [9].

One can combine multiple-shooting type methods with WR type methods, for example, parareal Schwarz WR methods have been proposed with a variety of first-order and higher-order RK methods used as fine solvers [46,15,92,47,48]. Consider a decomposition of a space-time domain as in Figure 4. One seeks to iteratively calculate increasingly accurate initial and boundary conditions for each space-time domain. This is accomplished by using a parareal approximation for the initial conditions of each space-time domain, and a Schwarz waveform relaxation algorithm for the boundary conditions.

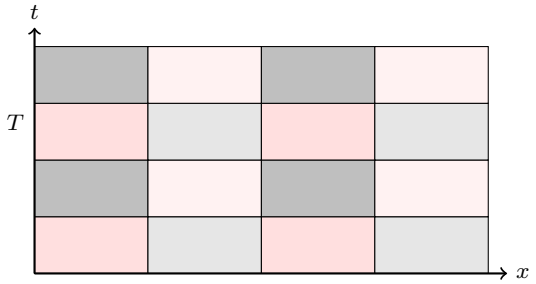

Fig. 4 Parallelism for Hybrid Parareal – Waveform Relaxation and for Space-Time Multigrid.

4 Multilevel Methods

This class of methods includes various multilevel approaches including the Parallel FAS[2] in Space-Time (PFASST) approach by Minion and Emmett [99,28] (section 4.1), space-time multigrid methods (section 4.2) and multigrid-in-time approaches (section 4.3). The effectiveness of these iterative approaches is due to the hierarchy of ever-coarser time grids that accelerate convergence to the solution on the finest time grid.

To describe these methods, take eq. (1) and write down a single step discretization,

$$u_{n+1} = \Phi(u_n, T_n) + g_{n+1}, \quad u(0) = u_0, \tag{4}$$

where T_n again satisfies $0 = T_0 < T_1 < \cdots < T_N = T$, u_n is the discrete state at time T_n, Φ represents the application of a time-stepping scheme, and g_n represents any forcing terms. When eq. (4) is viewed as a global space-time system in the linear case (although many of the multilevel methods also extend to the nonlinear setting), the following block $(N+1)\times(N+1)$ system is produced

$$H\,u = \begin{pmatrix} I & & & & \\ -\Phi & I & & & \\ & -\Phi & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi & I \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} u_0 \\ g_1 \\ g_2 \\ \vdots \\ g_N \end{pmatrix}. \tag{5}$$

4.1 PFASST

The PFASST approach uses a spectral deferred correction (SDC) [27] time integrator to define Φ. SDC integrates in time by spectrally computing the integral form of the initial value problem (Picard form in equation (3)). Specifically, SDC is an iterative scheme for computing a high-order collocation solution to equation (3), where each iteration is a correction sweep using a low-order time-stepping method. For example, consider computing an individual time-step from T_{n-1} to T_n, and let this nth time-step be represented as a state vector at $m+1$ high-order collocation points $\tau_{n,(0)} < \tau_{n,(1)} < \cdots < \tau_{n,(m)}$, where $T_{n-1} = \tau_{n,(0)}$ and $T_n = \tau_{n,(m)}$. The state vector is

$$\overline{U}_n = [u_{n,(0)}, u_{n,(1)}, \ldots, u_{n,(m)}].$$

Thus, $u_{n,(0)} = u_{n-1,(m)}$. Let $\overline{U}_{n,0}$ be an analogous vector, but with each entry equal to $u_{n,(0)}$, and

$$K(\overline{U}_n) = [f(u_{n,(0)}, \tau_{n,(0)}), \ldots, f(u_{n,(m)}, \tau_{n,(m)})]$$

represent the ODE right-hand-side evaluated at each collocation point. Then the nonlinear collocation problem to be solved is

$$\overline{U}_n = \overline{U}_{n,0} + \Delta t\; Q\; K(\overline{U}_n), \tag{6}$$

where $\Delta t = T_{n+1} - T_n$, and Q is an integration matrix. SDC solves equation (6) with a fixed-point iteration, where Q is preconditioned by the application of a lower-order integration scheme, such as backward or forward Euler. This makes the method relatively easy to implement, in that only a low-order integration scheme is needed. Next, we write the kth fixed-point iteration for solving (6) as

$$\overline{U}_n^k = F(\overline{U}_n^{k-1}, \overline{U}_{n,0}), \tag{7}$$

where the initial guess $\overline{U}_n^0$ can be computed with forward or backward Euler propagating $u_{n,(0)}$ forward to each collocation point. For typical cases when using forward or backward Euler to precondition, the order of the approximation can be shown to increase by one, for each iteration of (7).

Next following the compact PFASST description from [41], we consider the following SDC algorithm.

1 $u_{0,(m)}^K = u_0$
2 **for** $n = 1 : N$
3 *Initialize* $\overline{U}_n^0$ *with Euler on* $[T_{n-1}, T_n]$
4 **for** $k = 1 : K$
5 $\overline{U}_{n,0} = [u_{n-1,(m)}^K, \ldots, u_{n-1,(m)}^K]$
6 $\overline{U}_n^k = F(\overline{U}_n^{k-1}, \overline{U}_{n,0})$

Line 1 sets the initial condition with u_0 from equation (4). Line 2 is the loop over time-steps. Line 3 initializes $\overline{U}_n^0$ with Euler by propagating $u_{n-1,(m)}$ forward to u at each collocation point $u_{n,(i)}$ in $[T_{n-1}, T_n]$. Line 4 loops over the K SDC iterations. Line 5 sets $\overline{U}_{n,0}$ for use in equation (6). Line 6 carries out a fixed-point step.

The above algorithm is sequential, i.e., the SDC iterations in line 4 to compute time-step n require time-step $n-1$ to be complete. To make it parallel, a key insight of PFASST is to replace the index K with a lower-case k in line 5, first done in [99]. This allows for the SDC loop to be executed concurrently on many intervals $[T_{n-1}, T_n]$, i.e., on many processors. This modified algorithm is now inexact and forms the relaxation scheme for the FAS multigrid cycling that is PFASST. Each multigrid level is a successively coarser space-time discretization, where the order of the integration scheme typically drops by 1 when moving to a coarser level. For full PFASST pseudocode, see [28].

From the perspective of equation (5), the equation (6) forms the time-stepping scheme Φ, with the inexact

[2] FAS [12] is the full approximation scheme multigrid cycling strategy commonly used for nonlinear problems.

SDC sweeps forming the relaxation scheme on equation (5). For an example multilevel hierarchy, the finest time-level could use an order 4 SDC scheme, followed by three successively cheaper time-levels using order 3, 2, and 1 schemes, respectively. Thus the four time-levels are each associated with a system of the form (5), where the order of the time-integrator Φ changes between levels. Each coarser-level system is still of block-size $(N+1) \times (N+1)$, but each block is shrinking in size, as the number of collocation points shrinks. Importantly, each visit to a level in the hierarchy typically requires only one relaxation SDC sweep at each time point. This feature allows for good parallel efficiency for PFASST. For linear problems, the authors in [11] show the equivalence of PFASST to multigrid.

PFASST has been used in a variety of application areas. Regarding parabolic problems [28,100], the more recent work [100] observed speedups of 6×–9× with 24 processors in time for the heat equation, while using second- and fourth-order methods in space and in time with SDC. In [28], speedups of roughly 8× with 16 processors in time were observed for the the viscous Burgers equation, while using an 8th-order SDC method in time with a pseudo-spectral discretization in space. The work [100] reported parallel efficiencies (relative to sequential SDC integration) for the heat equation of over 60–70% when using 2 processors in time, going down to roughly 25% for 24 processors in time. Several additional strategies have been proposed to increase parallel efficiency [29,126,124].

PFASST has been coupled with the parallel Barnes-Hut tree code PEPC (Pretty Efficient Parallel Coulomb Solver [142]); speedups of 5×–7× with 32 processors in time and 10^5 to 4×10^6 particles in space are observed in [127], while using an 8th-order SDC method in time. The same idea is applied to vortex methods for solving the Navier–Stokes equations based on a Lagrangian particle discretization. The authors observed strong potential for speedup when using 10^4 particles in space [128] and 4th-, 8th-, and 12th-order SDC methods in time.

The field of PDE-constrained optimization is another promising area for PFASST (and parallel-in-time in general). Such problems often require repeated solves both forwards-in-time (for the primal variables) and backwards-in-time (for the adjoint variables), which can lead to prohibitively long time-serial run times. To address this, PFASST was applied to both the forwards-in-time and backwards-in-time equations for problems with parabolic PDE constraints [57,58]. Additional strategies to increase efficiency include reusing information from previous optimization iterations (warm restarts), solving the state and adjoint equations simultaneously, and a new mixed approach that takes advantage of the linearity of the adjoint equation. As a result, the work [58] observed a speedup of roughly 6× for the overall optimization procedure with 20 processors in time and a heat equation constraint, while using an 8th-order SDC method in time and a pseudo-spectral discretization in space. Furthermore, a speedup of roughly 8× with 32 processors in time was observed when using the nonlinear Nagumo equation as constraint, a 16th-order SDC method in time, and pseudo-spectral discretization in space.

The authors in [64] developed a multilevel SDC (MLSDC) approach for the rotating shallow water equations (RSWE), with a spherical harmonic spatial discretization that allows for efficient and effective spatial coarsening. The target applications here are weather, climate, and ocean models, which correspond to small viscosity values in the RSWE and thus historically difficult problems for parallel-in-time. This MLSDC work laid the foundation for application of PFASST to the RSWE. Thus, the authors in [65] went on to apply these concepts in parallel with PFASST for a suite of RSWE test problems, where speedups of 3×–10× were observed when using 16 processors in time and an 8th-order SDC method in time. Lastly by using the data redundancies in a multilevel setting, fault-tolerant extensions of PFASST were explored in [125].

4.2 Space-Time Multigrid

We categorize the following multigrid methods as either *space-time multigrid* or *multigrid in time* methods. The space-time multigrid methods apply multigrid to the full space-time problem (5), demanding the ability to access and coarsen each space-time degree of freedom in H. Here, techniques from classical spatial multigrid [133] must be appropriately modified before an efficient algorithm is obtained for the target problems, which are usually parabolic. First, the space-time matrix is recursively coarsened down to a trivially sized problem, where care must be taken to coarsen according to the level of anisotropy present in the space-time system, e.g., semi-coarsen in time, x, or y. For example, if the algebraic coupling in time is stronger than in space, then semi-coarsening in time is done. Multigrid interpolation techniques, such as constant in time or linear in space, move information between coarse and fine space-time grids. Again care must be taken to interpolate with respect to the level anisotropy, e.g., if semi-coarsening in time is done between two levels, then the interpolation could be constant in time, with no interpolation in space, as in [74]. Second, standard multigrid cycling is used to solve (5) with point-wise relaxation, such as

red-black Gauss-Seidel. In terms of parallelism, the domain is partitioned as in Figure 4, where each processor owns a space-time domain patch and is responsible for coarsening, interpolation, and point-wise relaxation on its local space-time domain patch.

Space–time multigrid was applied to parabolic problems by Hackbusch [63] in 1984, and then later in the 1990s [73,137,136,74]. However, these methods have enjoyed a renewed interest in recent years [141], in part because they are among the fastest parallel-in-time solvers available. For instance, the method by Horton and Vandewalle [74] enjoyed a 325× speedup for a model parabolic problem when using 16,384 processors in space-time, backward Euler in time, and second-order finite-differencing in space [31]. This speedup is computed as the comparison of a space-time parallel simulation using [74] against time-sequential, but spatially parallel, simulation.

The space-time multigrid method by Gander and Neumüller [50] for parabolic problems exhibited near perfect strong and weak scalability up to 262,144 cores, with a P1 linear finite element discretization in space and 3rd-order discontinuous Galerkin method in time. The authors also conducted non-parallel experiments with arbitrarily high-order DG discretizations in time, and verified that the multigrid method converges for various polynomial degrees (in time) up to order 45. The solver relies on multigrid components such as a parallel block Jacobi smoother with optimal damping factor introduced and analyzed in [50], and standard multigrid interpolation in space. The coarsening in space and time alternates (similarly to [74]) in order to maintain a balance between the spatial and temporal mesh spacings.

Space-time multigrid was applied to the time-periodic incompressible Navier-Stokes equations [7], where a lower-order preconditioner is used as a Vanka-style multigrid smoother [140]. The discretization uses fourth-order finite-differencing in space and time on a staggered 3D spatial grid. The speedup observed was approximately 50× when using 64 processors. As a prelude to expanding this solver to space-time linear systems with discontinuous Galerkin discretizations of the heat equation, the spectral symbol for diffusion matrices was derived [6].

Space-time multigrid solvers for parabolic problems discretized with isogeometric analysis began at least in [72] and were further developed in [71]. Here, higher-order space-time discretizations are considered (quadratic for the primary variable and cubic and quartic for the auxiliary variables). The same smooth higher-order splines are used to represent the computational domain and discretize the PDE. The space-time cylinder is decomposed into time-slices which are connected with a DG technique. A key contribution of this method is the smoother, which is an inexact damped block Jacobi. The method is verified on a linear parabolic model problem on a complicated, uniformly refined domain, although the paper [86] provides guaranteed error bounds and local indicators for local adaptivity. The authors observe nearly grid-independent convergence and enjoy good weak scaling out to 512 cores, with speedups of 1.5× and 10× reported for their two test problems. However, the excellent weak scaling indicates these speedups could significantly improve.

The authors in [129] consider both classical and modern algebraic multigrid (AMG) components for solving adaptively refined 4D linear finite element discretizations in space and time of a model parabolic equation. The lack of symmetry in the discretized system leads to the use of a GMRES outer iteration and Kaczmarcz relaxation. When GMRES is used, the number of iterations is either relatively flat, or modestly increasing, depending on the chosen AMG solver. The work [130] extends the above to consider more solvers such as classical algebraic multigrid, smoothed aggregation multigrid [139], and compatible relaxation multigrid [13,94].

The recent work [35] uses standard multigrid coarsening in both time and space, but explores different smoothing options, developing an adaptive smoothing strategy based on the degree of anisotropy in the discrete operator on each level. When the anisotropy is strong in the time dimension, a zebra relaxation ordering is used in space, and when the anisotropy is strong in space, a zebra relaxation ordering is used in time. For model parabolic equations fast, grid-independent multigrid convergence is observed when using backward Euler or Crank-Nicholson in time and second-order finite-differencing in space,

In an effort to compare all the various multigrid approaches, the authors in [31] compared the efficiency of space-time multigrid, with waveform multigrid, and multigrid reduction in time. All implementations were done through the hypre package [81] in an effort to provide uniform results. The test problem was the heat equation, discretized with backward Euler in time and second order finite-differencing in space. The result was that the speed of space-time multigrid was far superior to the other variants, often by 1 to 2 orders of magnitude.

The time-fractional heat equation offers a challenging opportunity for parallel-in-time, in part because standard discretizations here yield dense matrices. Nevertheless, the authors in [53] develop a multigrid method with waveform relaxation (line-relaxation in time) while exploiting the Toeplitz-like structure of the matrix to achieve nearly optimal $O(NM\log(M))$ complexity, where M is the number of time steps and N is the number of

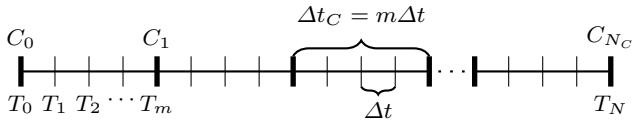

Fig. 5 MGRIT coarsening for a uniformly spaced grid with fine-grid points T_i (F-points), coarse-grid points C_i (C-points), and coarsening factor m. The C_i form the coarse-grid, while the F-points and C-points together form the fine-grid.

spatial grid points. The typical direct method would of course be cubic $O((NM)^3)$. The discretizations considered are backward Euler or Crank-Nicholson in time, with second order finite-differencing in space. The complexity and good convergence are confirmed with a semi-algebraic mode analysis. The authors in [79] extend this by exploiting H matrix properties to further reduce storage.

Lastly, while the above space-time methods all focused on parabolic problems, the recent effort by Manteuffel, Ruge, and Southworth [97] extended space-time multigrid methods to transport dominated problems, e.g., the time-dependent advection-diffusion equation, with very small (or no) diffusion. A key novelty here is that a local approximation to the so-called ideal restriction operator[3] is used. While no speedup numbers are given, the method does converge in a nearly grid-independent fashion for a model time-dependent recirculating flow problem. The discretization considered is backward Euler in time with streamline upwind Petrov-Galerkin (SUPG) in space and linear or quadratic finite element basis functions. This method has recently been implemented in parallel in the benchmark linear solver library hypre [81].

4.3 Multigrid in Time

An alternative multigrid approach is the multigrid reduction in time (MGRIT) method by Falgout et al. [36, 30], which applies multigrid only to the time-dimension (i.e., not space-time). MGRIT coarsens in time by coarsening the *block* rows in eq. (5). For example, if a coarsening factor of 2 in time were used, then the first coarse-level would yield a system similar to (5), only with $N/2$ block rows. For a general coarsening factor of m, an "ideal" coarse level operator would be

$$H_C = \begin{pmatrix} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{pmatrix}, \tag{8}$$

where H_C has N_C block rows and $N_C = N/m$ (for simplicity we assume N divides evenly by m). This coarsening by m corresponds to a partitioning of the fine level time-points into F-points and C-points, as depicted in Figure 5 for a uniform time-grid. The C-points (C_i) form the coarse time-grid and correspond to the block rows in H_C, whereas the union of the F- and C-points correspond to the block rows in H. Now, inverting H_C would be as expensive as inverting H, in terms of Φ applications, and would yield the exact solution at C-points. Thus, H_C is called "ideal." However, MGRIT makes a key approximation, substituting Φ^m with a single coarse time-step operation called Φ_C. For instance, Φ could be backward Euler with step size Δt, whereas Φ_C would use a step size of $m\Delta t$. With this substitution, inverting H_C is now relatively cheap, compared to inverting H.

We now describe a two-level MGRIT cycle, and note that a recursive application of the method on the coarse-level yields a multigrid V-cycle. The two-level cycle corresponds to (1) relaxation, followed by (2) coarse-grid correction. Relaxation is an unweighted block-Jacobi method. F-relaxation sets the residual equal to zero at all the block rows corresponding to F-points in equation (5), and C-relaxation does the same at C-point block rows. MGRIT typically uses an alternating FCF-relaxation.

After relaxation, the coarse-grid correction comes. The fine level solution is injected to the coarse level, where an error correction is computed based on the FAS formulation with H_C as the coarse level operator. This error correction is then interpolated to the fine level, where it is added to the current solution guess. Interpolation is injection, followed by F-relaxation. This interpolation is called "ideal", as well, because if the error correction is exact at C-points, then the interpolated solution at F-points will also be exact.

MGRIT is able to use existing time-stepping code because the individual space-time matrix entries are not required during this process, and the internals of Φ remain opaque to MGRIT. Only the ability to take a time-step with Φ on each time-level is required, similar to parareal. The nonlinear multigrid cycling is again handled with an FAS formulation.

In parallel, the domain is partitioned as in Figure 2, with each processor owning a set of time-values. For

[3] Ideal restriction (R) and interpolation (P) are defined to yield a Schur Complement coarse-grid operator with RAP.

further parallelism, coarsening and parallelism in the spatial dimension is also possible with the extensions in [31]. This yields a parallel decomposition as in Figure 4.

MGRIT has been used in a variety of application areas. Eddy-current problems for electromechanical energy converters was simulated using MGRIT [37]. A speedup of about 3× was observed when using 128 processors and either parareal or MGRIT with backward Euler in time and low-order finite elements in space. For parabolic problems [30,33,31], the more recent work [31] showed speedups for the heat equation of up to 53× when using 16,384 processors in space and time. The considered discretization was backward Euler in time and second-order finite-differencing in space. For the case of moving meshes and the heat equation, the work [34] successfully solved for both mesh points and solution values simultaneously in space and time, but did not report speedups. The discretization again was backward Euler and second-order finite-differencing in space.

The compressible Navier-Stokes equations with Reynolds number 100 was used to solve the classic problem of vortex shedding over a cylinder problem [32]. The observed speedup was 7× when using 4096 processors in time, with backward Euler in time and third-order finite-differencing in space. For a linearized elasticity problem, MGRIT was applied to model a linear beam attached to a wall [69]. The authors observed a speedup of 5× for 192 processors in time when using backward Euler in time and inf-sup stable Q^1-Q^2 Taylor-Hood elements in space. The work [78] solved the 1D inviscid Burgers equation with a novel adaptive spatial coarsening algorithm applied to the low-order discretization of backward Euler in time and first-order upwinding in space. The observed speedups go up to 6× and are the first observed speedups for MGRIT and pure advection, although the efficiencies are low, sometimes below 1%. The following work [131] considers only the linear case and carefully builds the coarse-level problem to optimize the MGRIT convergence estimate in [25]. The authors observed improved speedups of up to 18× for a model 1D advection equation when using 1024 processors in time. The considered discretizations ranged widely, including explicit Runge-Kutta methods of order 1–5 and singly diagonally implicit Runge-Kutta methods of order 1–4, coupled with finite-differencing in space of order 1–5. Looking forward, improved MGRIT performance for hyperbolic problems will likely remain an active research area.

Similar to PFASST, MGRIT can also be used to accelerate PDE-constrained optimization problems, by solving both the forwards- (primal) and backwards-in-time (adjoint) equations [59,60]. For greater efficiency, these works also developed a warm restart strategy that reuses information from previous optimization iterations, and explored the application of simultaneous optimization, which allows for inexact concurrent primal and adjoint solves. As a result, the work [60] observed speedups of up to 19× with 256 processors in time when solving an inverse design problem subject to a diffusion dominated PDE constraint that mimics flow past a cylindrical bluff body at low Reynolds number. The considered discretization was Crank-Nicholson in time and second-order finite-differencing in space. Such optimization approaches can be extended to ODE-inspired deep neural networks [62], where each layer in the network corresponds to a time-step. In this ODE neural network setting, the work [61] showed deep neural network training speedups of 4×–8× for various test problems when using 128 and 256 processors in time, respectively.

Moving to differential algebraic equations, MGRIT has been applied to power grid simulations [87,119], with the more recent work [119] showing a speedup of roughly 50× for a Western United States (WECC 179) model powergrid problem with scheduled events (i.e., discontinuities), while using RK4 and BDF2 in time.

Lastly, MGRIT and parareal have been applied to time-fractional heat equations [144], with observed speedups of 2×–5× when using 512 processors in time and linear space-time finite elements. The work [39] provides a performance model for the MGRIT methods, relative to its implementation in the open source XBraid package [143]. MGRIT methods have been extensively analyzed in [38,25,70,123] for parabolic and hyperbolic problems, and in multilevel settings.

4.3.1 Connection to Parareal

The multigrid approach to parallel in time has much in common with the methods in Sections 2 and 3. The work [52] showed equivalence between parareal and a two-grid nonlinear (FAS) multigrid method with a specific relaxation scheme. Given this connection, one can then state that parareal is equivalent to MGRIT with so-called F-relaxation. This connection between parareal and MGRIT is further investigated in [49] where the more powerful so-called FCF-relaxation in MGRIT is shown to be equivalent to parareal with an overlapping Schwarz relaxation. It has been shown that in some cases (but not all) this extra relaxation can be beneficial [25].

5 Direct Time-parallel Methods

Direct time parallel solvers comprise the last group of methods that we survey. Unlike the multiple shooting, multigrid and waveform relaxation ideas, these methods are non-iterative in nature. An early example of this approach is by Miranker and Liniger [101]. They transformed the predictor-corrector formula,

$$y^p_{n+1} = y^c_n + \frac{h}{2}\left(3f(y^c_n) - f(y^c_{n-1})\right),$$
$$y^c_{n+1} = y^c_n + \frac{h}{2}\left(f(y^p_{n+1}) + f(y^c_n)\right),$$

which is sequential in nature, into the modified formula

$$y^p_{n+1} = y^c_{n-1} + 2h\, f(y^p_n),$$
$$y^c_n = y^c_{n-1} + \frac{h}{2}\left(f(y^p_n + f(y^c_{n-1})\right),$$

which can be evaluated in parallel to give second- and third-order schemes. Similar predictor-corrector approaches resulting in parallel, iterated high-order RK methods were studied in [135,77,76,122]. Building on these ideas, a family of integral deferred correction methods [19,21, 106,20] was adapted to create arbitrary high-order parallel time integrators for multicore architectures. The key advance is a moving difference stencil which facilitates solving multiple error IVPs in parallel with high efficiency. These parallel-in-time integrators were used to solve parabolic PDEs, in combination with domain decomposition for spatial parallelism in [18,68]. It has been noted that these parallel integral deferred correction methods can be used for semi-Lagrangian Vlasov simulations [17], moving mesh PDEs [134], and simulations of vesicle suspensions [109]. A software release for this family of parallel time integrator is available [104]. Unlike the previously discussed parallel time integrators based on multiple shooting, multigrid or waveform relaxation methods, parallel RK and RIDC methods only offer small-scale parallelism of up to 12 time-parallel tasks.

Sheen, Sloan and Thomée [120] introduced a parallel-in-time method for parabolic problems based on contour integral representation and quadrature. Consider IVPs of the form

$$u_t + A\,u = 0, \quad u(0) = u_0. \tag{9}$$

Taking the Laplace transform of eq. (9) with parameter s gives

$$s\hat{u} + A\hat{u} = u_0 \implies \hat{u} = (sI + A)^{-1}u_0. \tag{10}$$

Applying the inverse Laplace transform of eq. (10) gives

$$u(t) = \frac{1}{2\pi\, i}\int_\Gamma e^{st}\hat{u}(s)\, ds.$$

This integral can be approximated with a quadrature formula with nodes s_j. This computation is completely independent and can be computed on separate processors. Some higher-order extensions are discussed in [121,132]. Lai [85] introduced transformation methods to further improve the parallel properties. Higher-order schemes for the Laplace transformations are considered in [26].

Another class of direct time-parallel methods leverages the tensor-product structure of space-time solvers [96]. Suppose $u_t = L\,u$ is discretized using the backward Euler integrator and time-step sizes Δt_i, such that

$$B := \begin{pmatrix} \left(\frac{1}{\Delta t_1} - L\right) & & & \\ -\frac{1}{\Delta t_2} & \left(\frac{1}{\Delta t_2} - L\right) & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\Delta t_N} & \left(\frac{1}{\Delta t_N} - L\right) \end{pmatrix},$$
$$u := \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}, \text{ and } f := \begin{pmatrix} f_1 + \frac{1}{\Delta t_1}u_0 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} = f,$$

which yields the equation

$$B\,u = f. \tag{11}$$

If the matrix B is diagonalizable, i.e., $B = SDS^{-1}$, then we can solve eq. (11) in three steps:

$$S\,g = f, \qquad \left(\frac{1}{\Delta t_n} - L\right) w_n = g_n, \qquad S^{-1}u = w.$$

If the time steps are all equal, then B is not diagonalizable. This begs the question how one should choose Δt_j? In [45], the authors introduce a geometric mesh, where each $\Delta t_n = (1+\epsilon)^{n-1}\Delta t_1$, then show that the solution can be computed by diagonalization using this geometric mesh, and finally analyzed the error that arises from this choice of mesh. An optimized variant (i.e., choice of ϵ) is proposed by balancing the round-off and truncation error. They further generalize this parallel-in-time approach to solving the wave equation using a second-order Newmark time integration scheme [40].

In [42], the authors describe a new method, ParaExp, for the parallel integration of linear IVPs of the form $u'(t) = A\,u(t) + g(t)$. They observe that the original problem can be decomposed into decoupled inhomogeneous and homogeneous subproblems. The solution to the homogeneous IVP $w'(t) = A\,w(t)$ can be given in terms of a matrix exponential; this computation can be parallelized efficiently by using a rational

Krylov approximation to the matrix exponential. Borrowing ideas from nonlinear PDE analysis, ParaExp can be extended to solve non-linear IVPs [43]. The authors show that the nonlinear variant converges in a finite number of steps, and that the nonlinear ParaExp algorithm can be interpreted as a parareal algorithm, if the coarse integrator solves the linear part of the evolution problem.

The REXI methods (rational approximation of exponential integrators) are another group of methods based on rational approximations, and they provide two levels of parallelism. First, the computation of the exponential approximation decouples into many terms that may be computed independently in parallel, and second, this level of parallelism could be combined with parareal-like parallelism. The REXI approach in [67] was designed for hyperbolic problems, such as the classical wave equation and linearized shallow water equation. The work [67] then formed the basis for [117], where REXI was studied and applied to solve the linearized rotating shallow water equations with large observed speedups of approximately 188× for spectral methods and 1503× speedup for finite-difference methods. However, these speedups do not consider the nonlinear terms present in the target weather, ocean, and climate models. Further REXI work in [116] solves the linearized shallow water equations on a rotating sphere, where the method provides a supported time step size increase from 100 seconds to one day and significant speedups. A Cauchy contour integral REXI (CI-REXI) is used in [118] to integrate the linear parts of the full nonlinear shallow water equations on the rotating sphere, with an approximately 3× parallel speedup. The target applications for these REXI methods applied to shallow water equations are weather, climate, and ocean models.

6 Outlook

As modern supercomputers continue to increase their capability through increased concurrency, the need for time-parallel integration is expected to manifest itself across a broad range of evolutionary computational science problems. Time-parallel integration is indeed a very active field of research, evidenced by the growing number of publications in this research area [4] and the increased attendance at the annual parallel-in-time workshop. Although this survey article attempts to summarize recent applications of parallel time integration and their varying levels of success in achieving parallel speedup, a systematic apples-to-apples comparison of the various approaches is still lacking, and caution should be taken when directly comparing speedup numbers across methods and implementations. In particular, some of the speedup and efficiency numbers are only theoretical in nature, and many of the presented parallel time methods do not address the storage or communication overhead of the parallel time integrator.

[4] http://parallel-in-time.org

References

1. Agboh, W.C., Ruprecht, D., Dogar, M.R.: Combining coarse and fine physics for manipulation using parallel-in-time integration (2019). URL https://arxiv.org/abs/1903.08470
2. Bal, G., Maday, Y.: A "Parareal" time discretization for non-linear PDE's with application to the pricing of an American Put. In: L. Pavarino, A. Toselli (eds.) Recent Developments in Domain Decomposition Methods, *Lecture Notes in Computational Science and Engineering*, vol. 23, pp. 189–202. Springer Berlin (2002). URL http://dx.doi.org/10.1007/978-3-642-56118-4_12
3. Bast, D., Kulchytska-Ruchka, I., Schoeps, S., Rain, O.: Accelerated steady-state torque computation for induction machines using parallel-in-time algorithms. IEEE Transactions on Magnetics pp. 1–1 (2020). DOI 10.1109/tmag.2019.2945510. URL https://doi.org/10.1109%2Ftmag.2019.2945510
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 International Publishing (2014). URL http://dx.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. Journal of Computational Physics **279**(0), 67–79 (2014). URL http://dx.doi.org/10.1016/j.jcp.2014.08.037
6. Benedusi, P., Garoni, C., Krause, R., Li, X., Serra-Capizzano, S.: Space-time FE-DG discretization of the anisotropic diffusion equation in any dimension: the spectral symbol. SIAM Journal on Matrix Analysis and Applications **39**(3), 1383–1420 (2018)
7. Benedusi, P., Hupp, D., Arbenz, P., Krause, R.: A Parallel Multigrid Solver for Time–periodic Incompressible Navier–Stokes Equations in 3D. In: Numerical Mathematics and Advanced Applications ENUMATH 2015, pp. 265–273. Springer (2016). DOI 10.1007/978-3-319-39929-4_26. URL https://doi.org/10.1007/978-3-319-39929-4_26
8. Bennequin, D., Gander, M.J., Gouarin, L., Halpern, L.: Optimized Schwarz waveform relaxation for advection reaction diffusion equations in two dimensions. Numer. Math. **134**(3), 513–567 (2016). DOI 10.1007/s00211-015-0784-8. URL https://doi.org/10.1007/s00211-015-0784-8
9. Blayo, E., Rousseau, A., Tayachi, M.: Boundary conditions and schwarz waveform relaxation method for linear viscous shallow water equations in hydrodynamics. The SMAI journal of computational mathematics **3**, 117–137 (2017). DOI 10.5802/smai-jcm.

22. URL https://smai-jcm.centre-mersenne.org/item/SMAI-JCM_2017__3__117_0
10. Blumers, A.L., Li, Z., Karniadakis, G.E.: Supervised parallel-in-time algorithm for long-time lagrangian simulations of stochastic dynamics: Application to hydrodynamics. Journal of Computational Physics **393**, 214 – 228 (2019). DOI https://doi.org/10.1016/j.jcp.2019.05.016. URL http://www.sciencedirect.com/science/article/pii/S0021999119303419
11. Bolten, M., Moser, D., Speck, R.: A multigrid perspective on the parallel full approximation scheme in space and time. Numerical Linear Algebra with Applications **24**(6), e2110 (2017). DOI 10.1002/nla.2110. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2110. E2110 nla.2110
12. Brandt, A.: Multi–level adaptive solutions to boundary–value problems. Math. Comp. **31**(138), 333–390 (1977)
13. Brandt, A.: General highly accurate algebraic coarsening. Electron. Trans. Numer. Anal. **10**, 1–20 (2000)
14. Burrage, K.: Parallel and sequential methods for ordinary differential equations. Numerical Mathematics and Scientific Computation. The Clarendon Press, Oxford University Press, New York (1995). Oxford Science Publications
15. Cadeau, T., Magoules, F.: Coupling the Parareal Algorithm with the Waveform Relaxation Method for the Solution of Differential Algebraic Equations. In: Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2011 Tenth International Symposium on, pp. 15–19 (2011). URL http://dx.doi.org/10.1109/DCABES.2011.34
16. Chartier, P., Philippe, B.: A parallel shooting technique for solving dissipative ODE's. Computing **51**(3-4), 209–236 (1993). URL http://dx.doi.org/10.1007/BF02238534
17. Christlieb, A., Guo, W., Morton, M., Qiu, J.M.: A high order time splitting method based on integral deferred correction for semi-lagrangian vlasov simulations. Journal of Computational Physics **267**, 7 – 27 (2014). DOI https://doi.org/10.1016/j.jcp.2014.02.012. URL http://www.sciencedirect.com/science/article/pii/S0021999114001326
18. Christlieb, A.J., Haynes, R.D., Ong, B.W.: A Parallel Space-Time Algorithm. SIAM Journal on Scientific Computing **34**(5), C233–C248 (2012). URL http://dx.doi.org/10.1137/110843484
19. Christlieb, A.J., Macdonald, C.B., Ong, B.W.: Parallel high-order integrators. SIAM Journal on Scientific Computing **32**(2), 818–835 (2010). URL http://dx.doi.org/10.1137/09075740X
20. Christlieb, A.J., MacDonald, C.B., Ong, B.W., Spiteri, R.J.: Revisionist integral deferred correction with adaptive step-size control. Communications in Applied Mathematics and Computational Science **10**, 1–25 (2015). URL http://dx.doi.org/10.2140/camcos.2015.10.1
21. Christlieb, A.J., Ong, B.W.: Implicit parallel time integrators. Journal of Scientific Computing **49**(2), 167–179 (2011). URL http://dx.doi.org/10.1007/s10915-010-9452-4
22. Clarke, A.T., Davies, C.J., Ruprecht, D., Tobias, S.M.: Parallel-in-time integration of kinematic dynamos (2019). URL https://arxiv.org/abs/1902.00387
23. Courvoisier, Y., Gander, M.J.: Time domain Maxwell equations solved with Schwarz waveform relaxation methods. In: Domain decomposition methods in science and engineering XX, *Lect. Notes Comput. Sci. Eng.*, vol. 91, pp. 263–270. Springer, Heidelberg (2013). DOI 10.1007/978-3-642-35275-1_30. URL https://doi.org/10.1007/978-3-642-35275-1_30
24. Descombes, S., Dolean, V., Gander, M.J.: Schwarz waveform relaxation methods for systems of semi-linear reaction-diffusion equations. In: Domain decomposition methods in science and engineering XIX, *Lect. Notes Comput. Sci. Eng.*, vol. 78, pp. 423–430. Springer, Heidelberg (2011). DOI 10.1007/978-3-642-11304-8_49. URL https://doi.org/10.1007/978-3-642-11304-8_49
25. Dobrev, V.A., Kolev, T., Petersson, N.A., Schroder, J.B.: Two-level convergence theory for multigrid reduction in time (mgrit). SIAM Journal on Scientific Computing **39**(5), S501–S527 (2017). DOI 10.1137/16M1074096. URL https://doi.org/10.1137/16M1074096
26. Douglas, C., Kim, I., Lee, H., Sheen, D.: Higher-order schemes for the Laplace transformation method for parabolic problems. Comput. Vis. Sci. **14**(1), 39–47 (2011). DOI 10.1007/s00791-011-0156-6. URL https://doi.org/10.1007/s00791-011-0156-6
27. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT Numerical Mathematics **40**(2), 241–266 (2000)
28. Emmett, M., Minion, M.L.: Toward an efficient parallel in time method for partial differential equations. Commun. Appl. Math. Comput. Sci. **7**(1), 105–132 (2012)
29. Emmett, M., Minion, M.L.: Efficient implementation of a multi-level parallel in time algorithm. In: Domain Decomposition Methods in Science and Engineering XXI, *Lecture Notes in Computational Science and Engineering*, vol. 98, pp. 359–366. Springer International Publishing (2014). URL http://dx.doi.org/10.1007/978-3-319-05789-7_33
30. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM J. Sci. Comput. **36**(6), C635–C661 (2014). LLNL-JRNL-645325
31. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B., Vandewalle, S.: Multigrid methods with space–time concurrency. Computing and Visualization in Science **18**(4), 123–143 (2017). DOI 10.1007/s00791-017-0283-9. URL https://doi.org/10.1007/s00791-017-0283-9
32. Falgout, R.D., Katz, A., Kolev, T.V., Schroder, J.B., Wissink, A.M., Yang, U.M.: Parallel Time Integration with Multigrid Reduction for a Compressible Fluid Dynamics Application. Tech. rep., Lawrence Livermore National Laboratory (2014). URL https://computation.llnl.gov/project/parallel-time-integration/pubs/strand2d-pit.pdf
33. Falgout, R.D., Manteuffel, T.A., O'Neill, B., Schroder, J.B.: Multigrid reduction in time for nonlinear parabolic problems: A case study. SIAM Journal on Scientific Computing **39**(5), S298–S322 (2017). DOI 10.1137/16M1082330. URL https://doi.org/10.1137/16M1082330
34. Falgout, R.D., Manteuffel, T.A., Southworth, B., Schroder, J.B.: Parallel-in-time for moving meshes. Tech. rep., Lawrence Livermore National Laboratory (2016). DOI 10.2172/1239230. URL http://www.osti.gov/scitech/servlets/purl/1239230
35. Franco, S.R., Gaspar, F.J., Pinto, M.A.V., Rodrigo, C.: Multigrid method based on a space-time approach with standard coarsening for parabolic problems. Applied Mathematics and Computation **317**(Supplement

C), 25–34 (2018). DOI 10.1016/j.amc.2017.08.043. URL https://doi.org/10.1016/j.amc.2017.08.043

36. 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, Mar 17 - Mar 22, 2013 (2013). URL http://www.osti.gov/scitech/servlets/purl/1073108
37. Friedhoff, S., Hahne, J., Kulchytska-Ruchka, I., Schöps, S.: Exploring parallel-in-time approaches for eddy current problems. In: Progress in Industrial Mathematics at ECMI 2018, pp. 373–379. Springer (2019)
38. Friedhoff, S., MacLachlan, S.: A generalized predictive analysis tool for multigrid methods. Numerical Linear Algebra with Applications **22**(4), 618–647 (2015)
39. Gahvari, H., Dobrev, V.A., Falgout, R.D., Kolev, T.V., Schroder, J.B., Schulz, M., Meier Yang, U.: A performance model for allocating the parallelism in a multigrid-in-time solver. In: 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (2016). DOI 10.1109/PMBS.2016.8. URL http://dx.doi.org/10.1109/PMBS.2016.8
40. Gander, M., Halpern, L., Rannou, J., Ryan, J.: A Direct Time Parallel Solver by Diagonalization for the Wave Equation. SIAM Journal on Scientific Computing **41**(1), A220–A245 (2019). DOI 10.1137/17M1148347. URL https://doi.org/10.1137/17M1148347
41. Gander, M.J.: 50 years of Time Parallel Time Integration. In: Multiple Shooting and Time Domain Decomposition. Springer (2015). URL http://dx.doi.org/10.1007/978-3-319-23321-5_3
42. Gander, M.J., Güttel, S.: PARAEXP: A Parallel Integrator for Linear Initial-Value Problems. SIAM Journal on Scientific Computing **35**(2), C123–C142 (2013). URL http://dx.doi.org/10.1137/110856137
43. Gander, M.J., Güttel, S., Petcu, M.: A nonlinear paraexp algorithm. In: P.E. Bjørstad, S.C. Brenner, L. Halpern, H.H. Kim, R. Kornhuber, T. Rahman, O.B. Widlund (eds.) Domain Decomposition Methods in Science and Engineering XXIV, pp. 261–270. Springer International Publishing, Cham (2018)
44. Gander, M.J., Halpern, L., Labbé, S., Santugini-Repiquet, K.: An optimized Schwarz waveform relaxation algorithm for micro-magnetics. In: Domain decomposition methods in science and engineering XVII, *Lect. Notes Comput. Sci. Eng.*, vol. 60, pp. 203–210. Springer, Berlin (2008). DOI 10.1007/978-3-540-75199-1_22. URL https://doi.org/10.1007/978-3-540-75199-1_22
45. Gander, M.J., Halpern, L., Ryan, J., Tran, T.T.B.: A Direct Solver for Time Parallelization. In: T. Dickopf, M.J. Gander, L. Halpern, R. Krause, L.F. Pavarino (eds.) Domain Decomposition Methods in Science and Engineering XXII, pp. 491–499. Springer International Publishing (2016). DOI 10.1007/978-3-319-18827-0_50
46. Gander, M.J., Jiang, Y.L., Li, R.J.: Parareal Schwarz waveform relaxation methods. In: Domain decomposition methods in science and engineering XX, *Lect. Notes Comput. Sci. Eng.*, vol. 91, pp. 451–458. Springer, Heidelberg (2013). DOI 10.1007/978-3-642-35275-1_53. URL https://doi.org/10.1007/978-3-642-35275-1_53
47. Gander, M.J., Jiang, Y.L., Li, R.J.: Parareal Schwarz Waveform Relaxation Methods. In: R. Bank, M. Holst, O. Widlund, J. Xu (eds.) Domain Decomposition Methods in Science and Engineering XX, *Lecture Notes in Computational Science and Engineering*, vol. 91, pp. 451–458. Springer Berlin Heidelberg (2013). URL http://dx.doi.org/10.1007/978-3-642-35275-1_53
48. Gander, M.J., Jiang, Y.L., Song, B.: A superlinear convergence estimate for the parareal schwarz waveform relaxation algorithm. SIAM Journal on Scientific Computing **41**(2), A1148–A1169 (2019). DOI 10.1137/18M1177226. URL https://doi.org/10.1137/18M1177226
49. Gander, M.J., Kwok, F., Zhang, H.: Multigrid interpretations of the parareal algorithm leading to an overlapping variant and mgrit. Computing and Visualization in Science (2018). DOI 10.1007/s00791-018-0297-y. URL https://doi.org/10.1007/s00791-018-0297-y
50. Gander, M.J., Neumüller, M.: Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM Journal on Scientific Computing **38**(4), A2173–A2208 (2016)
51. Gander, M.J., Stuart, A.M.: Space-Time Continuous Analysis of Waveform Relaxation for the Heat Equation. SIAM Journal on Scientific Computing **19**(6), 2014–2031 (1998). URL http://dx.doi.org/10.1137/S1064827596305337
52. Gander, M.J., Vandewalle, S.: Analysis of the Parareal Time-Parallel Time-Integration Method. SIAM Journal on Scientific Computing **29**(2), 556–578 (2007). URL http://dx.doi.org/10.1137/05064607X
53. Gaspar, F.J., Rodrigo, C.: Multigrid waveform relaxation for the time-fractional heat equation. SIAM Journal on Scientific Computing **39**(4), A1201–A1224 (2017). DOI 10.1137/16M1090193. URL https://doi.org/10.1137/16M1090193
54. Gear, C.W.: Parallel methods for ordinary differential equations. CALCOLO **25**(1-2), 1–20 (1988). URL http://dx.doi.org/10.1007/BF02575744
55. Gear, C.W.: Waveform methods for space and time parallelism. In: Proceedings of the International Symposium on Computational Mathematics (Matsuyama, 1990), vol. 38, pp. 137–147 (1991)
56. Giladi, E., Keller, H.B.: Space-time domain decomposition for parabolic problems. Numer. Math. **93**(2), 279–313 (2002). DOI 10.1007/s002110100345. URL https://doi.org/10.1007/s002110100345
57. Götschel, S., Minion, M.L.: Parallel-in-time for parabolic optimal control problems using PFASST. In: P.E. Bjørstad, S.C. Brenner, L. Halpern, H.H. Kim, R. Kornhuber, T. Rahman, O.B. Widlund (eds.) Domain Decomposition Methods in Science and Engineering XXIV, pp. 363–371. Springer International Publishing (2018). DOI 10.1007/978-3-319-93873-8_34. URL https://doi.org/10.1007/978-3-319-93873-8_34
58. Gotschel, S., Minion, M.L.: An efficient parallel-in-time method for optimization with parabolic pdes. SIAM Journal on Scientific Computing **41**(6), C603–C626 (2019)
59. Günther, S., Gauger, N.R., Schroder, J.B.: A non-intrusive parallel-in-time adjoint solver with the XBraid library. Computing and Visualization in Science **19**(3-4), 85–95 (2018)
60. Günther, S., Gauger, N.R., Schroder, J.B.: A non-intrusive parallel-in-time approach for simultaneous optimization with unsteady PDEs. Optimization Methods and Software **0**(0), 1–16 (2018)
61. Günther, S., Ruthotto, L., Schroder, J.B., Cyr, E.C., Gauger, N.R.: Layer-parallel training of deep residual

neural networks. SIAM Journal on Mathematics of Data Science **2**(1), 1–23 (2020). DOI 10.1137/19M1247620. URL https://doi.org/10.1137/19M1247620

62. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. Inverse Problems **34**(1), 014004 (2017)
63. Hackbusch, W.: Parabolic multigrid methods. In: Computing methods in applied sciences and engineering, VI (Versailles, 1983), pp. 189–197. North-Holland, Amsterdam (1984)
64. Hamon, F.P., Schreiber, M., Minion, M.L.: Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere. Journal of Computational Physics **376**, 435 – 454 (2019)
65. Hamon, F.P., Schreiber, M., Minion, M.L.: Parallel-in-time multi-level integration of the shallow-water equations on the rotating sphere (2020). DOI https://doi.org/10.1016/j.jcp.2019.109210. URL http://www.sciencedirect.com/science/article/pii/S0021999119309155
66. Haut, T., Wingate, B.: An asymptotic parallel-in-time method for highly oscillatory PDEs. SIAM Journal on Scientific Computing **36**(2), A693–A713 (2014). URL http://dx.doi.org/10.1137/130914577
67. Haut, T.S., Babb, T., Martinsson, P.G., Wingate, B.A.: A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator. IMA Journal of Numerical Analysis **36**(2), 688–716 (2015)
68. Haynes, R.D., Ong, B.W.: MPI-OpenMP algorithms for the parallel space-time solution of time dependent PDEs. In: Domain Decomposition Methods in Science and Engineering XXI, *Lecture Notes in Computational Science and Engineering*, vol. 98, pp. 179–187. Springer International Publishing (2014). URL http://dx.doi.org/10.1007/978-3-319-05789-7_14
69. Hessenthaler, A., Nordsletten, D., Röhrle, O., Schroder, J.B., Falgout, R.D.: Convergence of the multigrid reduction in time algorithm for the linear elasticity equations. Numerical Linear Algebra with Applications **25**(3), e2155 (2018). DOI 10.1002/nla.2155. URL https://dx.doi.org/10.1002/nla.2155
70. Hessenthaler, A., Southworth, B.S., Nordsletten, D., Röhrle, O., Falgout, R.D., Schroder, J.B.: Multilevel convergence analysis of multigrid-reduction-in-time. arXiv:1812.11508 [math.NA] (2018). URL https://arxiv.org/abs/1812.11508
71. Hofer, C., Langer, U., Neumüller, M., Schneckenleitner, R.: Parallel and robust preconditioning for space-time isogeometric analysis of parabolic evolution problems. SIAM Journal on Scientific Computing **41**(3), A1793–A1821 (2019)
72. Hofer, C., Langer, U., Neumüller, M., Toulopoulos, I.: Time-multipatch discontinuous galerkin space-time isogeometric analysis of parabolic evolution problems. Electronic Transactions on Numerical Analysis **49**, 126–150 (2018)
73. Horton, G.: The time-parallel multigrid method. Comm. Appl. Numer. Methods **8**(9), 585–595 (1992)
74. Horton, G., Vandewalle, S.: A space-time multigrid method for parabolic partial differential equations. SIAM J. Sci. Comput. **16**(4), 848–864 (1995)
75. Horton, G., Vandewalle, S., Worley, P.: An Algorithm with Polylog Parallel Complexity for Solving Parabolic Partial Differential Equations. SIAM Journal on Scientific Computing **16**(3), 531–541 (1995). URL http://dx.doi.org/10.1137/0916034
76. van der Houwen, P., Sommeijer, B.: Analysis of parallel diagonally implicit iteration of Runge-Kutta methods. Applied Numerical Mathematics **11**(1), 169–188 (1993). URL http://dx.doi.org/10.1016/0168-9274(93)90047-U
77. van der Houwen, P.J., Sommeijer, B.P.: Iterated Runge-Kutta Methods on Parallel Computers. SIAM Journal on Scientific and Statistical Computing **12**(5), 1000–1028 (1991). DOI 10.1137/0912054. URL http://dx.doi.org/10.1137/0912054
78. Howse, A., Sterck, H., Falgout, R., MacLachlan, S., Schroder, J.: Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. SIAM Journal on Scientific Computing **41**(1), A538–A565 (2019). DOI 10.1137/17M1144982. URL https://dx.doi.org/10.1137/17M1144982
79. Hu, X., Rodrigo, C., Gaspar, F.J.: Using hierarchical matrices in the solution of the time-fractional heat equation by multigrid waveform relaxation. arXiv:1706.07632v1 [math.NA] (2017). URL https://arxiv.org/pdf/1706.07632.pdf
80. Hulbert, G.M., Hughes, T.J.: Space-time finite element methods for second-order hyperbolic equations. Computer Methods in Applied Mechanics and Engineering **84**(3), 327 – 348 (1990). DOI https://doi.org/10.1016/0045-7825(90)90082-W. URL http://www.sciencedirect.com/science/article/pii/004578259090082W
81. *hypre*: High performance preconditioners. Http://www.llnl.gov/CASC/hypre/
82. Janssen, J., Vandewalle, S.: Multigrid waveform relaxation on spatial finite element meshes. In: Contributions to multigrid (Amsterdam, 1993), *CWI Tract*, vol. 103, pp. 75–86. Math. Centrum Centrum Wisk. Inform., Amsterdam (1994)
83. Kaber, S.M., Maday, Y.: Parareal in time approximation of the Korteveg-deVries-Burgers' equations. PAMM **7**, 1026403–1026404 (2007). URL http://dx.doi.org/10.1002/pamm.200700574
84. Kwok, F., Ong, B.: Schwarz waveform relaxation with adaptive pipelining. SIAM Journal on Scientific Computing **41**(1), A339–A364 (2019). DOI 10.1137/17M115311X. URL https://doi.org/10.1137/17M115311X
85. Lai, C.H.: On Transformation Methods and the Induced Parallel Properties for the Temporal Domain. In: Substructing Techniques and Domain Decomposition Methods, Computational Science, Engineering & Technology Series, pp. 45–70 (2010). URL http://dx.doi.org/10.4203/csets.24.3
86. Langer, U., Matculevich, S., Repin, S.: Guaranteed error bounds and local indicators for adaptive solvers using stabilised space–time iga approximations to parabolic problems. Computers & Mathematics with Applications (2019)
87. Lecouvez, M., Falgout, R.D., Woodward, C.S., Top, P.: A parallel multigrid reduction in time method for power systems. In: 2016 IEEE Power and Energy Society General Meeting (PESGM), pp. 1–5 (2016). DOI 10.1109/PESGM.2016.7741520. URL https://dx.doi.org/10.1109/PESGM.2016.7741520
88. Legoll, F., Lelièvre, T., Samaey, G.: A Micro-Macro Parareal Algorithm: Application to Singularly Perturbed Ordinary Differential Equations. SIAM Journal on Scientific Computing **35**(4), A1951–A1986 (2013). URL http://dx.doi.org/10.1137/120872681

89. Lelarasmee, E., Ruehli, A.E., Sangiovanni-Vincentelli, A.L.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE CAD **1**(3) (1982)
90. Lindelöf, E.: Sur l'application des méthodes d'approximations succesives à l'étude des intégrales réelles des équations différentielles ordinaires. Journal de matheématiques pures et appliquées **10**(4), 117–128 (1894)
91. Lions, J.L., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. Comptes Rendus de l'Académie des Sciences - Series I - Mathematics **332**, 661–668 (2001). URL http://dx.doi.org/10.1016/S0764-4442(00)01793-6
92. Liu, J., Jiang, Y.L.: A parareal algorithm based on waveform relaxation. Mathematics and Computers in Simulation **82**(11), 2167–2181 (2012). URL http://dx.doi.org/10.1016/j.matcom.2012.05.017
93. Liu, Y., Hu, J.: Modified propagators of parareal in time algorithm and application to Princeton Ocean model. Int. J. for Numerical Methods in Fluids **57**(12), 1793–1804 (2008). URL http://dx.doi.org/10.1002/fld.1703
94. Livne, O.E.: Coarsening by compatible relaxation. Numer. Linear Algebra Appl. **11**(2-3), 205–227 (2004)
95. Lorin, E.: Domain decomposition method for the n-body time-independent and time-dependent schrödinger equations. Numerical Algorithms **81**(2), 655–694 (2019). DOI 10.1007/s11075-018-0566-3. URL https://doi.org/10.1007/s11075-018-0566-3
96. Maday, Y., Rønquist, E.M.: Parallelization in time through tensor-product space-time solvers. Comptes Rendus Mathematique **346**(1–2), 113–118 (2008). URL http://dx.doi.org/10.1016/j.crma.2007.09.012
97. Manteuffel, T., Ruge, J., Southworth, B.: Nonsymmetric algebraic multigrid based on local approximate ideal restriction. SIAM Journal on Scientific Computing **40**(6), A4105–A4130 (2018)
98. Mercerat, D., Guillot, L., Vilotte, J.P.: Application of the parareal algorithm for acoustic wave propagation. In: AIP Conference Proceedings, vol. 1168, pp. 1521–1524 (2009). URL http://dx.doi.org/10.1063/1.3241388
99. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Comm. App. Math. and Comp. Sci. **5**(2), 265–301 (2010)
100. Minion, M.L., Speck, R., Bolten, M., Emmett, M., Ruprecht, D.: Interweaving PFASST and parallel multigrid. SIAM Journal on Scientific Computing **37**, S244 – S263 (2015). URL http://dx.doi.org/10.1137/14097536X
101. Miranker, W.L., Liniger, W.: Parallel methods for the numerical integration of ordinary differential equations. Mathematics of Computation **21**(99), 303–320 (1967). URL http://dx.doi.org/10.1090/S0025-5718-1967-0223106-8
102. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. Commun. ACM **7**(12), 731–733 (1964). URL http://dx.doi.org/10.1145/355588.365137
103. Ong, B., Kwok, F., High, S.: Pipeline schwarz waveform relaxation. In: Methods in Science and Engineering XXII, Lecture Notes in Computational Science and Engineering. Spring–Verlag (2015)
104. Ong, B.W., Haynes, R.D., Ladd, K.: Algorithm 965: RIDC methods: A family of parallel time integrators. ACM Trans. Math. Softw. **43**(1), 8:1–8:13 (2016). DOI 10.1145/2964377. URL http://dx.doi.org/10.1145/2964377
105. Ong, B.W., Mandal, B.C.: Pipeline implementations of Neumann–Neumann and Dirichlet–Neumann waveform relaxation methods. Numerical Algorithms **78**(1), 1–20 (2018). DOI 10.1007/s11075-017-0364-3. URL https://doi.org/10.1007/s11075-017-0364-3
106. Ong, B.W., Melfi, A., Christlieb, A.J.: Parallel Semi-Implicit Time Integrators (2012). URL http://arxiv.org/abs/1209.4297. ArXiv:1209.4297 [cs.DC]
107. Peddle, A.G., Haut, T., Wingate, B.: Parareal convergence for oscillatory PDElowercases with finite time-scale separation (2019). DOI 10.1137/17m1131611. URL https://doi.org/10.1137%2F17m1131611
108. Picard, E.: Remarques sur les équations différentielles. Acta Math. **17**(1), 297–300 (1893). DOI 10.1007/BF02391996. URL https://doi.org/10.1007/BF02391996. Extrait d'une lettre adressée à M. Mittag-Leffler
109. Quaife, B., Biros, G.: Adaptive time stepping for vesicle suspensions. Journal of Computational Physics **306**, 478 – 499 (2016). DOI https://doi.org/10.1016/j.jcp.2015.11.050. URL http://www.sciencedirect.com/science/article/pii/S0021999115007901
110. Quarteroni, A.: Mathematical models in science and engineering. Notices Amer. Math. Soc. **56**(1), 10–19 (2009)
111. Reynolds-Barredo, J.M., Newman, D.E., Sánchez, R.S., Berry, L.A.: Modelling parareal convergence in 2D drift wave plasma turbulence. In: High Performance Computing and Simulation (HPCS), 2012 International Conference on, pp. 726–727 (2012). URL http://dx.doi.org/10.1109/HPCSim.2012.6267004
112. Reynolds-Barredo, J.M., Newman, D.E., Sánchez, R.S., Samaddar, D., Berry, L.A., Elwasif, W.R.: Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations. Journal of Computational Physics **231**(23), 7851–7867 (2012). URL http://dx.doi.org/10.1016/j.jcp.2012.07.028
113. Samaddar, D., Newman, D.E., Sánchez, R.S.: Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. Journal of Computational Physics **229**, 6558–6573 (2010). URL http://dx.doi.org/10.1016/j.jcp.2010.05.012
114. Samaey, G., Slawig, T.: A micro/macro parallel-in-time (parareal) algorithm applied to a climate model with discontinuous non-monotone coefficients and oscillatory forcing. arXiv:1806.04442 [math.NA] (2018). URL https://arxiv.org/abs/1806.04442
115. Schmitt, A., Schreiber, M., Peixoto, P., Schäfer, M.: A numerical study of a semi-lagrangian parareal method applied to the viscous burgers equation. Computing and Visualization in Science **19**(1), 45 – 57 (2018). DOI 10.1007/s00791-018-0294-1. URL https://doi.org/10.1007/s00791-018-0294-1
116. Schreiber, M., Loft, R.: A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. Numerical Linear Algebra with Applications **26**(2), e2220 (2019). DOI 10.1002/nla.2220. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2220. E2220 nla.2220
117. Schreiber, M., Peixoto, P.S., Haut, T., Wingate, B.: Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. The

International Journal of High Performance Computing Applications **32**(6), 913–933 (2018). DOI 10.1177/1094342016687625. URL https://doi.org/10.1177/1094342016687625

118. Schreiber, M., Schaeffer, N., Loft, R.: Exponential integrators with parallel-in-time rational approximations for shallow-water equations on the rotating sphere. Parallel Computing (2019). DOI 10.1016/j.parco.2019.01.005. URL https://dx.doi.org/10.1016/j.parco.2019.01.005
119. Schroder, J.B., Falgout, R.D., Woodward, C.S., Top, P., Lecouvez, M.: Parallel-in-time solution of power systems with scheduled events. In: 2018 IEEE Power & Energy Society General Meeting (PESGM), pp. 1–5. IEEE (2018)
120. Sheen, D., Sloan, I.H., Thomée, V.: A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature. Math. Comp. **69**(229), 177–195 (2000). DOI 10.1090/S0025-5718-99-01098-4. URL https://doi.org/10.1090/S0025-5718-99-01098-4
121. Sheen, D., Sloan, I.H., Thomée, V.: A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. IMA J. Numer. Anal. **23**(2), 269–299 (2003). DOI 10.1093/imanum/23.2.269. URL https://doi.org/10.1093/imanum/23.2.269
122. Sommeijer, B.: Parallel-iterated Runge-Kutta methods for stiff ordinary differential equations. Journal of Computational and Applied Mathematics **45**(1), 151–168 (1993). URL http://dx.doi.org/10.1016/0377-0427(93)90271-C
123. Southworth, B.S.: Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. SIAM Journal on Matrix Analysis and Applications **40**(2), 564–608 (2019)
124. Speck, R.: Parallelizing spectral deferred corrections across the method. Computing and Visualization in Science (2018). DOI 10.1007/s00791-018-0298-x. URL https://doi.org/10.1007/s00791-018-0298-x
125. Speck, R., Ruprecht, D.: Toward fault-tolerant parallel-in-time integration with PFASST. Parallel Computing **62**, 20 – 37 (2017). DOI 10.1016/j.parco.2016.12.001. URL http://dx.doi.org/10.1016/j.parco.2016.12.001
126. Speck, R., Ruprecht, D., Emmett, M., Minion, M.L., Bolten, M., Krause, R.: A multi-level spectral deferred correction method. BIT Numerical Mathematics **55**, 843–867 (2015). URL http://dx.doi.org/10.1007/s10543-014-0517-x
127. Speck, R., Ruprecht, D., Krause, R., Emmett, M., Minion, M.L., Winkel, M., Gibbon, P.: A massively space-time parallel N-body solver. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '12, pp. 92:1–92:11. IEEE Computer Society Press, Los Alamitos, CA, USA (2012). URL http://dx.doi.org/10.1109/SC.2012.6
128. Speck, R., Ruprecht, D., Krause, R., Emmett, M., Minion, M.L., Winkel, M., Gibbon, P.: Integrating an N-body problem with SDC and PFASST. In: Domain Decomposition Methods in Science and Engineering XXI, *Lecture Notes in Computational Science and Engineering*, vol. 98, pp. 637–645. Springer International Publishing (2014). URL http://dx.doi.org/10.1007/978-3-319-05789-7_61
129. Steinbach, O., Yang, H.: Comparison of algebraic multigrid methods for an adaptive space–time finite-element discretization of the heat equation in 3d and 4d. Numerical Linear Algebra with Applications **25**(3), e2143 (2018)
130. Steinbach, O., Yang, H.: Comparison of algebraic multigrid methods for an adaptive space-time finite-element discretization of the heat equation in 3D and 4D. Numerical Linear Algebra with Applications **25**(3), e2143–17 (2018)
131. Sterck, H.D., Falgout, R.D., Friedhoff, S., Krzysik, O.A., MacLachlan, S.P.: Optimizing mgrit and parareal coarse-grid operators for linear advection (2019)
132. Thomée, V.: A high order parallel method for time discretization of parabolic type equations based on Laplace transformation and quadrature. Int. J. Numer. Anal. Model. **2**(1), 85–96 (2005)
133. Trottenberg, U., Oosterlee, C., Schüller, A.: Multigrid. Academic Press, London, UK (2001)
134. Uddin, B.: Revisionist integral deferred correction methods with application to the moving method of lines. Master's thesis, Memorial University (2018)
135. Van Der Houwen, P.J., Sommeijer, B.P.: Parallel iteration of high-order Runge-Kutta methods with stepsize control. Journal of Computational and Applied Mathematics **29**(1), 111 – 127 (1990). URL http://dx.doi.org/10.1016/0377-0427(90)90200-J
136. Vandewalle, S., Horton, G.: Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. Computing **54**(4), 317–330 (1995)
137. Vandewalle, S., Piessens, R.: Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. SIAM J. Sci. Statist. Comput. **13**(6), 1330–1346 (1992)
138. Vandewalle, S., Van de Velde, E.: Space-time concurrent multigrid waveform relaxation. Annals of Numerical Mathematics **1**(1-4), 347–360 (1994). URL http://dx.doi.org/10.13140/2.1.1146.1761
139. Vaněk, P., Mandel, J., Brezina, M.: Algebraic multigrid based on smoothed aggregation for second and fourth order problems. Computing **56**, 179–196 (1996)
140. Vanka, S.P.: Block-implicit multigrid solution of navier-stokes equations in primitive variables. Journal of Computational Physics **65**(1), 138–158 (1986)
141. Weinzierl, T., Köppl, T.: A geometric space-time multigrid algorithm for the heat equation. Numer. Math. Theory Methods Appl. **5**(1), 110–130 (2012)
142. Winkel, M., Speck, R., Hübner, H., Arnold, L., Krause, R., Gibbon, P.: A massively parallel, multi-disciplinary barnes–hut tree code for extreme-scale n-body simulations. Computer physics communications **183**(4), 880–889 (2012)
143. XBraid: Parallel multigrid in time. https://github.com/XBraid/xbraid/
144. Yue, X., Shu, S., Xu, X., Bu, W., Pan, K.: Parallel-in-time with fully finite element multigrid for 2-d space-fractional diffusion equations. arXiv preprint arXiv:1805.06688 (2018)