LA-UR-21-26105
Accepted Manuscript

Asynchronous Truncated Multigrid-Reduction-in-Time

Hahne, Jens
Southworth, Benjamin Scott
Friedhoff, Stephanie

To be published in: SIAM Journal on Scientific Computing

DOI to publisher's version: 10.1137/21M1433149

Permalink to record:
https://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-21-26105

ASYNCHRONOUS TRUNCATED MULTIGRID-REDUCTION-IN-TIME (AT-MGRIT)[*]

JENS HAHNE[†], BEN S. SOUTHWORTH [‡], AND STEPHANIE FRIEDHOFF [†]

Abstract. In this paper, we present the new "asynchronous truncated multigrid-reduction-in-time" (AT-MGRIT) algorithm for introducing time parallelism to the solution of discretized time-dependent problems. The new algorithm is based on the multigrid-reduction-in-time (MGRIT) approach, which, in certain settings, is equivalent to another common multilevel parallel-in-time method, Parareal. In contrast to Parareal and MGRIT that both consider a global temporal grid over the entire time interval on the coarsest level, the AT-MGRIT algorithm uses truncated local time grids on the coarsest level, each grid covering certain temporal subintervals. These local grids can be solved completely in an independent way from each other, which reduces the sequential part of the algorithm and, thus, increases parallelism in the method. Here, we study the effect of using truncated local coarse grids on the convergence of the algorithm, both theoretically and numerically, and show, using challenging nonlinear problems, that the new algorithm consistently outperforms classical Parareal/MGRIT in terms of time to solution.

Key words. Parallel-in-time integration, Parareal, MGRIT, truncated coarsest grids

AMS subject classifications. 65F10, 65M22, 65M55

1. Introduction.

Time-dependent problems are classically solved by a time-stepping procedure that propagates the solution stepwise forward in time. The method is optimal, i. e., of order $\mathcal{O}(N_t)$ for N_t time steps. However, this method quickly becomes a parallel bottleneck when using modern computer architectures, which have an increasing number of processors, yet stagnating processor clock speed. Due to the sequential nature of classical time stepping, parallelization is limited to the spatial domain, and, as the number of processors grows, spatial parallelization becomes exhausted even if more resources are available. Parallel-in-time methods use these resources of modern computer architectures to compute multiple time steps simultaneously, enabling spatial *and* temporal parallelization.

The development of the first parallel-in-time method goes back over 50 years [30], and an overview of the field can be found in [13]. Two of the best known methods are the Parareal method [24] and the multigrid-reduction-in-time (MGRIT) algorithm [11], both of which are based on multigrid reduction principles [33] applied in the time dimension. Parareal can be interpreted as a two-level multigrid method, and MGRIT generalizes the approach to a multilevel setting. The ideas of both methods are similar, and both methods are equivalent in certain settings. On the "fine" level(s), time integration is simultaneously (i. e., in parallel) applied to non-overlapping temporal subdomains, and on the coarsest level, the entire time interval is solved with sequential time stepping. The choice of the number of levels and the choice of the coarsest grid is both critical and challenging. The typical choice of the coarse grid in the two-level setting is based on the number of processes, choosing as many points on the coarse grid as there are processes available [24]. With this strategy, the fine level can be perfectly parallelized, but for a large number of processes, the

[*] Submitted to the editors DATE.

Funding: This work is supported by the BMBF (project PASIROM; grant 05M18PXB). BSS was supported as a Nicholas C. Metropolis Fellow under the Laboratory Directed Research and Development program of Los Alamos National Laboratory.

[†] Department of Mathematics, Bergische Universität Wuppertal, Germany (jens.hahne@math.uni-wuppertal.de, friedhoff@math.uni-wuppertal.de).

[‡] Theoretical Division, Los Alamos National Laboratory, USA (southworth@lanl.gov).

serial work on the coarsest level dominates the runtime.

Strategies to reduce the runtime of two-level schemes include variants of the Parareal algorithm, such as asynchronous Parareal [39, 26], a modified version enhanced by the asynchronous iterative scheme [6], or an adaptive Parareal algorithm, which increases the accuracy of the fine solver over the Parareal iterations. Using more than two grid levels can significantly reduce the serial work by using a coarsest grid with only a few time points, but the resulting very large time steps can be very expensive, if not infeasible, to compute for some applications [4] and/or may affect the convergence of the algorithm [8].

MGRIT and Parareal are primarily effective on parabolic-type problems [31, 37], which have a naturally dissipative behavior over long time intervals. Here, we make the observation that, due to the dissipative behavior inherent to these problems, the coarsest grid probably does *not* need to represent the full time domain. Indeed, the solution at time $t = 0$ will often have a negligible effect on the solution at much later times. Thus, in many cases we believe that computing a global coarse grid introduces an unnecessary sequential computational effort to an otherwise parallel algorithm.

In this paper, we introduce a new way to define the coarsest level in Parareal and MGRIT, emphasizing reducing the serial work while avoiding large time steps. Instead of solving the entire time interval serially on the coarsest grid, we define multiple *independent local* coarse grids each consisting of k coarse-grid time points that can be propagated independently and simultaneously. This idea was originally motivated by similar processor-local multigrid hierarchies used in geometric and algebraic multigrid for elliptic problems [2, 27, 28]. Such an approach offers both improved parallelism and reduced computational cost compared with a global coarse-grid solve, while still providing sufficient coarse-level information to each processor for rapid convergence of the global problem. Due to the asynchronous nature of computing the truncated coarsest grids, we refer to the new algorithm as “asynchronous truncated MGRIT” (AT-MGRIT).

Section 2 introduces the algorithm in a two-level and multilevel context, providing an FAS interpretation of the multilevel variant in Algorithm 2.2. In Section 3, we analyze the new algorithm theoretically, derive two-level error propagators, and present two-level convergence bounds in Subsection 3.2. We then describe various properties of the algorithm in Section 4, including describing the implementation with associated communication scheme in Subsection 4.1 and performing a parameter study for a model problem in Subsection 4.3. Finally, we apply the new algorithm to two challenging nonlinear problems, a chemical reaction in Subsection 5.1 and the simulation of a realistic model of an electrical machine in Subsection 5.2. AT-MGRIT consistently offers a 5–30% reduction in wallclock time compared with traditional MGRIT and Parareal, and we expect the speedup to be greater if the algorithms were applied on GPUs.

2. An overlapping and asynchronous coarse grid.

Consider an initial value problem of the form

$$\mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)), \quad \mathbf{u}(t_0) = \mathbf{g}_0, \quad t \in (t_0, t_f]. \tag{2.1}$$

We discretize (2.1) on a uniformly-spaced temporal grid $t_i = i\Delta t,\ i = 0, 1, \ldots, N_t$, with constant step size $\Delta t = (t_f - t_0)/N_t$, and let $\mathbf{u}_i \approx \mathbf{u}(t_i)$ for $i = 0, \ldots, N_t$ with $\mathbf{u}_0 = \mathbf{u}(0)$. A general form of a single step time integration method for the time-discrete initial value problem is

$$\mathbf{u}_i = \mathbf{\Phi}_i(\mathbf{u}_{i-1}) + \mathbf{g}_i, \quad i = 1, 2, \ldots, N_t, \tag{2.2}$$

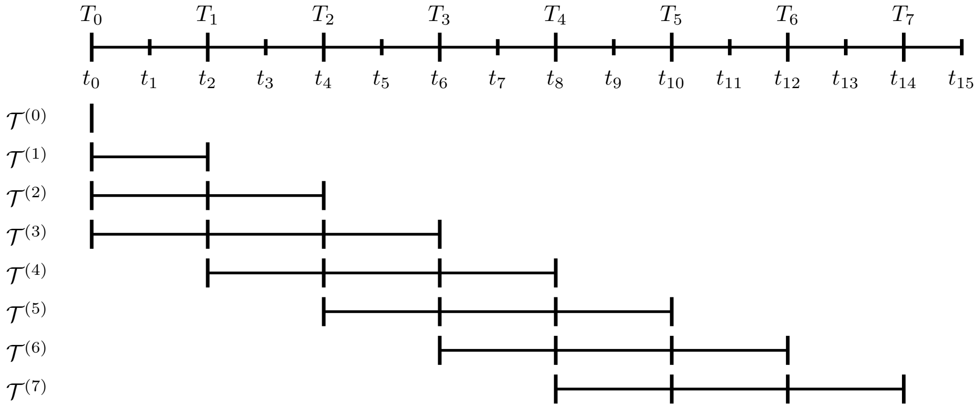


Fig. 1: Two-level temporal grid-hierarchy example for the AT-MGRIT algorithm with $N_t = 15$, $m = 2$ and $k = 4$. The C-points (long markers) define the global coarse grid. For each point $p = 0, \ldots, 7$ on the global coarse grid, a local coarse grid $\mathcal{T}^{(p)}$ is created.

where $\mathbf{\Phi}_i$ is a one-step time integrator, propagating a solution $\mathbf{u}_{i-1}$ from a time point t_{i-1} to time point t_i, and $\mathbf{g}_i$ contains forcing terms. Equation (2.2) can be written as a semi-linear matrix equation

$$A(\mathbf{u}) \equiv \begin{bmatrix} I & & & \\ -\mathbf{\Phi}_1(\cdot) & I & & \\ & \ddots & \ddots & \\ & & -\mathbf{\Phi}_{N_t}(\cdot) & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N_t} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{N_t} \end{bmatrix} \equiv \mathbf{g},$$

where $\mathbf{\Phi}_i(\cdot)$ indicates that $\mathbf{\Phi}_i$ is nonlinearly evaluated at the corresponding (block) vector entry. This system can be solved by a (linear) sequential block forward solve.

In contrast, the iterative AT-MGRIT algorithm solves the problem by updating multiple time points simultaneously. In the following, we first introduce the idea of the algorithm in Subsection 2.1 for the two-level case and explain how the algorithm works. We then discuss how the two-level method can be extended to a multilevel setting.

2.1. Two-level AT-MGRIT algorithm. For a given time grid $t_i = i\Delta t, i = 0, 1, ..., N_t$, and a given coarsening factor $m > 1$, we define a splitting of all time-points into F- and C-points, such that every m-th point is a C-point (note, non-uniform coarsening is also possible; uniform coarsening is used here to simplify presentation). This defines a global coarse grid of C-points $T_i = i\Delta T, i = 0, 1, ..., N_T$, with time step $\Delta T = m\Delta t$; all other non-C-points are F-points. Based on this global coarse grid, we define $N_T + 1$ overlapping *local* coarse grids. Given local grid size k, the pth local coarse grid, $\mathcal{T}^{(p)}$ for $p = 0, ..., N_T$, is given by

$$\mathcal{T}^{(p)} = \{i\Delta T : i \in [\max(0, p - k + 1), p]\},$$

with time step size $\Delta T = m\Delta t$, as depicted in Figure 1.

The two-level AT-MGRIT algorithm uses this time-grid hierarchy to solve time-dependent problems of the form (2.2) and is based on the following procedure: Given an initial solution $\mathbf{u}$ and the right-hand side $\mathbf{g}$, the first step of the algorithm applies a block relaxation, the so-called F-relaxation, to the fine space-time system of equations $A\mathbf{u} = \mathbf{g}$. The F-relaxation propagates the solution from a C-point to all following F-points preceding the next C-point (analogously to standard MGRIT/Parareal [11]). The relaxation of each interval of F-points can be executed in parallel and consists of $m - 1$ sequential applications of the time integrator. In the next step, the global residual vector $\mathbf{r}$ is computed and restricted by injection ($R_I^{(p)}$) to all local coarse grids. For each local coarse grid, the coarse system $A_c^{(p)}\mathbf{u}_c^{(p)} = \mathbf{r}_c^{(p)}$ is solved, which consists of $k - 1$ sequential applications of the coarse time integrator. Since the coarse-grid problems are independent of each other, they can be solved simultaneously. Then, the global solution vector is corrected using “selective ideal” interpolation, $P_S^{(p)}$. The selective ideal interpolation is the transpose of an injection followed by an F-relaxation starting from exactly one point in time. More precisely, the approximation of the solution at the last time point of each local coarse grid is interpolated to the fine grid and then, an F-relaxation is performed using these interpolated points. The steps are applied iteratively until a desired quality of the solution is achieved. The two-level AT-MGRIT algorithm is summarized in Algorithm 2.1.

Algorithm 2.1 AT-MGRIT $(A, \mathbf{u}, \mathbf{g})$

```
1: repeat
2:     Apply $F$-relaxation to $A\mathbf{u} = \mathbf{g}$
3:     Compute residual $\mathbf{r} = \mathbf{g} - A\mathbf{u}$
4:     For $p = 0$ to $N_T$:
5:         Restrict residual, $\mathbf{r}_c^{(p)} = R_I^{(p)}\mathbf{r}$
6:         Solve local system $A_c^{(p)}\mathbf{u}_c^{(p)} = \mathbf{r}_c^{(p)}$
7:         Correct using $\mathbf{u} = \mathbf{u} + P_S^{(p)}\mathbf{u}_c^{(p)}$
8: until stopping criterion is reached
```

We follow the typical MGRIT notation here and therefore specify the F-relaxation in line 2. From the second iteration on, this F-relaxation can be skipped, since the updates are already performed as part of the selective ideal interpolation of the previous iteration. Note, that the AT-MGRIT algorithm solves for the exact solution in N_T iterations if $k > 1$. Furthermore, the algorithm is equivalent to the Parareal method if $k = N_T + 1$, i. e., if all local coarse grids contain all C-points before in time. All components of the AT-MGRIT algorithm are highly parallel. The only communication needed is for the residual computation and the distribution of the residual (performed by the matrix-vector product $\mathbf{r}_c^{(p)} = R_I^{(p)}\mathbf{r}$ in Algorithm 2.1). Moreover, the coarse-level solve is communication-free (except for any communication that arises in spatial parallelism). This is particularly relevant for emerging heterogeneous computing architectures, where communication to and from GPU nodes can be quite expensive, and high efficiency is obtained with a low communication to computation ratio. For the coarse time integrator Φ_{i_c}, here we choose a re-discretization of the problem with step size ΔT, but other choices such as coarsening in space [34, 25, 20] or order of discretization [29, 12] can also be used.

2.2. Multilevel FAS AT-MGRIT algorithm. The two-level AT-MGRIT algorithm can easily be extended to the multilevel setting and the full approximation storage (FAS) framework [5] can be used to solve both linear and nonlinear problems. Analogously to MGRIT, a multilevel hierarchy of temporal grids is constructed recur-

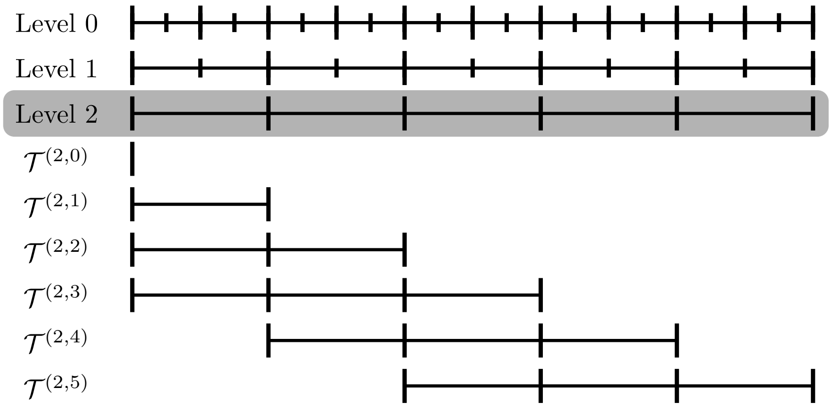


Fig. 2: Example of a three-level time grid hierarchy for the AT-MGRIT algorithm for a fine grid with 21 time points, $m = 2$ and $k = 4$. At the coarsest level, a local coarse grid is generated for each C-point of the global coarse grid (gray box). These local grids ($\mathcal{T}^{(2,p)}, p = 0, \dots, 5$) replace the global coarse grid used in the classical MGRIT algorithm.

sively using a uniform or non-uniform coarsening strategy. AT-MGRIT uses the same levels, coarsening, relaxation, and transfer operators as MGRIT on all finer levels in the hierarchy, but the coarsest MGRIT grid is replaced by local grids. Figure 2 shows an example grid hierarchy for three-level AT-MGRIT with $N_t = 20$, $m = 2$, and $k = 4$. While MGRIT utilizes the global coarse grid on level 2, AT-MGRIT uses local grids $\mathcal{T}^{(2,p)}, p = 0, \dots, 5$.

In the following, we assume that all problem-dependent forcing terms are included in the time integrator. Then, the multilevel FAS AT-MGRIT V-cycle algorithm is given in Algorithm 2.2, where $N_t^{(\ell)}$ denotes the number of time points, and $\mathcal{A}^{(\ell)}\mathbf{u}^{(\ell)} = \mathbf{g}^{(\ell)}$ and $\mathcal{A}^{(\ell,p)}\mathbf{u}^{(\ell,p)} = \mathbf{g}^{(\ell,p)}$ specifies the space-time system of equations on levels $\ell = 0, 1, \dots, L-1$ and on the local coarse grids $p = 0, 1, \dots, N_t^{(\ell)}$, respectively. On all except for the coarsest level, we use restriction by injection ($R_I^{(\ell)}$), "ideal" interpolation ($P^{(\ell)}$), and $F(CF)^\nu$-relaxation. For more details on MGRIT (and thus AT-MGRIT on finer levels), see [11]. At the coarsest level, restriction and interpolation to and from the local coarse grids is done by injection, denoted by $R_I^{(\ell,p)}$ and $P_I^{(\ell,p)}$, respectively. Note that at the coarsest level, the residual is first transferred to the global coarse grid and then to the local coarse grids, allowing for a nicer notation of the algorithm. AT-MGRIT can also be used with other common MGRIT cycle types, such as F-cycles [38] or nested iterations [23, 22]. While F-cycles visit the coarsest level several times per iteration, nested iterations compute an improved initial guess by starting on the coarsest level and interpolating the solution to the finer levels, applying one V-cycle per level. For all cycle types, the standard MGRIT coarsest level can be replaced by local coarse grids. Analogous to the two-level setting, AT-MGRIT is equivalent to MGRIT if $k = N_t^{(L-1)} + 1$.

3. Theory.

This section develops convergence theory for AT-MGRIT in the linear two-level setting. The analysis is built on two-level MGRIT/Parareal theory developed in [9, 36], and gives insight on the effects of truncating the coarse-grid time grid.

Algorithm 2.2 AT-MGRIT FAS(ℓ)

```
repeat
    if $\ell$ is the coarsest level:
        For $p = 0$ to $N_t^{(\ell)}$:
            Restrict to local grids
                $\mathbf{v}^{(\ell,p)} = R_I^{(\ell,p)}(\mathbf{v}^{(\ell)})$
                $\mathbf{g}^{(\ell,p)} = R_I^{(\ell,p)}(\mathbf{g}^{(\ell)})$
            Solve local problem $\mathcal{A}^{(l,p)}(\mathbf{u}^{(\ell,p)}) = \mathcal{A}^{(\ell,p)}(\mathbf{v}^{(\ell,p)}) + \mathbf{g}^{(\ell,p)}$
            Update $\mathbf{u}^{(\ell)} = P_I^{(\ell,p)}\mathbf{u}^{(\ell,p)}$
    else
        Apply $F$-relaxation to $\mathcal{A}^{(\ell)}(\mathbf{u}^{(\ell)}) = \mathbf{g}^{(\ell)}$
        For 0 to $\nu$:
            Apply $CF$-relaxation to $\mathcal{A}^{(\ell)}(\mathbf{u}^{(\ell)}) = \mathbf{g}^{(\ell)}$
        Inject the approximation and its residual to the coarse grid
            $\mathbf{u}^{(l+1)} = R_I^{(\ell)}(\mathbf{u}^{(\ell)})$
            $\mathbf{v}^{(l+1)} = \mathbf{u}^{(l+1)}$
            $\mathbf{g}^{(l+1)} = R_I^{(\ell)}(\mathbf{g}^{(\ell)} - \mathcal{A}^{(\ell)}\mathbf{u}^{(\ell)})$
        Solve on next level: AT-MGRIT $(\ell + 1)$
        Compute the error approximation: $\mathbf{e} = \mathbf{u}^{(\ell+1)} - \mathbf{v}^{(\ell+1)}$
        Correct using ideal interpolation: $\mathbf{u}^{(\ell)} = \mathbf{u}^{(\ell)} + P^{(\ell)}(\mathbf{e})$
until stopping criterion is reached
```

We begin by introducing the error-propagation operator in the case of exact solves on a truncated coarse grid (Subsection 3.1.1) and inexact coarse-grid solves (Subsection 3.1.2). Formal two-level convergence bounds are provided in Subsection 3.2. Because we are in the two-level setting, we drop ℓ superscripts from Subsection 2.2.

3.1. Error propagation. Following from [11], the two-level error propagation operator for linear AT-MGRIT with an exact coarse-grid solve is given by:

$$\mathcal{E} := \left(I - \sum_{p=0}^{N_T} P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A \right) P R_I, \tag{3.1}$$

where $A_c^{(p)}$ represents the local coarse grid systems, $R_I^{(p)}$ is the restriction operator to the local coarse grids, and $P_S^{(p)}$ defines the interpolation from the local coarse grids that updates the fine grid using *selective ideal interpolation*, i. e., for one specific C-point, this C-point and the following interval of F-points are updated. We see that (3.1) is analogous to that derived in [11, Eq. 2.12], but here we must sum over C-points, as each C-point is updated by a unique local coarse-grid. The operators P and R_I, corresponding to ideal interpolation and restriction by injection, respectively, are given by

$$P := I \otimes \left[\begin{array}{c} I \\ \Phi \\ \vdots \\ \Phi^{m-1} \end{array}\right] \left.\vphantom{\begin{array}{c} I \\ \Phi \\ \vdots \\ \Phi^{m-1} \end{array}}\right\}{\scriptstyle m}, \qquad P_S^{(p)} := \begin{array}{c} \overbrace{\hphantom{\quad\quad\quad}}^{\min(p,k-1)}\hphantom{\quad\quad} \\ \left[\begin{array}{ccc} & & \\ & & \\ & & I \\ & & \Phi \\ & & \vdots \\ & & \Phi^{m-1} \\ & & \\ & & \end{array}\right] \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c} \\ \end{array}}\right\}{\scriptstyle pm} \\ \left.\vphantom{\begin{array}{c} I \\ \Phi \\ \vdots \\ \Phi^{m-1} \end{array}}\right\}{\scriptstyle m} \\ \left.\vphantom{\begin{array}{c} \\ \end{array}}\right\}{\scriptstyle N_t+1-(p+1)m} \end{array}.$$

$$R_I := \left[\begin{array}{ccccccccc} I & & & & & & & & \\ & \mathbf{0} & \cdots & \mathbf{0} & I & & & & \\ & & & & & \ddots & & & \\ & & & & & & \mathbf{0} & \cdots & \mathbf{0} \; I \end{array}\right] \begin{array}{l} \}{\scriptstyle 1} \\ \left.\vphantom{\begin{array}{c} \\ \\ \end{array}}\right\}{\scriptstyle N_T} \end{array}, \qquad \text{with } \overbrace{\mathbf{0} \cdots \mathbf{0} \; I}^{m}$$

Note, that the operator PR_I is equivalent to error propagation for F-relaxation [11]. Recall that the fine-grid operator has block dimension $(N_t + 1) \times (N_t + 1)$, with each block being a square operator the size of Φ. Letting $N_t = mN_T$ for coarse-grid points $0, ..., N_T$, the fine-grid size can be written as $(mN_T + 1) \times (mN_T + 1)$, which we will use to express error propagation largely in terms of $m \times m$ coarse blocks. Each of these blocks represents a block of one C-point and $m - 1$ following F-points. At the end, there is a single block containing only one C-point. Note that the structure for this block is always a submatrix of the $m \times m$ blocks, containing only the part corresponding to the C-point.

3.1.1. Exact local coarse grid solve. First, we consider the effect of the local coarse grids using exact solves on the coarse time steps. For this purpose, we define the local coarse-grid problem as

$$A_c^{(p)} := R_I^{(p)} A P^{(p)},$$

where $P^{(p)}$ and $R_I^{(p)}$ define the transfer between the fine grid and the local coarse grids and are submatrices of P and R_I. For $P^{(p)}$, only columns of P associated to points lying on this local coarse grid are considered. Equivalently, only the associated rows are considered for the restriction. Then, the coarse-grid problems are given by

$$R_I^{(p)} A P^{(p)} = \overbrace{\left[\begin{array}{ccccc} I & & & & \\ -\Phi^m & I & & & \\ & -\Phi^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi^m & I \end{array}\right]}^{\min(p+1,k)} \left.\vphantom{\begin{array}{c} I \\ I \\ I \\ \ddots \\ I \end{array}}\right\}{\scriptstyle \min(p+1,k)}.$$

Here, it is important to note that all local coarse-grid systems $R_I^{(p)} A P^{(p)}$ have the same structure, but consider different time intervals. In fact, the exact local coarse-grid systems are principal submatrices of the Schur complement corresponding to a standard Parareal/MGRIT coarse-grid with exact solves [11].

We can now examine the error-propagator $\mathcal{E}_e$ using exact solves on the local coarse grids. We refer to Appendix A for detailed algebraic derivations. In forming $\mathcal{E}_e$ by summing over $p = 1, \ldots, N_T$, we obtain a block lower triangular matrix, whereby each

p updates m rows of $\mathcal{E}$, and the error-propagator using ideal local coarse grids can be written in block form as

(3.2)

$$\mathcal{E}_e = \begin{array}{c} \begin{array}{ccc} \overbrace{\quad}^{m} & \overbrace{\qquad\qquad\qquad}^{m(N_T-2)} & \overbrace{\quad}^{m+1} \end{array} \\ \left[\begin{array}{ccccccc} \mathbf{0} & \cdots & & & & & \mathbf{0} \\ \vdots & \cdots & & & & & \vdots \\ \mathbf{0} & \cdots & & & & & \mathbf{0} \\ \mathcal{G} & \mathbf{0} & \cdots & & & & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathcal{G} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \bar{\mathcal{G}} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] \end{array} \begin{array}{l} \}\, m \\ \left.\begin{array}{c} \\ \\ \end{array}\right\}\, (k-1)m \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\}\, (N_T-k)m \\ \}\, 1 \end{array}, \qquad \mathcal{G} = \begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad}^{m} \\ \left[\begin{array}{cccc} \Phi^{km} & \mathbf{0} & \cdots & \mathbf{0} \\ \Phi^{km+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^{km+m-1} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] \end{array} \Bigg\}\, m\,, \qquad \bar{\mathcal{G}} = \begin{array}{c} \overbrace{\qquad\qquad\qquad}^{m} \\ \left[\begin{array}{cccc} \Phi^{km} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] \end{array} \Big\}\, 1\,,$$

where $\bar{\mathcal{G}}$ is identical to the first row of $\mathcal{G}$ and represents the additional block consisting of one C-point. Note that the error propagator $\mathcal{E}_e$ is nonzero, so unlike Parareal/two-level MGRIT, AT-MGRIT using exact local coarse-grid inverses is not a direct method. For all $p > k$, we have some error perturbation that results from truncating the exact (Schur-complement) coarse grid.

3.1.2. Approximate local coarse grid solve.

As typical for multigrid reduction techniques, we do not invert $R_I^{(p)} A P^{(p)}$ exactly, but approximate $R_I^{(p)} A P^{(p)} \approx \widetilde{A}_c^{(p)}$. Specifically, we approximate the powers Φ^m, which correspond to m applications of the fine time integrator Φ, with a coarse operator Ψ. This results in the approximation $\widetilde{A}_c^{(p)}$ given by

$$\widetilde{A}_c^{(p)} := \begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad}^{\min(p+1,k)} \\ \left[\begin{array}{cccc} I & & & \\ -\Psi & I & & \\ & \ddots & \ddots & \\ & & -\Psi & I \end{array}\right] \end{array} \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\}\, \min(p+1,k)\,.$$

Using this approximation, we can formulate the error-propagation $\mathcal{E}_a$ using the approximated local coarse-grid inverse. Again, we refer to Appendix B for derivations. The error-propagation operator with approximate coarse grid, $\mathcal{E}_a$, is then given by

$$(3.3)\qquad \mathcal{E}_a = \left[\begin{array}{cccccccc|l}
m & \multicolumn{6}{c}{m(N_T-1)} & 1 & \\
\mathbf{0} & \cdots & & & & & & \mathbf{0} & m \\
\mathcal{Z}_0 & \mathbf{0} & \cdots & & & & & \mathbf{0} & (k-1)m \\
\vdots & \ddots & \ddots & \cdots & & & & \mathbf{0} & \\
\mathcal{Z}_{k-2} & \cdots & \mathcal{Z}_0 & \mathbf{0} & \cdots & & & \mathbf{0} & \\
\mathcal{W} & \mathcal{Z}_{k-2} & \cdots & \mathcal{Z}_0 & \mathbf{0} & \cdots & & \mathbf{0} & (N_T-k)m \\
\mathbf{0} & \ddots & \ddots & \vdots & \ddots & \ddots & & \vdots & \\
\mathbf{0} & \ddots & \mathcal{W} & \mathcal{Z}_{k-2} & \cdots & \mathcal{Z}_0 & \mathbf{0} & \mathbf{0} & \\
\mathbf{0} & \cdots & \mathbf{0} & \bar{\mathcal{W}} & \bar{\mathcal{Z}}_{k-2} & \cdots & \bar{\mathcal{Z}}_0 & \mathbf{0} & 1
\end{array}\right],$$

with block matrices $\mathcal{Z}_j$ and $\mathcal{W}$ given by

$$(3.4)\qquad \mathcal{Z}_j = \left[\begin{array}{cccc|l}
\multicolumn{4}{c}{m} & \\
\Phi^0\Psi^j(\Phi^m-\Psi) & \mathbf{0} & \cdots & \mathbf{0} & m \\
\vdots & \vdots & \vdots & \vdots & \\
\Phi^{m-1}\Psi^j(\Phi^m-\Psi) & \mathbf{0} & \cdots & \mathbf{0} &
\end{array}\right] \quad \mathcal{W} = \left[\begin{array}{cccc|l}
\multicolumn{4}{c}{m} & \\
\Phi^0\Psi^{k-1}\Phi^m & \mathbf{0} & \cdots & \mathbf{0} & m \\
\vdots & \vdots & \vdots & \vdots & \\
\Phi^{m-1}\Psi^{k-1}\Phi^m & \mathbf{0} & \cdots & \mathbf{0} &
\end{array}\right],$$

and $\bar{\mathcal{W}}$ and $\bar{\mathcal{Z}}_j$ are again identical to first row of $\mathcal{W}$ and $\mathcal{Z}_j$, respectively.

3.2. Convergence bounds.

To avoid multiple subscripts, we omit the subscript a of $\mathcal{E}_a$ from (3.3) in the sequel. Using f and c subscripts to denote F- and C-points, respectively, $\mathcal{E}$ (3.3) can be reordered and partitioned into 2×2 block form. Moreover, notice that F-points columns of $\mathcal{Z}_j$ and $\mathcal{W}$ (3.4) (and therefore also of $\mathcal{E}_a$ (3.3)) are all zero. If we then consider powers of the matrix, which correspond to several iterations, we get

$$\mathcal{E}^\ell := \begin{bmatrix} \mathcal{E}_{ff} & \mathcal{E}_{fc} \\ \mathcal{E}_{cf} & \mathcal{E}_{cc} \end{bmatrix}^\ell = \begin{bmatrix} \mathbf{0} & \mathcal{E}_{fc} \\ \mathbf{0} & \mathcal{E}_{cc} \end{bmatrix}^\ell = \begin{bmatrix} \mathbf{0} & \mathcal{E}_{fc}\mathcal{E}_{cc}^{\ell-1} \\ \mathbf{0} & \mathcal{E}_{cc}^{\ell} \end{bmatrix}.$$

It follows from above that for multiple iterations, convergence is fully defined by $\mathcal{E}_{cc} \in \mathbb{R}^{N_T+1\times N_T+1}$, that is, $\mathcal{E}^\ell$ will be convergent in some norm, that is, $\|\mathcal{E}^\ell\| < 1$, if and only if $\mathcal{E}_{cc}^\ell$ is as well. To that end, we consider analyzing the C-C principle submatrix of (3.3),

$$(3.5)\qquad \mathcal{E}_{cc} =$$

$$\begin{bmatrix}
\mathbf{0} & & & & & & \\
(\Phi^m-\Psi) & \mathbf{0} & & & & & \\
\vdots & (\Phi^m-\Psi) & \ddots & & & & \\
\Psi^{k-2}(\Phi^m-\Psi) & \vdots & \ddots & \mathbf{0} & & & \\
\Psi^{k-1}\Phi^m & \Psi^{k-2}(\Phi^m-\Psi) & \dots & (\Phi^m-\Psi) & \mathbf{0} & & \\
 & \ddots & \ddots & \vdots & \ddots & \ddots & \\
 & & \Psi^{k-1}\Phi^m & \Psi^{k-2}(\Phi^m-\Psi) & \dots & (\Phi^m-\Psi) & \mathbf{0}
\end{bmatrix}.$$

Now consider the case of Φ and Ψ being simultaneously diagonalizable, as would occur if the same (diagonalizable) spatial matrix is used on the fine and coarse grid. Let U denote the shared eigenvector matrix of Φ and Ψ, with eigenvalues $\mu \in \sigma(\Psi)$ and $\lambda \in \sigma(\Phi)$, where $\sigma(\Psi)$ and $\sigma(\Phi)$ denote the spectrum of Ψ and Φ, respectively. Following the frameworks developed in [9, 36], let $\widetilde{U}$ denote a block-diagonal matrix, with diagonal blocks given by eigenvectors U. Then,

$$\|\mathcal{E}_{cc}\|_{(\widetilde{U}\widetilde{U}^*)^{-1}} = \max_{\{\mu,\lambda\}} \|\widetilde{\mathcal{E}}_{cc}\|,$$

where $\|\cdot\|$ here corresponds to the ℓ^2-norm, and $\widetilde{\mathcal{E}}_{cc}$ is defined as follows for a fixed pair of eigenvalues $\{\mu, \lambda\}$:

$$\widetilde{\mathcal{E}}_{cc} \coloneqq$$

$$\begin{bmatrix}
\mathbf{0} & & & & & & \\
(\lambda^m - \mu) & \mathbf{0} & & & & & \\
\vdots & (\lambda^m - \mu) & \ddots & & & & \\
\mu^{k-2}(\lambda^m - \mu) & \vdots & \ddots & \mathbf{0} & & & \\
\mu^{k-1}\lambda^m & \mu^{k-2}(\lambda^m - \mu) & \dots & (\lambda^m - \mu) & \mathbf{0} & & \\
 & \ddots & \ddots & \vdots & \ddots & \ddots & \\
 & & \mu^{k-1}\lambda^m & \mu^{k-2}(\lambda^m - \mu) & \dots & (\lambda^m - \mu) & \mathbf{0}
\end{bmatrix}.$$

If the spatial matrix is normal, then $(\widetilde{U}\widetilde{U}^*)^{-1} = I$. In general, bounding $\widetilde{\mathcal{E}}_{cc}$ in the ℓ^2-norm for each eigenvalue pair guarantees convergence of $\mathcal{E}_{cc}$ in a certain eigenvector induced norm, where "convergence" corresponds to a guaranteed reduction in error every iteration (in contrast to, e. g., nilpotency, where convergence is eventually guaranteed, but error could in principle diverge significantly for many iterations before sudden convergence to the exact solution).

Recall the inequality $\|\widetilde{\mathcal{E}}_{cc}\|^2 \leq \|\widetilde{\mathcal{E}}_{cc}\|_1 \|\widetilde{\mathcal{E}}_{cc}\|_\infty$. Given that $\widetilde{\mathcal{E}}_{cc}$ is Toeplitz, the maximum row and column sums are equal, yielding the bound

$$\begin{aligned}
\|\widetilde{\mathcal{E}}_{cc}\| \leq \|\widetilde{\mathcal{E}}_{cc}\|_1 &= |\lambda^m - \mu| \sum_{\ell=0}^{k-2} |\mu^\ell| + |\lambda^m \mu^{k-1}| \\
&= \frac{|\lambda^m - \mu|(1 - |\mu|^{k-1})}{1 - |\mu|} + |\lambda^m \mu^{k-1}|. \qquad (3.6)
\end{aligned}$$

Results are summarized in the following theorem.

Theorem 3.1 (Two-level convergence). *Let Φ and Ψ be simultaneously diagonalizable with eigenvectors U, and consider two-level AT-MGRIT with coarsening factor m and local coarse-grid size k. For a given CF-splitting of time points, error propagation takes the form*

$$\mathcal{E}^\ell \coloneqq \begin{bmatrix} \mathcal{E}_{ff} & \mathcal{E}_{fc} \\ \mathcal{E}_{cf} & \mathcal{E}_{cc} \end{bmatrix}^\ell = \begin{bmatrix} \mathbf{0} & \mathcal{E}_{fc}\mathcal{E}_{cc}^{\ell-1} \\ \mathbf{0} & \mathcal{E}_{cc}^\ell \end{bmatrix}.$$

Further, assume Φ and Ψ are stable in an eigenvalue sense, that is, $|\mu|, |\lambda| < 1$ for all $\mu \in \sigma(\Psi)$, $\lambda \in \sigma(\Phi)$, and define

$$\varphi_{tg}(\mu, \lambda) := \frac{|\lambda^m - \mu|}{1 - |\mu|}$$

for eigenvalue pairs (with shared eigenvector) $\{\mu, \lambda\}$. Then,

$$\|\mathcal{E}_{cc}\|_{(\widetilde{U}\widetilde{U}^*)^{-1}} \leq \max_{\{\mu,\lambda\}} \left(\varphi_{tg}(\mu, \lambda) + |\mu^{k-1}| \left(|\lambda^m| - \varphi_{tg}(\mu, \lambda)\right)\right). \tag{3.7}$$

Proof. The proof follows from a simple expansion of (3.6). □

Corollary 3.2. *Under the same assumptions as Theorem 3.1,*

$$\|\mathcal{E}_{cc}\|_{(\widetilde{U}\widetilde{U}^*)^{-1}} \leq \max_{\{\mu,\lambda\}} \left(\varphi_{tg}(\mu, \lambda) + |\mu^k| \left(1 - \varphi_{tg}(\mu, \lambda)\right)\right). \tag{3.8}$$

Proof. Note that

$$\frac{|\lambda^m - \mu|(1 - |\mu|^{k-1})}{1 - |\mu|} = \frac{|\lambda^m - \mu|(1 - |\mu|^k)}{1 - |\mu|} - |\mu^{k-1}||\lambda^m - \mu|$$

In addition, we have $|\lambda^m \mu^{k-1}| = |\mu^{k-1}(\lambda^m - \mu) + \mu^k| \leq |\mu^{k-1}||\lambda^m - \mu| + |\mu^k|$. Plugging these two results into (3.6), yields an upper bound

$$\|\mathcal{E}_{cc}\|_{(\widetilde{U}\widetilde{U}^*)^{-1}} \leq \max_{\{\mu,\lambda\}} \left(\frac{|\lambda^m - \mu|(1 - |\mu|^k)}{1 - |\mu|} + |\mu^k|\right).$$

An analogous expansion as used in Theorem 3.1 completes the proof. □

Note that the first term, φ_{tg}, in (3.7) and (3.8), to $\mathcal{O}(1/N_T)$, provides necessary and sufficient conditions for convergence of two-level Parareal and MGRIT [9, 36, 37], while the second term introduces an error perturbation that results from truncating the coarse grid. Although Corollary 3.2 is less tight than Theorem 3.1, it provides a more intuitive description of convergence. Note that error modes which converge fast for traditional Parareal/MGRIT, $\varphi_{tg}(\mu, \lambda) \approx 0$, lead to the largest perturbation of convergence for AT-MGRIT , $\approx |\mu^k|$ (this also suggests convergence will be better for a more "diffusive" coarse solver, that is, a coarse solver with generally smaller eigenvalues). In contrast, there will be much less degradation in convergence for modes that are relatively slow to converge for traditional Parareal/MGRIT.

This leads to a further important observation on convergence of AT-MGRIT : with some algebra,[1] one can show that the "error" subdiagonal, that is, the subdiagonal of $\widetilde{\mathcal{E}}_{cc}$ that lacks a $\lambda^m - \mu$ scaling, is propagated out of the matrix after $\lceil (N_T + 1)/k \rceil$ iterations (i. e., all matrix entries then have at least one power of $\lambda^m - \mu$). This suggests a natural heuristic to choose k:

Choice of k: *choose k at least large enough so that $\lceil (N_T + 1)/k \rceil$ approximates the number of iterations to converge for traditional two-level Parareal.*

[1] There are various ways to show this; perhaps the most formal is in noting that multiplication of Toeplitz matrices such as $\widetilde{\mathcal{E}}_{cc}^{\ell}$ corresponds to finite discrete convolutions. One can also simply expand $\widetilde{\mathcal{E}}_{cc}^{p} = (\widehat{\mathcal{E}}_{cc} + \mu^{k-1}\lambda^m I_{-k})^p$, where I_{-k} is a diagonal of ones on the kth subdiagonal.

The number of iterations for Parareal convergence is defined by the slowest converging modes, which are in turn the least affected in convergence of AT-MGRIT by the perturbation term in (3.7) and (3.8) (i. e., we expect these modes to converge in a roughly similar number of iterations for AT-MGRIT). As mentioned previously, the fastest-converging modes for Parareal/MGRIT, however, can suffer significant degradation of convergence in AT-MGRIT. Thus, we choose k so that the error perturbation for these terms is eliminated via nilpotency at (approximately) the same number of iterations as the slowest converging modes for standard Paraeal/MGRIT will have converged. After this nilpotency is achieved, these “fast” modes will rapidly converge (if they have not already) due to $\varphi_{tg}(\mu, \lambda) \ll 1$. These observations can be seen in practice in Figure 6(b), where at almost exactly $\lceil (N_T + 1)/k \rceil$ iterations, the residual drops precipitously, decreasing by as much as four orders of magnitude on the following iteration (see Figures 6(a) and 6(b) and surrounding text for a more detailed discussion).

Last, it is important to remember that theory developed in this section is for two-level AT-MGRIT applied to linear problems. The resulting heuristic for choosing k provides a good starting point, but multilevel AT-MGRIT or application to nonlinear problems as considered in Section 5 may require larger k (e. g., see Table 4).

4. Algorithmic properties.

This section examines nuances of the AT-MGRIT algorithm, including a communication scheme for distributing residuals on the coarsest level (Subsection 4.1), the implicit propagation of initial conditions across the time domain (Subsection 4.2), and a study on the new parameter k (Subsection 4.3).

4.1. Implementation.

We have implemented Algorithm 2.2 in parallel as part of the `PyMGRIT` package [19, 4] framework. When applying the algorithm in parallel, we assume that at the coarsest level at most one C-point of the global grid lies on one process. In principle this is not necessary, but ensures that the solves of the local problems can be perfectly parallelized. To solve the local problems, the fine(r)-level residuals must be distributed. Let $p_0, p_1, ..., p_{N_T}$, where N_T is equal to the number of points on the coarsest global coarse grid, be all processes containing a C-point on the coarsest grid. Then, we define two groups of MPI communicators. The first decomposition divides all processes based on the local grid size k into $\lceil (N_T + 1)/k \rceil$ groups, where the first group consists of processes $p_0, p_1, ..., p_{k-1}$, the second consists of processes $p_k, p_{k+1}, ..., p_{2k-1}$, and so on. The second decomposition divides the processes $p_{k-1}, ..., p_{N_T}$ into groups of size k, so that the first group consists of processes $p_{k-1}, ..., p_{2k-2}$, the next of processes $p_{2k-1}, ..., p_{3k-1}$, and so on. Then, the distribution of residuals is given by a communication within all groups of the first decomposition, followed by a communication within all groups of the second decomposition. Note that the groups of a decomposition do not overlap, allowing for parallel communication within each group. Figure 3 shows an example of the residual communication for a two-level AT-MGRIT algorithm with a global coarse grid with $N_T = 5$, along with a description of the communication stages.

4.2. Propagation of the initial condition.

A key feature of the AT-MGRIT algorithm is the implicit propagation of the initial condition through the iterations of the method, which allows for using local coarse grids that do not include the initial time point. The idea of implicit propagation of the initial condition is best explained in a two-level example with F-relaxation. Figure 4 shows an example of the distribution of local coarse grids for $N_t = 20$, $m = 7$ and $k = 3$. Only the first three local coarse grids have direct access to the initial condition and, thus, the initial

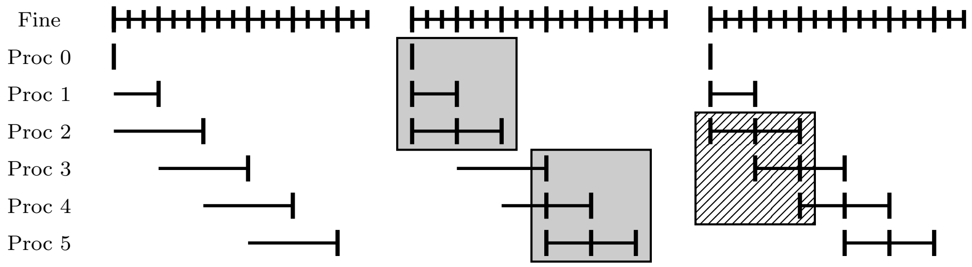
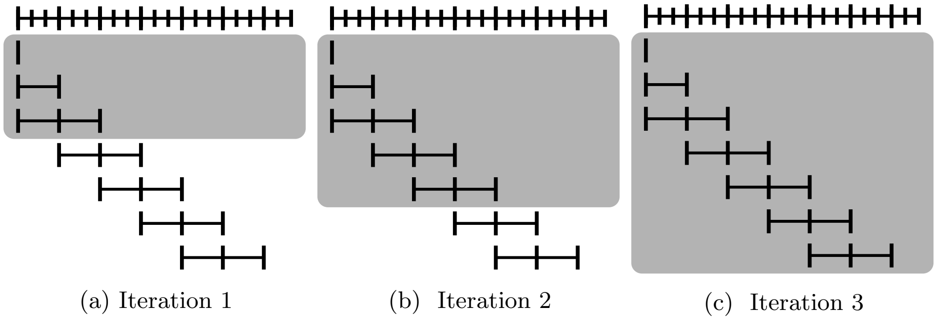


Fig. 3: Illustration of the communication scheme for $P = 6$ processes and local grid size $k = 3$. First (left), each process computes the residual of its rightmost C-point. In a second step (middle), the residuals are distributed in parallel within the groups of the first decomposition (gray boxes). Last (right), the residuals are distributed within the groups of the second decomposition (shaded boxes), after which each process has all the required residuals.


(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

Fig. 4: Implicit propagation (gray box) of the initial condition for a two-level AT-MGRIT variant with F-relaxation and $N_t = 20$, $m = 3$ and $k = 3$.

condition is distributed over the first three local coarse grids in the first iteration. All other local coarse grids do not have access to the initial condition at this point. In the second iteration, again only the first three local coarse grids directly contain the initial condition. However, the next two local coarse grids now contain C-points of local coarse grids, which directly depend on the initial condition from the previous iteration, making them implicitly depend on the initial condition as well. In the next iteration, the next two local coarse grids implicitly depend on the initial condition, and so on. In the end, the two-level AT-MGRIT algorithm with F-relaxation requires $\lceil (N_T + 1)/k \rceil$ iterations until all local coarse grids depend implicitly on the initial condition. Note that in the two-level variant with FCF-relaxation, the initial value is also propagated to the next C-point on the fine grid due to the additional CF-relaxation and, thus, the initial condition is propagated faster. Also note that the propagation in the multi-level case is even faster than in the two-level case, since the initial condition is additionally propagated by the relaxations on the intermediate level(s).

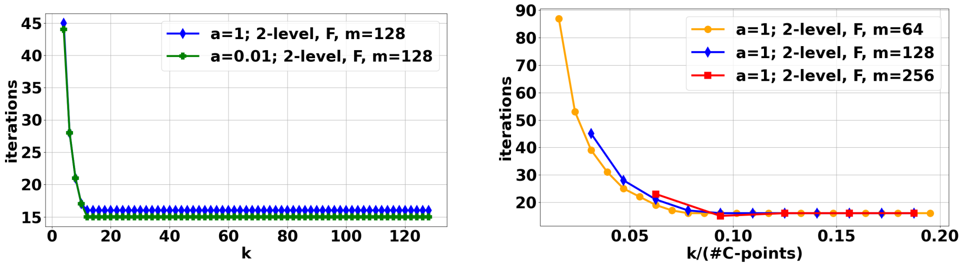
4.3. Size of local coarse grids. The parameter k defines the size of the local coarse grids and, thus, the number of sequential solves needed on the coarse grid. In the following, we consider the influence of the parameter k on the convergence of AT-MGRIT applied to a standard model problem for parallel-in-time methods, the one-dimensional heat equation,

$$u_t - a u_{xx} = b(x,t) \quad \text{in} \quad [0,3] \times [0,\pi], \tag{4.1}$$

where a is the thermal conductivity, and subject to the initial condition $u(x,0) = \sin(\pi x)$ and homogeneous Dirichlet boundary conditions in space. The forcing term is choosen as $b(x,t) = -\sin(\pi x)(\sin(t) - \pi^2 \cos(t))$, such that the exact solution is given by $u(x,t) = \sin(\pi x)\cos(t)$ for $a = 1$. We first examine the behavior of k for $a = 1$ and $a = 0.01$, and then choose $a = 1$ for more detailed results.

We discretize (4.1) using second-order central finite differences with 1,025 degrees of freedom in space and on an equidistant time grid with 16,384 time points using backward Euler. We investigate the behavior of the AT-MGRIT algorithm for the two-level case, and choose different coarsening factors m and local grid sizes k. We restrict ourselves to the two-level case, since we want to study the effect of using local coarse grids of various sizes k. For all simulations, the stopping tolerance is based on the discrete 2-norm of the absolute space-time residual with a tolerance of 10^{-7} and a random initial guess is chosen for all time points except for the initial condition. This choice guarantees that no knowledge of the right-hand side is used that could affect the convergence. Note that this is only a good choice for investigating the behavior of the algorithm and is not recommended in practice.

Figure 5(a) shows the required number of iterations to reach the stopping criterion for a two-level AT-MGRIT variant with F-relaxation and a coarsening factor of $m = 128$ as a function of size k for the two choices of the thermal conductivity. Note that while the variant with $k = 128$, which is equivalent to Parareal, performs 127 sequential time steps on the coarse level, equivalent convergence can be obtained with $k = 12$, 10× less coarse-grid solves. Note, that the behavior is similar for both choices of a. Figure 5(b) presents iterations to convergence as a function of *the ratio* of local to global coarse-grid points. For three different coarsening factors, we see that convergence does not improve beyond the same ratio of $k/(\#C\text{-points})$, in this case about 0.08. Although this parameter is likely problem specific, Figure 5(b) does suggest the choice of k is relatively agnostic to that of the coarsening factor by posing it relative to the global coarse-grid size.

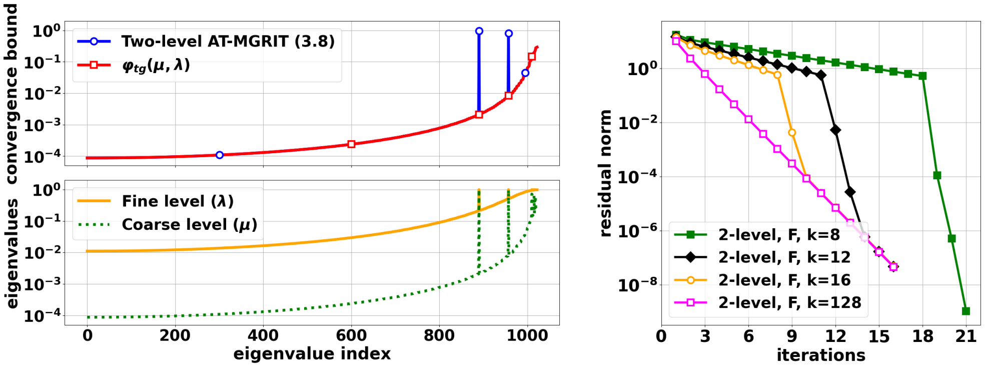


(a) [[New image]] Iterations to convergence as a function of k.

(b) [[New image]] Iterations to convergence as a function of k divided by the number of C-points.

Fig. 5: Required iterations for AT-MGRIT variants for the 1D heat equation.

To better understand the convergence behavior described in Subsection 3.2, for each spatial eigenvalue Figure 6(a) plots the theoretical convergence rate (3.8), the (asymptotic in N_T) two-grid rate φ_{tg}, and the corresponding eigenvalues for the coarse- and fine-propagator for two-level AT-MGRIT with $m = 128$ and $k = 12$. Notice that on an eigenvector basis, theoretical convergence of AT-MGRIT almost exactly matches that of Parareal, with (in this case) two exceptions, given by the blue dots with convergence ≈ 1. Each of these spatial eigenmodes correspond with a coarse-propagator eigenvalue $|\mu| \approx 1$ and $\varphi_{tg} \ll 1$, which (see Corollary 3.2) leads to a significant degradation in convergence (in a single-iteration sense). Figure 6(b) plots *observed* convergence behavior for two-level AT-MGRIT with coarsening factor $m = 128$ and various choices for k. The variant with $k = 128$ (i. e., Parareal) has uniform convergence behavior, while convergence for smaller k is split into three parts. Initially, convergence is much slower than for $k = 128$, due to the spatial eigenmodes with $|\mu^k| \approx 1$ and $\varphi_{tg} \ll 1$ discussed above. The smaller k is chosen, the slower is the convergence, since the convergence perturbation of $|\mu^k|(1 - \varphi_{tg})$ in Corollary 3.2 decreases with increased k. However, after almost exactly $(N_T + 1)/k$ iterations (see Theorem 3.1 and surrounding discussion), once the initial condition has been implicitly propagated over all local coarse grids, these problematic modes are eliminated via nilpotency and a drastic improvement in convergence can be seen for all three variants; e. g., for $k = 8$, one iteration suddenly reduces the residual norm almost four orders of magnitude. This rapid convergence lasts until the residual norm matches that of Parareal, and convergence rates thereafter follow Parareal. Comparing the theoretical and numerical results for $m = 12$, the theoretical bound (see Figure 6(a)) is given by 0.94987 and the maximum numerical convergence factor between two iterations for the equivalent setting (see Figure 6(b)) is given by 0.7418. Note, observed convergence is better than the theoretical bound due to many modes being rapidly attenuated. Only a few modes are very slow to converge, with rates likely close to the theoretical bound, but these modes degrade the average (across all error modes) convergence rate.



(a) Theoretical bound on convergence rate based on (3.8), φ_{tg}, and eigenvalues sorted by φ_{tg}, for $m = 128$ and $k = 12$.

(b) Residual norm as a function of iteration for two-level AT-MGRIT with coarsening factor $m = 128$.

Fig. 6: Results of AT-MGRIT variants in terms of theoretical error bound (3.8) and residual norm for the heat equation with $a = 1$.

5. Parallel results.

In this section, we present numerical results for AT-MGRIT applied to two challenging, nonlinear time-dependent problems: the 2D Gray Scott example of a chemical reaction of two substances, and a realistic model of an electrical machine. In addition, we apply different two- and three-level variants of AT-MGRIT and compare runtimes and iteration counts with the corresponding variants of Parareal and MGRIT, respectively. For the two-level variants, we apply F-relaxation and we choose the coarsening factor such that the number of coarse-grid points is equal to the number of processes used for the simulation enabling perfect parallelization on the fine level. For the three-level algorithms, we apply non-uniform coarsening strategies with a coarsening factor of $m_0 = 64$ between the fine and the intermediate level, and different factors between the intermediate and the coarse level. All simulations were performed on an Intel Xeon Phi Cluster consisting of four 1.4 GHz Intel Xeon Phi processors. The code for all experiments can be found in the `PyMGRIT` repository [18], and this package is also used for simulations with Parareal and MGRIT. For all experiments, we use all possible resources for temporal parallelization, i. e., we do not use spatial parallelization (largely due to limited resources). For a brief discussion on the effect of spatial parallelism for the different algorithms, see Appendix C.

5.1. Gray Scott.

We consider the 2D Gray-Scott problem [32] of a chemical reaction of two components $\mathcal{U}$ and $\mathcal{V}$, given by

$$\begin{aligned} u_t &= D_u \Delta u - uv^2 + F(1-u), \\ v_t &= D_v \Delta v + uv^2 - (K+F)u, \end{aligned}$$

where $u = u(x, y, t)$ and $v = v(x, y, t)$ are the concentration of $\mathcal{U}$ and $\mathcal{V}$, respectively, D_u and D_v are the diffusion rates, F is the feed rate, and K is the removal rate. For our simulations, we choose the spatial domain $[0, 2.5]^2$ with periodic boundary conditions, and the time interval $[0, 256]$. Further, we choose the parameters $F = 0.024$, $K = 0.06$, $D_u = 8 \times 10^{-5}$, and $D_v = 4 \times 10^{-5}$, and we consider the initial value

$$\begin{aligned} u(x, y, 0) &= 1 - 2(0.25 \sin(4\pi x)^2 \sin(4\pi y)^2), \qquad && (x, y) \in [1, 1.5]^2 \\ v(x, y, 0) &= 0.25 \sin(4\pi x)^2 \sin(4\pi y)^2, && (x, y) \in [1, 1.5]^2, \end{aligned}$$

and $u(x, y, 0) = 1$ and $v(x, y, 0) = 0$ otherwise. The problem is discretized using central finite differences with 128^2 points in space and on an equidistant time grid with 16,384 points using backward Euler. We solve the resulting nonlinear problem using Newton's method of `PETSc` [1] with a relative and absolute tolerance of 10^{-10}.

We apply two-level and three-level AT-MGRIT variants, and compare the two-level variants with Parareal variants and the three-level variants with MGRIT. Furthermore, we use nested iterations to compute an improved initial guess. In the nested iteration strategy, Parareal and MGRIT solve the global coarse-grid problem at the coarsest level, while AT-MGRIT uses the local coarse grids instead of the global grid. The stopping criterion for all variants is based on the discrete 2-norm of the space-time residual with a tolerance of 10^{-7}.

Table 1 shows the number of iterations and runtimes of the setup and solve phases of two-level AT-MGRIT and Parareal variants. The setup time consists of computing an improved initial guess and the solve time consists of applying the algorithm. The results show that iteration counts of AT-MGRIT are equal to iteration counts of Parareal with the same coarsening strategy. Furthermore, a finer coarse grid significantly reduces the number of iterations required. While 12 iterations are needed for the two-level variants with a coarse grid of only 32 points, this number is reduced to

Method	m	k	# Procs	# Iters	Setup time	Solve time	Speedup w.r.t Parareal
Parareal	512	-	32	12	1,338 s	172,068 s	-
	256	-	64	10	2,308 s	89,288 s	-
	128	-	128	9	3,958 s	66,485 s	-
	64	-	256	7	7,646 s	66,272 s	-
Two-level AT-MGRIT	512	16	32	12	701 s	165,351 s	1.04
	256	32	64	10	1,167 s	78,230 s	1.15
	128	64	128	9	2,022 s	48,675 s	1.39
	64	128	256	7	3,812 s	39,895 s	1.69

Table 1: Iteration counts, setup times (for computing an improved initial guess), and runtimes of the solve phase of two-level AT-MGRIT and Parareal variants applied to the 2D Gray-Scott problem for various numbers of processes.

seven iterations for the variants with 256 coarse-grid points. However, this reduction in iterations is accompanied by significantly more expensive sequential coarse-grid solves, reflected in increasing setup times with increasing points on the coarse grid. However, if the number of points on the coarse grid doubles, the setup time does not double. This is because a smaller time step requires fewer Newton iterations and, thus, affects the duration of the application of each time integration. The setup time of each AT-MGRIT variant is about half as long as that of the corresponding Parareal variant due to the choice of k. Looking at the runtimes of the solve phase, we see that AT-MGRIT is always faster than the corresponding Parareal variant, achieving a speedup of up to a factor of 1.69 compared to Parareal. Furthermore, we see that while the Parareal algorithm does not scale for more than 128 processes, since the serial part of the algorithm dominates the benefit of the additional parallelization of the fine level, the AT-MGRIT algorithm shows good parallel scaling up to 256 processes.

Table 2 presents similar results to Table 1 for four different three-level variants of AT-MGRIT and MGRIT with FCF-relaxation on 256 processes. The number of iterations here does not depend as much on the coarsest grid as in the two-level case, but we still see that the MGRIT variant with the coarsening strategy $(64, 2)$, i. e., the variant with the most points on the second level, requires the fewest iterations. The corresponding AT-MGRIT variant needs one additional iteration, but after the sixth iteration the stopping criterion is slightly missed. A minimal increase in k would likely eliminate this extra iteration. In terms of solve times, we see that all variants of the AT-MGRIT algorithm are faster than the corresponding MGRIT variants, even the variant that requires an additional iteration. Again, the more points on the coarsest level, the higher the speedup of AT-MGRIT over MGRIT. Note that for this problem, adding a coarser level to the three-level MGRIT variants does not guarantee further reduction of the runtime. Rather, adding a level may increase the runtime. For example, the four-level MGRIT with the coarsening strategy $(64, 8, 2)$, which adds

Method	m	k	# Iters	Setup time	Solve time	Speedup w.r.t MGRIT
MGRIT	(64,16)	-	7	3,525 s	43,604 s	-
	(64,8)	-	7	3,498 s	38,420 s	-
	(64,4)	-	7	4,980 s	42,285 s	-
	(64,2)	-	6	8,075 s	45,131 s	-
AT-MGRIT	(64,16)	8	7	2,864 s	41,054 s	1.07
	(64,8)	16	7	2,174 s	33,688 s	1.17
	(64,4)	32	7	2,713 s	34,063 s	1.29
	(64,2)	64	7	4,247 s	38,571 s	1.24

Table 2: Iteration counts and runtimes of the setup and solve phase on 256 processes of three-level AT-MGRIT and MGRIT variants with FCF-relaxation and different non-uniform coarsening strategies applied to the 2D Gray-Scott problem.

Method	m	k	# Iters	Setup time	Solve time	Speedup w.r.t MGRIT
AT-MGRIT	(64,8)	4	10	1,140 s	42,063 s	0.97
	(64,8)	6	9	1,357 s	38,978 s	1.04
	(64,8)	8	8	1,543 s	35,767 s	1.12
	(64,8)	10	8	1,716 s	36,509 s	1.10
	(64,8)	12	8	1,877 s	37,060 s	1.08
	(64,8)	14	8	2,047 s	37,786 s	1.05
	(64,8)	16	7	2,174 s	33,688 s	1.17

Table 3: Iteration counts and runtimes of the setup and solve phase on 256 processes of three-level AT-MGRIT with FCF-relaxation, coarsening factor $(64, 8)$, and different choices of k applied to the 2D Gray-Scott problem.

another level with a coarsening factor of two to the variant with the coarsening factor $(64, 8)$ from Table 2, requires eight iterations and the overall runtime (setup and solve) is 52,370 s.

Table 3 extends the results from Table 2 and shows the effect of different choices of k for the three-level AT-MGRIT with coarsening factor $(64, 8)$. We see that the number of iterations increases slightly as k decreases. Despite the increasing number of iterations, the runtime for all $k \geq 6$ is smaller than the runtime of the corresponding MGRIT variant from Table 2. For smaller k, the runtime is larger compared to the MGRIT variant because the cost of the additional iterations is more expensive than the cost reduction due to the local coarse grids.

Figure 7 shows the overall runtime for one AT-MGRIT variant (blue line) and the corresponding MGRIT variant (orange line) as a function of the number of processes

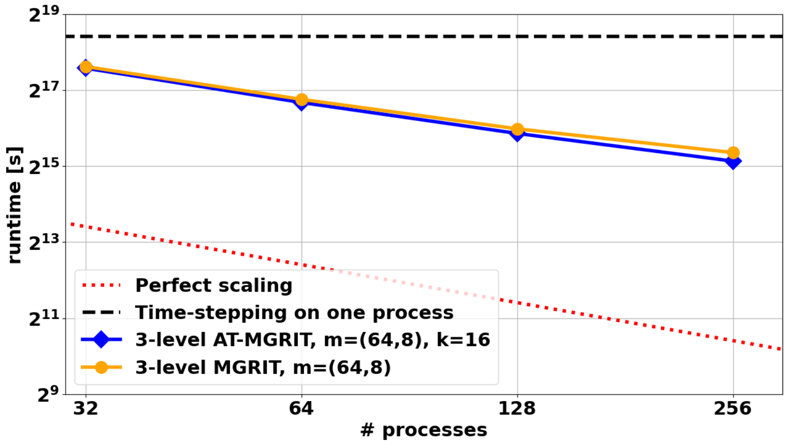


Fig. 7: Strong scaling results for one three-level AT-MGRIT variant, the corresponding MGRIT variant, and sequential time-stepping on one process applied to the 2D Gray-Scott problem. The red dotted line indicates the perfect scaling based on the runtime of time-stepping.

and the runtime of time-stepping using only one process (black dashed line) which is about four days. For reference, the red dotted line indicates the behavior of perfect scaling based on the runtime of time-stepping. While the runtime almost halves when using 32 and 64 processes, the scaling curve starts to flatten slightly with a higher number of processes. This is mainly because only the fine level computations have an additional benefit from more processes due to the chosen coarsening strategy, and the runtime of the coarser levels becomes more and more dominant. However, compared to the corresponding MGRIT variant, the AT-MGRIT variant scales better due to its reduced work at the coarsest level.

5.2. Induction machine. The standard approach for the simulation of electrical machines is based on neglecting the displacement current in Maxwell's equations [21]. The resulting magnetoquasistatic approximation, the so-called eddy current problem, is defined in terms of the unknown magnetic vector potential $\mathbf{A} : \Omega \times (t_0, t_f] \to \mathbb{R}^3$ as

$$\begin{aligned} \sigma\partial_t\mathbf{A} + \nabla\times(\nu(\cdot)\nabla\times\mathbf{A}) &= \mathbf{J}_s \text{ in } \Omega\times(t_0,t_f],\\ \mathbf{n}\times\mathbf{A} &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

with initial value $\mathbf{A}(\mathbf{x}, t_0) = \mathbf{A}_0(\mathbf{x})$, spatial domain Ω, consisting of the rotor, stator, and the air gap in between, and where $(t_0, t_f]$ is the time interval. The magnetic flux $\mathbf{B} = \nabla\times\mathbf{A}$ vanishes at the boundaries $\partial\Omega$ of the spatial domain (Dirichlet boundary condition). Three $(n_s = 3)$ distributed stranded conductors are modeled by the source current density $\mathbf{J}_s = \sum_{s=1}^{n_s} \chi_s i_s$, with winding functions $\chi_s : \Omega \to \mathbb{R}^3$ and currents $i_s : (t_0, t_f] \to \mathbb{R}^3$. An attached electrical network provides a connection between the so-called flux-linkage, i. e., the spatially integrated time derivative of the magnetic vector potential, and the voltage. The scalar electrical conductivity $\sigma(\mathbf{x}) \geq 0$ and the (nonlinear) magnetic reluctivity $\nu(\mathbf{x}, |\nabla\times\mathbf{A}|) > 0$ encode the geometry. To consider the rotation of the rotor, the problem is extended by an additional equation of motion; we refer to [4] for more details.

In the following, we consider a cross-section in the x, y-plane to reduce the spatial domain to a two-dimensional (2D) domain $\Omega_{2D} \subset \mathbb{R}^2$. Discretizing in space using finite elements with n_a degrees of freedom yields a system of differential-algebraic equations of the form

$$M\mathbf{u}'(t) + K(\mathbf{u}(t))\mathbf{u}(t) = \mathbf{f}(t), \quad t \in (t_0, t_f] \tag{5.4a}$$

$$\mathbf{u}(t_0) = \mathbf{u}_0, \tag{5.4b}$$

with unknown $\mathbf{u}^\top = [\mathbf{a}^\top, \mathbf{i}^\top, \theta, \omega] : (t_0, t_f] \to \mathbb{R}^n$. At each point in time $t \in (t_0, t_f]$, the solution $\mathbf{u}(t) \in \mathbb{R}^n$ consists of the magnetic vector potentials $\mathbf{a}\,(t) \in \mathbb{R}^{n_a}$, the currents of the three phases $\mathbf{i}\,(t) \in \mathbb{R}^3$, the rotor angle $\theta\,(t) \in \mathbb{R}$, and the angular velocity of the rotor $\omega\,(t) \in \mathbb{R}$. The given voltages $\mathbf{v}(t) \in \mathbb{R}^3$ and the mechanical excitation define the right-hand side $\mathbf{f}(t)$. Note, that (5.4) is a differential-algebraic equation of index-1 [3, 7] due to the presence of non-conducting materials, i. e., regions with $\sigma = 0$, in the domain, which can be treated by standard techniques in a parallel-in-time setting [35].

The multi-slice finite element model "im_3_kw" [17] of a four-pole squirrel cage induction machine is used for modeling the semi-discrete problem (5.4). A mesh representation with $n_a = 4449$ degrees of freedom is generated using Gmsh [15, 16]. Further, we choose $t_0 = 0$, $t_f = 0.2$ and a time grid with $N_t = 16{,}384$ points, which corresponds to a time-step size of $\delta t \approx 10^{-5}$. Note that this time interval is required approximately to reach the steady-state of the machine. For the simulations, only a quarter of the machine is considered with periodic boundary conditions and the `GetDP` library [10, 14], which implements the time integration using backward Euler, is used for the computations. At each time step, the `GetDP` library is called and a nonlinear problem is solved by applying Newton's method with damping. For the stopping criterion of the Newton solver, we choose a relative error of 10^{-6}. The machine is supplied by a three-phase sinusoidal voltage source, and, as proposed in [17], an initial ramp-up of the applied voltage is used for reducing the transient behavior of the motor for the time interval $[0, 0.04]$.

In the following, we present results for one Parareal variant and several two-level AT-MGRIT variants. For all experiments, we use an improved initial guess given by a global coarse-grid solve. Unfortunately, the use of too large time steps on the coarse level in the simulation of the electrical machine in the time-parallel setting causes at least one nonlinear solve within `GetDP` to fail to converge. Note that this also prevents the use of another, even coarser level. To overcome this problem, we apply subcycling at the coarse level, i. e., we apply three smaller steps per time step at the coarse level, reducing the time step size and improving the accuracy of the solution. For all algorithms, we use a convergence criterion based on the relative change in Joule losses, an important quantity of an electrical machine, at all C-points of two successive iterations; see [4] for details. The algorithm terminates when the maximum norm of the relative difference of two successive iterations is less than 1%. Note that this criterion does not guarantee convergence to the discrete time-stepping solution, but for each variant it has been verified that it does indeed iterate to the discrete time-stepping solution.

Table 4 shows the number of iterations, total runtimes, and the speedup compared to sequential time-stepping on one process for different AT-MGRIT variants and Parareal. Furthermore, the speedup compared to Parareal is shown for all AT-MGRIT variants. Comparing the number of iterations, the Parareal algorithm and AT-MGRIT with $k = 24$ both require five iterations to convergence. For $16 \geq k \geq 22$,

Method	k	Iterations	Total time (Setup + Solve)	Speedup w.r.t Parareal	Speedup w.r.t time-stepping on one process
Parareal	-	5	40,544 s	-	4.64
Two-level AT-MGRIT	12	8	39,480 s	1.03	4.77
	14	7	36,188 s	1.12	5.2
	16	6	32,710 s	1.24	5.75
	18	6	33,337 s	1.22	5.64
	20	6	33,996 s	1.19	5.53
	22	6	34,626 s	1.17	5.43
	24	5	30,582 s	1.33	6.15

Table 4: Iteration counts and total runtimes on 64 processes of Parareal and various two-level AT-MGRIT variants with a coarsening factor of 256 for the simulation of the induction machine.

six iterations are needed to reach the stopping criterion, and for $k = 14$ and $k = 16$, seven and eight iterations, respectively. Despite the increased number of iterations for some variants, the total runtime of all AT-MGRIT variants is smaller than that of Parareal, with the largest speedup of a factor of approximately 1.33. For this nonlinear problem, a slightly larger choice of k than the minimum choice proposed in Subsection 3.2 gives the best results in terms of runtime. Note that the time for the setup phase is the same for all variants and is about 2,891 s. Note also that both algorithms treat the fine level identically, and the improvement comes only from using local coarse grids instead of one global coarse grid. For comparison, the simulation time using serial time-stepping on one process is 188,123 s, which is more than two days. The fastest AT-MGRIT variant needs less than nine hours, which corresponds to a speedup of a factor of 6.15.

6. Conclusion. In this paper, we introduce the new AT-MGRIT algorithm, an iterative parallel-in-time algorithm for solving time-dependent problems. While the fine level(s) are treated as in the Parareal/MGRIT algorithm, the AT-MGRIT algorithm modifies the coarsest level computations. Instead of considering one global time grid that covers the entire time interval and is solved sequentially at the coarsest level, the AT-MGRIT algorithm uses a number of truncated, overlapping local coarse grids, one for each point on the global coarse grid. Each of these time grids is independent and covers only a fraction of the global time interval, allowing each problem to be solved simultaneously and reducing the serial work of the algorithm at the coarsest level.

Theoretical and numerical investigations of the algorithm show that the use of local coarse grids, which are not all connected to the initial value of the problem, introduces an additional error term compared to classical Parareal/MGRIT, which may affect the convergence at the beginning of the algorithm. However, the AT-MGRIT algorithm takes advantage of its iterative nature and eliminates this additional error term during several iterations, achieving convergence in the same number of iterations as with Parareal/MGRIT while significantly reducing the serial cost on the coarse level. Simulation of challenging nonlinear problems shows that the MGRIT algorithm can provide significant speedup compared to Parareal/MGRIT. Future work involves implementing and studying AT-MGRIT on GPUs and emerging shared-memory computing architectures, where the local and asynchronous aspect of coarse grid solves is likely to be particularly advantageous.

Appendix A. Error propagation for ideal local coarse problems. We consider error-propagation $\mathcal{E}_e$ for one C-point $p = 0, \dots, N_T$ using the ideal local coarse-grid problem (i. e., exact coarse grid and inverses). The structure of the matrices differs for the first k C-points from all other C-points, since the local coarse grids corresponding to the first k C-points contain all C-points prior in time. Here, we want to study the effect of local coarse grids that do not extend back to $t = 0$. Therefore, we start by considering all local coarse grids $N_T > p \geq k$ and subsequently discuss the structure for $p < k$. Note that the structure of the matrices of $p = N_T$ is always a submatrix of $N_T > p \geq k$ (see Section 3.1) and therefore is not explicitly stated. For $N_T > p \geq k$ the matrix $R_I^{(p)} A$ is given by

$$\text{(A.1)} \quad \begin{array}{c}
\begin{array}{cccc}
\overbrace{\hphantom{XXXX}}^{(p-k)m} & \overbrace{\hphantom{\mathbf{0}_{1\times(m-1)} \; -\Phi}}^{m} & \overbrace{\hphantom{I \;\; \mathbf{0}_{1\times(m-1)} \quad \mathbf{0}_{1\times(m-1)} \; -\Phi \quad I \;\; \mathbf{0}_{1\times(m-1)}}}^{km} & \overbrace{\hphantom{XXXX}}^{N_t+1-(p+1)m}
\end{array} \\
\left.\left[\begin{array}{c:c:c:c:c:c}
\hphantom{XXXX} & \mathbf{0}_{1\times(m-1)} \; -\Phi & I \;\; \mathbf{0}_{1\times(m-1)} & & & \hphantom{XXXX} \\ \hdashline
 & & \ddots & \ddots & & \\ \hdashline
 & & & \mathbf{0}_{1\times(m-1)} \; -\Phi & I \;\; \mathbf{0}_{1\times(m-1)} &
\end{array}\right]\right\}{\scriptstyle k}
\end{array},$$

which initially contains $(p-k)m+m$ columns corresponding to the omitted points on the local coarse grid. The following km columns correspond to the C-points present on the local coarse grid and their corresponding following interval of F-points. Next, we consider

$$P_S^{(p)} (A_c^{(p)})^{-1} = \begin{array}{c}
\overbrace{\hphantom{\Phi^{(k-1)m+1} \;\cdots\; \Phi^{2m+1} \; \Phi^{m+1} \;\; \Phi}}^{k} \\
\left[\begin{array}{cccc}
 & & & \\ \hdashline
\Phi^{(k-1)m} \; \cdots & \Phi^{2m} & \Phi^{m} & I \\
\Phi^{(k-1)m+1} \; \cdots & \Phi^{2m+1} & \Phi^{m+1} & \Phi \\
\vdots & \vdots & \vdots & \vdots \\
\Phi^{km-1} \; \cdots & \Phi^{3m-1} & \Phi^{2m-1} & \Phi^{m-1} \\ \hdashline
 & & &
\end{array}\right]
\end{array}
\begin{array}{l}
\}\,{\scriptstyle pm} \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\}{\scriptstyle m}, \\
\}\,{\scriptstyle N_t+1-(p+1)m}
\end{array}$$

with $A_c^{(p)}$ as in (3.1.1), which defines the effect of selective ideal interpolation multiplied by the inverse of the coarse-grid problem. Due to the selective ideal interpolation operator, exactly m points are considered, namely the C-point to be updated and the following F-interval consisting of $m-1$ points. All other points are not changed by the update of one p and the corresponding rows are therefore zero. As a consequence, the product $P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A$ also has only m nonzero rows. Furthermore, we have exactly $k+1$ blocks of $m \times m$ matrices which are not equal to zero. The matrix $P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A$ in block form is given by

(A.2)

$$\begin{array}{cccccccc}
\overbrace{\hphantom{xxx}}^{(p-k)m} & \overbrace{\hphantom{xxx}}^{m} & \multicolumn{3}{c}{\overbrace{\hphantom{xxxxxxxxx}}^{(k-1)m}} & \overbrace{\hphantom{xxx}}^{m} & \overbrace{\hphantom{xxx}}^{N_t+1-(p+1)m} & \\
\multicolumn{7}{c}{\left[\begin{array}{c:c:c:c:c:c:c}
\hphantom{xxx} & \hphantom{xxx} & & & & \hphantom{xxx} & \hphantom{xxx} \\ \hdashline
 & \mathcal{D} & \mathcal{V}_{k-2} & \cdots & \mathcal{V}_0 & \mathcal{S} & \\ \hdashline
 & & & & & &
\end{array}\right]} & \begin{array}{l} \}\, pm \\ \}\, m\,, \\ \}\, N_t+1-(p+1)m \end{array}
\end{array}$$

with block $m \times m$ inner matrices

$$\text{(A.3)} \qquad \mathcal{S} = \begin{bmatrix} I & \mathbf{0} & \cdots & \mathbf{0} \\ \Phi & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \Phi^{m-1} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \qquad \mathcal{D} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{(k-1)m+1} \\ \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{(k-1)m+2} \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{km} \end{bmatrix},$$

$$\mathcal{V}_j = \begin{bmatrix} \Phi^{(j+1)m} & \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{jm+1} \\ \Phi^{(j+1)m+1} & \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{jm+2} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^{(j+2)m-1} & \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{(j+1)m} \end{bmatrix}.$$

Here, $\mathcal{D}$ comes from the truncated coarse-grid points, $\mathcal{V}_{k-2}, \ldots, \mathcal{V}_0$ represent the first $k-1$ local coarse-grid points, and $\mathcal{S}$ corresponds to the last point of the local coarse grid. Note, the $\mathcal{S}$-block is the diagonal block of the larger matrix. Now, we consider the operator PR_I, which is equivalent to an F-relaxation, and globally given by

$$PR_I = \begin{bmatrix} \mathcal{S} & & \\ & \ddots & \\ & & \mathcal{S} \end{bmatrix}.$$

We can now calculate the error-propagation $(I - \sum_{p=0}^{N_T} P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A) PR_I$ by exploiting the structure of the matrices $P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A$ and PR_I. Instead of computing the complete matrix, we can compute the blocks $-\mathcal{DS}, -\mathcal{V}_j\mathcal{S}$ for $j = k-2, \ldots, 0$, and $(I - \mathcal{S})\mathcal{S}$. Note that the identity term is added to $-\mathcal{S}$ because $\mathcal{S}$ is the diagonal block of $P_S^{(p)} (A_c^{(p)})^{-1} R_I^{(p)} A$. Working through the algebra yields $-\mathcal{V}_j\mathcal{S} = \mathbf{0}$ for $j = k-2, \ldots, 0$, $(I - \mathcal{S})\mathcal{S} = \mathcal{S}^2 - \mathcal{S} = \mathbf{0}$, and the block $m \times m$ matrix

$$-\mathcal{DS} = \begin{bmatrix} \Phi^{km} & \mathbf{0} & \cdots & \mathbf{0} \\ \Phi^{km+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^{km+m-1} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix},$$

which is equivalent to the definition of $\mathcal{G}$ in (3.2). Note that for the case $p < k$ in matrix (A.2) the operator $\mathcal{D}$ is omitted, since for these C-points all previous C-points are contained in the local coarse grid.

Appendix B. Error propagation for approximate local coarse problems. The definition $R_I^{(p)}A$ is the same as in (A.1), but now

$$P_S^{(p)}\widetilde{A_c}^{(p)} = \left[\begin{array}{ccccc} \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{k}} \\ & & & & \\ \hdashline \Phi^0\Psi^{k-1} & \cdots & \Phi^0\Psi^2 & \Phi^0\Psi & \Phi^0 \\ \vdots & & \vdots & \vdots & \vdots \\ \Phi^{m-1}\Psi^{k-1} & \cdots & \Phi^{m-1}\Psi^2 & \Phi^{m-1}\Psi & \Phi^{m-1} \\ \hdashline & & & & \end{array}\right] \begin{array}{l} \left.\begin{array}{l} \\ \end{array}\right\}{\scriptstyle pm} \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\}{\scriptstyle m} \\ \left.\begin{array}{l} \\ \end{array}\right\}{\scriptstyle N_t+1-(p+1)m} \end{array} .$$

As a result, we get a block matrix equivalent to (A.2), but this time with $m\times m$ block matrices $\widetilde{\mathcal{V}_j}$ and $\widetilde{\mathcal{D}}$ given by

$$\widetilde{\mathcal{V}_j} = \begin{bmatrix} \Phi^0\Psi^{(j+1)} & \mathbf{0} & \cdots & \mathbf{0} & -\Phi^0\Psi^j\Phi \\ \vdots & \vdots & & \vdots & \vdots \\ \Phi^{m-1}\Psi^{(j+1)} & \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{m-1}\Psi^j\Phi \end{bmatrix} \quad \widetilde{\mathcal{D}} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & -\Phi^0\Psi^{k-1}\Phi \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & -\Phi^{m-1}\Psi^{k-1}\Phi \end{bmatrix}$$

Note, that $\mathcal{S}$ is the same as given in (A.3). Again, we use the structure of the matrices and calculate $m\times m$ block submatrices of $P_S^{(p)}(\widetilde{A_c}^{(p)})^{-1}R_I^{(p)}APR_I$ given by

$$-\widetilde{\mathcal{V}_j}\mathcal{S} = \begin{bmatrix} \Phi^0\Psi^j(\Phi^m-\Psi) & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^{m-1}\Psi^j(\Phi^m-\Psi) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \qquad -\widetilde{\mathcal{D}}\mathcal{S} = \begin{bmatrix} \Phi^0\Psi^{k-1}\Phi^m & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi^{m-1}\Psi^{k-1}\Phi^m & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

where $-\widetilde{\mathcal{V}_j}\mathcal{S}$ is equivalent to $\mathcal{Z}_j$ from (3.4) and $-\widetilde{\mathcal{D}}\mathcal{S}$ is equivalent to $\mathcal{W}$ from (3.4).

Appendix C. Discussion spatial parallelism. Here we demonstrate that the use of spatial parallelism has comparable effects on sequential time-stepping (before saturation) as it does on AT-MGRIT . In particular, we emphasize that when spatial parallelism saturates, the observed near-perfect speedup obtained by spatial parallelism before saturation will extend to AT-MGRIT. Table 5 presents overall runtimes for using one and four processes in space for time-stepping, Parareal, and two-level AT-MGRIT, the last two using 64 processes in time (same variants as in Table 1). We see that for all algorithms we get a speedup of about 3.8 by using four spatial processes compared to one process. Note that this problem scales well with spatial parallelization, and spatial parallelism (as in most cases) should be the first choice. However, spatial parallelization is exhausted at some point and temporal parallelization can then provide additional speedups.

Acknowledgments. We would like to thank Dr. Wayne Mitchell for helpful discussions that initiated this research. Los Alamos National Laboratory report number LA-UR-21-26105.

REFERENCES

Space processes	Time-stepping one time process	Parareal 64 time processes	Two-level AT-MGRIT 64 time processes
1	347,666 s	91,596 s	79,397 s
4	92,473 s	23,708 s	20,411 s

Table 5: Total runtimes using one and four processes in space for time-stepping, Parareal, and AT-MGRIT with $k = 32$, the latter two using a coarsening factor of 256 and 64 processes in time.

[1] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Report ANL-95/11 - Revision 3.13, Argonne National Laboratory, 2020.

[2] R. Bank, R. Falgout, T. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid domain and range decomposition (AMG-DD/AMG-RD)*, SIAM Journal on Scientific Computing, 37 (2015), pp. S113–S136.

[3] A. Bartel, S. Baumanns, and S. Schöps, *Structural analysis of electrical circuits including magnetoquasistatic devices*, Applied Numerical Mathematics, 61 (2011), pp. 1257–1270.

[4] M. Bolten, S. Friedhoff, J. Hahne, and S. Schöps, *Parallel-in-time simulation of an electrical machine using MGRIT*, Comput. Vis. Sci., 23 (2020), pp. Paper No. 14, 14.

[5] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[6] D. Chazan and W. Miranker, *Chaotic relaxation*, Linear Algebra and its Applications, 2 (1969), pp. 199–222.

[7] I. Cortes Garcia, H. De Gersem, and S. Schöps, *A structural analysis of field/circuit coupled problems based on a generalised circuit element*, Numerical Algorithms, (2019), pp. 1–22.

[8] F. Danieli and S. MacLachlan, *Multigrid reduction in time for non-linear hyperbolic equations*, 2021, https://arxiv.org/abs/2104.09404.

[9] V. A. Dobrev, T. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM Journal on Scientific Computing, 39 (2017), pp. S501–S527.

[10] P. Dular, C. Geuzaine, F. Henrotte, and W. Legros, *A general environment for the treatment of discrete problems and its application to the finite element method*, IEEE Transactions on Magnetics, 34 (1998), pp. 3395–3398.

[11] R. D. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[12] R. D. Falgout, M. Lecouvez, and C. S. Woodward, *A parallel-in-time algorithm for variable step multistep methods*, Journal of Computational Science, 37 (2019), pp. 101029, 12.

[13] M. J. Gander, *50 years of Time Parallel Time Integration*, in Multiple Shooting and Time Domain Decomposition, Springer, 2015, pp. 69–113.

[14] C. Geuzaine, *GetDP: a general finite-element solver for the de Rham complex*, PAMM, 7 (2007), pp. 1010603–1010604.

[15] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1309–1331.

[16] C. Geuzaine and J.-F. Remacle, *Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, 2020. http://www.gmsh.info, Online; accessed June 21, 2021.

[17] J. Gyselinck, L. Vandevelde, and J. Melkebeek, *Multi-slice FE modeling of electrical machines with skewed slots-the skew discretization error*, Magnetics, IEEE Transactions on, 37 (2001), pp. 3233 – 3237.

[18] J. Hahne and S. Friedhoff, *Github repository for PyMGRIT*, 2020. https://github.com/pymgrit/pymgrit, Online; accessed June 21, 2021.

[19] J. Hahne, S. Friedhoff, and M. Bolten, *Algorithm 1016: PyMGRIT: A python package for the parallel-in-time method mgrit*, ACM Trans. Math. Softw., 47 (2021).

[20] A. Howse, H. De Sterck, R. D. Falgout, S. MacLachlan, and J. B. Schroder, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations*, SIAM Journal on Scientific Computing, 41 (2019), pp. A538–A565.

[21] J. D. Jackson, *Classical electrodynamics*, Wiley, New York, NY, 3rd ed., 1998.

[22] L. Kronsjö, *A note on the "nested iterations" methods*, Nordisk Tidskr. Informationsbehandling (BIT), 15 (1975), pp. 107–110.

[23] L. Kronsjö and G. Dahlquist, *On the design of nested iterations for elliptic difference equations*, Nordisk Tidskr. Informationsbehandling (BIT), 12 (1972), pp. 63–71.

[24] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel"*, Comptes Rendus de l'Académie des Sciences. Série I. Mathématique, 332 (2001), pp. 661–668.

[25] T. Lunet, J. Bodart, S. Gratton, and X. Vasseur, *Time-parallel simulation of the decay of homogeneous turbulence using parareal with spatial coarsening*, Computing and Visualization in Science, 19 (2018), pp. 31–44.

[26] F. Magoulès, G. Gbikpi-Benissan, and Q. Zou, *Asynchronous iterations of parareal algorithm for option pricing models*, Mathematics, 6 (2018).

[27] W. Mitchell and T. Manteuffel, *Advances in implementation, theoretical motivation, and numerical results for the nested iteration with range decomposition algorithm*, Numerical Linear Algebra with Applications, 25 (2018), p. e2149.

[28] W. B. Mitchell, R. Strzodka, and R. D. Falgout, *Parallel performance of algebraic multigrid domain decomposition*, Numerical Linear Algebra with Applications, 28 (2021), p. e2342.

[29] A. S. Nielsen, G. Brunner, and J. S. Hesthaven, *Communication-aware adaptive Parareal with application to a nonlinear hyperbolic system of partial differential equations*, Journal of Computational Physics, 371 (2018), pp. 483–505.

[30] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Comm. ACM, 7 (1964), pp. 731–733.

[31] B. W. Ong and J. B. Schroder, *Applications of time parallelization*, Comput. Vis. Sci., 23 (2020), pp. Paper No. 11, 15.

[32] J. E. Pearson, *Complex patterns in a simple system*, Science, 261 (1993), pp. 189–192.

[33] M. Ries, U. Trottenberg, and G. Winter, *A note on MGR methods*, Linear Algebra Appl., 49 (1983), pp. 1–26.

[34] D. Ruprecht, *Convergence of Parareal with spatial coarsening*, PAMM, 14 (2014), pp. 1031–1034.

[35] S. Schöps, I. Niyonzima, and M. Clemens, *Parallel-in-time simulation of eddy current problems using Parareal*, IEEE Transactions on Magnetics, 54 (2018), pp. 1–4.

[36] B. S. Southworth, *Necessary conditions and tight two-level convergence bounds for Parareal and Multigrid Reduction in Time*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 564–608.

[37] B. S. Southworth, W. Mitchell, A. Hessenthaler, and F. Danieli, *Tight two-level convergence of Linear Parareal and MGRIT: Extensions and implications in practice*, arXiv e-prints, (2020). arXiv:2010.11879.

[38] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[39] Q. Zou, G. Gbikpi-Benissan, and F. Magoulès, *Asynchronous parareal algorithm applied to european option pricing*, in 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2017, pp. 37–40.