Electronic Transactions on Numerical Analysis.
Volume 54, pp. 210–233, 2021.

ISSN 1068–9613.
DOI: 10.1553/etna_vol54s210

ETNA
Kent State University and
Johann Radon Institute (RICAM)

MULTIGRID REDUCTION IN TIME WITH RICHARDSON EXTRAPOLATION*

R. D. FALGOUT†, T. A. MANTEUFFEL‡, B. O'NEILL§, AND J. B. SCHRODER§

Abstract. The advent of exascale computing will leave many users with access to more computational resources than they can simultaneously use, e.g., billion-way parallelism. In particular, this is true for time-dependent simulations that limit parallelism to the spatial domain. One method to add parallelism in time to existing simulation codes and thus take advantage of ever larger compute resources is Multigrid Reduction in Time (MGRIT). The goal is to achieve a smaller time-to-solution through parallelism in time. In this paper, MGRIT is enhanced with Richardson extrapolation in a cost-efficient way to produce a parallel-in-time method with improved accuracy. Overall, this leads to a large improvement in the accuracy per computational cost of MGRIT.

Key words. parallel time integration, high-performance-computing, multigrid-reduction-in-time, extrapolation-methods

AMS subject classifications. 65M20, 65M55, 65F08, 65F10, 65Y05

1. Introduction.

Based on current trends, it is reasonable to assume that future increases in compute power will be available through increases in concurrency rather than increases in clock speed. This is a troubling trend for those using sequential time integration schemes that limit parallelism to the spatial domain because it creates a sequential bottleneck. One remedy to this bottleneck is to consider parallel-in-time methods, which include a wide variety of direct and indirect methods. Work on parallel-in-time methods actually began at least 50 years ago [23], but interest in parallel-in-time was rekindled with the parareal method [21] in 2001, followed by the multilevel spectral deferred correction approach (also called PFASST) [6, 22]. For a detailed review of parallel-in-time methods, see the review papers [12, 25].

This work focuses on the Multigrid Reduction in Time (MGRIT) method [7, 10], an iterative multilevel algorithm that uses multigrid reduction techniques and temporal domain decomposition to add large-scale temporal parallelism to existing time integration codes. In terms of time step evaluations, MGRIT is an optimal algorithm, i.e., $O(N_t)$, where N_t is the total number of time steps [7]; however, the constant for MGRIT is higher than that of sequential time integration. This creates a crossover point where the additional computational resources eventually overcome the extra computational work. Beyond this crossover point,

*Received January 22, 2020. Accepted November 18, 2020. Published online on February 18, 2021. Recommended by Stefan Vandewalle.
The work of R. D. Falgout was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. LLNL-JRNL-752911.
The work of T. A. Manteuffel was performed under the auspices of the U.S. Department of Energy under grant numbers (SC) DE-FC02-03ER25574 and (NNSA) DE-NA0002376 and Lawrence Livermore National Laboratory under contract B600360.

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551 (rfalgout@llnl.gov).

‡Department of Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado (tmanteuf@colorado.edu).

§Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico (jbschroder@unm.edu).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

MGRIT provides a speedup over sequential methods. This work uses XBraid [33], an open source implementation of MGRIT developed at the Lawrence Livermore National Laboratory (LLNL). Some recent applications of XBraid include linear and nonlinear parabolic problems [7, 9], compressible fluid dynamics [8], power grids [20, 29], eddy-current problems [11], linear advection [16, 30], and machine learning [14].

This work seeks to improve the accuracy and efficiency of MGRIT through Richardson extrapolation (RE)-based time integration, which is a non-intrusive algorithm that uses coarse-grid approximations in conjunction with RE to improve the convergence order of the underlying time-integration scheme. MGRIT is also a non-intrusive iterative algorithm [7], and it uses a coarse-grid error correction to accelerate convergence to the solution at a desired fine scale. In this paper, we present a new, improved MGRIT algorithm that combines RE-based time integration and MGRIT in a way that capitalizes on those similarities. The result is a method that delivers higher accuracy per computational cost (APCC) when compared to the original MGRIT algorithm, while continuing to offer the temporal concurrency of MGRIT.

The use of RE with time-parallelism has some history. The work [31] extracts small-scale parallelism from RE by processing corresponding fine and coarse RE steps in parallel, while the work [32] incorporates RE with parareal in a manner analogous to the proposed approach. Since MGRIT with so-called F-relaxation and parareal are equivalent in a two-level setting [13], the proposed work adds further insight into the two-level method from [32], while extending the use of RE to multilevel MGRIT. One key difference between [32] and the proposed method is that while parareal solves the coarse-level problem needed for RE sequentially (exactly), MGRIT solves it inexactly in parallel with a multigrid hierarchy. We also provide a sharp two-level convergence analysis for linear problems that applies to both the parareal and MGRIT settings and includes the effects of FCF-relaxation, which is not a part of parareal. Lastly, the proposed work considers the impact of nonuniform time-grids, which [32] did not.

The spatial multigrid community refers to a similar approach as τ-extrapolation [2, 4, 15] because it modifies the τ-term in full approximation storage (FAS) multigrid [3]. The further works [17, 18, 28] connect multilevel finite element discretizations directly with multigrid τ-extrapolation solvers. Although the proposed approach uses Richardson extrapolation, which differs slightly from τ-extrapolation, we name the new MGRIT algorithm τ-MGRIT because our eventual algorithm also results in a simple modification of the τ-term. The results presented here are based on [24].

In summary, the τ-MGRIT algorithm developed here is a non-intrusive, parallel-in-time algorithm that uses RE to improve the accuracy of the underlying time integration scheme, while also allowing for parallel speedups. This is accomplished without a significant increase in the computational cost when compared to the original MGRIT algorithm. The method is generally applicable whenever RE is an effective technique for the underlying problem and discretization. Examining the applicability of RE-based time integration for a particular problem is beyond the scope of this paper, but we note that Richardson-type methods have a long history of improving solutions and providing error estimates for a wide variety of problems [1, 19, 27].

The outline of this paper is as follows. First, a brief overview of RE-based time integration is given. Next, the two- and multilevel τ-MGRIT algorithms are derived. For brevity, this derivation also acts as an introduction to the standard MGRIT algorithm. The following section presents a cost analysis of τ-MGRIT for uniform time grids. This includes a derivation of a convergence bound (Section 4.2), two numerical examples (Sections 5.1 and 5.2), and a strong scaling study (Section 5.4). Finally, in Section 6, the application of τ-MGRIT on a non-uniform temporal grid is considered. Overall, the results in this paper show how the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

τ-MGRIT algorithm can be used to significantly improve the APCC of the MGRIT algorithm, whenever RE-based time integration is appropriate.

2. Richardson extrapolation. Richardson extrapolation (RE), first introduced in 1911 [26], is a non-intrusive extrapolation-based technique designed to estimate and eliminate the lowest-order error term from the underlying numerical approximations. Let $U(\delta t)$ be an approximation of U with a step size δt and, by Taylor's theorem, an error of the form

$$U - U(\delta t) = C_0(\delta t)^{k_0} + C_1(\delta t)^{k_1} + \dots, \tag{2.1}$$

where each $k_0 < k_1 < \dots$ is a known constant and each C_i is an unknown constant. We assume that the expansions here are convergent, i.e., δt is small enough. For a larger step size $m\,\delta t$, the same error formula becomes

$$U - U(m\,\delta t) = C_0(m\,\delta t)^{k_0} + C_1(m\,\delta t)^{k_1} + \dots \tag{2.2}$$

Eliminating the exact solution U and ignoring higher-order terms, we find

$$(\delta t)^{k_0} C_0 = \frac{U(\delta t) - U(m\,\delta t)}{m^{k_0} - 1} + O((\delta t)^{k_1}).$$

Substituting for $(\delta t)^{k_0} C_0$ in equation (2.1) gives

$$U = \frac{m^{k_0} U(\delta t) - U(m\,\delta t)}{m^{k_0} - 1} + O((\delta t)^{k_1}), \tag{2.3}$$

thus improving the order of accuracy from k_0 to k_1. This process can be repeated to further improve the accuracy. The general recurrence relationship for RE is

$$\begin{aligned} U_0 &= U(\delta t), \\ U_{i+1}(\delta t) &= \frac{m^{k_i} U_i(\delta t) - U_i(m\,\delta t)}{m^{k_i} - 1}, \end{aligned} \tag{2.4}$$

where U_i is the ith RE-based approximation of U and $U = U_{i+1}(\delta t) + O((\delta t)^{k_{i+1}})$.

It is important to note that RE does not *guarantee* an improvement. For example, equations (2.2) and (2.1) imply that one level (i.e, $i = 1$) of RE will not improve the solution if the next error term C_1 is relatively large,

$$\left| C_1 \frac{m^{k_0} - m^{k_1}}{m^{k_0} - 1} \right| (\delta t)^{k_1} \geq |C_0| (\delta t)^{k_0}.$$

Moving forward, the two-step RE-based time integration method is formulated as a two-level sequential time integration scheme defined on the following nested two-level grid: Define a uniform temporal grid with time step δt and nodes $t_j,\ j = 0, \dots, N_t$ (non-uniform time grids are discussed in Section 6). Further, define a coarse temporal grid with time step $\Delta T = m\,\delta t$ and nodes $T_j = j\Delta T,\ j = 0, 1, \dots, N_t/m$, for some coarsening factor m. As shown in Figure 2.1, coarse-grid points (red) reside on both the fine and coarse grids. Fine-grid time points (shown in black) exist only on the fine grid. This perspective will allow us to highlight the striking similarities between the RE- and MGRIT-based time integration methods, while also acting as a stepping stone toward developing the cost-efficient τ-MGRIT algorithm.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

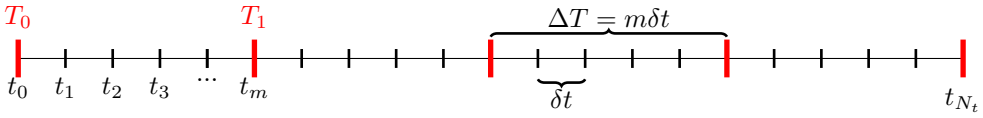


FIG. 2.1. *Fine- and coarse-grid temporal meshes. Fine-grid points (black) are present on only the fine-grid, whereas coarse-grid points (red) are present on both the fine- and coarse-grid.*

2.1. RE-based time integration. Both MGRIT and RE solve the time-discretized equations of a general first-order ordinary differential equation (ODE):

$$\begin{aligned} \frac{du}{dt} &= f(u,t), \quad u(0) = u_0, \qquad t \in [0,T], \\ u_{j+1} &= \Phi(u_j, \delta t) + g_{j+1}, \qquad j = 0, \ldots, N_t - 1, \end{aligned} \tag{2.5}$$

where u_j represents the discrete solution at $t = t_j$, Φ is an operator representing the chosen time-stepping method, δt is the time step size, and g_j represents the collection of all time-dependent but solution-independent terms needed to make (2.5) consistent at time $t = t_j$.

In general, the error formula for a time integration scheme of order k_g can be obtained using Taylor's theorem. The only difficulty is that, in most cases, the expansion coefficients vary in both space and time. Thus, RE-based time integration must be completed under the assumption that the solution is sufficiently smooth and that δt is small, so that each $C_i(x,t)$ can be assumed to be nearly constant in time.

To minimize the domain size on which these assumptions must hold, RE-based time integration is usually completed by chopping the global time domain into a discrete set of sequentially ordered temporal subdomains. In the notation above, each temporal subdomain corresponds to an interval of length $m\delta t$ lying in between two consecutive coarse points (see Figure 2.1). RE is used to update the solution at each coarse-grid time point. That updated solution is then used to propagate the solution forward in time to the next coarse-grid point, where the process is repeated. Solving the system in equation (2.5) with a single stage (i.e., $i = 0$ in equation (2.4)) of RE-based time integration equates to solving the following system of equations

$$\mathbf{u}_{f,mj+1} = \Phi(\mathbf{u}_{*,mj}, \delta t) + g_{mj}, \tag{2.6}$$

$$\mathbf{u}_{f,mj+2} = \Phi(\mathbf{u}_{f,mj+1}, \delta t) + g_{mj+1}, \tag{2.7}$$

$$\vdots \;=\; \vdots \tag{2.8}$$

$$\mathbf{u}_{f,m(j+1)} = \Phi(\mathbf{u}_{f,m(j+1)-1}, \delta t) + g_{m(j+1)}, \tag{2.9}$$

followed by a step with a coarse time-spacing and a subsequent error correction,

$$\mathbf{u}_{c,m(j+1)} = \Phi(\mathbf{u}_{*,mj}, m\delta t) + g_{m(j+1)}, \tag{2.10}$$

$$\mathbf{u}_{*,m(j+1)} = \frac{m^{k_g}\mathbf{u}_{f,m(j+1)} - \mathbf{u}_{c,m(j+1)}}{m^{k_g} - 1}, \tag{2.11}$$

$$= a\mathbf{u}_{f,m(j+1)} - b\mathbf{u}_{c,m(j+1)}, \tag{2.12}$$

for $j = 0, 1, 2, \ldots, N_t/m$, $\mathbf{u}_{*,0} = \mathbf{u}_{f,0}$. We let $a = m^{k_g}/(m^{k_g} - 1)$ and $b = 1/(m^{k_g} - 1)$, such that $a - b = 1$. The notation $\mathbf{u}_{f/c/*,i}$ indicates a fine-grid (f), coarse-grid (c), or RE-enhanced ($*$) solution at time $t = t_i$. Figure 2.2 provides a graphical representation of this algorithm with $m = 2$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

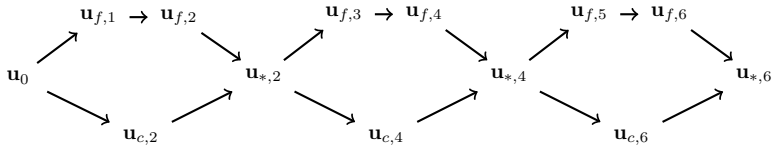


FIG. 2.2. *Time integration with RE and $m = 2$. The notation $\mathbf{u}_{f/c/*,a}$ denotes a fine-grid (f), coarse-grid (c), or RE-enhanced ($*$) solution at time $t = t_a$. Coarse-grid solutions are used to enhance the fine-grid solution at the C-points in order to generate the RE-enhanced solutions $u_{*,a}$.*

MGRIT- and RE-based time integration are both non-intrusive algorithms; MGRIT adds temporal parallelism to existing time integration schemes, whereas RE-based time integration improves the accuracy. The goal of this paper is to develop a method that allows RE-based time integration to be used efficiently inside the MGRIT framework. In particular, we develop a non-intrusive, parallel-in-time solver that improves the accuracy of the underlying time integration scheme without a significant increase in the total cost when compared to the standard MGRIT algorithm. This new MGRIT variant, known as τ-MGRIT, is derived in the following sections.

3. MGRIT with Richardson extrapolation. In the spatial multigrid community, the algorithm resulting from integrating multigrid and RE is known as τ-extrapolation. Multigrid τ-extrapolation uses the extrapolation principle to exploit the multilevel structure of spatial multigrid algorithms, resulting in a fast solver with improved approximation properties.

Multigrid τ-extrapolation is an extension of the FAS multigrid method, which was originally designed to solve general nonlinear problems. MGRIT is also built using FAS. A basic two-level FAS cycle is described in Algorithm 1, where $A_h\mathbf{u}_h = \mathbf{g}_h$ and $A_H\mathbf{v}_H = \mathbf{g}_H$ represent the discretization of a (perhaps nonlinear) differential equation on fine and coarse grids, respectively. We use $\mathbf{v}_H$ (and not $\mathbf{u}_H$) for the coarse-grid variable, as is common in the FAS literature. For simplicity, we further assume that A_h and A_H were discretized on uniform time-grids with step sizes δt and $m\delta t$, respectively. Both R and $\hat{R}$ are restriction operators interpolating from fine to coarse ($h \to H$), and P is the prolongation operator interpolating from coarse to fine ($H \to h$). FAS allows for these two restriction operators to be distinct, although they are often equal, as for the case here where we use injection. Algorithm 1 relies on the complementary relationship between relaxation and coarse-grid correction, where any error that is not effectively reduced by relaxation is mapped to the coarse grid and accurately computed there. Of particular importance here is τ, which allows for RE to be integrated into the coarse-grid correction straightforwardly.

Algorithm 1 FAS(A, $\mathbf{u}$, $\mathbf{g}$).

1: Relax on $A_h\mathbf{u_h} = \mathbf{g_h}$ using a (non)linear relaxation method.
2: Solve the coarse-grid problem
$A_H\mathbf{v}_H = \mathbf{g}_H + \tau_h^H$,
where $\tau_h^H = A_H R\mathbf{u}_h - \hat{R}A_h\mathbf{u}_h$.
3: Compute the coarse-grid error approximation
$\mathbf{e}_H = \mathbf{v}_H - R\mathbf{u}_h$.
4: Correct the fine-grid approximation
$\mathbf{u}_h \leftarrow \mathbf{u}_h + P\mathbf{e}_H$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The τ-term (τ_h^H) can be interpreted as a correction of the coarse-grid solution $\mathbf{v}_H$ to make it coincide with the fine-grid solution $\mathbf{u}_h$ [2]. Based on the principles of RE, in the τ-extrapolation algorithm, τ_h^H is multiplied by a factor $a = m^p/(m^p - 1)$, where p is the order of the approximation, so that the coarse-grid solution better approximates the true solution $\mathbf{u}$. Using the standard MGRIT interpolation operator, this error correction can be carried up to the finest grid, giving a fine-grid solution with improved approximation properties [2]. We now derive the τ-extrapolation principle from the perspective of time integration with RE and MGRIT.

3.1. Derivation of τ-MGRIT. The two-level τ-MGRIT algorithm for problems with a linear, time-independent time integrator defined on a uniform temporal grid will now be presented. The nonlinear and time-dependent variations follow easily. The extension of τ-MGRIT to non-uniform temporal grids is discussed in Section 6. Note that this derivation assumes that the spatial grid at each time point is fixed across all levels. Spatial coarsening with τ-MGRIT is an area of research.

In block-triangular form, the original time integration problem, equation (2.5), is

$$A\mathbf{u} = \begin{bmatrix} I & & & & \\ -\Phi & I & & & \\ & \ddots & \ddots & & \\ & & -\Phi & I & \\ & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \\ \vdots \end{bmatrix} = \bar{\mathbf{g}}, \tag{3.1}$$

where Φ is a matrix. The RE-based time-integration problem, equations (2.6)–(2.12), can be written in block triangular form as

$$A^\tau\mathbf{u} = \begin{bmatrix} I & & & & & & & & \\ -\Phi & I & & & & & & & \\ & \ddots & \ddots & & & & & & \\ b\Phi_\Delta & & -a\Phi & I & & & & & \\ & 0 & & -\Phi & I & & & & \\ & & \ddots & & \ddots & \ddots & & & \\ & & & b\Phi_\Delta & & -a\Phi & I & & \\ & & & & \ddots & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \\ \mathbf{u}_{m+1} \\ \vdots \\ \mathbf{u}_{2m} \\ \vdots \end{bmatrix} = \bar{\mathbf{g}}, \tag{3.2}$$

where Φ_Δ is a matrix and $\Phi_\Delta\mathbf{u} = \Phi(\mathbf{u}, m\delta t)$ represents the coarse-grid time step. Note that $\mathbf{u} = [\mathbf{u}_0, \ldots, \mathbf{u}_{N_t}]$ is a vector of vectors, where each $\mathbf{u}_i$ has a length equal to the number of spatial unknowns at time point i. In what follows, setting $a = 1$ and $b = 0$ yields the original MGRIT algorithm.

RE-based time integration is equivalent to a block forward solve of this system. This approach allows for spatial parallelism but must propagate forward sequentially in time[1]. The τ-MGRIT algorithm solves this system iteratively, in parallel, using a coarse-grid correction scheme based on multigrid reduction. To solve equation (3.2) with FAS multigrid, one must first define the coarse-grid operator, a method of relaxation, and a method of restriction and interpolation.

[1] Technically, the two (or more) solutions can proceed in parallel. However, this is a low-level form of temporal parallelism that does not scale.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3.1.1. The τ-MGRIT coarse-grid operator. The τ-MGRIT algorithm uses multigrid reduction strategies to determine the coarse-grid operator by successively eliminating unknowns from the fine-grid system. The elimination targets F-points, where the time points have been partitioned into F- and C-points based on the temporal coarsening factor m. For example with $m = 4$, the circles in Figure 3.1 depict F-points, while the squares are C-points. Each time point refers to a block row in equation (3.2). The elimination is based on the following recursion relation:

$$\begin{aligned}\mathbf{u}_{f,mj} &= \Phi\mathbf{u}_{f,mj-1} + \mathbf{g}_{mj} = \Phi(\Phi\mathbf{u}_{mj-2} + \mathbf{g}_{mj-1}) + \mathbf{g}_{mj} = \dots, \\ &= \Phi^m\mathbf{u}_{f,m(j-1)} + \hat{\mathbf{g}}_{mj},\ j = 1, 2, \dots, N_t/m,\end{aligned}$$

where $\hat{\mathbf{g}}_{mj} = \mathbf{g}_{mj} + \Phi\mathbf{g}_{mj-1} + \cdots + \Phi^{m-1}\mathbf{g}_{m(j-1)+1}$. In block triangular form, the ideal coarse-grid operator and equation is

(3.3)
$$A^{\tau}_{\Delta}\mathbf{u}_{\Delta} := \begin{bmatrix} I & & & & \\ (b\Phi_{\Delta} - a\Phi^m) & I & & & \\ & (b\Phi_{\Delta} - a\Phi^m) & I & & \\ & & \ddots & \ddots & \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\Delta,0} \\ \mathbf{u}_{\Delta,1} \\ \mathbf{u}_{\Delta,2} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ a\hat{\mathbf{g}}_m - b\mathbf{g}_m \\ a\hat{\mathbf{g}}_{2m} - b\mathbf{g}_{2m} \\ \vdots \end{bmatrix}.$$

The coarse-grid operator A^{τ}_{Δ} is referred to as ideal because the exact solution of equation (3.3) yields the exact solution to equation (3.2) at the coarse points. If this is followed by interpolation (i.e., injection from the coarse-level to the fine-level at the C-points, followed by the F-relaxation depicted in Figure 3.1), then the exact solution is also available at the fine points (F-points). Here, the "exact solution" refers to the solution found at that time point using the RE-based sequential time integration scheme. That is to say, $\mathbf{u}_{\Delta,j} = \mathbf{u}_{mj}$ in equation (3.2), for $j = 0, 1, 2, \dots, N_t/m$.

The limitation of this exact reduction method is that the coarse-grid problem (3.3) is, in general, as expensive to solve as the original fine-grid problem (3.2) (because of the Φ^m-evaluations). Approximating Φ^m with Φ_{Δ} and noting that $a - b = 1$ gives a cheap approximation of this exact coarse-grid operator

$$B_{\Delta} = \begin{bmatrix} I & & & \\ -\Phi_{\Delta} & I & & \\ & \ddots & \ddots & \\ & & -\Phi_{\Delta} & I \end{bmatrix},$$

where Φ_{Δ} is an approximate coarse-grid time step operator. This choice of approximating Φ^m with Φ_{Δ} has been used in various MGRIT algorithms to good effect [7, 9, 20]. Additionally, the coarse-grid operator B_{Δ} is independent of the constants a and b, indicating that the coarse-grid operator is identical for both the τ-MGRIT and MGRIT algorithms. In fact, the only difference between the two-level τ-MGRIT and MGRIT algorithms is the calculation of the coarse-grid right-hand side (RHS) and a process known as fine-grid C-relaxation (introduced below).

One obvious choice for defining Φ_{Δ} is to re-discretize the problem on the coarse grid so that a coarse-grid time step is roughly as expensive as a fine-grid time step. For instance, with a backward Euler method, one simply uses a larger time step size. Convergence of MGRIT is ultimately determined by the approximation $A^{\tau}_{\Delta} \approx B_{\Delta}$, and this choice of re-discretizing Φ with $\Delta T = m\delta t$ has proved effective when applied to the MGRIT algorithm [7, 8, 9].

Note that while the definition of this algorithm relies upon Φ and Φ_{Δ}, the internals of these functions need not be known. This is the non-intrusive aspect of MGRIT. The user defines the time step operator and can use a library τ-MGRIT code.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

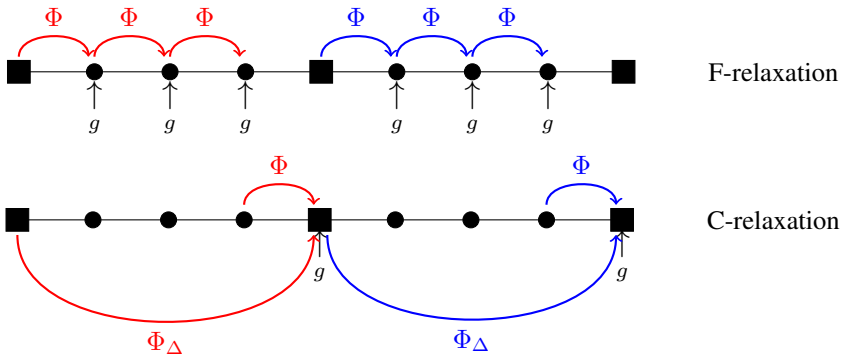


FIG. 3.1. *F- and C-relaxation with $m = 4$. F-relaxation propagates the solution forward across each fine-grid interval. C-relaxation needs two solution propagations for RE, from the last F-point and preceding C-point for each coarse-grid interval. The Φ_Δ steps are not required when using the standard MGRIT algorithm. Also note that each separate series of time steps (red and blue) can be completed simultaneously on separate processors.*

As per the original MGRIT algorithm [7], restriction is completed using injection. Similarly, interpolation is equivalent to injection from the coarse to the fine grid at the C-points, followed by F-relaxation (see Figure 3.1). For the purposes of this paper, the restriction and interpolation operators are denoted R_I (injection) and P, respectively. Comparing to Algorithm 1, A_H corresponds to B_Δ discussed here, and A_h corresponds to one of the operators (3.1) or (3.2), depending on whether MGRIT or τ-MGRIT is being used.

3.1.2. F- and C-relaxation. MGRIT uses two basic types of relaxation, namely F- and C-relaxation. Figure 3.1 shows the actions of F- and C-relaxation on a temporal grid with $m = 4$.

F-relaxation propagates the solution forward in time from each coarse point to the neighboring F-points. On each coarse-grid interval, F-relaxation includes $m - 1$ sequential time integration steps. For a linear problem with a backward Euler time integration scheme, this constitutes $m-1$ sequentially linked spatial inverses. For a nonlinear time integration operator, updating each interval of F-points requires $m - 1$ sequentially linked nonlinear solves, each of which will likely require several nonlinear iterations, e.g., Newton iterations. Most importantly, F-relaxation is highly parallel; each interval of F-points can be updated independently during F-relaxation. Note that the sequential nature of F-relaxation places a bound on the number of temporal processors for which we expect to see a speedup. To be precise, F-relaxation obtains a maximal parallel concurrency when N_t/m temporal processors are used. At this point, additional temporal processors only act to increase communication costs associated with F-relaxation. However, there may be an advantage in memory consumption [9].

C-relaxation is the process of updating the coarse time points. As with F-relaxation, each C-point can be updated independently, in parallel. The standard C-relaxation used by the original MGRIT algorithm requires exactly one time step evaluation per C-point (remove the Φ_Δ application in Figure 3.1). For τ-MGRIT, C-relaxation on the finest grid requires two time steps per C-point, one on the fine grid and one on the coarse grid, i.e., Φ and Φ_Δ in equation (3.2), respectively.

In [7], it was shown that F-relaxation was not enough to ensure a scalable multilevel algorithm. To fix this, a composite relaxation known as FCF-relaxation was introduced. FCF-relaxation consists of an F-relaxation, followed by a C-relaxation, followed by another F-relaxation.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3.1.3. The two-level τ-MGRIT algorithm with FAS. To accommodate both linear and nonlinear problems, the τ-MGRIT algorithm uses FAS multigrid as outlined in Algorithm 1. In the τ-MGRIT notation, the FAS coarse-grid problem, defined in Algorithm 1, is: find $\mathbf{v}_\Delta$ such that

$$B_\Delta \mathbf{v}_\Delta = B_\Delta R_I \mathbf{u} + R_I(\mathbf{g} - A^\tau \mathbf{u}). \tag{3.4}$$

Expanding this equation, the jth row of the coarse-grid problem is

$$\begin{aligned}
[B_\Delta \mathbf{v}_\Delta]_j &= [B_\Delta R_I \mathbf{u}]_j + [R_I(\mathbf{g} - A^\tau \mathbf{u})]_j, \\
&= [B_\Delta R_I \mathbf{u}]_j + \mathbf{g}_{mj} - [A^\tau \mathbf{u}]_{mj}, \\
&= \mathbf{u}_{mj} - \Phi_\Delta \mathbf{u}_{m(j-1)} + \mathbf{g}_{mj} - [A^\tau \mathbf{u}]_{mj}, \\
&= \mathbf{u}_{mj} - \Phi_\Delta \mathbf{u}_{m(j-1)} + \mathbf{g}_{mj} + a\Phi \mathbf{u}_{mj-1} - \mathbf{u}_{mj} - b\Phi_\Delta \mathbf{u}_{m(j-1)}, \\
&= \mathbf{g}_{mj} + a\Phi \mathbf{u}_{mj-1} - (1+b)\Phi_\Delta \mathbf{u}_{m(j-1)}, \\
&= \mathbf{g}_{mj} + a(\Phi \mathbf{u}_{mj-1} - \Phi_\Delta \mathbf{u}_{m(j-1)}), \\
&= \mathbf{g}_{mj} + a(\mathbf{u}_{mj} - \Phi_\Delta \mathbf{u}_{m(j-1)} - \mathbf{u}_{mj} + \Phi \mathbf{u}_{mj-1}), \\
&= \mathbf{g}_{mj} + a([B_\Delta R_I \mathbf{u}]_j - [R_I A \mathbf{u}]_j),
\end{aligned}$$

where $j = 1, 2, \ldots, N_t/m$, the operator A is as defined in equation (3.1), and the notation $[X]_j$ denotes the jth block row of the matrix X. Given this, the τ-MGRIT coarse-grid problem is: find $\mathbf{v}_\Delta$ such that

$$B_\Delta \mathbf{v}_\Delta = R_I \mathbf{g} + a(B_\Delta R_I \mathbf{u} - R_I A \mathbf{u}). \tag{3.5}$$

Thus, the two-level τ-MGRIT algorithm is identical to the standard MGRIT algorithm with the exception that the second term in the coarse-grid RHS is scaled by the factor $a = m^{k_g}/(m^{k_g} - 1)$ (compare equations (3.4) and (3.5)). This result shares a similar form to that outlined in the original τ-extrapolation paper for spatial multigrid. Namely, that the accuracy of the standard MGRIT algorithm can be improved by simply scaling the RHS of the coarse-grid equation by the appropriate factor.

Algorithm 2 outlines the two-level τ-MGRIT algorithm. The multilevel extension of the two-level τ-MGRIT algorithm is discussed in Remark 3.1.

Algorithm 2 τ-MGRIT(A, $\mathbf{u}$, $\mathbf{g}$).

1: Apply F- or FCF-relaxation to $A(\mathbf{u}) = \mathbf{g}$.
2: Calculate $\mathbf{u}_{f,mi} = \Phi(\mathbf{u}_{mi-1}) + \mathbf{g}_{mi}$.
3: Calculate $\mathbf{u}_{c,mi} = \Phi_\Delta(\mathbf{u}_{m(i-1)}) + \mathbf{g}_{mi}$.
4: Inject $\mathbf{u}_{f,mi}$ and $\mathbf{u}_{c,mi}$ to the coarse grid.
   $\hat{\mathbf{u}}_{\Delta,i} \leftarrow \mathbf{u}_{f,mi}, \quad \bar{\mathbf{u}}_{\Delta,i} \leftarrow \mathbf{u}_{c,mi}$.
5: Solve $B_\Delta(\mathbf{v}_\Delta) = \mathbf{g}_\Delta + a(\hat{\mathbf{u}}_\Delta - \bar{\mathbf{u}}_\Delta)$.
6: Compute the error approximation:
   $\mathbf{e}_{\Delta,i} = \mathbf{v}_{\Delta,i} - \hat{\mathbf{u}}_{\Delta,i}$.
7: Correct $\mathbf{u}$ at C-points: $\mathbf{u}_{mi} = \mathbf{u}_{f,mi} + \mathbf{e}_{\Delta,i}$.
8: If converged, then update F-points: apply F-relaxation to $A(\mathbf{u}) = \mathbf{g}$.
9: Else go to step 1.

It is important to note that the original RE equations were derived under the assumption that the solution is smooth and that the time step size is small. Of course, this is not always

ETNA
Kent State University and
Johann Radon Institute (RICAM)

true, and as such, RE-based time integration cannot be used in all situations. However, as will be shown in Section 4.2, the τ-MGRIT scheme is guaranteed to converge to the solution obtained through sequential time integration with Richardson extrapolation in a finite number of iterations. This means that the stability and error analysis of RE-based time integration methods is in no way altered by the τ-MGRIT algorithm. That is, τ-MGRIT will return a stable solution with improved convergence order if sequential time integration with RE also returns a stable solution with improved convergence order. This ability to exploit existing analysis is a major advantage of the non-intrusive MGRIT approach. As such, determining the applicability of RE-based time integration to a particular problem is beyond the scope of this paper. Rather, our goal is to show that, when applicable, RE time integration can be used with MGRIT to dramatically improve the APCC of the MGRIT algorithm.

REMARK 3.1 (Multilevel extension). The two-level τ-MGRIT algorithm can be extended to multiple levels in several ways. Of course, with more levels comes more opportunities to further enhance the solution with RE. However, in a parallel setting, the extra coarse-grid time steps required for multilevel RE are not free. For this reason, the multilevel τ-MGRIT algorithm does not combine multiple stages of RE-based time integration. Instead, the multilevel τ-MGRIT algorithm uses the two-level τ-MGRIT algorithm on the fine grid but solves the coarse-grid problem using a recursive call to the original MGRIT algorithm. This is possible because the coarse-grid equations are identical for τ-MGRIT and MGRIT.

4. Computational cost of the τ-MGRIT algorithm.

The accuracy improvements of τ-MGRIT would be of no use if the cost of the method increased commensurately. To study this issue, we consider the cost of (τ-)MGRIT, which is a function of the cost of a V-cycle and the number of iterations required for convergence.

4.1. Cost of a single V-cycle.

The main computational kernel of all MGRIT-based algorithms is the evaluation of Φ. In terms of Φ-evaluations, the only difference between the two-level τ-MGRIT and MGRIT algorithms is the collection of extra time integration steps completed during fine-grid C-relaxation. To be precise, N_t/m extra time steps are completed per V-cycle for the τ-MGRIT algorithm when compared to the MGRIT algorithm if FCF-relaxation is used. If p processors are used, then this equates to a maximum of $\lceil N_t/(mp) \rceil$ extra time steps per processor. The cost associated with scaling the RHS is negligible. If F-relaxation is used, then there is no difference in the number of Φ-evaluations completed per iteration for the two methods.

As mentioned in Remark 3.1, the multilevel extension of the two-level τ-MGRIT algorithm uses the original MGRIT algorithm to solve the coarse-grid problem. This means that the extra time steps completed on the fine grid are the only extra time steps taken, per iteration, by the τ-MGRIT algorithm when compared to the original MGRIT algorithm. In the following section, a bound for the convergence rate is derived for the two-level τ-MGRIT algorithm, and it is compared to a similar bound derived in [5] for the original MGRIT algorithm. With this bound, we can approximate the number of iterations required for the convergence of both methods.

REMARK 4.1. As a practical point, the XBraid implementations of the MGRIT and τ-MGRIT algorithms use the norm of the residual, i.e., $||\bar{\mathbf{g}} - A\mathbf{u}||$, to measure the error and determine when to stop the iteration. Unlike the original MGRIT algorithm, τ-MGRIT does not require a direct calculation of the residual. However, it can easily be obtained at a cost of one vector addition per C-point.

4.2. Convergence analysis.

In the previous section, it was demonstrated that the computational cost for τ-MGRIT and MGRIT is very similar. In this section, we compare the algebraic convergence rates of the two methods. The analysis in this section develops a

ETNA
Kent State University and
Johann Radon Institute (RICAM)

convergence bound for the τ-MGRIT algorithm applied on an evenly spaced time grid, with no spatial coarsening, and a time integration operator that is both linear and constant in time. For comparison, we first give two analytic bounds for the convergence rate of the standard MGRIT algorithm from [5].

Consider solving equation (2.5) on the temporal grid shown in Figure 2.1, and assume $g_j = 0$ for all j. The case with nonzero g_j is similar. Next, assume $||\Phi|| \leq 1$, i.e., Φ is a convergent time integration scheme, and that Φ and Φ_Δ are diagonalized by the same unitary transform,

$$\hat{\Phi} = X^*\Phi X = \text{diag}(\lambda_1, \ldots, \lambda_{N_x}) \quad \text{and} \quad \hat{\Phi}_\Delta = X^*\Phi_\Delta X = \text{diag}(\mu_1, \ldots, \mu_{N_x}), \tag{4.1}$$

where N_x represents the number of spatial unknowns associated with each time step, $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{N_x})$, and $X^*X = I$.

Finally, let E_Δ^F and E_Δ^{FCF} denote the C-point error propagation matrices for MGRIT with F- and FCF-relaxation, and define $\mathbf{e}_j$ to be the error at $t = t_j$ between the current approximation of the solution $\hat{\mathbf{u}}_j$ and the exact discrete solution $\mathbf{u}_j$. Then, the global space-time error vector at the C-points, $\bar{\mathbf{e}} = [\mathbf{e}_0^T, \mathbf{e}_m^T, \ldots, \mathbf{e}_{N_t/m}^T]^T$, satisfies

$$||E_\Delta^F \bar{\mathbf{e}}||_2 \leq \max_{\omega=1,2,\ldots,N_x} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1 - |\mu_\omega|^{N_t/m}}{(1 - |\mu_\omega|)} \right\} ||\bar{\mathbf{e}}||_2, \tag{4.2}$$

for MGRIT with F-relaxation and

$$||E_\Delta^{FCF} \bar{\mathbf{e}}||_2 \leq \max_{\omega=1,2,\ldots,N_x} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1 - |\mu_\omega|^{N_t/m}}{(1 - |\mu_\omega|)} |\lambda_\omega|^m \right\} ||\bar{\mathbf{e}}||_2, \tag{4.3}$$

for MGRIT with FCF-relaxation [5]. The fraction in this bound, $(1-|\mu_\omega|^{N_t/m})/(1-|\mu_\omega|)$, is the result of a geometric sum. As $\mu_\omega \to 1$, this terms grows like N_t. Fortunately for many common time integration schemes, this is compensated because $|\lambda_\omega^m - \mu_\omega| \to 0$ as $N_t \to \infty$ (see Section 4.2.1).

Next, similar estimates for the τ-MGRIT algorithm are derived. Following F-relaxation, the residual at $t = mj$ for the τ-MGRIT method is

$$\mathbf{r}_{mj} = -[A(\hat{\mathbf{u}})]_{mj} = -\hat{\mathbf{u}}_{mj} + (a\Phi^m - b\Phi_\Delta)\hat{\mathbf{u}}_{m(j-1)},$$

where we have assumed, without loss of generality, that $\bar{\mathbf{g}} = \mathbf{0}$. The exact solution of equation (3.2) satisfies $\mathbf{u}_{mj} = (a\Phi^m - b\Phi_\Delta)\mathbf{u}_{m(j-1)}$. Hence, the residual can be rewritten in terms of the error $\mathbf{e}_{mj} = \mathbf{u}_{mj} - \hat{\mathbf{u}}_{mj}$ as

$$\mathbf{r}_0 = \mathbf{e}_0, \tag{4.4}$$

$$\mathbf{r}_{mj} = \mathbf{e}_{mj} - (a\Phi^m - b\Phi_\Delta)\mathbf{e}_{m(j-1)}, \tag{4.5}$$

for $j = 1, 2, \ldots, N_t/m$. For two-level τ-MGRIT, the coarse-grid problem is solved using forward substitution. This is an $O(N_t)$ sequential operation. Multilevel τ-MGRIT, which instead solves the coarse-grid problem recursively and scalably, is discussed in Remark 3.1. The exact coarse-grid correction, $\mathbf{c}_{mj}$, is given by

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{r}_0, \\ \mathbf{c}_{mj} &= \Phi_\Delta(\mathbf{c}_{m(j-1)}) + \mathbf{r}_{mj}, \\ &= \Phi_\Delta^j(r_0) + \Phi_\Delta^{j-1}(\mathbf{r}_m) + \cdots + \mathbf{r}_{mj}, \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

for $j = 1, 2, \ldots, N_t/m$. Substituting for each $\mathbf{r}$ with a subscript using equations (4.4) and (4.5) gives

$$\begin{aligned}\mathbf{c}_0 &= \mathbf{e}_0,\\ \mathbf{c}_{mj} &= \Phi_\Delta^j(\mathbf{e}_0) + \Phi_\Delta^{j-1}(\mathbf{e}_m + (b\Phi_\Delta - a\Phi^m)\mathbf{e}_0)\\ &\quad + \Phi_\Delta^{j-2}(\mathbf{e}_{2m} + (b\Phi_\Delta - a\Phi^m)\mathbf{e}_m) + \ldots\\ &\quad + \mathbf{e}_{mj} + (b\Phi_\Delta - a\Phi^m)\mathbf{e}_{m(j-1)}.\end{aligned}$$

Collecting like terms and noting that $b + 1 = a$ gives

$$\begin{aligned}\mathbf{c}_0 &= \mathbf{e}_0,\\ \mathbf{c}_{mj} &= a\Phi_\Delta^{j-1}(\Phi_\Delta - \Phi^m)\mathbf{e}_0 + a\Phi_\Delta^{j-2}(\Phi_\Delta - \Phi^m)\mathbf{e}_m + \ldots\\ &\quad + a(\Phi_\Delta - \Phi^m)\mathbf{e}_{m(j-1)} + \mathbf{e}_{mj}.\end{aligned}$$

After the correction, the updated error at the C-points $\mathbf{f}_{mj} = \mathbf{e}_{mj} - \mathbf{c}_{mj}$ satisfies:

$$\mathbf{f}_0 = 0, \tag{4.6}$$

$$\mathbf{f}_{mj} = a\sum_{q=0}^{j-1} \Phi_\Delta^{j-1-q}(\Phi^m - \Phi_\Delta)\mathbf{e}_{mq}, \tag{4.7}$$

for $j = 1, 2, \ldots, N_t/m$. The unitary transformation given in equation (4.1) can be used to decompose $\mathbf{e}_{mj}$ as $\mathbf{e}_{mj} = \sum_{\omega=1}^{N_x} \hat{e}_{j,\omega}\,\mathbf{x}_\omega$ and $\mathbf{f}_{mj}$ as $\mathbf{f}_{mj} = \sum_{\omega=1}^{N_x} \hat{f}_{mj,\omega}\,\mathbf{x}_\omega$. Substituting these decompositions into equation (4.6) and (4.7) gives

$$\begin{aligned}\hat{f}_{0,\omega} &= 0,\\ \hat{f}_{mj,\omega} &= a\sum_{q=0}^{j-1} \mu^{j-1-q}(\lambda_\omega^m - \mu_\omega)\hat{e}_{mq,\omega},\end{aligned}$$

for $j = 1, 2, \ldots, N_t/m$ and $\omega = 1, 2, \ldots, N_x$. Defining $\hat{\mathbf{e}}_\omega = (\hat{e}_{0,\omega}, \hat{e}_{m,\omega} \ldots \hat{e}_{N_T,\omega})^T$ and $\hat{\mathbf{f}}_\omega = (\hat{f}_{0,\omega}, \hat{f}_{m,\omega} \ldots \hat{f}_{N_T,\omega})^T$ gives $\hat{\mathbf{f}}_\omega = E_\omega^F \hat{\mathbf{e}}_\omega$, where

$$E_\omega^F = a(\lambda_\omega^m - \mu_\omega)\begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ \mu_\omega & 1 & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \mu_\omega^{N_T/m-1} & \ldots & \mu_\omega & 1 & 0 \end{bmatrix}.$$

The reader will note that the matrix $\mathbf{E}_\omega^F$ is nilpotent. As such, τ-MGRIT with F-relaxation is guaranteed to converge to the exact solution in at most N_t/m iterations. That is, after at most N_t/m iterations, the error in each eigenmode is guaranteed to be zero. A similar convergence guarantee holds for τ-MGRIT with FCF-relaxation and for the original MGRIT algorithm with both F- and FCF-relaxation. However, speedups are obtained when MGRIT converges in $O(1)$ iterations to within the desired halting tolerance.

The reader should also notice how the eigenmodes are decoupled. This allows us to calculate a convergence bound for each eigenmode independently, where the overall convergence bound will be given by the maximum bound over all the eigenmodes. Following [5], a convergence bound for each eigenmode is then given by

$$||E_\omega||_1 = ||E_\omega||_\infty = a|\lambda_\omega^m - \mu_\omega|\frac{1 - |\mu_\omega|^{N_t/m-1}}{1 - |\mu_\omega|}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

(a) SDIRK-1

$$\begin{array}{c|cc} 1-\alpha & 1-\alpha & \\ \alpha & 2\alpha-1 & 1-\alpha \\ \hline & 1/2 & 1/2 \end{array}$$

(b) SDIRK-2, $\alpha = 1/\sqrt{2}$

$$\begin{array}{c|ccc} a & a & & \\ c & c-a & a & \\ 1 & b & 1-a-b & a \\ \hline & b & 1-a-b & a \end{array}$$

(c) SDIRK-3, $a = 0.435866\ldots$, $b = 1.20846\ldots$, $c - 0.717933\ldots$

FIG. 4.1. *Butcher tableaux for SDIRK-1, SDIRK-2 and SDIRK-3.*

Next, it is straightforward to apply Theorem 3.3 from [5] to obtain a bound for the global error vector $\bar{\mathbf{e}} = [\mathbf{e}_0^T, \mathbf{e}_m^T, \ldots, \mathbf{e}_{N_T m}^T]^T$ for τ-MGRIT with F-relaxation:

$$||\bar{E}_\Delta^F \bar{\mathbf{e}}||_2 \leq \max_{\omega=1,2,\ldots,N_x} \frac{m^{k_g}}{m^{k_g}-1} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1-|\mu_\omega|^{N_T/m}}{(1-|\mu_\omega|)} \right\} ||\bar{\mathbf{e}}||_2, \tag{4.8}$$

where $||E_\Delta^F \mathbf{e}||_2$ is defined as in equation (4.2). A similar calculation gives the global error bound for τ-MGRIT with FCF-relaxation:

$$||\bar{E}_\Delta^{FCF} \bar{\mathbf{e}}||_2 \leq \max_{\omega=1,2,\ldots,N_x} \frac{m^{k_g}}{m^{k_g}-1} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1-|\mu_\omega|^{N_T/m}}{(1-|\mu_\omega|)} \left| \frac{m^{k_g}\lambda_\omega^m - \mu_\omega}{m^{k_g}-1} \right| \right\} ||\bar{\mathbf{e}}||_2. \tag{4.9}$$

The F-relaxation bound (4.8) is a scaling by $m^{k_g}/(m^{k_g}-1)$ of the MGRIT bound (4.2), but there does not exist such a simple scaling between the error bounds for τ-MGRIT (4.9) and MGRIT (4.3) when FCF-relaxation is used.

4.2.1. Singly diagonally implicit Runge-Kutta (SDIRK) methods. In order to understand these bounds, we examine the convergence rate of the τ-MGRIT algorithm for various coarse- and fine-grid time integrators. Consider solving

$$\mathbf{u}' + G\mathbf{u} = \mathbf{b},$$

where G is a symmetric positive definite matrix with real eigenvalues γ_ω, using the first-, second-, and third-order singly diagonally implicit Runge-Kutta (SDIRK) time-integration schemes. Note that an operator with real eigenvalues was chosen because the scope of this paper is parabolic problems. However, there is no reason that this analysis could not be applied to complex eigenvalues in future work. Figure 4.1 shows the Butcher tableaux for each of the SDIRK methods.

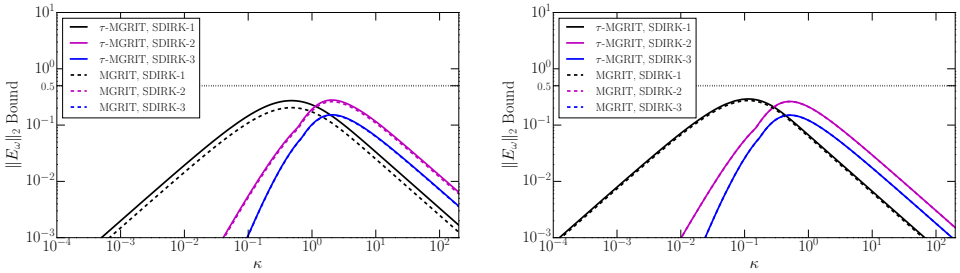
REMARK 4.2. The first order SDIRK-1 method is equivalent to the backward Euler method. In this case, the fine- and coarse-grid operators are $\Phi = (I + \delta t G)^{-1}$ and $\Phi_\Delta = (I + m\delta t G)^{-1}$, respectively. Likewise, the fine- and coarse-grid eigenvalues are

$$\lambda_\omega = \frac{1}{(1+\delta t \gamma_\omega)}, \qquad \mu_\omega = \frac{1}{(1+m\delta t \gamma_\omega)},$$

respectively. Hence, for $\omega = 1, 2, \ldots, N_x$, it follows that $\gamma_\omega \in (0, \infty)$, $\lambda_\omega \in (0, 1)$, and $\mu_\omega \in (0, 1)$. Notice that the eigenvalues can get very close to 1. In that limit, the fraction in the error bound (equations (4.8) and 4.9)) grows like $N_t = T/\delta t$. Fortunately, the first term, $|\lambda_\omega^m - \mu_\omega|$ tends to zero with an order of at least $O(\delta t^2)$.

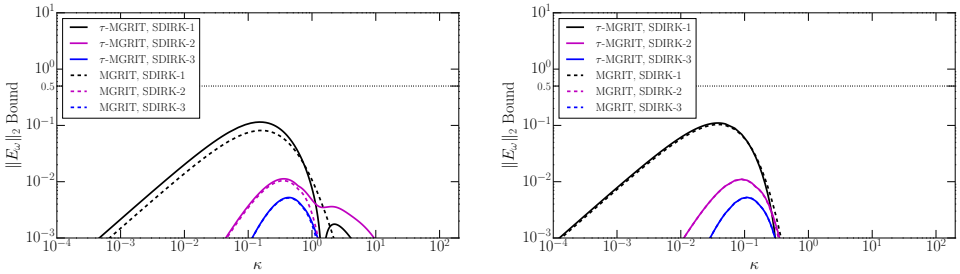
Figures 4.2 and 4.3 display the convergence bound $||E_\omega||_2$ versus $\kappa = \delta t|\gamma_\omega|$, for $\gamma_\omega > 0$ and for F- and FCF-relaxation, respectively. Two coarsening factors are considered in each plot, $m = 4$ on the left and $m = 16$ on the right. Solid lines represent the bounds derived

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $m = 4$ (b) $m = 16$

FIG. 4.2. *$||E_\omega||_2$ vs $\kappa = \delta t \gamma_\omega$ for the SDIRK algorithms with τ-MGRIT and F-relaxation. Dashed lines represent the convergence bounds for MGRIT; solid lines are the convergence bounds for τ-MGRIT.*



(a) $m = 4$ (b) $m = 16$

FIG. 4.3. *$||E_\omega||_2$ vs $\kappa = \delta t \gamma_\omega$ for the SDIRK algorithms with τ-MGRIT and FCF-relaxation. Dashed lines represent the convergence bounds for MGRIT; solid lines are the convergence bounds for τ-MGRIT.*

for the τ-MGRIT algorithm, and dashed lines represent the bounds for the standard MGRIT algorithm. The overall convergence bound for a dataset is the maximum value attained on the y-axis. For both F- and FCF-relaxation, there is only a slight convergence degradation for τ-MGRIT when $m = 4$, which is due to the term $a = m^{k_g}/(m^{k_g} - 1)$ in the convergence bounds. Notably, this degradation decreases for higher-order methods (larger k_g). While there is almost no degradation when $m = 16$, we will see that a larger m lead to worse discretization errors in the numerical tests.

Importantly, these bounds show that the τ-MGRIT algorithm is capable of increasing the convergence order of the underlying time-integration scheme without significantly increasing the convergence factor, including higher-order methods. Combined with the limited increase in time step evaluations, this gives the τ-MGRIT algorithm a sizable advantage when considering the APCC of the method. In the following sections, several numerical examples are used to compare the APCC of the τ-MGRIT and MGRIT algorithms.

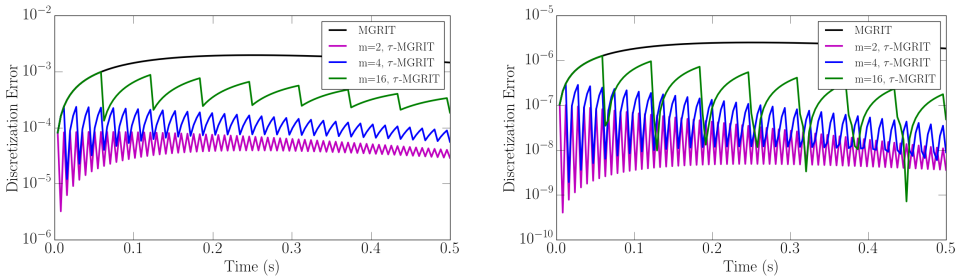
5. τ-MGRIT results.

5.1. Numerical example: first-order ODE.

Consider the following IVP:

$$y' + 4y = 1 - t, \qquad t \in [0, 1], \quad \text{with } y(0) = 1, \tag{5.1}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) SDIRK-1 (b) SDIRK-2

FIG. 5.1. *Error against time for the solution of equation (5.1) using τ-MGRIT and 128 time steps. The Richardson-based error corrections occur only at the C-points, giving rise to the sawtooth appearance of the plots.*

and the exact solution $y(t) = (1/16)(-4t + 11e^{-4t} + 5)$. Figure 5.1 displays the error against time for the solution found when solving this equation using τ-MGRIT with the SDIRK-1 and SDIRK-2 methods and 128 time steps. The error profile appears to be discontinuous because the τ-MGRIT algorithm corrects the solution at each C-point, allowing the error to increase at F-points. Notice that in all but a few local regions (i.e., near $t = 0.45$ for τ-MGRIT with SDIRK-2), the accuracy of the τ-MGRIT solution becomes worse as the coarsening factor m increases. This is not surprising because letting $k_g = k_0 = 1$ in equation (2.3) implies that the lowest-order error term is $mC_1\delta t^2$, i.e., the error increases linearly with m. Any exceptions are likely due to a fortuitous cancellation of errors. This result is unfortunate for the low-order SDIRK-1 method since the analysis in the previous section predicts that decreasing m leads to a degradation in the convergence rate. However for the higher-order SDIRK-3 method, the analysis predicts that the convergence rates are almost identical for τ-MGRIT and MGRIT and small m, hence, making this less of a problem.

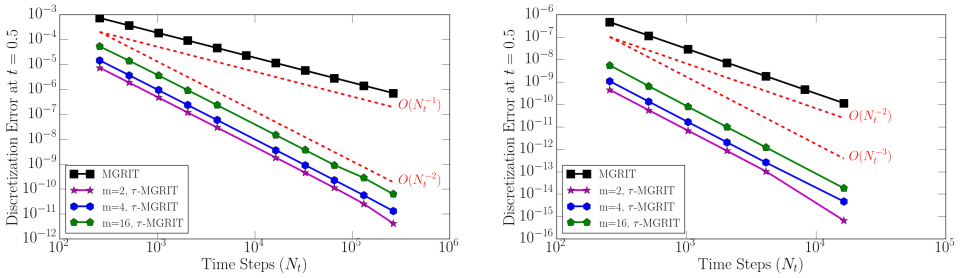
In Figure 5.2, the error with respect to the time step size at $t = 0.5$ is presented, highlighting the algorithmic scalability of the τ-MGRIT algorithm. For SDIRK-1, a globally first-order method, τ-MGRIT scales with an accuracy equivalent to a second-order method. Likewise, τ-MGRIT improves the accuracy of the second-order SDIRK-2 algorithm to that of a third-order method. It is important to remember that the improved error scaling presented in Figure 5.2 can be entirely attributed to the use of RE-based time integration. That is, identical results would be obtained using a sequential two-stage RE-based time integrator. The most significant benefit of the τ-MGRIT algorithm, namely that it provides a cost-efficient mechanism for improving the APCC of the MGRIT algorithm, is covered in the following section.

5.2. One-dimensional heat equation. In order to analyze a representative PDE, τ-MGRIT is applied to the one-dimensional (1D) heat equation,

$$\begin{aligned} u_t - u_{xx} &= f(x,t), && \forall x \in [0,\pi],\ t \in [0,2\pi], \\ u(0,t) = g_1(t),\ u(\pi,t) &= g_2(t), && \forall t \in [0,2\pi], \\ u(x,0) &= h(x), && \forall x \in [0,\pi], \end{aligned}$$

with Dirichlet boundary conditions. The source term $f(x,t)$ as well as the boundary terms $g_1(t)$ and $g_2(t)$ and the initial condition $h(x)$ are chosen to yield the exact solution $u_{exact}(x,t) = \sin(x)\cos(t)$. The temporal discretization uses SDIRK-1, while the spatial

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) SDIRK-1 (b) SDIRK-2

FIG. 5.2. *Algorithmic scaling study for τ-MGRIT applied to equation (5.1). The τ-MGRIT algorithm significantly increases the accuracy of the final solution.*

discretization uses standard second-order central differences. A direct spatial solver was used, limiting parallelism to the temporal domain.[2]

Figure 5.3a displays the error against time for τ-MGRIT with SDIRK-1, 128 time steps, and 16385 spatial grid points. The error represents the Euclidean norm of the difference between the exact and the approximate space-time solutions after the final MGRIT iteration. A relatively large number of spatial grid points is used so that the second-order errors associated with the spatial discretization did not dominate the overall error.

The reader will notice that generally the error for τ-MGRIT is significantly smaller than that for MGRIT, except for points near $t = 2$ and $t = 5$. As discussed above, the τ-MGRIT algorithm converges to the exact solution using a sequential implementation of RE. Assessing the suitability of RE for solving particular ODEs and PDEs is beyond the scope of this paper.

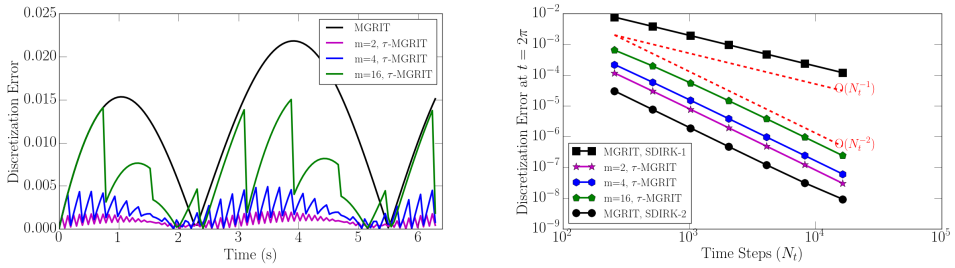
Figure 5.3b displays the error at final time $t = 2\pi$. As shown in Section 5.1, the τ-MGRIT algorithm returned a solution with second-order accuracy, the magnitude of which depends on the coarsening factor. Figure 5.3b also depicts the error at final time for a time integration with SDIRK-2. The accuracy of the solution obtained with τ-MGRIT is lower than that of the SDIRK-2 method. This suggests that, if accuracy is the most important aspect of the solution, then the user would be better off switching to a higher-order time integration scheme.

It is important to note however, that a single step of SDIRK-2 is twice as expensive as a single time step with SDIRK-1. As such, a direct comparison between the two methods should also take into account the cost of the method. A study comparing the APCC of the two approaches is given in Section 5.4, with the conclusion being that the higher-order SDIRK-2 method outperforms τ-MGRIT with SDIRK-1 in terms of both accuracy and APCC. However, the study also shows that τ-MGRIT with SDIRK-2 outperforms MGRIT with SDIRK-2, highlighting the fact that τ-MGRIT is a simple, cost-effective way to improve the accuracy of the MGRIT algorithm, no matter what the order of the time integration scheme is. That is to say, if RE-based time integration is appropriate for a problem, then τ-MGRIT can be used to improve the APCC over the final solution found with the standard MGRIT algorithm.

REMARK 5.1. While we only consider the classic MGRIT test problem, the 1D heat equation, we note that there are no expected changes in the behavior, either theoretically or practically, when moving to 2D or 3D. Previous MGRIT work, e.g., [5, 7], indicates that the dimensionality of the heat equation does not affect the behavior of MGRIT. Moreover,

[2]The focus of this paper is a description and validation of τ-MGRIT, hence we do not consider spatial parallelism. Generally, the speedup due to time parallelism is not affected by spatial parallelism and is in addition to any speedup from spatial parallelism.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Error against time for $N_t = 128$

(b) Error at $t = 2\pi$ against N_t

FIG. 5.3. *Error analysis for the solution of the 1D heat equation found using τ-MGRIT with 16384 spatial grid points and SDIRK-1. Figure (a) shows the error at the final iteration when using the MGRIT and τ-MGRIT algorithms; Figure (b) shows that τ-MGRIT with SDIRK-1 scales as a second-order method.*

the rigorous theoretical results from Section 4.2 indicate consistent performance across all problems with real-valued spatial eigenvalues (not just the heat equation) and the considered time-stepping schemes.

5.2.1. Convergence rate of two-level τ-MGRIT. Tables 5.1 and 5.2 illustrate the convergence rates of an SDIRK-1 implementation of τ-MGRIT and MGRIT with both F- and FCF-relaxation. The final column, labeled "Est", shows the estimated convergence factor as derived in equations (4.8) and (4.9). These numerically determined convergence factors represent the average convergence factor over the last 5 iterations for runs using two-level MGRIT and τ-MGRIT, solved to a residual tolerance of 10^{-10} with a random initial guess.

TABLE 5.1
Two-level numerical convergence factors for τ-MGRIT (yes in first column) and MGRIT (no in first column) with F-relaxation. The final column shows the error estimate derived in Section 4.2 and agrees well with the observed convergence.

F-Relaxation		Time Steps						
τ	m	256	512	1024	2048	4096	8192	Est.
No	2	0.1206	0.1216	0.122	0.1222	0.1224	0.1216	0.1249
Yes	2	0.2416	0.2442	0.2446	0.2446	0.2450	0.2438	0.2499
No	4	0.1956	0.1958	0.199	0.1994	0.1994	0.1996	0.2038
Yes	4	0.2606	0.2614	0.2652	0.2658	0.2658	0.2662	0.2719
No	16	0.1409	0.2548	0.258	0.2602	0.2662	0.2668	0.2729
Yes	16	0.1511	0.2718	0.2756	0.2776	0.2842	0.2848	0.2929

These tables highlight two important facts. First in all cases, the analysis in the previous section provides a good bound for the convergence rate. Second, the convergence rate of both the τ-MGRIT and MGRIT algorithms does not vary noticeably with respect to the problem size. In other words, the two-level τ-MGRIT algorithm exhibits good weak scalability with respect to iteration counts.

5.3. Multilevel results. As discussed in Remark 3.1, the multilevel τ-MGRIT algorithm solves the coarse-grid problem using standard multilevel MGRIT, compared to the two-level approach where direct sequential time integration is used. Thus, the number of total MGRIT

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 5.2
Two-level numerical convergence factors for τ-MGRIT (yes in first column) and MGRIT (no in first column) with FCF-relaxation. The final column indicates the error estimate derived in Section 4.2.

FCF-Relaxation		Time Steps						
τ	m	256	512	1024	2048	4096	8192	Est.
No	2	0.0480	0.0524	0.0520	0.0519	0.0520	0.0520	0.0527
Yes	2	0.0938	0.0982	0.0975	0.0968	0.0972	0.0982	0.1547
No	4	0.0740	0.0749	0.0802	0.0794	0.0787	0.0792	0.0812
Yes	4	0.0924	0.0981	0.1040	0.1030	0.1020	0.1026	0.1147
No	16	0.0114	0.0367	0.0922	0.0961	0.1030	0.1020	0.1038
Yes	16	0.01231	0.0373	0.0966	0.1020	0.1096	0.1092	0.1157

levels increases with the problem size in the multilevel case, so that a fixed coarsest-grid size of 2 is always maintained. Table 5.3 presents the number of iterations required for the multilevel MGRIT and τ-MGRIT algorithms to converge to within a relative residual tolerance of 10^{-10} when solving the 1D heat equation from the previous section with a random initial guess. In each test, the spatial problem size was fixed at 16384 spatial unknowns as described earlier.

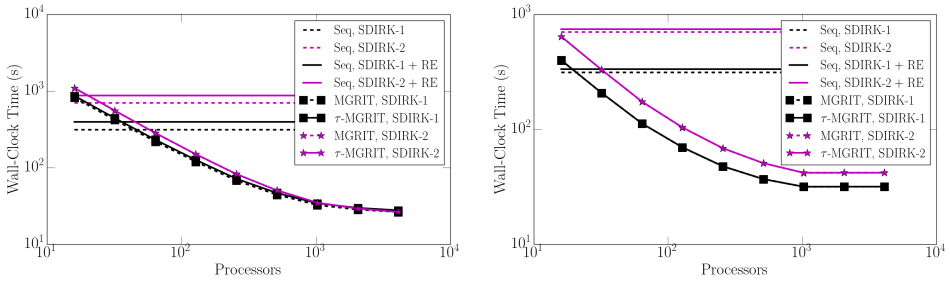
TABLE 5.3
Iterations required to reduce the residual by 10^{-10} for the multilevel τ-MGRIT and MGRIT algorithms when used with SDIRK-1. The first column indicates whether τ-MGRIT (Yes) or MGRIT (No) was used. Iteration counts for both τ-MGRIT and MGRIT are bounded independently of the problem size.

Method			Time Steps					
τ	Relax	m	256	512	1024	2048	4096	8192
No	F	4	18	20	21	23	23	24
Yes	F	4	21	22	24	24	25	25
No	F	16	15	18	18	18	18	18
Yes	F	16	15	18	18	19	19	19
No	FCF	4	10	11	11	11	12	12
Yes	FCF	4	11	12	12	12	12	12
No	FCF	16	8	9	11	11	11	11
Yes	FCF	16	8	9	11	11	12	12

In general, the MGRIT algorithm appears to converge faster than the τ-MGRIT algorithm. On the other hand, the τ-MGRIT algorithm returns a solution with improved accuracy. When one takes this into account (see Section 5.4), τ-MGRIT becomes more efficient than MGRIT. Either way, the iteration counts associated with the FCF-based variant of τ-MGRIT appear to be scalable and bounded independently of the problem size.

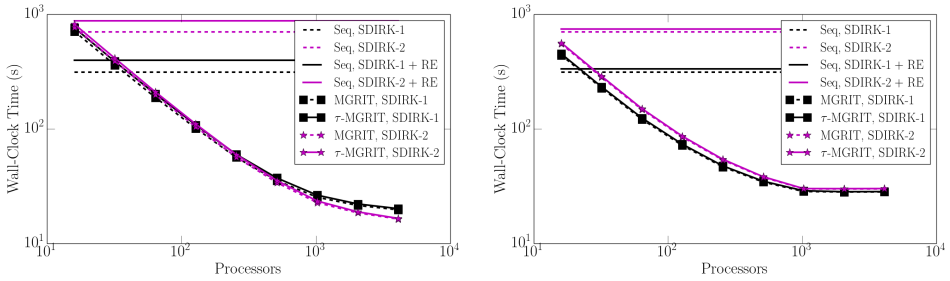
5.4. Strong scaling analysis. As already mentioned, MGRIT and most parallel-in-time schemes require more computational work than sequential time-stepping, but they allow for greater concurrency. This combination of increased overall cost and increased potential for concurrency become manifest in a crossover point at which the additional temporal concurrency eventually overcomes the additional work. These crossover points will be visible during our parallel scaling studies and are important markers indicating where there is a benefit in using MGRIT.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $m = 4$ (b) $m = 16$

FIG. 5.4. *Strong scaling study for τ-MGRIT applied to the 1D heat equation with F-relaxation. Black lines represent SDIRK-1 methods. Magenta lines represent results found using SDIRK-2.*



(a) $m = 4$ (b) $m = 16$

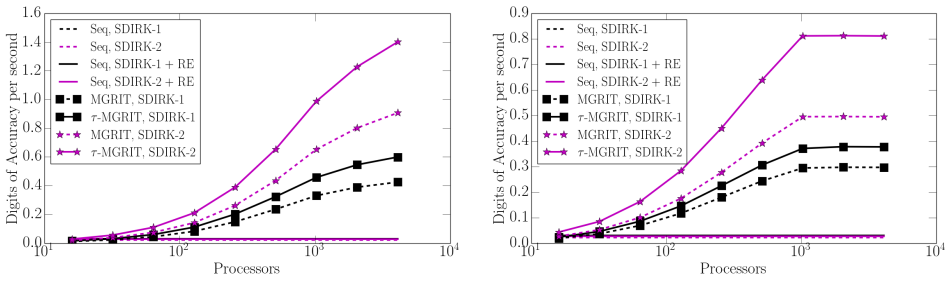
FIG. 5.5. *Strong scaling study for τ-MGRIT applied to the 1D heat equation with FCF-relaxation. Reducing the coarsening factor delays the crossover point, but provides more opportunities for temporal concurrency and hence, maximum parallel speedup.*

Figures 5.4 and 5.5 present a strong scaling study for τ-MGRIT applied to the 1D heat equation with F- and FCF-relaxation, respectively. In all cases, the problem size included 16384 time steps and 16384 spatial unknowns. For these tests, the forcing function, $f(x,t)$ and the associated boundary conditions were chosen to ensure an exact solution of $u(x,t) = \sin(x)\cos(10t)$. The machine used for all numerical tests was Vulcan, an IBM BG/Q machine at LLNL.

The time-to-solution and the scaling behavior for τ-MGRIT and MGRIT is very similar. As expected, both τ-MGRIT and MGRIT performed better when FCF-relaxation is used. Comparing the two figures shows that a larger coarsening factor ($m = 16$) leads to an earlier crossover point. Reducing the coarsening factor delays the crossover point but allows for greater temporal concurrency and hence higher speedups. Overall, the largest speedup was obtained for $m = 4$ with 16384 temporal processors. The speedup factor was 19 when comparing τ-MGRIT with FCF-relaxation to sequential time integration without RE and 20 when comparing to sequential time integration with RE.

The strength of the τ-MGRIT algorithm is that it returns a solution with improved accuracy. To illustrate this, the APCC of each algorithm was calculated. For this comparison, cost is measured in seconds, and the accuracy is defined to be $|\log(1/err)|$ where err is the error at time $t = 2\pi$. In this way, APCC measures the digits of accuracy per second of runtime.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $m = 4$ (b) $m = 16$

FIG. 5.6. *Digits of accuracy per second for the τ-MGRIT and MGRIT algorithms applied to the 1D heat equation with FCF-relaxation. Black lines represent SDIRK-1 methods. Magenta lines represent results found using SDIRK-2.*

Figure 5.6 displays the digits of accuracy per second for the τ-MGRIT and MGRIT algorithms with FCF-relaxation. In all cases, as anticipated by the analysis in the previous section, the switch from MGRIT to τ-MGRIT (dashed lines to solid lines) dramatically increased the APCC of the associated method. The APCC for the results using F-relaxation are similar.

Another, arguably more interesting comparison is found by considering the results for second-order MGRIT with SDIRK-2 (dotted magenta lines) and second-order τ-MGRIT with SDIRK-1 (solid black lines). The analysis presented in Section 4.2 suggests that the convergence factor of MGRIT with SDIRK-2 will be about half that of the τ-MGRIT algorithm with SDIRK-1. On the other hand, each SDIRK-2 time integration step requires two spatial inverses, compared to one spatial inverse for SDIRK-1. However, MGRIT with SDIRK-2 still outperforms τ-MGRIT with SDIRK-1. As such, it appears that switching to a higher-order time integration scheme is still a better approach to increase the APCC of the method. However in that case, as shown by the results for τ-MGRIT with SDIRK-2, τ-MGRIT can then be used to further increase the APCC of that method. Overall, τ-MGRIT is an excellent approach for improving the APCC of MGRIT by allowing for either greater accuracy with the same runtime or a similar accuracy in a reduced runtime by using fewer time steps.

6. Extension to non-uniform time grids. The analysis and derivation presented above was developed under the assumption that the temporal grids are uniform. In what follows, we modify the derivation of the τ-MGRIT algorithm to accommodate non-uniform temporal grids. To do this, the key assumption that must be made is that the local truncation errors accumulate linearly across each temporal coarse-grid interval. This is not an uncommon assumption in the sequential time integration community. In fact, this is the assumption underlying the common statement that the global order of a time integration scheme is one less than the local order. Despite this, it is important to note that the use of RE on non-uniform temporal grids is somewhat unconventional. In some cases, this may limit the accuracy improvements that can be achieved with the τ-MGRIT algorithm. This is an unfortunate compromise that must be made to accommodate unrestricted non-uniform temporal grids.

The alternative, albeit more restrictive, approach is to allow for a non-uniform coarse grid but to only allow uniform refinement by the coarsening factor across each coarse-grid interval. This ensures uniform time steps across each coarse-grid interval while still allowing for some level of non-uniformity in the temporal grid (see Figure 6.1). This approach, referred to as the "semi-uniform" approach, does not require any modification of the τ-MGRIT algorithm. Results comparing both approaches are presented in Section 6.1.1.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

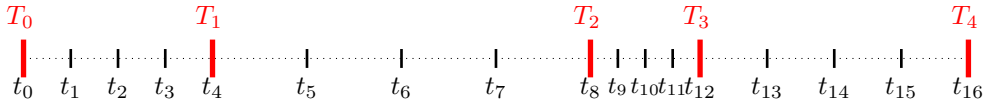


FIG. 6.1. *Example of a semi-uniform temporal mesh. Coarse-grid points (red) are distributed non-uniformly across the temporal domain, but the time step size is fixed inside each coarse-grid interval (for comparison, see the full non-uniform grid in Figure 6.2).*

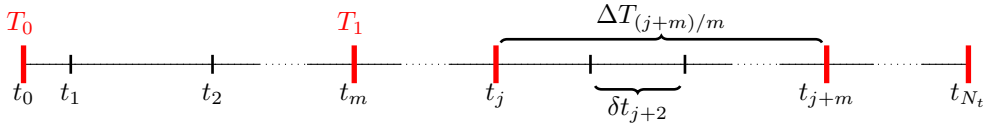


FIG. 6.2. *Fully non-uniform fine- and coarse-grid temporal meshes.*

6.1. τ-MGRIT for non-uniform temporal grids. Define a non-uniform temporal grid with time steps δt_j and nodes $t_0 = 0$, $t_j = t_{j-1} + \delta t_j$, $j = 1, 2, \ldots, N_t$. Further, define a coarse temporal grid with time steps $\Delta T_j = \sum_{i=m(j-1)+1}^{mj} \delta t_i$ and nodes $T_0 = 0$, $T_j = T_{j-1} + \Delta T_j$, $j = 1, 2, \ldots, N_{t/m}$, for some fixed coarsening factor m. This is depicted in Figure 6.2.

Let $\Phi_j = \Phi_j(u_{j-1}, t_j, \delta t_j)$ represent the time integration operator at $t = t_j$. Next, assume the error introduced by a single time step with Φ from $t = t_{j-1}$ to $t = t_j$ is of the form $C_j \delta t_j^{k_l}$ where C_j is a constant. Here k_l represents the *local* order of the time integration scheme. Finally, assume that each δt_j is small enough and that the solution $u(x,t)$ is sufficiently smooth enough such that C_j is constant across the given coarse-grid interval. Then, the global error introduced by time integration across the jth coarse-grid interval on the non-uniform temporal mesh is

$$u_{f,mj} - u(t_{mj}) = C_{mj} \sum_{i=0}^{m-1} \delta t_{mj-i}^{k_l}.$$

Likewise, the error introduced by a single time step on the coarse grid is

$$u_{c,mj} - u(t_{mj}) = C_{mj} \Delta T_j^{k_l}.$$

As in Section 2, these equations can be combined to eliminate C_{mj} and give an enhanced solution

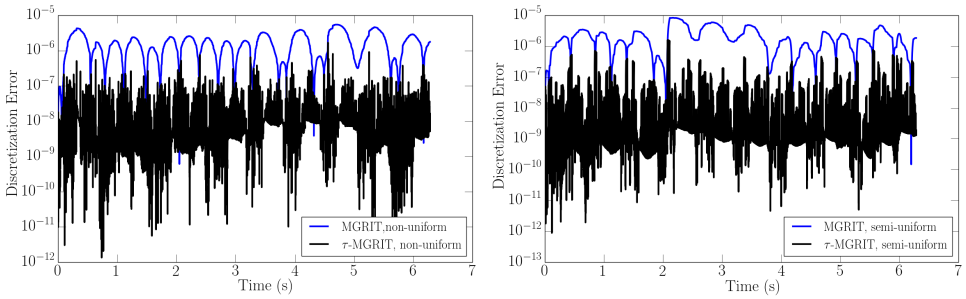
$$u_{*,mj} = \bar{a} u_{f,mj} - \bar{b} u_{c,mj},$$

where

$$\bar{a} = \frac{\Delta T_j^{k_l}}{\Delta T_j^{k_l} - \sum_{i=0}^{m-1} \delta t_{mj-i}^{k_l}}, \qquad \bar{b} = \frac{\sum_{i=0}^{m-1} \delta t_{mj-i}^{k_l}}{\Delta T_j^{k_l} - \sum_{i=0}^{m-1} \delta t_{mj-i}^{k_l}}.$$

An important difference between τ-MGRIT for non-uniform and uniform grids is that the non-uniform approach uses the local order of the time integration scheme k_l instead of the global order k_g. In general, the global order of the time integration scheme is assumed to be one less than the local order. For example, the local truncation error of a backward Euler time step is $O(\delta t^2)$ ($k_l = 2$) and the global order of the method is $k_g = 1$. Statements such as these

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) full non-uniform (b) semi-uniform

FIG. 6.3. *Error in the final solution when applying MGRIT and τ-MGRIT to the 1D heat equation using the semi-uniform (left) and full non-uniform (right) grids.*

are based on the assumption that the lowest-order error term dominates the error equation and that the local truncation errors accumulate linearly across the temporal domain. Comparing to the uniform time step case, we have

$$\bar{a} = \frac{(m\delta t)^{k_l}}{(m\delta t)^{k_l} - m\delta t^{k_l}} = \frac{m^{k_l-1}}{m^{k_l-1} - 1} = \frac{m^{k_g}}{m^{k_g} - 1} = a.$$

A similar equivalence holds for $\bar{b}$. Hence, the τ-MGRIT algorithm for non-uniform temporal grids can be used on both uniform and non-uniform grids. The remainder of the derivation for τ-MGRIT on non-uniform grids is identical to that of τ-MGRIT for uniform grids.

6.1.1. Numerical tests. In this section, results of τ-MGRIT applied on a semi-uniform and a full non-uniform temporal grid are presented. The semi-uniform temporal grid was created as follows. First, $N_t/m - 2$ points were randomly distributed across the temporal domain, where N_t is the total number of time points and m is the coarsening factor. These points, along with the end points $t = 0$ and $t = 2\pi$, are set as coarse-grid points. Fine-grid points are then determined by a uniformly splitting of each coarse-grid interval into m uniform time steps. The result is a "semi-uniform" temporal grid consisting of N_t time steps. The full non-uniform temporal grid consists of $t_0 = 0$, $t_{N_t} = 2\pi$, and $t_j = \text{rand}(0, 2\pi)$, for $j = 1, 2, \ldots, N_t - 1$. The results shown below use $N_t = 16384$, SDIRK-1, and standard second-order differencing in space.

Figure 6.3 displays the error across the temporal domain for MGRIT and τ-MGRIT applied on the two non-uniform temporal grids described above. In both cases, the τ-MGRIT algorithm is an effective approach for reducing the error.

Figure 6.4 displays a strong scaling study and the digits of accuracy per second for the error at time $t = 2\pi$. In this case, FCF-relaxation is used with a coarsening factor of $m = 4$. Clearly, τ-MGRIT still leads to an improvement in the APCC of the method, even on a non-uniform grid. While there may be cases where the full non-uniform grid will outperform the semi-uniform case, this was not observed here. For example when the grid is chosen to reflect the solution (rather than randomly as done here), one would expect the full non-uniform grid to produce a solution with better accuracy than the semi-uniform grid.

7. Conclusion. The non-intrusive and multilevel nature of both the RE and MGRIT algorithms allows the two algorithms to be combined in a simple and efficient way. The resulting τ-MGRIT algorithm is capable of improving the accuracy of the underlying time-integration scheme with little increase in the overall cost when compared to the standard

ETNA
Kent State University and
Johann Radon Institute (RICAM)

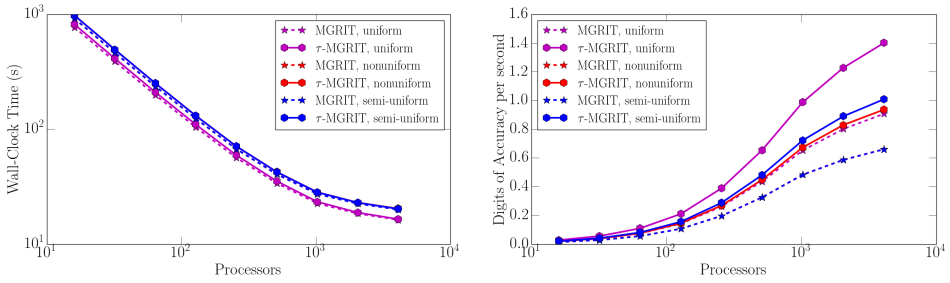


FIG. 6.4. *Wall time and digits of accuracy per second for the τ-MGRIT and MGRIT algorithms applied to the 1D heat equation with a uniform, semi-uniform, and full non-uniform temporal grid. The lines for non-uniform MGRIT (red stars) are hidden beneath the lines for uniform MGRIT (blue stars) in both plots.*

MGRIT algorithm. It is important to note that not all problems are suitable for use with τ-MGRIT; however, τ-MGRIT is guaranteed to converge towards the discrete solution that is obtained when using sequential time integration with RE. As such, the analysis required to determine the suitability of τ-MGRIT for a particular problem is no more difficult than that required to determine if RE can be used in the sequential case. For problems where sequential time integration with RE improves the convergence order of the time integration scheme, the τ-MGRIT algorithm is guaranteed to also improve the convergence order when compared to standard MGRIT, while also allowing for a parallel speedup over sequential time integration.

Moreover, with some simple modifications, the τ-MGRIT algorithm can be extended to include problems with non-uniform temporal grids. Current research is focused on developing an adaptive, MGRIT-based parallel-in-time solver that uses RE as a means of temporal error estimation [24].

REFERENCES

[1] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484–512.
[2] K. BERNERT, *τ-extrapolation—theoretical foundation, numerical experiment, and application to Navier–Stokes equations*, SIAM J. Sci. Comput., 18 (1997), pp. 460–478.
[3] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
[4] ———, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., vol. 960 of Lecture Notes in Math., Springer, Berlin, 1982, pp. 220–312.
[5] V. A. DOBREV, T. KOLEV, N. A. PETERSSON, AND J. B. SCHRODER, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM J. Sci. Comput., 39 (2017), pp. S501–S527.
[6] M. EMMETT AND M. L. MINION, *Toward an efficient parallel in time method for partial differential equations*, Commun. Appl. Math. Comput. Sci., 7 (2012), pp. 105–132.
[7] R. D. FALGOUT, S. FRIEDHOFF, T. V. KOLEV, S. P. MACLACHLAN, AND J. B. SCHRODER, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661.
[8] R. D. FALGOUT, A. KATZ, T. KOLEV, J. B. SCHRODER, A. WISSINK, AND U. M. YANG, *Parallel time integration with multigrid reduction for a compressible fluid dynamics application.*, Tech. Report LLNL-JRNL-663416, Lawrence Livermore National Laboratory, Livermore, 2015.
[9] R. D. FALGOUT, T. A. MANTEUFFEL, B. O'NEILL, AND J. B. SCHRODER, *Multigrid reduction in time for nonlinear parabolic problems: a case study*, SIAM J. Sci. Comput., 39 (2017), pp. S298–S322.
[10] S. FRIEDHOFF, R. D. FALGOUT, T. V. KOLEV, S. P. MACLACHLAN, AND J. B. SCHRODER, *A multigrid-in-time algorithm for solving evolution equations in parallel*, Tech. Report LLNL-CONF-606952, Lawrence Livermore National Laboratory, Livermore, 2013. .
[11] S. FRIEDHOFF, J. HAHNE, I. KULCHYTSKA-RUCHKA, AND S. SCHÖPS, *Exploring parallel-in-time approaches for eddy current problems*, in Progress in Industrial Mathematics at ECMI 2018, I. Faragó, F. Izsák, and P. L. Simon, eds., vol. 30 of The European Consortium for Mathematics in Industry, Springer,

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Cham, 2019, pp. 373–379.

[12] M. J. GANDER, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, T. Carraro, M. Geiger, S. Körkel, and R. Rannachev, eds., vol. 9 of Contrib. Math. Comput. Sci., Springer, Cham, 2015, pp. 69–113.

[13] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578.

[14] S. GÜNTHER, L. RUTHOTTO, J. B. SCHRODER, E. C. CYR, AND N. R. GAUGER, *Layer-parallel training of deep residual neural networks*, SIAM J. Math. Data Sci., 2 (2020), pp. 1–23.

[15] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer, Berlin, 1985.

[16] A. J. HOWSE, H. DE STERCK, R. D. FALGOUT, S. MACLACHLAN, AND J. SCHRODER, *Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid Burgers equations*, SIAM J. Sci. Comput., 41 (2019), pp. A538–A565.

[17] M. JUNG AND U. RÜDE, *Implicit extrapolation methods for multilevel finite element computations*, SIAM J. Sci. Comput., 17 (1996), pp. 156–179.

[18] ———, *Implicit extrapolation methods for variable coefficient problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1109–1124.

[19] S. KAMKAR, A. WISSINK, V. SANKARAN, AND A. JAMESON, *Combined feature-driven Richardson-based adaptive mesh refinement for unsteady vortical flows*, AIAA J., 50 (2012), pp. 2834–2847.

[20] M. LECOUVEZ, R. D. FALGOUT, C. S. WOODWARD, AND P. TOP, *A parallel multigrid reduction in time method for power systems*, in 2016 Power and Energy Society General Meeting (PESGM), IEEE Conference Proceedings, Los Alamitos, 2016, 5 pages.

[21] J.-L. LIONS, Y. MADAY, AND G. TURINICI, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.

[22] M. L. MINION AND S. A. WILLIAMS, *Parareal and spectral deferred corrections*, in Numerical Analysis and Applied Mathematics, T. E. Simos, G. Psihoyios, and Ch. Tsitouras, eds., no. 1048 of AIP Conference Proceedings, AIP Publishing, Melville, 2008, pp. 388–391.

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

[24] B. O'NEILL, *Multigrid Reduction in Time for Nonlinear Parabolic Problems*, PhD. Thesis, Dept. Appl. Math., University of Colorado at Boulder, Boulder 2017.

[25] B. W. ONG AND J. B. SCHRODER, *Applications of time parallelization*, Comput. Vis. Sci., 23 (2020), Paper No. 11, 15 pages.

[26] L. F. RICHARDSON, *The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam*, Philos. Trans. Roy. Soc. London Ser. A., 210 (1911), pp. 307–357.

[27] C. ROY, *Review of discretization error estimators in scientific computing*, in 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, AIAA Meeting papers, 2010. https://doi.org/10.2514/6.2010-126

[28] U. RÜDE, *The hierarchical basis extrapolation method*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 307–318.

[29] J. B. SCHRODER, R. D. FALGOUT, C. S. WOODWARD, P. TOP, AND M. LECOUVEZ, *Parallel-in-time solution of power systems with scheduled events*, in 2018 IEEE Power and Energy Society General Meeting (PESGM), IEEE Conference Proceedings, Los Alamitos, 2018, 5 pages.

[30] H. D. STERCK, R. D. FALGOUT, S. FRIEDHOFF, O. A. KRZYSIK, AND S. P. MACLACHLAN, *Optimizing MGRIT and parareal coarse-grid operators for linear advection*, Preprint on arXiv, 2019. https://arxiv.org/abs/1910.03726

[31] P. J. VAN DER HOUWEN, *Parallel step-by-step methods*, Appl. Numer. Math., 11 (1993), pp. 69–81.

[32] S. WU, B. SHI, AND C. HUANG, *Parareal-Richardson algorithm for solving nonlinear ODEs and PDEs*, Commun. Comput. Phys., 6 (2009), pp. 883–902.

[33] *XBraid: Parallel multigrid in time*, Software. http://llnl.gov/casc/xbraid.