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 faster 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 [22], but interest in
parallel-in-time was rekindled with the parareal method [20] in 2001, followed by the
multilevel spectral deferred correction approach (also called PFASST) [21, 5]. For a
detailed review of parallel-in-time methods, see the review paper [11].

This work focuses on the multigrid reduction in time (MGRIT) method [9, 6], an
iterative multilevel algorithm that uses multigrid reduction techniques and temporal
domain decomposition to add large-scale temporal parallelism to existing time integra-
tion 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 [6]; however, the constant for
MGRIT is higher than that of sequential time integration. This creates a crossover

∗Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box
808, L-561, Livermore,CA 94551. This work 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

†Department of Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado. This
work 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.

‡Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico
point where the additional computational resources eventually overcome the extra computational work. Beyond this crossover point MGRIT provides a speedup over sequential methods. This work uses XBraid [30], an open source implementation of MGRIT developed at Lawrence Livermore National Laboratory (LLNL). Some recent applications of XBraid include linear and nonlinear parabolic problems [6, 8], compressible fluid dynamics [7], power grids [19, 27], eddy-current problems [10], linear advection [15, 28], and machine learning [13].

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 [6] 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 greater accuracy per computational cost (APCC) when compared to the original MGRIT algorithm, while continuing to offer the temporal concurrency of MGRIT.

The spatial multigrid community refers to a similar approach as \( \tau \)-extrapolation [4, 14, 2] because it modifies the \( \tau \)-term in full approximation storage (FAS) multigrid [3]. The further works [26, 16, 17] connect multilevel finite element discretizations directly with multigrid \( \tau \)-extrapolation solvers. Although the proposed approach uses Richardson extrapolation, which differs slightly from \( \tau \)-extrapolation, we name the new MGRIT algorithm \( \tau \)-MGRIT because our eventual algorithm also results in a simple modification to the \( \tau \)-term. The results presented here are based on [23].

In summary, the \( \tau \)-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, 25, 18]. Lastly, this approach could be applied to Parareal, which is equivalent in a certain setting to two-level MGRIT [12].

The outline of this paper is as follows. First, a brief overview of RE based time integration is given. Next, the two- and multi-level \( \tau \)-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 \( \tau \)-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 \( \tau \)-MGRIT on a non-uniform temporal grid is considered. Overall, the results in this paper show how the \( \tau \)-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 [24], 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 to \( U \) with a step size \( \delta t \) and error of the form, through
Taylor’s theorem,
\begin{equation}
U - U(\delta t) = C_0(\delta t)^k_0 + C_1(\delta t)^k_1 + \ldots,
\end{equation}
where each \(k_0 < k_1 < \ldots\) is a known constant and each \(C_i\) is an unknown constant.

We assume that the asymptotic expansions here are convergent, i.e., \(\delta t\) is small enough.

For a larger step size, \(m \delta t\), the same error formula becomes
\begin{equation}
U - U(m \delta t) = C_0(m \delta t)^k_0 + C_1(m \delta t)^k_1 + \ldots.
\end{equation}

Eliminating the exact solution, \(U\), and ignoring higher order terms, we find
\begin{equation}
(\delta t)^k_0 C_0 = U(\delta t) - U(m \delta t) - \frac{m}{k_0 - 1} + O((\delta t)^{k_1}).
\end{equation}

Substituting for \((\delta t)^k_0 C_0\) in equation (2.1) gives
\begin{equation}
U = \frac{m^{k_0}U(\delta t) - U(m \delta t)}{m^{k_0} - 1} + O((\delta t)^{k_1}),
\end{equation}
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{equation}
U_0 = U(\delta t),
\end{equation}
\begin{equation}
U_{i+1}(\delta t) = \frac{m^{k_i}U_i(\delta t) - U_i(m \delta t)}{m^{k_i} - 1},
\end{equation}
where \(U = U_{i+1}(\delta t) + O((\delta t)^{k_i+1})\).

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,
\begin{equation}
|C_1 \frac{m^{k_0} - m^{k_1}}{m^{k_0} - 1}| (\delta t)^{k_1} \geq |C_0|(\delta t)^{k_0}.
\end{equation}

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 \(\delta t\) and nodes \(t_j, j = 0, \ldots, N_t\) (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 \(T_j = j \Delta T, j = 0, 1, \ldots, N_t/m\), 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 \(\tau\)-MGRIT algorithm.

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{equation}
\frac{du}{dt} = f(u, t), \quad u(0) = u_0, \quad t \in [0, T],
\end{equation}
\begin{equation}
u_{j+1} = \Phi(u_{j+1}, u_j, \delta t) + g(t_{j+1}), \quad j = 0, \ldots, N_t - 1,
\end{equation}
where $u_j$ represents the discrete solution at $t = t_j$, $\Phi$ is an operator representing the chosen time-stepping method and $g$ is a time dependent function that incorporates all the solution independent terms.

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 $\delta t$ 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 given above, each temporal subdomain corresponds to an interval of length $m\delta t$, 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 presented in equation (2.9) with a single stage (i.e., $i = 0$ in equation (2.6)) of RE-based time integration equates to solving the following system of equations

\begin{equation}
    u_{f,mj+1} = \Phi(u_{*,mj}, \delta t) + g_{mj},
\end{equation}

\begin{equation}
    u_{f,mj+2} = \Phi(u_{f,mj+1}, \delta t) + g_{mj+1},
\end{equation}

\vdots

\begin{equation}
    u_{f, mj} = \Phi(u_{f,mj}-1, \delta t) + g_{m(j+1)},
\end{equation}

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

\begin{equation}
    u_{c,m(j+1)} = \Phi(u_{*,mj}, m\delta t) + g_{m(j+1)},
\end{equation}

\begin{equation}
    u_{*, j+1} = \frac{m^{k_g}u_{f,m(j+1)} - u_{c,m(j+1)}}{m^{k_g} - 1},
\end{equation}

\begin{equation}
    = a u_{f,m(j+1)} - bu_{c,m(j+1)},
\end{equation}

for $j = 0, 1, 2, \ldots, N_t/m$, $u_{*,0} = u_{f,0}$ and $g_{mj} = g(t_{mj})$. We let $a = m^{k_g}/(m^{k_g} - 1)$ and $b = 1/(m^{k_g} - 1)$, such that $a - b = 1$. The notation $u_{f/c/*}$ indicates a fine grid (f), coarse grid (c), or RE enhanced (*) solution at time $t = t_i$. Algorithm 1 outlines a standard single stage RE based time integration scheme and Figure 2.2 provides a graphical representation of this approach with $m = 2$.

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...
Algorithm 1 RE based time integration \((u, g(x,t), \Phi, t_{stop})\)

1: While \(( t < t_{stop} )\) do.
2: Take a time step of size \( m\delta t \):
   \[ u_c = \Phi(u) + g(t + m\delta t). \]
3: Take \( m \) time steps of size \( \delta t \) to get \( u_f = \Phi(\Phi(\Phi(u) + g(t + \delta t)) \ldots )) + g(t + m\delta t). \]
4: Calculate the enhanced solution \( u = (m^k u_f - u_c)/(m^k - 1) \)
5: \( t \leftarrow t + m\delta t. \)
6: Goto Step 1.

<table>
<thead>
<tr>
<th>( u_0 )</th>
<th>( u_{f,1} )</th>
<th>( u_{f,2} )</th>
<th>( u_{c,2} )</th>
<th>( u_{c,4} )</th>
<th>( u_{f,4} )</th>
<th>( u_{f,6} )</th>
<th>( u_{c,6} )</th>
</tr>
</thead>
</table>
| \( u_{f,3} \rightarrow u_{f,4} \) | \( u_{f,5} ightarrow u_{f,6} \) | \( u_{c,3} \rightarrow u_{c,4} \)

Fig. 2.2: Time integration with RE and \( m = 2 \). The notation \( u_{f/c/*,a} \) indicates a fine grid (f), coarse grid (c), or RE enhanced (∗) solution at time \( t = t_a \). Coarse-grid estimates are used to enhance the fine grid solution at the coarse-grid points.

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 \( \tau \)-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 \( \tau \)-extrapolation. Multigrid \( \tau \)-extrapolation uses the extrapolation principle to exploit the multilevel structure of spatial multigrid algorithms, resulting in a fast solver with improved approximation properties.

Multigrid \( \tau \)-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 2, where \( A_h u_h = g_h \) and \( A_H v_H = g_H \) represent the discretization of a (perhaps nonlinear) differential equation on fine and coarse grids, respectively. We use \( v_H \) (and not \( u_H \)) 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 \( \delta t \) and \( m\delta t \), respectively. Both \( R \) and \( \hat{R} \) are restriction operators, interpolating from fine to coarse \((h \rightarrow H)\), and \( P \) is the prolongation operator, interpolating from coarse to fine \((H \rightarrow h)\). 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 2 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 \( \tau \), which allows for RE to be integrated into the coarse-grid correction straightforwardly.

The \( \tau \)-term \((\tau^H_h)\) can be interpreted as a correction to the coarse grid solution, \( v_H \), to make it coincide with the fine grid solution, \( u_h \) \[2\]. Based on the principles of RE, the \( \tau \)-extrapolation algorithm will multiply \( \tau^H_h \) by a factor \( a = m^p/(m^p - 1) \) where \( p \) is the order of the approximation, so that the coarse grid solution better
Algorithm 2 FAS($A$, $u$, $g$)

1: Relax on $A_h u_h = g_h$ using a (non)linear relaxation method.
2: Solve the coarse grid problem

$$A_H v_H = g_H + \tau_H,$$

where $\tau_H = A_H R u_h - \hat{R} A_h u_h$.
3: Compute the coarse grid error approximation:

$$e_H = v_H - R u_h.$$
4: Correct the fine grid approximation

$$u_h \leftarrow u_h + P e_H.$$

approximates the true solution, $u$. 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 $\tau$-extrapolation principle from the perspective of time integration with RE and MGRIT.

3.1. Derivation of $\tau$-MGRIT. The two-level $\tau$-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 $\tau$-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 $\tau$-MGRIT is an area of research.

In block-triangular form, the original time integration problem, equation (2.9), is

$$A u = \begin{bmatrix} I & 0 & \cdots & 0 \\ -\Phi & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -\Phi & \cdots & -\Phi & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_m \end{bmatrix} = \bar{g},$$

The RE based time-integration problem, equations (2.10)-(2.16), can be written in block triangular form as

$$A^T u = \begin{bmatrix} I \\ -\Phi & I \\ \vdots \\ b\Phi\Delta & -a\Phi & I \\ 0 \\ -\Phi & I \\ \vdots \\ b\Phi\Delta & \cdots & -a\Phi & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_m \\ u_{m+1} \\ \vdots \\ u_{2m+1} \end{bmatrix} = \bar{g},$$

where $\Phi\Delta(u) = \Phi(u, \eta \delta t)$ represents the coarse grid time step. Note that $u = [u_0, \ldots, u_N]$ is a vector of vectors, where each $u_i$ 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 \(\tau\)-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.

### 3.1.1. The \(\tau\)-MGRIT Coarse Grid Operator

The \(\tau\)-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:

\[
\begin{align*}
\mathbf{u}_{f,mj} &= \Phi \mathbf{u}_{f,mj-1} + \mathbf{g}_{mj} = \Phi(\Phi \mathbf{u}_{mj-2} + \mathbf{g}_{mj-1}) + \mathbf{g}_{mj} = \ldots, \\
\mathbf{u}_{f,mj} &= \Phi^m \mathbf{u}_{f,m(j-1)} + \hat{\mathbf{g}}_{mj}, \quad j = 1, 2, \ldots, N_t/m, \\
\end{align*}
\]

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

\[
\mathbf{A}_\Delta \mathbf{u}_\Delta = \begin{bmatrix}
I \\
(b\Phi_\Delta - a\Phi^m) \\
(b\Phi_\Delta - a\Phi^m) & I \\
& \ddots & \ddots \\
& & \ddots & I \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_{\Delta,0} \\
\mathbf{u}_{\Delta,1} \\
\mathbf{u}_{\Delta,2} \\
\vdots
\end{bmatrix}
\]

The coarse-grid operator, \(A_\Delta\), is referred to as ideal because the exact solution of equation (3.5) yields the exact solution to equation (3.2) at the coarse points. If this is followed by interpolation (i.e., F-relaxation in Figure 3.1), then the exact solution is also available at 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 \(j = 0, 1, 2, \ldots, N_t/m\).

The limitation of this exact reduction method is that the coarse-grid problem (3.5) is, in general, as expensive to solve as the original fine-grid problem (3.2) (because of the \(\Phi^m\) evaluations). Approximating \(\Phi^m\) with \(\Phi_\Delta\) and noting that \(a - b = 1\) gives a cheap approximation to this exact coarse grid operator

\[
\mathbf{B}_\Delta = \begin{bmatrix}
I \\
-\Phi_\Delta & I \\
& \ddots & \ddots \\
& & \ddots & -\Phi_\Delta \\
& & & I
\end{bmatrix}
\]

where \(\Phi_\Delta\) is an approximate coarse-grid time-step operator. This choice of approximating \(\Phi^m\) with \(\Phi_\Delta\) has been used in various MGRIT algorithms to good effect \([6, 8, 19]\). Additionally, the coarse grid operator, \(B_\Delta\), is independent of the constants \(a\) and \(b\), indicating that the coarse grid operator is identical for both the \(\tau\)-MGRIT and MGRIT algorithms. In fact, the only difference between the two-level \(\tau\)-MGRIT

\(^1\)Technically, the two (or more) solutions can proceed in parallel. However, this is a low level form of temporal parallelism that does not scale.
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 $\Phi_\Delta$ 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 backward Euler, one simply uses a larger time-step size. Convergence of MGRIT is ultimately determined by the approximation, $A_\Delta^\tau \approx B_\Delta$, and this choice of re-discretizing $\Phi$ with $\Delta T = m\delta t$ has proved effective when applied to the MGRIT algorithm [6, 8, 7].

Note that while the definition of this algorithm relies upon $\Phi$ and $\Phi_\Delta$, 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 $\tau$-MGRIT code.

As per the original MGRIT algorithm [6], restriction is completed using injection. Similarly, interpolation is equivalent to injection from the coarse to the fine grid, 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.

### 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. 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 [8].

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 $\Phi_\Delta$ application in Figure 3.1). For $\tau$-MGRIT, C-relaxation on the finest grid requires 2 time steps per C-point, one on the fine grid and one on the coarse grid, i.e., $\Phi$ and $\Phi_\Delta$ in equation (3.2), respectively.

In [6], 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 a F-relaxation, followed by a C-relaxation, followed by another F-relaxation.

### 3.1.3. The two-level $\tau$-MGRIT algorithm with FAS

To accommodate both linear and nonlinear problems, the $\tau$-MGRIT algorithm uses FAS multigrid as outlined in Algorithm 2. In the $\tau$-MGRIT notation, the FAS coarse grid problem, defined in Algorithm 2, is: find $v_\Delta$ such that

$B_\Delta v_\Delta = B_\Delta R_I u + R_I (g - A^\tau u)$.

(3.7)
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 the last C-point on each coarse grid interval. The $\Phi_\Delta$ 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.

Expanding this equation, the $j$’th row of the coarse grid problem is

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

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 $j$’th row of the matrix $X$. Given this, the $\tau$-MGRIT coarse grid problem is: find $\mathbf{v}_\Delta$ such that

\begin{align}
B_\Delta \mathbf{v}_\Delta = R_I \mathbf{g} + a(B_\Delta R_I \mathbf{u} - R_I A \mathbf{u}).
\end{align}

Thus, the two-level $\tau$-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 $a = m^{k}\tau/(m^{k} - 1)$ (compare equations (3.7) and (3.16)). This result shares a similar form to that outlined in the original $\tau$-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 3 outlines the two-level $\tau$-MGRIT algorithm. The multilevel extension of the two-level $\tau$-MGRIT algorithm is discussed in Remark 3.1.

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 true, and as such, RE based time integration cannot be used in all situations. However, as will be shown in Section 4.2, the $\tau$-MGRIT is
Algorithm 3 \( \tau \)-MGRIT\((A, u, g) \)

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

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 \( \tau \)-MGRIT algorithm. That is, \( \tau \)-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 \( \tau \)-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 multi-level \( \tau \)-MGRIT algorithm does not combine multiple stages of RE-based time
integration. Instead, the multi-level \( \tau \)-MGRIT algorithm uses the two-level \( \tau \)-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 \( \tau \)-MGRIT and MGRIT.

4. Computational cost of the \( \tau \)-MGRIT algorithm. The accuracy improve-
ments of \( \tau \)-MGRIT would be of no use if the cost of the method increased commensu-
rately. To study this issue, we consider the cost of \((\tau-)MGRIT\), 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 \( \Phi \). In terms of \( \Phi \) evaluations, the only difference
between the two-level \( \tau \)-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 \( \tau \)-MGRIT algorithm when compared to
the MGRIT algorithm, if FCF-relaxation is used. If \( p \) processors are used, 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, there is no difference in the
number of \( \Phi \) evaluations completed per iteration for the two methods.

As mentioned in Remark 3.1, the multilevel extension of the two-level \( \tau \)-MGRIT
algorithm uses the original MGRIT algorithm to solve the coarse grid problem. This
Then, the global space-time error vector at the C-points, $N$, where we compare the convergence rates of the two methods. The analysis in this section relaxes the residual at because $|||s|||\text{satisfies the current approximation to the solution, }MGRIT$ with F- and FCF-relaxation and define $X = (1-\lambda/\omega_m)^{1/\mu_m}|||t\right)$.

**Remark 4.1.** As a practical point, the XBraid implementations of the MGRIT algorithms do 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 MGRIT and MGRIT is very similar. In this section, we compare the convergence rates of the two methods. The analysis in this section develops a convergence bound for the 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 on the convergence rate of the standard MGRIT algorithm from [29].

Consider solving equation (2.9) on the temporal grid shown in Figure 2.1 and assume $g(x,t) = 0$. The case with nonzero $g(x,t)$ is similar. Next, assume $|||\Phi||| \leq 1$, i.e., $\Phi$ is a convergent time integration scheme, and that $\Phi$ and $\Phi_\Delta$ are diagonalized by the same unitary transform,

\begin{align}
\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}),
\end{align}

where $N_x$ represents the number of spatial unknowns associated with each time-step, $X = (x_1, \ldots, x_{N_x})$, and $X^*X = I$.

Finally, let $E^F_\Delta$ and $E^{FCF}_\Delta$ denote the coarse grid error propagation matrices for MGRIT with F- and FCF-relaxation and define $e_j$ to be the error at $t = t_j$ between the current approximation to the solution, $\hat{u}_j$, and the exact discrete solution, $u_j$.

Then, the global space-time error vector at the C-points, $\hat{e} = [e_1^T, e_2^T, \ldots, e_N^T]^T$, satisfies

\begin{align}
||E^F_\Delta \hat{e}||_2 \leq \max_{\omega = 1, 2, \ldots, N_x} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1 - |\mu_\omega|^{N_\omega}/m}{(1 - |\mu_\omega|)} \right\} ||\hat{e}||_2,
\end{align}

for MGRIT with F-relaxation, and

\begin{align}
||E^{FCF}_\Delta \hat{e}||_2 \leq \max_{\omega = 1, 2, \ldots, N_x} \left\{ |\lambda_\omega^m - \mu_\omega| \frac{1 - |\mu_\omega|^{N_\omega}/m}{(1 - |\mu_\omega|)} |\lambda_\omega^m| \right\} ||\hat{e}||_2,
\end{align}

for MGRIT with FCF-relaxation [29]. Note that the fraction in this bound, $(1 - |\mu_\omega|^{N_\omega}/m)/(1 - |\mu_\omega|)$ is the result of a geometric sum. As $\mu_\omega \to 1$, this terms grows like $N_\omega$. Fortunately for many common time integration schemes, this is compensated because $|\lambda_\omega^m - \mu_\omega| \to 0$ as $N_\omega \to \infty$ (see Section 4.2.1).

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

\begin{align}
r_{mj} &= -[A(\hat{u})]_{mj}, \\
&= -\hat{u}_{mj} + (a\Phi^m - b\Phi_\Delta)\hat{u}_{m(j-1)},
\end{align}

means that the extra time steps completed on the fine grid are the only extra time steps taken, per iteration, by the MGRIT algorithm when compared to the original MGRIT algorithm. In the following section, a bound on the convergence rate is derived for the two-level MGRIT algorithm, and compared to a similar bound derived in [29] for the original MGRIT algorithm. With this bound, we can approximate the number of iterations required for the convergence of both methods.

As a practical point, the XBraid implementations of the MGRIT algorithms use the norm of the residual, i.e., $||g - Au||$, to measure the error and determine when to stop iterating. Unlike the original MGRIT algorithm, 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.
where we have assumed, without loss of generality, that \( g = 0 \). The exact solution of equation (3.2) satisfies \( u_{mj} = (a\Phi^m - b\Phi)u_{m(j-1)} \). Hence, the residual can be rewritten in terms of the error, \( e_{mj} = u_{mj} - \tilde{u}_{mj} \), as

\[
(4.6) \quad r_0 = e_0, \\
(4.7) \quad r_{mj} = e_{mj} - (a\Phi^m - b\Phi)u_{m(j-1)},
\]

for \( j = 1, 2, \ldots, N_t/m \). For two-level \( \tau \)-MGRIT, the coarse grid problem is solved using forward substitution. This is an \( O(N_t) \) sequential operation. A multilevel variant of \( \tau \)-MGRIT is discussed in Section 5.3. The exact coarse grid correction, \( c_{mj} \), is given by

\[
(4.8) \quad c_0 = r_0, \\
(4.9) \quad c_{mj} = \Phi_\Delta(c_{m(j-1)}) + r_{mj}, \\
(4.10) \quad = \Phi_\Delta^j(r_0) + \Phi_\Delta^{j-1}(r_m) + \ldots + r_{mj},
\]

for \( j = 1, 2, \ldots, N_t/m \). Substituting for each \( r \) with a subscript using equations (4.6) and (4.7) gives

\[
(4.11) \quad e_0 = e_0, \\
(4.12) \quad e_{mj} = \Phi_\Delta^j(e_0) + \Phi_\Delta^{j-1}(e_m + (b\Phi - a\Phi^m)e_0) \\
+ \Phi_\Delta^{j-2}(e_{2m} + (b\Phi - a\Phi^m)e_m) + \ldots \\
+ e_{mj} + (b\Phi - a\Phi^m)e_{m(j-1)},
\]

Collecting like terms and noting that \( b + 1 = a \) gives

\[
(4.13) \quad e_0 = e_0, \\
(4.14) \quad e_{mj} = a\Phi_\Delta^{j-1}(\Phi - \Phi^m)e_0 + a\Phi_\Delta^{j-2}(\Phi - \Phi^m)e_m + \ldots \\
+ a(\Phi - \Phi^m)e_{m(j-1)} + e_{mj}.
\]

After the correction, the updated error at the C-Points, \( f_{mj} = e_{mj} - c_{mj} \), satisfies:

\[
(4.15) \quad f_0 = 0, \\
(4.16) \quad f_{mj} = a \sum_{q=0}^{j-1} \Phi_\Delta^{j-1-q}(\Phi^m - \Phi^m)e_{mq},
\]

for \( j = 1, 2, \ldots, N_t/m \). The unitary transformation given in equation (4.1) can be used to decompose \( e_{mj} \) as \( e_{mj} = \sum_{\omega=1}^{N_x} \tilde{\epsilon}_{j,\omega} x_{\omega} \) and \( f_{mj} \) as \( f_{mj} = \sum_{\omega=1}^{N_x} \tilde{f}_{mj,\omega} x_{\omega} \). Substituting these decompositions into equation (4.15) and (4.16) gives

\[
(4.17) \quad \tilde{f}_{0,\omega} = 0, \\
(4.18) \quad \tilde{f}_{mj,\omega} = a \sum_{q=0}^{j-1} \mu^{j-1-q}(\lambda^m_{\omega} - \mu_{\omega})\tilde{\epsilon}_{mq,\omega},
\]
for \( j = 1, 2, \ldots, N_t/m \) and \( \omega = 1, 2, \ldots, N_x \). Defining \( \hat{\mathbf{e}}_\omega = (\hat{\mathbf{e}}_{0,\omega}, \hat{\mathbf{e}}_{m,\omega}, \ldots, \hat{\mathbf{e}}_{N_T,\omega})^T \) and
\[
\hat{\mathbf{f}}_\omega = (\hat{f}_{0,\omega}, \hat{f}_{m,\omega}, \ldots, \hat{f}_{N_T,\omega})^T
\]
gives \( \hat{\mathbf{f}}_\omega = \mathbf{E}_\omega^F \hat{\mathbf{e}}_\omega \) where
\[
\mathbf{E}_\omega^F = a(\lambda^m_\omega - \mu_\omega) \begin{pmatrix} 0 & 1 & \mu_\omega & 1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mu_\omega^{N_T/m-1} & \cdots & \cdots & \cdots & \mu_\omega \\ \end{pmatrix}.
\]

The reader will note that the matrix \( \mathbf{E}_\omega^F \) is nilpotent. As such, \( \tau \)-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 \( \tau \)-MGRIT with FCF-relaxation and for the original MGRIT algorithm with both F- and FCF-relaxation. Speedups are obtained when MGRIT converges in \( O(1) \) iterations.

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 [29], a convergence bound for each eigenmode is then given by
\[
\|\mathbf{E}_\omega\|_1 = \|\mathbf{E}_\omega\|_\infty = a|\lambda^m_\omega - \mu_\omega| \frac{1 - |\mu_\omega|^{N_t/m-1}}{1 - |\mu_\omega|}.
\]

Next, it is straightforward to apply Theorem 3.3 from [29] to obtain a bound on the global error vector, \( \mathbf{e} = [\mathbf{e}_0^F, \mathbf{e}_m^F, \ldots, \mathbf{e}_{N_T/m}^F]^T \), for \( \tau \)-MGRIT with F-relaxation:

\[
\|\hat{\mathbf{E}}^F_{\Delta} \mathbf{e}\|_2 \leq \max_{\omega=1,2,\ldots,N_x} \frac{m^{k_b}}{m^{k_b} - 1} \left\{ |\lambda^m_\omega - \mu_\omega| \frac{1 - |\mu_\omega|^{N_T/m}}{(1 - |\mu_\omega|)} \right\} \|\mathbf{e}\|_2 = \frac{m^{k_b}}{m^{k_b} - 1} \|\hat{\mathbf{E}}^F_{\Delta} \mathbf{e}\|_2,
\]

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

\[
\|\hat{\mathbf{E}}^{FCF}_{\Delta} \mathbf{e}\|_2 \leq \max_{\omega=1,2,\ldots,N_x} \frac{m^{k_b}}{m^{k_b} - 1} \left\{ |\lambda^m_\omega - \mu_\omega| \frac{1 - |\mu_\omega|^{N_T/m}}{(1 - |\mu_\omega|)} \right\} \|\mathbf{e}\|_2.
\]

Unlike equation (4.19), there does not exist a simple scaling between the error bounds for \( \tau \)-MGRIT (4.20) and MGRIT (4.3) when FCF-relaxation is used.

### 4.2.1. Singularly diagonal implicit Runge-Kutta (SDIRK) methods.

To understand these bounds, we examine the convergence rate of the \( \tau \)-MGRIT algorithm for various coarse and fine grid time integrators. Consider solving

\[
\mathbf{u}' + \mathbf{G}\mathbf{u} = \mathbf{b},
\]

where \( \mathbf{G} \) is a symmetric positive definite matrix with real eigenvalues, \( \gamma_\omega \), using the first, second and third order Singularly Diagonal 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 this analysis could not be applied to complex eigenvalues as 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. Here, the fine and coarse grid operators are \( \Phi = (I + \delta t\mathbf{G})^{-1} \) and...
\[ \Phi_\Delta = (I + m \delta t G)^{-1}, \] respectively. Likewise, the fine and coarse grid eigenvalues are

\[ \lambda_\omega = \frac{1}{1 + \delta t \gamma_\omega}, \quad \mu_\omega = \frac{1}{1 + m \delta t \gamma_\omega}, \]

respectively. Hence, for \( \omega = 1, 2, \ldots, N_x \), \( \gamma_\omega \in (0, \infty) \), \( \lambda_\omega \in (0, 1) \) and \( \mu_\omega \in (0, 1) \).

Notice that the eigenvalues can get very close to one. In that limit, the fraction in the error bound (equations (4.19) and 4.20) grows like \( N_t = T/\delta t \). Fortunately, the first term, \( |\lambda_m^\omega - \mu_\omega| \), goes to zero with at least \( O(\delta t^2) \).

Figures 4.2 and 4.3 plot the convergence bound, \( ||E_\omega|| \), vs. \( \kappa = \delta t |\gamma_\omega| \) for \( \gamma_\omega > 0 \), 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 for the \( \tau \)-MGRIT algorithm, and dashed lines represent the bounds for the standard MGRIT algorithm. The overall convergence bound for a dataset is the maximum value achieved on the y-axis. For both F- and FCF-relaxation, there is only a slight convergence degradation for \( \tau \)-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 larger \( m \) lead to worse discretization error in the numerical tests.

Importantly, these bounds show that the \( \tau \)-MGRIT algorithm is capable of increasing the convergence order of the underlying time-integration scheme without significantly increasing the convergence factor, including for higher order methods. Combined with the limited increase in time step evaluations, this gives the \( \tau \)-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 \( \tau \)-MGRIT and MGRIT algorithms.

5. \( \tau \)-MGRIT results.

5.1. Numerical example: first order ODE. Consider the following IVP:

\[ y' + 4y = 1 - t, \quad t \in [0, 1], \quad \text{with } y(0) = 1 \] (5.1) and exact solution, \( y(t) = (1/16)(-4t + 11e^{-4t} + 5) \). Figure 5.1 plots the error against time for the solution found when solving this equation using \( \tau \)-MGRIT with the SDIRK-1 and SDIRK-2 methods and 128 time steps. The error profile appears discontinuous because the \( \tau \)-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 \( \tau \)-MGRIT with SDIRK-2), the accuracy of the \( \tau \)-MGRIT solution worsens as the coarsening factor \( m \) increases. This is not surprising, as substituting \( k = 1 \) into equation (2.4) suggests the accuracy of the improved solution
Fig. 4.2: $||E_\omega||$ vs $\kappa = \delta t \gamma_\omega$ for the SDIRK algorithms with $\tau$-MGRIT and F-relaxation. Dashed lines represent the convergence bounds for MGRIT; solid lines are the convergence bounds for $\tau$-MGRIT.

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

varies as $m^k(1 - m)/(m^k - 1)\delta t^{k+1} = mC_1\delta t^{k+1}$. 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 $\tau$-MGRIT and MGRIT and small $m$, hence, making this less of a problem.

Figure 5.2 shows the error with respect to the time step size, at $t = 0.5$, highlighting the algorithmic scalability of the $\tau$-MGRIT algorithm. For SDIRK-1, a globally first order method, $\tau$-MGRIT scales with an accuracy equivalent to a second order method. Likewise, $\tau$-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 $\tau$-MGRIT algorithm, 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. To analyze for a representative PDE, \( \tau \)-MGRIT was applied to the one dimensional (1D) heat equation,

\[
\begin{align*}
(5.2) & \quad u_t - u_{xx} = f(x,t), \quad \forall x \in [0, \pi], \ t \in [0, 2\pi], \\
(5.3) & \quad u(0, t) = g_1(t), \ u(\pi, t) = g_2(t), \quad \forall t \in [0, 2\pi], \\
(5.4) & \quad u(x, 0) = h(x), \quad \forall x \in [0, \pi],
\end{align*}
\]

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, \( h(x) \), were chosen to prescribe the exact solution \( u_{exact}(x,t) = \sin(x) \cos(t) \). The temporal discretization uses SDIRK-1, while the spatial discretization uses standard second-order central differences. A direct spatial solver was used, limiting parallelism to the temporal domain.\(^2\)

\(^2\)The focus of this paper is a description and validation of \( \tau \)-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.
Fig. 5.3: Error analysis for the solution of the 1D heat equation found using $\tau$-MGRIT with 16384 spatial grid points and SDIRK-1. Figure (a) shows the error at the final time when using the MGRIT and $\tau$-MGRIT algorithms; Figure (b) shows that $\tau$-MGRIT with SDIRK-1 scales as a second order method.

Figure 5.3a plots the error against time for $\tau$-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 approximate space-time solutions. A relatively large number of spatial grid points were 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 $\tau$-MGRIT is significantly less than that for MGRIT, except for points near $t = 2$ and $t = 5$. As discussed above, the $\tau$-MGRIT algorithm converges to the exact solution found 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 5.3a also depicts the error at the final time for time integration with SDIRK-2. The accuracy of the solution obtained with $\tau$-MGRIT is less than that of the SDIRK-2 method. This suggests that, if accuracy is the most important aspect of the solution, 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 $\tau$-MGRIT with SDIRK-1 in terms of both accuracy and APCC. However, the study also shows that $\tau$-MGRIT with SDIRK-2 outperforms MGRIT with SDIRK-2, highlighting the fact that $\tau$-MGRIT is a simple, cost effective way to improve the accuracy of the MGRIT algorithm, no matter the order of the time integration scheme. That is to say, if RE based time integration is appropriate for a problem, then $\tau$-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 behavior, either theoretically or practically, when moving to 2D or 3D. Previous MGRIT work, e.g., [29, 6], indicates that the dimensionality of the heat equation does not affect the
behavior of MGRIT. Moreover, 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 $\tau$-MGRIT. Tables 5.1 and 5.2 show the convergence rates of an SDIRK-1 implementation of $\tau$-MGRIT and MGRIT with both F- and FCF-relaxation. The final column, labeled “Est”, shows the estimated convergence factor as derived in equations (4.19) and (4.20). These numerically determined convergence factors represent the average convergence factor over the last 5 iterations for runs using two level $\tau$-MGRIT, solved to a residual tolerance of $10^{-10}$, with a random initial guess.

<table>
<thead>
<tr>
<th>F-Relaxation</th>
<th>Time Steps</th>
<th>$\tau$</th>
<th>m</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
<th>Est.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>2</td>
<td>0.1206</td>
<td>0.1216</td>
<td>0.122</td>
<td>0.1222</td>
<td>0.1224</td>
<td>0.1216</td>
<td>0.1249</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>2</td>
<td>0.2416</td>
<td>0.2442</td>
<td>0.246</td>
<td>0.2466</td>
<td>0.2450</td>
<td>0.2438</td>
<td>0.2499</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>4</td>
<td>0.1956</td>
<td>0.1958</td>
<td>0.199</td>
<td>0.1994</td>
<td>0.1994</td>
<td>0.1996</td>
<td>0.2038</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>4</td>
<td>0.2606</td>
<td>0.2614</td>
<td>0.2652</td>
<td>0.2658</td>
<td>0.2658</td>
<td>0.2662</td>
<td>0.2719</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>16</td>
<td>0.1409</td>
<td>0.2548</td>
<td>0.258</td>
<td>0.2602</td>
<td>0.2662</td>
<td>0.2668</td>
<td>0.2729</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>16</td>
<td>0.1511</td>
<td>0.2718</td>
<td>0.2756</td>
<td>0.2776</td>
<td>0.2842</td>
<td>0.2848</td>
<td>0.2929</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>FCF-Relaxation</th>
<th>Time Steps</th>
<th>$\tau$</th>
<th>m</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
<th>Est.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>2</td>
<td>0.0480</td>
<td>0.0524</td>
<td>0.052</td>
<td>0.0519</td>
<td>0.0520</td>
<td>0.0520</td>
<td>0.0527</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>2</td>
<td>0.0938</td>
<td>0.0982</td>
<td>0.0975</td>
<td>0.0968</td>
<td>0.0972</td>
<td>0.0982</td>
<td>0.1547</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>4</td>
<td>0.0740</td>
<td>0.0749</td>
<td>0.0802</td>
<td>0.0794</td>
<td>0.0787</td>
<td>0.0792</td>
<td>0.0812</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>4</td>
<td>0.0924</td>
<td>0.0981</td>
<td>0.1040</td>
<td>0.1030</td>
<td>0.1020</td>
<td>0.1026</td>
<td>0.1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>16</td>
<td>0.0114</td>
<td>0.0367</td>
<td>0.0922</td>
<td>0.0961</td>
<td>0.1030</td>
<td>0.1020</td>
<td>0.1038</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>16</td>
<td>0.01231</td>
<td>0.0373</td>
<td>0.0966</td>
<td>0.1020</td>
<td>0.1096</td>
<td>0.1092</td>
<td>0.1157</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Two-level numerical convergence factors for $\tau$-MGRIT (yes in first column) and MGRIT (no is first column) with FCF-relaxation. The final column indicates the error estimate derived in Section 4.2.

These tables highlight two important facts. First in all cases, the analysis in the previous section provides a good bound on the convergence rate. Second, the convergence rate of both the $\tau$-MGRIT and MGRIT algorithms does not vary noticeably with respect to the problem size. In other words, the two-level $\tau$-MGRIT algorithm exhibits good parallel weak scalability.
5.3. Multilevel results. As discussed in Remark 5.3, the multilevel \( \tau \)-MGRIT algorithm solves the coarse grid problem using standard multilevel MGRIT, compared to the two-level approach where direct sequential time integration is used. Table 5.3 presents the number of iterations required for the multilevel MGRIT and \( \tau \)-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>
<thead>
<tr>
<th>Method</th>
<th>Relax</th>
<th>( m )</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau ) No</td>
<td>F</td>
<td>4</td>
<td>18</td>
<td>20</td>
<td>21</td>
<td>23</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>Yes</td>
<td>F</td>
<td>4</td>
<td>21</td>
<td>22</td>
<td>24</td>
<td>24</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>No</td>
<td>F</td>
<td>16</td>
<td>15</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Yes</td>
<td>F</td>
<td>16</td>
<td>15</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>No</td>
<td>FCF</td>
<td>4</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Yes</td>
<td>FCF</td>
<td>4</td>
<td>11</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>No</td>
<td>FCF</td>
<td>16</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Yes</td>
<td>FCF</td>
<td>16</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

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

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

5.4. Strong scaling analysis. For a fixed time step, standard time integration with Richardson extrapolation requires the completion of \( CN_t > N_t \) sequential time integration steps. The multi-level, iterative, and concurrent nature of \( \tau \)-MGRIT means that the user must complete \( DN_p > CN_t \) time integration steps split roughly uniformly across \( P \) temporal processors. This combination of increased overall cost and increased potential for concurrency manifests as a crossover point whereat the additional temporal concurrency eventually overcomes the additional work.

Figures 5.4 and 5.5 present a strong scaling study for \( \tau \)-MGRIT applied to the 1D heat equation with F- and FCF-relaxation, respectively. In all cases, the problem size included 16,384 time steps and 16,384 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 scaling behavior for \( \tau \)-MGRIT and MGRIT is very similar. As expected, both \( \tau \)-MGRIT and MGRIT performed better when FCF-relaxation was used. Comparing the two figures shows that a larger coarsening factor \( (m = 16) \) leads to an earlier crossover point. Reducing the coarsening factor delays
Fig. 5.4: Strong Scaling study for $\tau$-MGRIT applied to the 1D heat equation with F-relaxation. Black lines represent SDIRK-1 methods. Magenta lines represent results found using SDIRK-2.

Fig. 5.5: Strong Scaling study for $\tau$-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, parallel speedup.
for second order MGRIT with SDIRK-2 (black lines) and second order τ-MGRIT with SDIRK-1 (magenta 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 the 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 increasing 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.
6.1. $\tau$-MGRIT for non-uniform temporal grids. Define a non-uniform temporal grid with time steps $\delta t_j$ 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 $\Delta T_j = \sum_{i=0}^{m_j-1} \delta t_i$ 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 $\Phi_j = \Phi_j(u_{j-1}, t, \delta t_j)$ represent the time integration operator at $t = t_j$. Next, assume the error introduced by a single time step with $\Phi$ 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 $\delta t_j$ is small enough and that the solution, $u(x,t)$, is sufficiently smooth enough such that $C_j$ is constant across the given course grid interval. Then, the global error introduced by time integration across the $j$'th 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

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

An important difference between $\tau$-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 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

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

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

### 6.1.1. Numerical tests

In this section, results showing $\tau$-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 $t = 2\pi$, were set as coarse-grid points. Fine grid points were then determined by uniformly splitting 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 consisted 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 used $N_t = 16384$, SDIRK-1 and standard second order differencing in space.

Figure 6.3 shows the error across the temporal domain for MGRIT and $\tau$-MGRIT applied on the two non-uniform temporal grids described above. In both cases, the $\tau$-MGRIT algorithm is an effective approach for reducing the error. 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
Fig. 6.4: Wall time and digits of accuracy per second for the $\tau$-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.

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 $\tau$-MGRIT algorithm is capable of improving the accuracy of the underlying time-integration scheme with little increase in overall cost when compared to the standard MGRIT algorithm. It is important to note that not all problems are suitable for use with $\tau$-MGRIT; however, $\tau$-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 $\tau$-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 $\tau$-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 $\tau$-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 [23].

REFERENCES

time integration with multigrid reduction for a compressible fluid dynamics application.,

[8] R. D. Falgout, T. A. Manteuffel, B. O’Neill, and J. B. Schroeder, Multigrid reduction in
time for nonlinear parabolic problems: A case study, SIAM Journal on Scientific Computing,

Multigrid-in-Time Algorithm for Solving Evolution Equations in Parallel, in Presented at:
Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO,

approaches for eddy current problems, in Progress in Industrial Mathematics at ECMI 2018,


deep residual neural networks, SIAM Journal on Data Science, (2019 (accepted)). arXiv


multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers

[16] M. Jung and U. Rüde, Implicit extrapolation methods for multilevel finite element computations,


[18] S. Kamkar, A. Wissink, V. Sankaran, and A. Jameson, Combined feature-driven richardson-
based adaptive mesh refinement for unsteady vortical flows, AIAA journal, 50 (2012),
pp. 2834–2847.

in time method for power systems, Power and Energy Society General Meeting (PESGM),
(2016).


Analysis and Applied Mathematics, T. E. Simos, ed., no. 1048 in AIP Conference


[24] 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

Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition,
2010, p. 126.

[26] U. Rüde, The hierarchical basis extrapolation method, SIAM Journal on Scientific and Statistical

[27] J. B. Schroeder, 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 & Energy
Society General Meeting (PESGM), IEEE, 2018, pp. 1–5.

Optimizing mgrid and parareal coarse-grid operators for linear advection, 2019, https:

[29] N. A. P. V. Dobrev, Tz. Kolev and J. B. Schroeder, Two-level convergence theory for