Multigrid Interpretation of a Three-Level Parareal Algorithm

Stephanie Friedhoff, Martin J. Gander, and Felix Kwok

1 Introduction

Parallel-in-time methods, of which parareal [13] and multigrid reduction in time (MGRIT) [3] are well-known examples, are important tools for increasing parallelism beyond traditional spatially parallel methods, see [6, 14] and references therein. As a two-level method, parareal performs the fine but expensive integration independently (and in parallel) over many short time intervals, and it uses a cheap (but coarse) integrator to correct values across time subintervals sequentially. For linear ODE systems, parareal iterates are known to be equivalent to two-level MGRIT ones for a specific choice of initial guess, restriction/prolongation operators and relaxation scheme, cf. [3, 9, 10]. One can thus analyze parareal convergence in two ways: one can make hypotheses on Lipschitz constants and truncation errors, which is typical in the ODE community, cf. [1, 8, 13], or one can use spectral information of all-at-once matrices, as is common in the multigrid community, see [2, 3, 5, 15].

When parareal and MGRIT are used with many time subintervals, the coarse correction step becomes a computational bottleneck. To overcome this, one can parallelize the coarse solution by subdividing the coarse problem and using a coarser level to ensure global communication. For MGRIT, this leads to a multilevel variant [11]; for parareal, a three-level variant has been introduced and analyzed in [12]. In this paper, we show that there is a choice of restriction/prolongation operators and relaxation schemes such that the resulting MGRIT method is equivalent to three-level parareal when applied to linear problems. The existing MGRIT literature can thus add to our understanding of three-level parareal, beyond what is shown in [12].

Stephanie Friedhoff
Bergische Universität Wuppertal, e-mail: friedhoff@math.uni-wuppertal.de

Martin J. Gander
Université de Genève, e-mail: martin.gander@unige.ch

Felix Kwok
Université Laval, e-mail: felix.kwok@mat.ulaval.ca

2 The three-level parareal algorithm

Suppose one wishes to solve the *linear* system of ODEs $u' = \Phi u + f(t)$ with initial conditions $u(0) = u_0$ on the interval $[0, T]$. To obtain the temporal grid for both parareal and MGRIT, we subdivide the interval hierarchically as follows:[1]

- The interval $[0, T]$ is subdivided into p coarsest intervals $\mathcal{I}_i = [T_{i-1}, T_i]$, $i = 1, \dots, p$, each of length $\Delta T = T/p$;
- Each coarsest interval $\mathcal{I}_i$ is subdivided into m subintervals $\mathcal{I}_{i,j} = [t_{i,j}, t_{i,j+1}]$, $j = 0, 1, \dots, m-1$, of length $\Delta t = \Delta T/m$;
- Each $\mathcal{I}_{i,j}$ is divided into intervals $[t_{i,j,k}, t_{i,j,k+1}]$ $(0 \le k < n)$ of length $\delta t = \Delta t/n$.

We can now define the following *propagators*, which take an initial value at the beginning of $\mathcal{I}_i$, $\mathcal{I}_{i,j}$ or $\mathcal{I}_{i,j,k}$ and return the solution at the end of the interval:[2]

- F_0 is the action of the fine integrator over one fine time step δt. For a linear problem, we have $F_0 u_{i-1} = \Phi_0 u_{i-1} + f_i$.
- $F = F_0^n$ is the action of the fine integrator over one intermediate time step $\Delta t = n\delta t$. For a linear problem, we have $F u_{i-n} = \Phi_0^n u_{i-n} + \sum_{k=0}^{n-1} \Phi_0^k f_{i-k}$.
- G is the action of the intermediate integrator over one intermediate time step Δt. For a linear problem, we have $G U_{i,j-1} = \Phi_1 U_{i,j-1} + \gamma_{i,j}$.
- H is the action of the coarse integrator over one coarse time step $\Delta T = m\Delta t$. For a linear problem, we have $H Y_{i-1} = \Phi_2 Y_{i-1} + \eta_i$.

The three-level parareal algorithm, as introduced in [12], iterates on the level-1 state variables $U_{i,j}$ and level-2 state variables Y_i as follows:

1. Initialization (with iteration indices appearing as superscripts):

$$\begin{aligned} Y_0^0 &= u_0, & Y_i^0 &= HY_{i-1}^0 \\ U_{i,0}^0 &= Y_{i-1}^0, & U_{i,j}^0 &= GU_{i,j-1}^0 \qquad (1 \le j \le m), \end{aligned}$$

2. Iteration: for $\nu = 0, 1, 2, \dots$,

$$U_{i,0}^{\nu+1} = Y_{i-1}^{\nu}, \qquad U_{i,j}^{\nu+1} = FU_{i,j-1}^{\nu} + GU_{i,j-1}^{\nu+1} - GU_{i,j-1}^{\nu} \qquad (1 \le j \le m), \tag{1}$$

$$Y_0^{\nu+1} = u_0, \qquad Y_i^{\nu+1} = U_{i,m}^{\nu+1} + HY_{i-1}^{\nu+1} - HY_{i-1}^{\nu}. \tag{2}$$

This method is shown in [12] to converge to the fine solution *in finitely many steps*, i.e., $U_{i,j}^{\nu} = F^{(i-1)m+j} u_0$ for $\nu \ge i(m+1)$, *for any choice of G and H*. Note that this is not a nested iteration, where one needs to iterate U or Y to sufficient accuracy before switching levels; instead, only one parareal step on $U_{i,j}$ is performed before it is used in (2), and one coarse parareal step (2) is performed before the Y_i are used as new initial values in (1).

[1] For ease of explanation, we assume that all subdivisions have equal length, although it is easy to see that similar results hold for non-uniform subdivisions.

[2] To lighten the notation, the time index is only indicated in the variable on which the propagators are applied, and not in the propagators themselves.

Algorithm 1 MGRIT(ℓ, $\tilde{\mathbf{g}}^{(\ell)}$) (in correction form, as defined in [3])

```
if $\ell$ is the coarsest level $L$ then
    Solve coarse grid system $A_L \mathbf{u}^{(L)} = \tilde{\mathbf{g}}^{(L)}$
else
    Relax on $A_\ell \mathbf{u}^{(\ell)} = \tilde{\mathbf{g}}^{(\ell)}$ using $F$-relaxation
    Compute and restrict residual using injection: $\tilde{\mathbf{g}}^{(\ell+1)} = R_\ell^{\ell+1}(\tilde{\mathbf{g}}^{(\ell)} - A_\ell \mathbf{u}^{(\ell)})$
    Solve on the next level : MGRIT($\ell + 1$, $\tilde{\mathbf{g}}^{(\ell+1)}$)
    Correct: $\mathbf{u}^{(\ell)} \leftarrow \mathbf{u}^{(\ell)} + P_{\ell+1}^{\ell} \mathbf{u}^{(\ell+1)}$
end if
```

Algorithm 2 MGRIT-FAS(ℓ, $\mathbf{u}^{(\ell)}$, $\mathbf{g}^{(\ell)}$) (as defined in [4])

```
if $\ell$ is the coarsest level $L$ then
    Solve coarse grid system $A_L(\mathbf{u}^{(L)}) = \mathbf{g}^{(L)}$
else
    Relax on $A_\ell(\mathbf{u}^{(\ell)}) = \mathbf{g}^{(\ell)}$ using $F$-relaxation to obtain $\mathbf{v}^{(\ell)}$
    Compute FAS right hand side: $\mathbf{g}^{(\ell+1)} = R_\ell^{\ell+1}(\mathbf{g}^{(\ell)} - A_\ell(\mathbf{v}^{(\ell)})) + A_{\ell+1}(R_\ell^{\ell+1}\mathbf{v}^{(\ell)})$
    Solve on the next level : MGRIT-FAS($\ell + 1$, $\mathbf{u}^{(\ell+1)}$, $\mathbf{g}^{(\ell+1)}$)
    Correct: $\mathbf{u}^{(\ell)} \leftarrow \mathbf{v}^{(\ell)} + P_{\ell+1}^{\ell}(\mathbf{u}^{(\ell+1)} - R_\ell^{\ell+1}\mathbf{v}^{(\ell)})$
end if
```

3 Equivalence with the MGRIT V-cycle

The initial value problems that are solved by the propagators can also be written as linear systems of the type $A_\ell \mathbf{u}^{(\ell)} = \mathbf{g}^{(\ell)}$, where

$$A_\ell = \begin{bmatrix} I & & & \\ -\Phi_\ell & I & & \\ & \ddots & \ddots & \\ & & -\Phi_\ell & I \end{bmatrix}.$$

The index ℓ here indicates the level of coarseness of the temporal grid, with $\ell = 0$ being the finest grid, and $\ell = 2$ being the coarsest for a three-level method. Such systems can be solved using the MGRIT V-cycle with F-relaxation algorithm, which can be written in correction form [3] or as a full approximation scheme (FAS) [4], see Algorithms 1 and 2. Here, we consider the special case of $L = 2$, i.e., the three-level algorithm. For the purpose of writing the recurrence, we will index the fine grid (level-0) solution as $u_{i,j,k} \approx u(t_{i,j,k})$. The level-1 vectors will be double indexed as $u_{i,j} \approx u(t_{i,j})$, and level-2 vectors are singly indexed as $u_i \approx u(T_{i-1})$. If injection is used for $P_{\ell+1}^{\ell}$ and $R_\ell^{\ell+1} = (P_{\ell+1}^{\ell})^T$ in Algorithm 2, then one V-cycle of MGRIT-FAS with F-relaxation for solving $A_0\mathbf{u} = f$ updates the iterate $u_{i,j,k}$ as follows:

1. Relax on level 0:

$$v_{i,j,k} = \begin{cases} f_{i,j,k} + \Phi_0 v_{i,j,k-1}, & 1 \leq k \leq n-1, \quad \forall i, j, \\ u_{i,j,0}, & k = 0. \end{cases}$$

2. Compute FAS right-hand side for level 1:

$$g_{i,j} = \begin{cases} f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} - \Phi_1 u_{i,j-1,0}, & 1 \le j \le m-1, \\ f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} - \Phi_1 u_{i-1,m-1,0}, & j = 0. \end{cases}$$

3. Relax on level 1 using initial guess $(\mathbf{u}^{(1)})_{i,j} = v_{i,j,0} = u_{i,j,0}$:

$$v_{i,j} = \begin{cases} f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} + \Phi_1 (v_{i,j-1} - u_{i,j-1,0}), & 1 \le j \le m-1, \\ u_{i,0,0}, & j = 0. \end{cases}$$

4. Compute FAS right-hand side for level 2:

$$g_i = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) - \Phi_2 u_{i-1,0,0}.$$

5. Solve the level-2 system:

$$u_i^{\text{new}} = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2 (u_{i-1}^{\text{new}} - u_{i-1,0,0}).$$

6. Correct on level 1 and then on level 0, using injection for both levels: we set for all $1 \le i \le p$

$$u_{i,j,k}^{\text{new}} = \begin{cases} f_{i,j,k} + \Phi_0 v_{i,j,k-1}, & 1 \le k \le n-1, \ \forall j, \\ f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} + \Phi_1 (v_{i,j-1} - u_{i,j-1,0}), & k = 0, 1 \le j \le m-1, \end{cases}$$
$$u_{i,0,0}^{\text{new}} = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2 (u_{i-1}^{\text{new}} - u_{i-1,0,0}).$$

We can now prove the following equivalence theorem.

Theorem 1 *For the linear problem $u' = \Phi u + f(t)$, assume that $u_{i,j,k}^0$ satisfies*

$$u_{1,0,0}^0 = u_0, \quad u_{i,0,0}^0 = H u_{i-1,0,0}^0 \ \forall i \ge 1, \quad u_{i,j,0}^0 = G u_{i,j-1,0}^0 \ \forall j = 1, \dots, m-1.$$

Then for all $\nu \ge 0$, the three-level MGRIT-FAS V-cycle with F-relaxation and with injection as the prolongation operator is equivalent to three-level parareal via

$$u_{i,j,k}^{\nu+1} = \begin{cases} F_0^k U_{i,j}^\nu, & 1 \le k \le n-1, \quad \forall i, j, \\ U_{i,j}^{\nu+1}, & k = 0, 1 \le j \le m-1, \quad \forall i \ge 1, \\ Y_{i-1}^{\nu+1}, & j = k = 0, \quad \forall i \ge 1. \end{cases}$$

Proof From the initialization conditions, we have for $\nu = 0$ that $u_{i,j,0}^\nu = U_{i,j}^\nu$ for $1 \le j \le m-1$, and $u_{i,0,0}^\nu = Y_{i-1}^\nu$ for all i. We will prove by induction that these two equalities also hold for $\nu \ge 1$. To do so, we rewrite $u_{i,j,k}^{\text{new}}$ in terms of the propagators F_0, F, G and H. The update formula at step 6 leads us to consider three cases:

Case 1 ($k \neq 0$). Step 1 at iteration ν reads

$$u_{i,j,k}^{\text{new}} = v_{i,j,k} = F_0 v_{i,j,k-1} = \cdots = F_0^k v_{i,j,0} = F_0^k U_{i,j}^\nu.$$

Case 2 ($k = 0, j \neq 0$). This case is given by step 3, where

$$u^{\text{new}}_{i,j,0} = v_{i,j} = F_0 v_{i,j-1,n-1} + \Phi_1(v_{i,j-1} - U^{\nu}_{i,j-1}) = F_0^n U^{\nu}_{i,j-1} + G v_{i,j-1} - G U^{\nu}_{i,j-1}.$$

Here, we have replaced the difference of Φ_1 by a difference of G, because G is affine. Thus, we have $v_{i,j} = U^{\nu+1}_{i,j}$ for $1 \le j \le m-1$, since both quantities are initialized the same way (we have $v_{i,0} = u_{i,0,0} = Y^{\nu}_{i-1} = U^{\nu+1}_{i,0}$) and satisfy the same recurrence.

Case 3 ($j = k = 0$). Here we have $u^{\text{new}}_{i,0,0} = u^{\text{new}}_i$, so step 5 gives, for $i \ge 2$,

$$\begin{aligned} u^{\text{new}}_i &= f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1(v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2(u^{\text{new}}_{i-1} - u_{i-1,0,0}) \\ &= F_0 v_{i-1,m-1,n-1} + \Phi_1(v_{i-1,m-1} - U^{\nu}_{i-1,m-1}) + \Phi_2(u^{\text{new}}_{i-1} - u_{i-1,0,0}) \\ &= F_0^n U^{\nu}_{i-1,m-1} + G v_{i-1,m-1} - G U^{\nu}_{i-1,m-1} + H u^{\text{new}}_{i-1} - H u_{i-1,0,0} \\ &= U^{\nu+1}_{i-1,m} + H u^{\text{new}}_{i-1} - H Y^{\nu}_{i-2}. \end{aligned}$$

For $i = 1$, we have $u^{\text{new}}_1 = u_0 = Y^{\nu+1}_0$; thus, u^{new}_i and $Y^{\nu+1}_{i-1}$ satisfy the same recurrence with the same initial condition. This leads to $u^{\text{new}}_{i,0,0} = Y^{\nu+1}_{i-1}$ for all i, as claimed. □

We can now use the FAS formulation to deduce the equivalence in classical (correction) form. We define the following operators:

$$E_\ell = I - P^{\ell}_{\ell+1} R^{\ell+1}_{\ell}, \qquad M_\ell = \operatorname{diag}((A_\ell)_{11}, (A_\ell)_{22}, \ldots),$$

where $(A_\ell)_{ii}$ are diagonal blocks of A_ℓ corresponding to the ith subinterval, *starting* with the coarse point and including all the fine points until (but excluding) the next coarse point. In other words, M_ℓ is the block Jacobi smoother for level ℓ, and E_ℓ blanks out the coarse points and retains the fine points when applied to a vector of values at level ℓ. Similar operators were defined in [10], where the authors proved the equivalence between two-level parareal and a geometric multigrid method with block Jacobi smoothing and aggressive coarsening in the FAS setting; however, the blocks in [10] are defined differently, with the coarse points appearing at the end of the block rather than the beginning. We write the change in the solution at step 6 as

$$u^{\text{new}}_{i,j,k} - u_{i,j,k} = \begin{cases} v_{i,j,k} - u_{i,j,k} =: (\Delta \mathbf{u}^{(0)})_{i,j,k}, & k \neq 0, \\ v_{i,j} - u_{i,j,0} =: (\Delta \mathbf{u}^{(1)})_{i,j}, & k = 0, j \neq 0, \\ u^{\text{new}}_i - u_{i,0,0} =: (\Delta \mathbf{u}^{(2)})_i, & j = k = 0. \end{cases}$$

To compute $\Delta \mathbf{u}^{(0)}$, note that $v_{i,j,k} - u_{i,j,k} = 0$ when $k = 0$; for $k \neq 0$, we have

$$\begin{aligned} (\Delta \mathbf{u}^{(0)})_{i,j,k} = v_{i,j,k} - u_{i,j,k} &= f_{i,j,k} + \Phi_0(v_{i,j,k-1} - u_{i,j,k-1}) + \Phi_0 u_{i,j,k-1} - u_{i,j,k} \\ &= (\mathbf{f} - A_0 \mathbf{u})_{i,j,k} + \Phi_0 (\Delta \mathbf{u}^{(0)})_{i,j,k-1}. \end{aligned}$$

If we move $\Phi_0(\Delta \mathbf{u}^{(0)})_{i,j,k-1}$ to the left and recall the definition of M_0, we get

$$M_0 \Delta \mathbf{u}^{(0)} = E_0(\mathbf{f} - A_0 \mathbf{u}) \implies \Delta \mathbf{u}^{(0)} = M_0^{-1} E_0 \tilde{\mathbf{g}}^{(0)},$$

where $\tilde{\mathbf{g}}^{(0)} = \mathbf{f} - A_0\mathbf{u}$ is the initial residual. This is almost the same as in [10], except the residual is blanked before the smoothing, instead of after. Next, we calculate

$$(\Delta\mathbf{u}^{(1)})_{i,j} = v_{i,j} - u_{i,j,0} = \begin{cases} 0, & j = 0, \\ g_{i,j} + \Phi_1(v_{i,j-1} - u_{i,j-1,0}) + \Phi_1 u_{i,j-1,0} - u_{i,j,0}, & \\ & j \neq 0, \end{cases}$$

which implies

$$M_1\Delta\mathbf{u}^{(1)} = E_1(\mathbf{g}^{(1)} - A_1R_0^1\mathbf{u}) = E_1R_0^1(\mathbf{f}^{(0)} - A_0(\mathbf{u} + \Delta\mathbf{u}^{(0)})).$$

Thus, $\Delta\mathbf{u}^{(1)} = M_1^{-1}E_1\tilde{\mathbf{g}}^{(1)}$, where $\tilde{\mathbf{g}}^{(1)} = R_0^1(\tilde{\mathbf{g}}^{(0)} - A_0\Delta\mathbf{u}^{(0)})$. Finally, we have

$$(\Delta\mathbf{u}^{(2)})_i = u_i^{\text{new}} - u_{i,0,0} = g_i + \Phi_2(u_{i-1}^{\text{new}} - u_{i-1,0,0}) + \Phi_2 u_{i-1,0,0} - u_{i,0,0},$$

which leads to

$$A_2\Delta\mathbf{u}^{(2)} = \mathbf{g}^{(2)} - A_2R_0^2\mathbf{u} = R_1^2(\mathbf{g}^{(1)} - A_1(R_0^1\mathbf{u} + \Delta\mathbf{u}^{(1)})) = R_1^2(\tilde{\mathbf{g}}^{(1)} - A_1\Delta\mathbf{u}^{(1)}).$$

We conclude, by replacing $\Delta\mathbf{u}^{(1)}$ with $M_1^{-1}E_1\tilde{\mathbf{g}}^{(1)}$ in the last step, that

$$\begin{aligned} \mathbf{u}^{\text{new}} - \mathbf{u} &= \Delta\mathbf{u}^{(0)} + P_1^0\Delta\mathbf{u}^{(1)} + P_2^0\Delta\mathbf{u}^{(2)} \\ &= \Delta\mathbf{u}^{(0)} + P_1^0(\Delta\mathbf{u}^{(1)} + P_2^1A_2^{-1}R_1^2(\tilde{\mathbf{g}}^{(1)} - A_1\Delta\mathbf{u}^{(1)})) \\ &= \Delta\mathbf{u}^{(0)} + P_1^0((I - P_2^1A_2^{-1}R_1^2A_1)M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2)\tilde{\mathbf{g}}^{(1)} \end{aligned}$$

Defining $T = (I - P_2^1A_2^{-1}R_1^2A_1)M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2$, we continue to calculate

$$\begin{aligned} \mathbf{u}^{\text{new}} - \mathbf{u} &= \Delta\mathbf{u}^{(0)} + P_1^0TR_0^1(\tilde{\mathbf{g}}^{(0)} - A_0\Delta\mathbf{u}^{(0)}) \\ &= (P_1^0TR_0^1 + (I - P_1^0TR_0^1A_0)M_0^{-1}E_0)(\mathbf{f} - A\mathbf{u}) =: \mathcal{P}(\mathbf{f} - A\mathbf{u}). \end{aligned}$$

We conclude that the error propagator reads

$$\mathcal{S} = I - \mathcal{P}A_0 = (I - P_1^0TR_0^1A_0)(I - M_0^{-1}E_0A_0),$$

where the operator T satisfies $I - TA_1 = (I - P_2^1A_2^{-1}R_1^2A_1)(I - M_1^{-1}E_1A_1)$. Note that the preconditioners $\mathcal{P}$ and T can also be written as

$$\mathcal{P} = M_0^{-1}E_0 + P_1^0TR_0^1(I - A_0M_0^{-1}E_0), \qquad T = M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2(I - A_1M_1^{-1}E_1).$$

We can hence interpret the action of the preconditioner $\mathcal{P}$ as follows:

1. $M_0^{-1}E_0$: Take the fine residual, blank out the coarse points and apply block Jacobi.
2. $I - A_0M_0^{-1}E_0$: Update the residual after relaxation.
3. $P_1^0TR_0^1$: Restrict the new residual, recursively solve the coarse problem, then update the coarse points by injection.

Since T acts the same way but at a coarser level, the action of $\mathcal{P}$ corresponds to exactly one MGRIT V-cycle with F-relaxation, written in correction form.

Remark If one replaces injection with injection plus F-relaxation (like in standard MGRIT), then the equivalent parareal formulation at the νth iteration would be

$$\begin{aligned}
U_{i,0}^{\nu+1/2} &= Y_{i-1}^{\nu}, & U_{i,j}^{\nu+1/2} &= GU_{i,j-1}^{\nu+1/2} + FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu} & &(1 \le j \le m),\\
Y_0^{\nu+1} &= u_0, & Y_i^{\nu+1} &= U_{im}^{\nu+1/2} + HY_{i-1}^{\nu+1} - HY_{i-1}^{\nu}, & &\\
U_{i,0}^{\nu+1} &= Y_{i-1}^{\nu+1}, & U_{i,j}^{\nu+1} &= GU_{i,j-1}^{\nu+1} + FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu} & &(1 \le j \le m).
\end{aligned}$$

Note that the term $FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu}$ is used twice, but it only needs to be computed once using a fine propagation. The intermediate propagation G, however, needs to be computed twice, since it is applied once to $U_{i,j-1}^{\nu+1/2}$, and another time to $U_{i,j-1}^{\nu+1}$.

4 Numerical example

We present the numerical example in [7], where the advection-diffusion equation $u_t = u_x + \kappa u_{xx}$ with periodic boundary conditions $u(0,t) = u(2,t)$, $u_x(0,t) = u_x(2,t)$ is solved on $t \in (0,4)$, with $\kappa = 1/1024$ (advection-dominated case) and $u(x,0) = e^{-20(x-1)^2}$. We discretize the problem using second order finite difference in space and backward Euler in time, with $\Delta x = 1/20$ and $\delta t = 1/1280$. For two-level parareal, the coarse propagator is backward Euler with $\Delta T = 1/2$ (8 coarse steps with 640 fine steps per coarse step). For three-level parareal, we use an intermediate level with $\Delta t = 1/128$ (10 fine steps per intermediate step), while keeping $\Delta T = 1/2$ for the coarsest level (i.e., 64 intermediate steps per coarse step). In Figure 1, we compare two-level and three-level parareal, both with and without post-smoothing. We compare both the iteration count and the *idealized* running time, as measured by the number of *non-concurrent* backward Euler steps taken at all levels; this cost is normalized by that of sequential time-stepping, so that a cost of 1 means the same cost as sequential time-stepping without parallelization. We see that two-level parareal

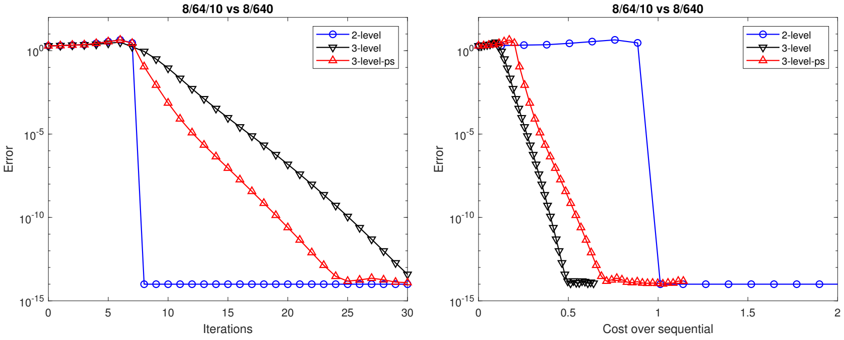

Fig. 1 Left: Iteration count for two-level parareal, and three-level parareal, with and without post-smoothing. Right: Computational cost of the three methods, as measured by the number of backward Euler steps taken, normalized by the cost of sequential time-stepping.

converges to the exact solution in 8 iterations, whereas the three-level variants take many more iterations. However, the three-level iterations are much more parallel and take less time to run than a two-level iteration. In particular, both three-level versions converge with cost much lower than 1; such speedup is not possible for two-level parareal. Finally, although post-smoothing reduces the number of three-level iterations, the higher cost per iteration (two intermediate propagations rather than one) makes it slower than no post-smoothing once the normalized cost is considered.

Acknowledgements F. Kwok acknowledges support from the National Science and Engineering Research Council of Canada (RGPIN- 2021-02595). The work described in this paper is partially supported by a grant from the ANR/RGC joint research scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and the French National Research Agency (Project no. A-CityU203/19). This project has received funding from the Federal Ministry of Education and Research and the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701, Time-X. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland.

References

1. Bal, G. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: *Domain Decomposition Methods in Science and Engineering XV*, 425–432. Springer (2005).
2. Dobrev, V., Kolev, T., Petersson, N. A., and Schroder, J. B. Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM J. Sci. Comput.* **39**(5), S501–S527 (2017).
3. Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. Parallel time integration with multigrid. *SIAM J. Sci. Comput.* **36**(6), C635–C661 (2014).
4. Falgout, R. D., Katz, A., Kolev, T. V., Schroder, J. B., Wissink, A., and Yang, U. M. Parallel time integration with multigrid reduction for a compressible fluid dynamics application. Tech. Rep. LLNL-JRNL-663416, Lawrence Livermore National Laboratory (2015).
5. Friedhoff, S. and MacLachlan, S. A generalized predictive analysis tool for multigrid methods. *Numer. Linear Algebra Appl.* **22**(4), 618–647 (2015).
6. Gander, M. J. 50 years of time parallel time integration. In: *Multiple Shooting and Time Domain Decomposition Methods*, 69–113. Springer (2015).
7. Gander, M. J. Five decades of time parallel time integration, and a note on the degradation of the performance of the Parareal algorithm as a function of the Reynolds number. *Oberwolfach Report* (2017).
8. Gander, M. J. and Hairer, E. Nonlinear convergence analysis for the parareal algorithm. In: *Domain Decomposition Methods in Science and Engineering XVII*, 45–56 (2008).
9. Gander, M. J., Kwok, F., and Zhang, H. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Comput. Visualization Sci.* **19**(3), 59–74 (2018).
10. Gander, M. J. and Vandewalle, S. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.* **29**(2), 556–578 (2007).
11. Hessenthaler, A., Southworth, B. S., Nordsletten, D., Röhrle, O., Falgout, R. D., and Schroder, J. B. Multilevel convergence analysis of multigrid-reduction-in-time. *SIAM J. Sci. Comput.* **42**(2), A771–A796 (2020).
12. Kwok, F. Analysis of a three-level variant of parareal (In preparation, 2022).
13. Lions, J.-L., Maday, Y., and Turinici, G. Résolution d'EDP par un schéma en temps «pararéel». *C. R. Acad. Sci., Ser. I: Math.* **332**(7), 661–668 (2001).
14. Ong, B. W. and Schroder, J. B. Applications of time parallelization. *Computi. Visualization Sci.* **23**(1), 1–15 (2020).
15. Southworth, B. S. Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. *SIAM J. Matrix Anal. Appl.* **40**(2), 564–608 (2019).