SIAM J. SCI. COMPUT.
Vol. 46, No. 4, pp. A2709–A2736

MULTILEVEL PARAREAL ALGORITHM WITH AVERAGING FOR OSCILLATORY PROBLEMS*

JULIANE ROSEMEIER†, TERRY HAUT‡, AND BETH WINGATE§

Abstract. The present study is an extension of the work done by Peddle, Haut, and Wingate [*SIAM J. Sci. Comput.*, 41 (2019), pp. A3476–A3497] and Haut and Wingate [*SIAM J. Sci. Comput.*, 36 (2014), pp. A693–A713], where a two-level Parareal method with mapping and averaging is examined. The method proposed in this paper is a multilevel Parareal method with arbitrarily many levels, which is not restricted to the two-level case. We give an asymptotic error estimate which reduces to the two-level estimate for the case when only two levels are considered. Introducing more than two levels has important consequences for the averaging procedure, as we choose separate averaging windows for each of the different levels, which is an additional new feature of the present study. The different averaging windows make the proposed method especially appropriate for nonlinear multiscale problems, because we can introduce a level for each intrinsic scale of the problem and adapt the averaging procedure such that we reproduce the behavior of the model on the particular scale resolved by the level. The method is applied to nonlinear differential equations. The nonlinearities can generate a range of frequencies in the problem. The computational cost of the new method is investigated and studied on several examples.

Key words. time-stepping, multilevel method, Parareal method, averaging, parallel-in-time

MSC codes. 68Q25, 68R10, 68U05

DOI. 10.1137/23M1547123

1. Introduction.

The main motivation for this study is the application and analysis of a multilevel Parareal method to oscillatory PDEs of type (1.1); see below. These types of PDEs are important in geophysical and astrophysical fluid dynamics and represent the underpinning equations used in climate and weather prediction. Furthermore, numerical discretizations of these equations are posing significant challenges for contemporary computer architectures, so there is a strong motivation to explore new ways of exploiting the increasing parallelism [15]. The multiscale equations of type (1.1) can have more than one oscillatory linear term and hence more than one small parameter, and have a nonlinear term that can generate a wide range of frequencies. In [12, 18] a two-level Parareal method was shown to have the potential for significant parallel speedups. However, given the wide range of frequencies possible

*Submitted to the journal's Numerical Algorithms for Scientific Computing section January 17, 2023; accepted for publication (in revised form) March 14, 2024; published electronically August 20, 2024.

https://doi.org/10.1137/23M1547123

Funding: The work of the first author was supported by the German Research Foundation (DFG) through the Walter Benjamin Programme, project "Formulation and numerical computation of the low frequency mean flow of fluids," project 463179503. This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 "Scaling Cascades in Complex Systems," project 235221301. This work was supported by UKRI EPSRC grant EP/R029628/1, by Leverhulme Trust Project RPG-2017-098, by Leverhulme Trust Fellowship RF-2022-013, by the University of Exeter, and by Lawrence Livermore National Laboratory. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Mathematical Multifaceted Integrated Capability Center (MMICC), award DE-SC0023164. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344, LLNL-JRNL-862718-DRAFT.

†University of Exeter, Exeter EX4 4PY, UK; Current address: Freie Universität Berlin, Berlin, Germany (j.rosemeier@exeter.ac.uk).

‡Lawrence Livermore National Laboratory, Livermore, CA 94551 USA (haut3@llnl.gov).

§University of Exeter, Exeter EX4 4PY, UK (b.wingate@exeter.ac.uk).

in PDEs of type (1.1), in this paper we ask the following question: Given a two-level Parareal method with a fixed number of iterations, can we find a multilevel Parareal method that achieves the same accuracy with a faster wall clock time, assuming we have as many processors as we need? This central question is illustrated graphically in Figure 1.

The focus of this paper is on the Parareal method, a time-parallel method first proposed in [17]. Since its publication a lot of research has been done on the method with the aim to exploit the advantages of parallelism in time. For instance, [18] and [12] combine the Parareal method with averaging to solve fluid-dominated problems. Several studies treat the well-known stability issues related to problems of this type; see, for instance, [19] or [24]. Convergence of the Parareal method is also discussed in [11], [9], or [1]. Moreover, the authors of [8] give several interpretations of the Parareal algorithm and especially show its relation to the MGRIT algorithm [6]. Standard Parareal has been shown to be a special case of MGRIT with particular projection and prolongation operators. Further comparisons of MGRIT and Parareal were made in [8, 10, 6, 23]. As such, MGRIT is a more general framework than multilevel Parareal. Our combination of multilevel Parareal with hierarchical nonlinear averaging methods can also be used with MGRIT and is a topic for future research. Investigations of the error and convergence can be found, for example, in [23], [22], or [7]. There are also attempts to improve the understanding of hyperbolic or advection-dominated problems; see [2], [13], or [25], just to name a few. The study [16] introduces a micro-macro Parareal algorithm. The coarse propagator computes a cheap approximation to a slow, macroscopic model, whereas the fine propagator solves the full microscopic problem. The idea to use simpler models for the coarse propagator is related to the construction of the coarse propagators in the present study, which relies on a mapping and averaging method.

The problems under consideration in the present study are oscillatory equations of the following form:

$$\frac{d\mathbf{u}}{dt} + \sum_{i=1}^{M} \frac{1}{\varepsilon_i} \mathcal{L}_i \mathbf{u} = \mathcal{N}(\mathbf{u}). \tag{1.1}$$

The linear operators $\mathcal{L}_i$ are skew Hermitian, i.e., they have purely imaginary eigenvalues and are responsible for temporal oscillations in the solution. The parameters ε_i with $\varepsilon_{i+1} \leq \varepsilon_i$ can be small and can make the system stiff. Furthermore, the nonlinear term $\mathcal{N}$ is a quadratic nonlinearity and can produce a wide range of temporal frequencies. In addition, a diffusive term $\mathcal{D}$ can be added in (1.1). In this study we are interested in developing a multilevel Parareal scheme, which we define below, that has the potential to handle a range of different frequencies. Though many choices are possible, for this work we consider the following cases:

1. a single linear oscillatory term $\varepsilon_1 = \varepsilon$ and a range of frequencies from $\mathcal{N}$ (see Examples 5.3 and 5.2),
2. two different linear oscillatory terms and a range of frequencies from $\mathcal{N}$ (see Example 5.4 with $\epsilon_1 = 1/10, \epsilon_2 = 1/100$),
3. two different linear oscillatory terms with $\epsilon_1 = \epsilon_2$ and a range of frequencies from $\mathcal{N}$ (see Example 5.4, $\epsilon_1 = \epsilon_2 = 1/10$). Though this case can be written as case 1, above, the comparison of this case with case 2 occurs in applications and will be considered in the examples.

In the following sections we analyze case 1 in the list above, but will provide example calculations for all three cases in section 5. We apply a transformation to (1.1); see subsection 2.1.1. In this paper, we will use transformation and mapping interchangeably. The resulting system is solved with a method combining the Parareal method and an averaging technique; see subsection 2.1.2.

The Parareal method was first presented in [17]. It is a time-parallel method with two levels. For the exposition in this study we enumerate the levels, i.e., for the two-level method we have level 1 and level 0. On level 1 a coarse time grid is introduced and on level 0 we have several fine time grids. On the coarse time grid of level 1 a coarse propagator, for instance, a Runge–Kutta method, is applied to compute a numerical approximation to a differential equation. This numerical approximation is then improved iteratively using solutions computed with the fine propagator on level 0 in parallel. We can define the multilevel Parareal algorithm recursively in the levels until we reach the two-level case. The idea of the multilevel Parareal algorithm with L levels is that the coarse propagator provides a solution on the coarsest level and the fine propagator is a multilevel Parareal algorithm with $L-1$ levels. More detailed descriptions of the two-level and multilevel methods can be found in subsections 2.2 and 2.3. To mitigate the oscillatory stiffness caused by the skew Hermitian linear operator $\mathcal{L}$, an averaging method is applied on coarse levels; see subsection 2.1.2. This leads the multilevel Parareal method with averaging.

A major achievement of the present work is an asymptotic convergence proof. The error estimate for a two-level Parareal method is extended to the case of multiple levels with and without averaging. This is a generalization of the classical proof found in [9] and the convergence proof for the APinT method found in [18]. The new proof has two main steps. In the mentioned literature, error estimates for the two-level case can be found; however, the fine propagator is assumed to be the exact propagator. When we introduce multiple levels we cannot make this assumption, because we want to know how the error contributions that emerge on the finer levels propagate through the different coarser levels and we will find that they are amplified by an amplification factor that depends on the details of the scheme. Consequently, the first step of the new proof is to refine the two-level estimate to the case where the fine propagator is not exact. The second main step is to apply an inductive argument to obtain an estimate for the multilevel case. This might be beneficial for identifying on which level the dominant error contribution emerges and therefore how the time-steps or the number of iterations on the different levels should be chosen to reach a certain error tolerance. For this idea error estimators would be needed. This is a possible future extension of the presented work.

We want to answer the question if multilevel methods can have a faster runtime than two-level methods while achieving the same accuracy. As a measure for the runtime, we count the number of serial steps, which are explained in the next paragraph and section 4. Let us suppose that we are given a two-level method with a fixed number of iterations and consequently a fixed accuracy. The aim is to design a multilevel method which is as accurate and faster. If we do not compute too many iterations with the multilevel method, the number of serial steps done with the multilevel method is less than the number of serial steps done with the two-level method. However, we expect that we have to do several iterations with the multilevel method to reach the accuracy of the two-level method. This is illustrated in Figure 1. An example will be discussed in subsection 5.4.

In the present study, a numerical time-stepping method is said to be efficient if it computes a solution with good accuracy in a short amount of time; thus efficiency relates accuracy and runtime. Accuracy is determined by the error of the numerical time-stepping method. The number of right-hand-side evaluations is a measure for the runtime if the code is not parallelized. In the special case of the Parareal method, some right-hand-side evaluations can be computed at the same time when a parallelized numerical code is run. Therefore, the notion of serial steps is introduced in section

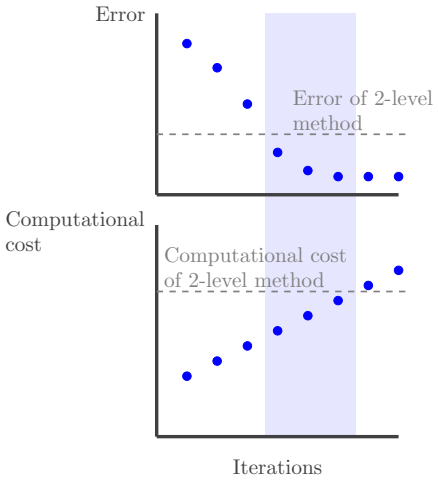

FIG. 1. *The blue dots in the upper/lower figure show that the error/computational cost of a multilevel Parareal method decreases/increases with an increasing number of iterations. For comparison, the dashed gray line in the upper/lower figure illustrates the error/computational cost of a two-level Parareal method with a fixed number of iterations. The shaded blue area shows how many iterations can be done when the multilevel method is both more exact and less computationally expensive than the two-level method.*

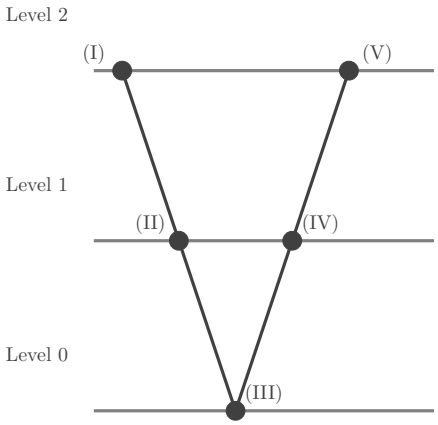

FIG. 2. *Parareal algorithm with three levels,* $k_2 = 1$ *iteration on level* 2 *and* $k_1 = 1$ *iteration on level* 1. (I) *Compute initial guess on coarsest level, level* 2; (II) *get initial values from level* 2, *compute initial guess on level* 1 *in parallel;* (III) *apply fine propagator in parallel, get initial values from level* 1; (IV) *compute Parareal iteration with fine solution from level* 0; (V) *compute Parareal iteration with fine solution from level* 1.

4. If two right-hand-side evaluations can be computed at the same time, they belong to the same serial step. Efficiency, as explained above, must be distinguished from parallel efficiency of a parallel algorithm, which is defined as the speedup divided by the number of processors used.

The paper is organized as follows. The algorithms are described in section 2. Especially, subsection 2.1 shows the reformulation of (1.1) for the case $M = 1$. The exposition in subsection 2.2 is about the two-level Parareal schemes, and subsection 2.3 contains a description of the new multilevel schemes. Asymptotic convergence results with and without averaging can be found in section 3. In section 4 the issue of computational cost of the multilevel Parareal schemes is discussed. Numerical examples can be found in section 5. Finally, in section 6 a discussion of the results and a conclusion are given.

2. Formulation of the algorithm.

The Parareal method was first formulated in [17]. Versions with mapping and averaging incorporated can be found in [12] or [18]. Further, subsection 2.2 summarizes the methods. Then, subsection 2.3 presents multilevel versions.

2.1. Reformulation of the problem.

2.1.1. The transformation. We analyze the problem by taking $M = 1$ in (1.1) and dropping the subscripts. The following transformation, also used in [21] and related to the *method of cancellation of oscillations,* is applied to the above system to eliminate the linear term

$$\mathbf{w}(t) = \exp\left(\frac{\mathcal{L}}{\varepsilon}t\right)\mathbf{u}(t). \tag{2.1}$$

The transformed system, denoted as the *modulation equation,* since its time evolution is more regular than (1.1), admits the form

$$\frac{d\mathbf{w}}{dt} = \exp\left(\frac{\mathcal{L}}{\varepsilon}t\right)\mathcal{N}\left(\exp\left(-\frac{\mathcal{L}}{\varepsilon}t\right)\mathbf{w}\right). \tag{2.2}$$

There exist numerous scientific applications which have the form of (1.1), including examples that occur in atmospheric and oceanic simulations, like the swinging spring [14], also called the elastic pendulum, or the rotating shallow water equations (RSWEs); see [4].

As the operator $\mathcal{L}$ is skew Hermitian, the norm of the right-hand side of (2.2) can be bounded independent of ϵ. Especially, applying the transformation (2.1) to the problem (1.1) eliminated the linear term and made the problem smoother. However, taking higher order derivatives of $\mathbf{w}$ we see that with each order we get an additional power of $1/\epsilon$ in the derivative. To further mitigate the oscillatory stiffness we apply averaging techniques. This is important when numerical time-stepping schemes are applied, since the truncation error depends on higher order derivatives.

Here, the importance of applying the transformation and solving the modulation equation instead of the system without transformation is explained. First, using the modulation equation is the first step which makes the equations smoother. Second, for an autonomous oscillatory problem, the information if the solution reaches a valley or a peak of an oscillation at time t is in the initial conditions. That means if we average the initial data of an autonomous, oscillatory differential equation, we cannot expect to compute the phase correctly. In subsection 5.2 we will show for the nonautonomous modulation equation that the phase is computed correctly with the proposed method, although the averaging removes almost all the information about the phase on the coarse levels.

2.1.2. Averaging. In order to explain how an averaging integral mitigates the oscillatory stiffness due to fast oscillations, let us assume that a slow function is superimposed by a fast periodic function with zero mean, that is, we consider a function $f(t) = f_{\text{slow}}(t) + f_{\text{fast}}(t)$. An example for a fast function would be $f_{\text{fast}}(t) = e^{2\pi i n t}$, where $|n| \gg 0$. Integrating the fast function over an interval of length η where η is as large as a few times the period of the fast periodic function, i.e., computing $1/\eta \int_{-\eta/2}^{\eta/2} f_{\text{fast}}(t+s)ds$, the positive and negative contributions cancel each other. However, for the averaging the integrand is weighted by a scaled kernel function ρ with compact support, which decays fast close to the boundary of the compact support. Moreover, the knowledge of the exact period is not assumed in the method. More precisely, an integral of the form $1/\eta \int_{-\eta/2}^{\eta/2} \rho(s/\eta) f(t+s)ds$ is evaluated. Therefore, in general we do not observe an exact cancellation of the oscillations but rather a mitigation; see, for example, Figure 4. The technical details can be found in [5] in Lemma 2.2.

Let us now formulate an averaged version of the modulation equation. In the averaged equation the right-hand side of (2.2) is replaced. In particular, when the coarse propagator is applied, an approximation to the following equation is computed:

$$\frac{d\bar{\mathbf{w}}(\mathrm{t})}{dt} = \frac{1}{\eta}\int_{-\eta/2}^{\eta/2} \rho\left(\frac{s}{\eta}\right)\exp\left(\frac{L}{\epsilon}(s+t)\right)\mathcal{N}\left(\exp\left(-\frac{L}{\epsilon}(s+t)\right)\bar{\mathbf{w}}(t)\right)ds, \tag{2.3}$$

where η is the averaging window and ρ is the kernel function. We use

$$\rho(s) = \frac{1}{\rho_0} \exp\left(\frac{1}{(s-1/2)(s+1/2)}\right), \tag{2.4}$$

where ρ_0 normalizes the function. In the right-hand side of (2.3) we write $\bar{\mathbf{w}}(t)$ to emphasize that the equation depends on t only and not on s. The oscillations in the equation with period $\leq \eta$ are averaged. The exponential oscillates fast in s, whereas the nonlinearity is slowly varying. The scaled filter function is slowly varying too. In that case, we integrate the oscillations which come from the exponential over a few periods. The length of the integration interval is not necessarily an integer multiple of the period and damped oscillations remain after the integration. The ideas of temporal averaging were also investigated in related contexts such as ODEs [20], in the context of heterogeneous multiscale methods [3], [5], and in PDEs analysis [4].

The averaging process requires the formulation of analytical equations. In particular, for each level $l, l \neq 0$, we formulate an averaged, analytical equation which depends on a level-dependent averaging window η_l and shows the same behavior as the original system, that is, the modulation equation, on the coarser scales up to the smallest scale that can still be resolved when the averaging window η_l is used, but whose behavior on the finer scales is different. Especially, the features on the finer scales are averaged in the averaged equation. The step of formulating analytical equations is skipped when other strategies to deal with the fast components, for example, applying implicit methods, are used. However, the analytical equations provide descriptions of physical phenomena and are therefore a link to the theory and modeling of the physical application considered.

2.2. Two-level Parareal and two-level Parareal with averaging. The two-level Parareal method has two levels, denoted as level 1 and level 0. Level 1 is the coarse level where we do $\hat{N}$ time-steps on a coarse grid, which has $\hat{N}+1$ grid points. Two neighboring grid points of the coarse grid form an interval. In total we have $\hat{N}$ such small intervals and on each small interval we introduce a fine grid. These are the fine grids of level 0.

The Parareal method is a parallel-in-time method which has two basic solvers, a coarse propagator denoted as G^1 and a fine propagator denoted as P^0. The upper indices refer to the levels on which the propagators are applied. The coarse and the fine propagators can be Runge–Kutta methods, but other choices are possible too. First, the coarse propagator is applied on level 1 to compute initial guesses U_n^0, where the index n counts the time-steps on level 1. The initial guesses are computed in serial and are improved iteratively. The values provided through the initial guess at the end of a coarse step are passed to the fine grids on level 0 as initial values. Then the fine propagator P_0 is applied in parallel. The results of the fine propagator are passed back to the coarse level 1. Applying the coarse propagator again, a Parareal iteration step can be computed:

$$U_{n+1}^{k+1} = G^1(U_n^{k+1}) + P^0(U_n^k) - G^1(U_n^k). \tag{2.5}$$

The upper index of the numerical solutions counts the iterations. Once the first iteration is computed, it can be used as a new initial guess and the next iteration can be done.

This procedure is modified in the APinT method where averaging is incorporated; see [12] and [18]. Here, the coarse propagator, denoted as $\bar{G}^1$, provides a numerical solution to an averaged problem (2.3) and not the original system (2.2). Solving

an averaged problem has the advantage that the fast oscillations, which are still in the modulation equation, are further mitigated, provided the averaging window η is chosen appropriately. Thus, the oscillatory stiffness is mitigated and taking large time-steps is possible when the numerical method of the coarse propagator is applied. This can be beneficial for the efficiency of the algorithm. We can formulate a Parareal iteration step with averaging incorporated as follows:

$$U_{n+1}^{k+1} = \bar{G}^1(U_n^{k+1}) + P^0(U_n^k) - \bar{G}^1(U_n^k), \tag{2.6}$$

where $\bar{G}^1$ denotes the coarse propagator, which provides a numerical solution to the averaged problem (2.3) on the coarse level, level 1.

2.3. Multilevel Parareal and multilevel Parareal with averaging. Here we state again the idea behind the multilevel Parareal algorithm: The multilevel Parareal algorithm uses a recursion in the levels. Especially, the multilevel Parareal algorithm with L levels is a two-level Parareal algorithm where we compute a coarse solution with the coarse propagator on the coarsest level and the fine propagator is a multilevel Parareal algorithm with $L-1$ levels applied on the finer levels. Then applying the multilevel Parareal algorithm can be iterated until we reach the two-level case, where the two-level Parareal algorithm, described in the previous subsection, is applied.

To distinguish between the different levels, we introduce the subsequent notation: When we consider a multilevel Parareal method with L levels, level $L-1$ denotes the coarsest level and level 0 is the finest level. If $l_1 > l_2$ for $l_1, l_2 \in \{0, \dots L-1\}$, then level l_1 is the coarser and level l_2 the finer level. The coarse propagator applied on level l is denoted by G^l or in the case when an averaged equation is solved by $\bar{G}^l$. The fine propagator needed to compute the Parareal solution on level l is denoted as P^{l-1} or $\bar{P}^{l-1}$ if the fine propagator solves an averaged problem.

Here, the multilevel Parareal method with and without averaging is explained. Algorithms 2.1, 2.3, and 2.4 show the method without averaging. The case with averaging can be found in Algorithms 2.2, 2.3, 2.4, 2.5, 2.6, and 2.7. Moreover, the multilevel algorithm with averaging included is illustrated in the supplemental material in Figure SM1.

We assume that a method with L levels is applied. On the coarsest level, which is denoted as level $L-1$, a coarse propagator is applied. The biggest time-steps are used on that level. For the method without averaging, the coarse propagator solves the unaveraged modulation equation (2.2). When the method with averaging is applied, the coarse propagator solves an averaged modulation equation of the form (2.3). For the averaging, the averaging window η_{L-1} is used. Note the index $L-1$.

The coarse propagator provides the initial values for the fine propagator, which can be applied in parallel. The following two cases must be distinguished for the method with L levels:

- $L > 2$: For that case, the fine propagator is a multilevel Parareal method with $L-1$ levels, which consists of a coarse propagator applied on level $L-2$ and a fine propagator. If the method without averaging is used, the coarse propagator on level $L-2$ is applied to problem (2.2). For the method with averaging, (2.3) is solved on level $L-2$ with an averaging window η_{L-2}, which satisfies $\eta_{L-2} \leq \eta_{L-1}$.
- $L = 2$: To compute solutions with the fine propagator, the unaveraged modulation equation (2.2) is solved for the method without averaging as well as with averaging on the finest level, level 0.

As the fine solution is computed, a correction iteration can be done. For this, the coarse propagator must be evaluated again. To improve the accuracy of the approximation, the process of computing corrections can be iterated.

Algorithm 2.1 Multilevel Parareal method without averaging.

multi_level_method$(w_0, t_0, N, \Delta t, l)$:
$\mathbf{w}^{0,l} \leftarrow$ propagator$(w_0, t_0, N, \Delta t)$
for $k = 0, \ldots, K_{\max} - 1$:
  parfor $n = 0, \ldots, N - 1$:
    if $l > 1$:
      $\mathbf{w}_n^{l-1} \leftarrow$ multi_level_method$(\mathbf{w}^{k,l}[n], t_0 + n\Delta t, N, \Delta t/N, l - 1)$
    else:
      $\mathbf{w}_n^{l-1} \leftarrow$ propagator$(\mathbf{w}^{k,l}[n], t_0 + n\Delta t, N, \Delta t/N)$
    end if
  end parfor
  for $n = 0, \ldots, N - 1$:
    $\mathbf{w}^{k+1,l}[n+1] \leftarrow$ step$(\mathbf{w}^{k+1,l}[n], t_n, \Delta t) + \mathbf{w}_n^{l-1}[N] - \mathbf{w}^{k,l}[n+1]$
  end for
end for
return $\mathbf{w}^{K_{\max},l}$

Algorithm 2.2 Multilevel Parareal method with averaging.

av_multi_level_method$(w_0, t_0, N, \Delta t, l, \eta_l)$:
$\mathbf{w}^{0,l} \leftarrow$ av_propagator$(w_0, t_0, N, \Delta t, \eta_l)$
for $k = 0, \ldots, K_{\max} - 1$:
  parfor $n = 0, \ldots, N - 1$:
    if $l > 1$:
      $\mathbf{w}_n^{l-1} \leftarrow$ av_multi_level_method$(\mathbf{w}^{k,l}[n], t_0 + n\Delta t, N, \Delta t/N, l - 1, \eta_{l-1})$
    else:
      $\mathbf{w}_n^{l-1} \leftarrow$ propagator$(\mathbf{w}^{k,l}[n], t_0 + n\Delta t, N, \Delta t/N)$
    end if
  end parfor
  for $n = 0, \ldots, N - 1$:
    $\mathbf{w}^{k+1,l}[n+1] \leftarrow$ av_step$(\mathbf{w}^{k+1,l}[n], t_n, \Delta t, \eta_l) + \mathbf{w}_n^{l-1}[N] - \mathbf{w}^{k,l}[n+1]$
  end for
end for
return $\mathbf{w}^{K_{\max},l}$

Algorithm 2.3 Basic propagator.

propagator$(w_0, t_0, N, \Delta t)$:
$\mathbf{w}[0] \leftarrow w_0$
for $n = 0, \ldots, N - 1$:
  $\mathbf{w}[n+1] \leftarrow$ step$(\mathbf{w}[n], t_n, \Delta t)$
end for
return $\mathbf{w}$

Algorithm 2.4 One step with RK2 as a basic propagator.

step$(x, t, \Delta t)$:
$k_1 \leftarrow e^{\mathcal{L}/\epsilon t}\mathcal{N}(e^{-\mathcal{L}/\epsilon t}x)$; $\quad k_2 \leftarrow e^{\mathcal{L}/\epsilon(t+1/2\Delta t)}\mathcal{N}(e^{-\mathcal{L}/\epsilon(t+1/2\Delta t)}(x + \Delta t/2k_1))$
$\tilde{x} \leftarrow x + \Delta t/2(k_1 + k_2)$
return $\tilde{x}$

Algorithm 2.5 Averaged basic propagator.

av_propagator$(w_0, t_0, N, \Delta t, \eta)$:
$\mathbf{w}[0] \leftarrow w_0$
for $n = 0, \ldots, N-1$:
  $\mathbf{w}[n+1] \leftarrow$ av_step$(\mathbf{w}[n], t_n, \Delta t, \eta)$
end for
return $\mathbf{w}$

Algorithm 2.6 One step for an averaged equation with RK2 as a basic propagator.

av_step$(x, t, \Delta t, \eta)$:
$k_1 \leftarrow$av_RHS(x, t, η), $\quad k_2 \leftarrow$av_RHS$(x + \Delta t/2k_1, t + \Delta t/2, \eta)$
$\tilde{x} \leftarrow x + \Delta t/2(k_1 + k_2)$
return $\tilde{x}$

Algorithm 2.7 Evaluate averaged right-hand side with trapezium rule.

av_RHS$(w, t, \eta, M = 100)$:
parfor $i = 1, \ldots, M-1$
  $s_i \leftarrow -\eta/2 + i/M\eta$
end parfor
$\bar{w} = 1/(\eta M)$ sum$(\rho(s_i/\eta)e^{\mathcal{L}/\epsilon(t+s_i)}\mathcal{N}(e^{-\mathcal{L}/\epsilon(t+s_i)}w))$
return $\bar{w}$

A basic principle of the Parareal method is that an initial guess is iteratively improved. These correction iterations are done on the levels $L-1, L-2, \ldots, 1$ in the multilevel case, and inspired by multigrid methods we can adopt the terminology of cycles. The case when only one iteration is computed on the levels $1, \ldots, L-1$ is denoted as a V-cycle. Illustrations of methods with three levels can be found in Figure 2 and in the supplemental material in Figure SM2. Figure 2 shows the case if one iteration is done on level 1 and level 2; in Figure SM2 we do two iterations on levels 1 and 2.

The algorithm with averaging is particularly promising for multiscale problems. The strategy for the application of the method to such problems can be formulated as follows. Suppose we are given a problem with several time scales. For each intrinsic scale of the problem we can introduce a level and resolve the properties of the system which are the specific for that scale, i.e., every scale is assigned a level and the scale specific properties are resolved on the level assigned. We average such that we keep the features of the original system, for instance, a modulation equation, on that scale, but the finer components of the equation vanish through the averaging process. This step requires a convenient choice of the averaging windows η_l. The averaging windows define the lengths of the intervals over which filter integrals are computed;

see subsection 2.1.2. In the multilevel case, they are level-dependent. As the averaging procedure makes the system more well-behaved for numerical time-stepping methods, the time-steps, which can be used for the averaged problems, are larger than the ones required for the unaveraged problem. Note that only on the finest level, level 0, the full, unaveraged system is solved.

3. Convergence results for the multilevel methods. The proof for the multilevel method is an inductive proof for which we can apply similar arguments as in the two-level case. The difference is that the fine propagator is not a basic ODE solver, like a Runge–Kutta scheme, but it is the multilevel Parareal method on the finer grids. Only on the finest level, level 0, the finest propagator is a basic ODE solver.

Moreover, we introduce a coarsening factor N which relates the time-steps on the different levels. Let ΔT_0 be the time-step on the finest grids on level 0. Then, the grids on level 1 have a coarser time-step, given by $\Delta T_1 = N\Delta T_0$. For level 2, we have a time-step $\Delta T_2 = N\Delta T_1 = N^2\Delta T_0$ and so on. See the relations (B.11) and (B.12).

THEOREM 3.1. *Let the coarse propagators G_l applied on the levels $l = 1, \dots, L-1$ satisfy the conditions*

$$E(x) - G^l(x) = c_{p+1}^{\ l}(x)\Delta T_l^{p+1} + c_{p+2}^{\ l}(x)\Delta T_l^{p+2} + \dots, \quad \text{and} \tag{3.1}$$

$$\|G^l(x) - G^l(y)\| \le (1 + C_2^l \Delta T_l)\|x - y\|, \tag{3.2}$$

where $E(x)$ denotes the exact propagator, i.e., $E(x)$ is the exact solution of the initial valued problem with initial value x when the time span ΔT_l has passed. Suppose that the error of the fine propagator, on level 0, is bounded by $c\Delta T_0^{p_0+1}$. Then we can show the following error bound for the multilevel Parareal method without averaging:

$$e_{n,l}^{k_l} \le \sum_{\bar{l}=1}^{l} E_{\bar{l}} \prod_{j=\bar{l}+1}^{l} A_j + \delta_0 \prod_{\bar{l}=1}^{l} A_{\bar{l}}, \tag{3.3}$$

where $e_{n,l}^{k_l}$ denotes the bound of an error on level l after k_l iterations. The index n counts the grid points. In addition, the following notation is used:

$$E_{\bar{l}} = \begin{cases} \binom{n}{k_{\bar{l}}+1}\gamma_{\bar{l}}\alpha_{\bar{l}}^{k_{\bar{l}}}\beta_{\bar{l}}^{N-k_{\bar{l}}-1}, & \bar{l} = l, \\ \binom{N}{k_{\bar{l}}+1}\gamma_{\bar{l}}\alpha_{\bar{l}}^{k_{\bar{l}}}\beta_{\bar{l}}^{N-k_{\bar{l}}-1}, & \bar{l} < l, \end{cases} \tag{3.4}$$

$$\delta_0 = c\Delta T_1 \Delta T_0^{p_0}, \tag{3.5}$$

$$A_l = N\beta_l^{N-1}(1 + \alpha_{0N,l})^{N-1}. \tag{3.6}$$

We have $\alpha_l = C_{1,l}\Delta T_l^{p+1}, \beta_l = (1 + C_{2,l}\Delta T_l), \gamma_l = C_{3,l}\Delta T_l^{p+1}$, and $\alpha_{0N,l} \in [0, \alpha_l]$.

Especially, the derived error estimate is a sum. The different terms of the sum are composed of the errors $E_{\bar{l}}$ emerging on level $\bar{l}$ and amplification factors $A_{\bar{l}}$ which amplify the errors made on the finer levels as the fine solutions are passed to the coarser levels. This result can be concluded from the more general result in Theorem 3.3 by setting $\eta = 0$.

COROLLARY 3.2. *Assuming that a constant coarsening factor N relates the different levels and the number of iterations on the levels is constant, i.e., $k_l = k$ for all $l = 1, \dots, L-1$, we get the following error bound:*

$$\begin{aligned}\|u(T_n)-U_n^{k+1}\| \le cT\Delta T_0^{p_0}\exp\left(C_2T\frac{1-1/N^L}{1-1/N}+C_1T\Delta T_1^p\frac{1-1/N^{L(p+1)}}{1-1/N^{p+1}}\right)\\ +\exp\left(\frac{C_2T}{1-1/N}+\frac{C_1T\Delta T_1}{1-1/N^{p+1}}\right)\\ \times C_3C_1^k\binom{N}{k+1}\frac{1}{1-1/N^{kp+k+p}}\Delta T_1^{kp+k+p+1}.\end{aligned} \tag{3.7}$$

(The notational conventions from Appendix B *are used. The coarse propagators are assumed to have the accuracy order* p*.) Particularly, we recover the accuracy order of the two-level scheme.*

Proof. The corollary is an immediate consequence of the results given in Theorem 3.1, Lemma B.5, and Lemma B.6. ∎

A convergence result for the multilevel Parareal algorithm with averaging is presented. Assuming we have L levels in total, the averaging is done on the levels $1,\dots,L-1$. Only on level 0, the level with the finest grids, the full, unaveraged system is solved by the finest propagator.

Especially, the finest propagator solves the following exact equation:

$$\frac{dw}{dt}(t)-e^{\frac{\mathcal{L}}{\epsilon}t}\mathcal{N}(e^{-t/\epsilon\mathcal{L}}w(t))=0. \tag{3.8}$$

The coarse propagators do not solve the exact equation. Instead, they solve averaged equations, given through

$$\frac{d\bar{w}_l(t)}{dt}(t)-e^{\frac{\mathcal{L}}{\epsilon}t}\frac{1}{\eta_l}\int_{-\eta_l/2}^{\eta_l/2}\rho\left(\frac{s}{\eta}\right)e^{\frac{\mathcal{L}}{\epsilon}s}\mathcal{N}(e^{-\frac{\mathcal{L}}{\epsilon}(t+s)}\bar{w}_l(t))\,ds=0. \tag{3.9}$$

This is the crucial difference between the Parareal algorithms with and without averaging. We will use the following notation to denote the propagators relevant for this section:

- E—analytically exact solver for the unaveraged equation (3.8).
- $\bar{E}^l$—analytically exact solver for the averaged equation (3.9).
- $\bar{G}^l$—numerical solver of the averaged problem (3.9) on level l, $l=1,\dots,L-1$, also the coarse propagator of the multilevel method $\bar{P}^l$.
- $\bar{P}^l$—multilevel Parareal method with l levels with averaging, the coarse propagator is $\bar{G}_l$, and the fine propagator is $\bar{P}^{l-1}$ for $l>1$ (or P^0 for $l=1$).
- P^0—numerical solver of the unaveraged problem (3.8) on level 0.

For the propagators $\bar{G}_l$, $l=1,\dots,L-1$, we impose the following conditions on the truncation error:

$$E(x)-\bar{G}^l(x)=\bar{c}_{p+1}^{\,l}(x)\Delta T_l^{p+1}+\bar{c}_{p+2}^{\,l}(x)\Delta T_l^{p+2}+\dots, \quad \text{and} \tag{3.10}$$

$$\|\bar{G}^l(x)-\bar{G}^l(y)\|\le(1+C_2^l\Delta T_l)\|x-y\|. \tag{3.11}$$

THEOREM 3.3. *Suppose that the coarse propagators* $\bar{G}^l$ *satisfy* (3.10) *and the Lipschitz condition* (3.11) *for* $l=1,\dots,L-1$. *Let the truncation error of the fine propagator, on level* 0*, be bounded by* $c\Delta T_0^{p_0+1}$. *Then we can show that the following error estimate for the multilevel Parareal method with averaging holds:*

$$e_{n,l}^{k_l}\le\sum_{\bar{l}=1}^{l}\bar{E}_{\bar{l}}\prod_{j=\bar{l}+1}^{l}A_j+\delta_0\prod_{\bar{l}=1}^{l}A_{\bar{l}}, \tag{3.12}$$

where

$$\bar{E}_{\bar{l}} = \begin{cases} \binom{n}{k_{\bar{l}}+1} \bar{\gamma}_{\bar{l}} \bar{\alpha}_{\bar{l}}^{k_{\bar{l}}} \beta_{\bar{l}}^{N-k_{\bar{l}}-1}, & \bar{l} = l, \\ \binom{N}{k_{\bar{l}}+1} \bar{\gamma}_{\bar{l}} \bar{\alpha}_{\bar{l}}^{k_{\bar{l}}} \beta_{\bar{l}}^{N-k_{\bar{l}}-1}, & \bar{l} < l, \end{cases} \tag{3.13}$$

$$\delta_0 = c \Delta T_1 \Delta t^{p_0}, \tag{3.14}$$

$$A_l = N \beta_l^{N-1} (1 + \bar{\alpha}_{0N,l})^{N-1}. \tag{3.15}$$

We have

$$\bar{\alpha}_l = \tilde{C}_{1,l} \eta_l \epsilon + C_{1,l} \Delta T_l^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_l, \omega) \right\}, \tag{3.16}$$

$$\beta_l = (1 + C_{2,l} \Delta T_l), \tag{3.17}$$

$$\bar{\gamma}_l = \tilde{C}_{3,l} \eta_l \epsilon \|\tilde{\mathcal{M}}_1\| + C_{3,l} \Delta T_l^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_l, \omega) \right\} \|\tilde{\mathcal{M}}_0\|, \ \textit{and} \tag{3.18}$$

$$\alpha_{0N,l} \in [0, \bar{\alpha}_l]. \tag{3.19}$$

Remarks.

1. The error constants, which arise as numerical time-stepping schemes, are applied to the differential equations specific to the different levels and should be level-dependent. For stiff problems the constants are big; for nonstiff problems they are smaller. Moreover, the error constants grow with the length of the integration interval. Consequently, when the averaging windows are chosen appropriately, the constants can be roughly the same for each level.
2. The structure of the amplification factor is the same as in the case without averaging. However, the terms $\bar{\alpha}_l$ and $\bar{\gamma}_l$, which cause the contraction in the error, have an additional term that accounts for the error due to averaging.

The proof can be found in Appendix A.1 and is a generalization of the two-level proofs given in [9, 18].

4. Computational cost. In this section, we distinguish between the total number of steps done on a level and the serial number of steps. Suppose on the coarsest level we do X steps with the coarse propagator, for example, to compute the initial guess. Then the total number of steps on that level is X and the serial number of steps on that level is X, too. Now, we go to the next finer level and the coarsening factor is N. We then have NX as the total number of steps and N as the number of serial steps (done on X grids in parallel). Refining again leads to N^2X as the total number of steps and N serial steps (done on NX grids in parallel) and so on.

Figure 3 shows the example of a two-level method with $X = 3$ steps on the coarse level and a coarsening factor $N = 3$. Consequently, on the coarse level, 3 steps must be computed sequentially. On the fine level, we have 9 steps in total, but parallelization leads to only 3 serial steps. (As an example for the parallelism on the fine level, all steps, which belong to serial step d, can be computed at the same time.)

We define the cost of an algorithm as the number of serial steps. The computational cost of a two-level method is given by

$$C^2 = k_1 (N_1 + N_0) + N_1. \tag{4.1}$$

The cost of the three-level Parareal algorithm can be defined as

$$C^3 = k_2 (N_2 + k_1 (N_1 + N_0) + N_1) + N_2, \tag{4.2}$$

where N_i is the number of serial steps on level i and k_i is the number of iterations on level i. For a V-cycle we have $k_i = 1$.

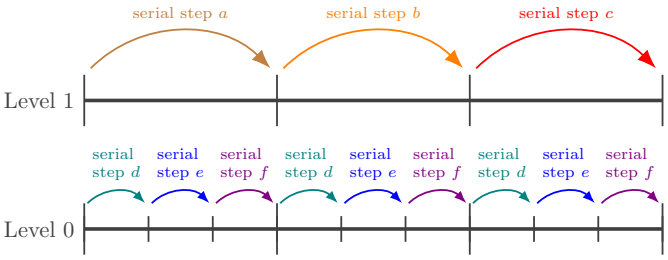

FIG. 3. *Illustration of serial steps. Same colors stand for steps which can be computed at the same time and therefore belong to the same serial step. The figure shows the example of a two-level method with $N_1 = 3$ steps on the coarse level and a coarsening factor of $N = 3$. The steps on the coarse level must be computed sequentially. On the fine level, 9 steps must be computed in total, but 3 steps can be computed at the same time. An example is the steps colored in bluish green, which belong to serial step d. So we have 3 serial steps, that is, serial steps d, e, and f, that must be computed sequentially.*

We may hypothesize that increasing the number of levels pays off when we have a broad range of scales or strong scale separation. We investigate the computational cost of a V-cycle in more detail. Suppose $\tilde{X}$ is the total number of fine steps on the finest level that must be done and N is the coarsening factor which relates the time-steps of the different levels, i.e., $\tilde{X}$ would be the number of serial steps that must be done when a nonparallelizable basic ODE solver is applied. (We might assume that two neighboring levels are related by the different coarsening factors. To keep the computation simple we will not do that here and leave it for future work.) We assume that we do N serial steps on all levels except the coarsest level. The computational cost for a V-cycle, which depends on the number of levels, can be computed as follows:

$$\text{1 level: } f_1(N) = \tilde{X} \text{ serial steps,} \tag{4.3}$$

$$\text{2 levels: } f_2(N) = N + 2\tilde{X}/N \text{ serial steps,} \tag{4.4}$$

$$\text{3 levels: } f_3(N) = N + 2N + 2\tilde{X}/N^2 = 3N + 2\tilde{X}/N^2 \text{ serial steps,} \tag{4.5}$$

$$\text{L levels: } f_L(N) = 2(L-2)N + N + 2\tilde{X}/N^{L-1} \text{ serial steps.} \tag{4.6}$$

Note that when a V-cycle is applied, only one correction iteration is computed on the levels $1, \ldots, L-1$. It is possible to find the coarsening factor N which minimizes the number of serial steps, depending on the number of levels. Solving $f_L'(N) = 0$, we find

$$N_{\text{opt}} = (\tilde{X} + \tilde{X}/(2L-3))^{1/L}. \tag{4.7}$$

However, when we choose a coarsening factor N which is computed in the described way, we adapt the algorithm to the behavior of the model on the finest scale only. It may be necessary to account for the behavior of the model on other scales too.

For a V-cycle, the total number of evaluations of the right-hand side for all the levels is not that much bigger than that of a serial time-stepper. On the finest level we do the same number of evaluations of the right-hand side that we would do with a serial time-stepping scheme, provided we apply the multilevel Parareal algorithm with averaging. When we do in total $\tilde{X}$ evaluations of the right-hand side on the finest level, level 0, the total number of right-hand-side evaluations on level 1 is $2\tilde{X}/N$, for level 2 we get $2\tilde{X}/N^2$, and so on. That means that the number of evaluations of the right-hand side decreases exponentially with the levels. Considering the multilevel

Parareal algorithm without averaging, we can possibly not coarsen in the same way because the time-step might become too large on the coarser levels and as a result the numerical solver might be unstable for the oscillatory problems.

The model for the computational cost does not distinguish between averaged and unaveraged right-hand-side evaluations; especially, the time needed to compute a serial step is the same for the different levels. The reason for this, as pointed out by [12], is that the averaging can be done in parallel and is therefore not assumed to have an impact on the wall clock speed. Furthermore, we make the assumption that the right-hand-side evaluations cause the computational cost. The exponential operator is both in the right-hand side of the averaged and unaveraged equations. As the evaluation of the averaging integral can be parallelized, we assume that the evaluation of the exponential of the linear operator causes the same cost for the averaged and unaveraged right-hand sides. In the present study, a spectral method was used and the $\mathcal{L}$ operator can be applied very rapidly via FFTs.

5. Numerical examples. For the numerical examples the explicit midpoint rule (RK2) is used as a basic integrator unless stated differently. The results presented in subsections 5.2, 5.3, and 5.4 are computed solving the modulation equations of the systems.

5.1. One-dimensional single scale example. Let us consider the problem

$$\frac{dx}{dt} = -x, \qquad x(0) = 1, \qquad t \in [0, 2], \tag{5.1}$$

which does not admit form (1.1). This problem does not have a skew Hermitian linear operator. The numerical tests will show results, which are not satisfactory. Equation (5.1) is to be solved with the multilevel Parareal algorithm without averaging and without transformation. We compute the error for one V-cycle and increase the number of levels. The time-step on the coarsest level for all the computations in Table 1 is constant $\Delta T = 0.25$. We choose a coarsening factor of $N = 10$. This means that when we have a time-step Δt on level l, the time-step on level $l-1$ is $\Delta t/10$.

We see in Table 1 that increasing the number of levels while keeping the time-step on the coarsest level constant barely changes the error of the approximations of problem (5.1). This example leads to the following observation. Adding levels while keeping the time-step on the coarsest level constant does not increase the accuracy but more computational work must be done. Therefore, using the multilevel method with a V-cycle is an unfortunate choice for problems of type (5.1). From the error bound in Theorem 3.1 it is known that every level contributes to the total error. The results for example (5.1) indicate that the main part of the error arises on the coarsest levels. For the next example in subsection 5.2 the error is not expected to come from the coarsest level. In contrast to the example in this subsection, the next example

TABLE 1

Error of the multilevel Parareal algorithm without averaging for a varying number of levels applied to problem (5.1) *where the step size on the coarsest level is* 0.25. *We see the relative error in the discrete* $l1$ *norm.*

Number of levels	Error	Number of levels	Error
2	1.2566212807763046e-05	6	1.9809615854133382e-05
3	1.9562958164422008e-05	7	1.9809616125891306e-05
4	1.9807099440426344e-05	8	1.980961620086837e-05
5	1.9809587023590493e-05		

exhibits oscillatory stiffness. Therefore, it is convenient to mitigate the stiffness with an averaging method.

5.2. One-dimensional example with fast oscillations. Let us now suppose the following situation. We have to solve a multiscale problem with very fast oscillations on the finest scale, which is resolved by the finest level. Therefore, the time-step on the finest level must be chosen small enough to resolve the very fast oscillations. In particular, the fastest scale dictates the time-step on level 0.

An example for such a problem is given by

$$\frac{dw}{dt} = -\exp(irt)w^2 \tag{5.2}$$

for large values of r. This equation can be interpreted as the one-dimensional modulation equation of the problem

$$\frac{du}{dt} = iru - u^2. \tag{5.3}$$

In contrast to the previous example, the tests for this example show benefits of the increased parallelism if the method with averaging and transformation is applied. Moreover, the effect of the transformation is explained in this subsection. In the following we will consider $r = 100$, $r = 1000$, and $r = 10000$. The exact solution of the problem (5.2) is given by

$$w(t) = \frac{rw_0}{-iw_0 \exp(irt) + iw_0 + r}. \tag{5.4}$$

The averaged problem obeys the following relation:

$$\frac{d\bar{w}_l}{dt} = (-1) \underbrace{\frac{1}{\eta_l} \int_{-\eta_l/2}^{\eta_l/2} \rho\left(\frac{s}{\eta_l}\right) \exp(irs)\, ds}_{\text{damping factor}} \exp(irt)\ \bar{w}_l(t)^2. \tag{5.5}$$

The problem (5.2) will be solved on the interval $[0, 1]$ with multilevel Parareal methods including averaging. Numerical studies show that for $r = 100$ the fastest period in the solution is of order 0.1. Therefore we choose averaging windows $\eta_1 = 0.2$ and $\eta_2 = 2$. This case is solved with a two- and a three-level method. In addition, for $r = 1000$ the fastest period in the solution is of order 0.01. Thus the averaging windows $\eta_1 = 0.02, \eta_2 = 0.2,$ and $\eta_3 = 2$ are a reasonable choice and methods with two, three, and four levels are applied. For the case $r = 10000$ the fastest period in the solution is of order 0.001. Therefore, the computations are done with the averaging windows $\eta_1 = 0.002, \eta_2 = 0.02, \eta_3 = 0.2,$ and $\eta_4 = 2$ and multilevel methods with two, three, four, and five levels are used to solve the problem. The length of the solution interval is always 1 and the coarsening factor is always 10. To solve problem (5.2) a V-cycle is applied. Depending on the value for r we choose different step sizes on the finest level:

- r = 100: The step size on the finest level is $\Delta T_0 = 10^{-3}$.
- r = 1000: The step size on the finest level is $\Delta T_0 = 10^{-4}$.
- r = 10000: The step size on the finest level is $\Delta T_0 = 2.5 \cdot 10^{-5}$.

Now consider the case $r = 1000$. A time-step of $\Delta T_0 = 10^{-4}$ means that the total number of steps which must be done on the finest level is $\tilde{X} = 10^4$. The number of serial steps for a V-cycle with a coarsening factor of $N = 10$ and two levels is given

TABLE 2

Errors at time $t = 1$ for a V-cycle including averaging with a varying number of levels applied to problem (5.2) with $r = 100, 1000,$ and 10000. The time-step on the finest level is $\Delta t = 1 \cdot 10^3, 1 \cdot 10^{-4},$ and $2.5 \cdot 10^{-5}$ depending on r. The coarsening factor to relate the levels is $N = 10$.

Number of levels	$r = 100$	$r = 1000$	$r = 10000$
2	0.0002169750591733674	2.1847140061040485e-06	3.0668862104273734e-07
3	0.00019811199764541986	2.106413305540747e-06	3.0480547757705495e-07
4		2.251750942815333e-06	3.0357016877934065e-07
5			2.7011063136189545e-07

by $f_2(10) = 2010$ according to (4.4). For three levels we get $f_3(10) = 230$ serial steps; applying (4.5) and with four levels $f_4(10) = 70$ serial steps must be done according to (4.6). However, we note that for the tests we did not choose the optimal coarsening factors, which can be computed with (4.7). In section 4 it was mentioned that for complex models possibly the time-step on the levels and thus the coarsening factor must be adapted to the problem behavior on different scales which might prevent an optimal choice of the coarsening factor according to (4.7). But also for optimal coarsening factors the number of serial steps is the smallest for the four-level method and the most steps must be done for the two-level method.

In Table 2, we find the errors at time $t = 1$. The time-step on the finest level was adapted to the fast oscillations in the problem which depend on the parameter r. Fixing one value for r, we see that the errors hardly change in the number of levels. We can draw the following conclusions. First, for the parameters chosen for the model runs the increased parallelism of the methods with 3, 4, or 5 levels should lead to a better efficiency due to a smaller number of serial steps and thus a shorter runtime. Second, having a third, fourth, or fifth level does not change the error significantly. The bound in Theorem 3.3 shows that the errors from all the levels contribute to the total error. A possible explanation for the findings in this subsection is that the main part of the error arises on the finest level, where the fast oscillations are resolved and computed numerically.

The results indicate that increasing the number of levels and thus the parallelism can lead to more efficient algorithms. A gain in efficiency, when a V-cycle is applied, should come from a combination of increased parallelism and larger time-steps on the coarser levels. For oscillatory problems larger time-steps can lead to unstable behavior of the algorithm. For example, when the case with $r = 1000$ is solved with RK2, which is an explicit Runge–Kutta method, using a time-step larger than the period of the oscillations leads to instabilities. This can be circumvented when the averaging is applied to mitigate the oscillations on the coarser levels.

In section 1, we mentioned the importance of the transformation, especially for the correct computation of the phase. This is now illustrated in Figure 4. The figure shows a solution of the averaged system (5.5) and unaveraged system (5.2), as well as the initial guess and the first two correction iterations of a three-level Parareal method with transformation and averaging. The initial guess, which was computed with the coarsest propagator of the three-level method, has a strongly attenuated amplitude. We can make the following observations for the method which solves the modulation equation (5.2): (1) The coarse propagator gives (almost) the same initial values for each fine time interval on which the fine propagator is applied. (2) Applying only the coarse propagator, we cannot compute solutions to (5.2) with the correct amplitude, as the solutions from the coarse propagator have strongly damped amplitudes. As Figure 4 shows, the Parareal method with averaging computes the

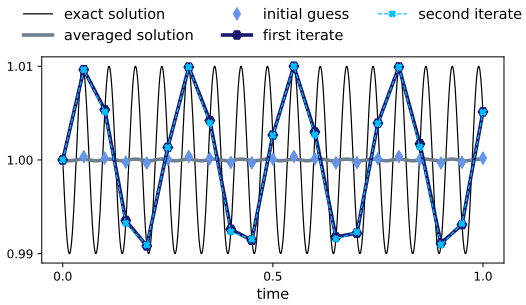

FIG. 4. *The figure shows the initial guess and the first two iterations of the multilevel Parareal method with averaging. The initial guess has strongly damped oscillations. Therefore, the fine propagator gets almost the same initial values for each interval. However, the Parareal iterations compute the phase correctly, that is, the departures from the mean behavior due to the fast oscillations are computed correctly.*

phase and amplitude correctly. Therefore, we can draw the following conclusions: (1) The correct amplitude comes from the corrections of the fine propagator, which computes a solution to the unaveraged system and (2) the information about the phase is in the right-hand side of the modulation equation, which is nonautonomous and depends on $e^{\mathcal{L}/\varepsilon t}$. Thus, the corrections with the fine propagator compute the correct phase, i.e., the departures from the mean behavior are computed correctly. In contrast, for the method with averaging applied to an autonomous problem, we cannot expect to compute the correct phase when the fine propagator starts with (almost) the same initial values on each fine grid.

5.3. Swinging spring (elastic pendulum). In this subsection we do a first test on a fluid-related, more complicated problem, the swinging spring, also denoted as the elastic pendulum. In the tests a two-level method is compared to three-level methods with different averaging windows to study the effect of the averaging. The swinging spring can be rewritten as a first order system and is given by

$$\frac{d\mathbf{u}}{dt} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{y}_1 \\ \dot{y}_2 \\ \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\omega_R^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\omega_R^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega_Z^2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \lambda x_1 z_1 \\ 0 \\ \lambda y_1 z_1 \\ 0 \\ \frac{1}{2}\lambda(x_1^2 + y_1^2) \end{pmatrix}. \tag{5.6}$$

For the numerical computation, we use the following parameters: $\omega_R = \pi, \omega_Z = 2\omega_R$, and $\lambda = 1.2\omega_Z^2$. The initial value of the modulation equation is given by $\mathbf{w}_0 = (0.006, 0., 0., 0.00489, 0.012, 0)$. Multilevel Parareal algorithms with transformation and averaging are applied to solve the system. We compare a three-level method with a two-level method and investigate the errors at time $T_{max} = 50$ for the first component of the system. For the two-level method we choose time-steps $\Delta T_1 = 5$ and $\Delta T_0 = 0.05$. The time-step ΔT_0 is small enough to resolve the fast oscillations in the problem. With the time-step ΔT_1 the coarse dynamics is resolved. Additionally, the averaging window $\eta = 2$ guarantees that the fast oscillations are averaged on the coarse level, level 1.

When applying the three-level method, two iterations are done on the intermediate level, level 1. The number of iterations on the coarse level, level 2, is varied in

the test. Moreover, the coarse and fine time-steps are the same as in the two-level case. Especially, we have $\Delta T_2 = 5$ and $\Delta T_0 = 0.05$. But we introduce an intermediate level and increase the parallelism. The time-step of the intermediate level, level 1, is $\Delta T_1 = 0.5$. In particular, the coarsening factor is $N = 10$. For the coarsest level the averaging window is chosen as $\eta_2 = 2$. We test different averaging windows on the intermediate level, that is, $\eta_1 = 0.2, 0.75$, and 2. When we choose $\eta_1 = 0.2$ we relate the averaging windows on the levels by the same coarsening factor that relates the time-steps. The choice $\eta_1 = 2$ is inspired by the dynamics of the system, because this window ensures that we average the fast oscillations also on level 1. The value $\eta_1 = 0.75$ is an intermediate choice to see how the error develops for changing averaging windows.

The reference solution is computed with the RK2 method with a time-step of $\Delta t = 0.001$. This choice makes sure that the reference solution is more accurate than the fine solver of the two-level method. The errors for the two- and three-level methods can be found in Table 3. We may observe that with an increasing number of iterations on the coarse level the accuracy of both the two- and three-level methods increases until the methods converge to the fine solver. The approximation error and thus also the efficiency of the three-level methods strongly depend on the choice of the averaging window on the intermediate level, η_1. For $\eta_1 = 2$ the convergence for the three-level method is almost the same as for the two-level method, whereas the accuracy deteriorates for smaller intermediate averaging windows. Consequently for the case $\eta_1 = 2$ we can benefit from more parallelism of the three-level method. Choosing $\eta_1 = \eta_2 = 2$ means that the same frequencies are averaged on the coarsest and second coarsest levels. Thus, for that case the solutions computed on level 1 do not represent the model behavior of an intermediate scale. Table 4 shows the number of serial steps done for the two- and three-level methods depending on the iterations. Especially, we see that we do fewer serial steps when the three-level method is applied.

TABLE 3
Errors of the first component at time $T_{\max} = 50$.

Iteration	$\eta_1 = 0.2$	$\eta_1 = 0.75$	$\eta_1 = 2$	Two-level method
1	5.96698264213838e-04	7.08281803419343e-04	7.175950573062584e-04	7.175766438230063e-04
2	1.4503391511486857e-04	4.627682726392884e-06	7.244961778282016e-06	7.2234487876014775e-06
3	1.6846111177755765e-04	3.6002717718981725e-05	2.4794370887603473e-05	2.4824342166950703e-05
4	1.5175374381607917e-04	1.977436923063236e-05	8.614816726031094e-06	8.641258080730602e-06
5	1.5190431144367772e-04	1.980929225765815e-05	8.639280904016583e-06	8.665761458456767e-06

TABLE 4
Computational cost. Number of serial steps for a two- and a three-level method, computed with (4.1) and (4.2). For the three-level method the number of iterations on level 1 is $k_1 = 2$. Varying number of iterations on the coarsest level for both the two- and three-level methods.

Iteration	Two-level method	Three-level method
1	120	70
2	230	130
3	340	190
4	450	250
5	560	310

5.4. One-dimensional rotating shallow water equations. Here solutions to the RSWE

$$\frac{\partial v_1}{\partial t} + -\frac{1}{\epsilon_1} v_2 + \frac{1}{\epsilon_2}\frac{\partial h}{\partial x} + v_1\frac{\partial v_1}{\partial x} = \mu \partial_x^4 v_1, \tag{5.7}$$

$$\frac{\partial v_2}{\partial t} + \frac{1}{\epsilon_1} v_1 + v_1\frac{\partial v_2}{\partial x} = \mu \partial_x^4 v_2, \tag{5.8}$$

$$\frac{\partial h}{\partial t} + \frac{1}{\epsilon_2}\frac{\partial v_1}{\partial x} + \frac{\partial}{\partial x}(h v_1) = \mu \partial_x^4 h \tag{5.9}$$

are computed. This is the same example investigated in [12] where convergence of the two-level method with averaging is shown. A similar notation is used here. With $h(x,t)$ we denote the surface height and $v_1(x,t), v_2(x,t)$ are the horizontal velocity. Moreover, a hyperviscosity term with diffusion coefficient $\mu = 10^{-4}$ is used. Additionally, we impose periodic boundary conditions and use the initial data

$$\begin{aligned} &v1 = 0, \qquad v2 = 0, \\ &h = c_1\left(e^{(-4(x-\pi/4)^2)}\sin(3(x-\pi/2)) + e^{(-2(x-\pi)^2)}\sin(8(x-\pi))\right) + c_2, \end{aligned} \tag{5.10}$$

where the constants c_1 and c_2 are chosen such that

$$\int_0^{2\pi} h(x,0)dx = 0, \qquad \max_x |h(x,0)| = 1. \tag{5.11}$$

To solve the RSWE numerically a pseudospectral method with 128 spatial Fourier modes is applied. For the time-stepping we use a three-level Parareal scheme with averaging and a second order Strang splitting method as a basic integrator. The numerical time-stepping method is tested for two different parameter regimes. The Rossby number, ϵ_1, is chosen as 0.1 in both cases. The Froude number, ϵ_2, is given by the relation $F^{1/2}\epsilon_1$, where F is the Burger number. In the first test, an important limiting case in geophysical fluid dynamics, the Burger number F is chosen to be 1, thus $\epsilon_1 = \epsilon_2 = 1/10$. This test is done with coarsening factors $N = 10, 20, 40$ and an interval length 48. In the second test the Burger number F is chosen to be $1/100$, thus there are two linear intrinsic frequencies with $\epsilon_1 = 1/10$ and $\epsilon_2 = 1/100$. The coarsening factors are $N = 10, 20, 30$ and the interval length is 45. For both tests, the time-step on the finest level is $1/2000$. The time-steps on levels 1 and 2 are determined by the fine time-step and the coarsening factor. The averaging windows on levels 1 and 2 are chosen to be equal to the time-steps used on levels 1 and 2. The initial guess is computed on the entire length of the solution interval before the first correction iteration. In Figures 5 and 6 the errors in the relative L^∞ norm, the same norm used for the errors in [12], are shown. We can see that the numerical method converges with an increasing number of iterations. For the case $F = 1/100$ fewer iterations are needed for convergence compared to the case $F = 1$.

Suppose we are given a two-level method to solve the case with $F = 1$. The two-level method uses a coarsening factor $N = 40$ and two correction iterations are done. The solution interval in time is [0,48]. Then we want to answer the question if it is possible to design a three-level method which is more efficient than the two-level method. The two-level method gives an error of $1.80e-05$. The computational cost of the two-level method is computed by applying (4.1) and is given by $C^2 = 7280$ serial steps. We compare it to a three-level method with a coarsening factor of $N = 20$. With k_2 we denote the number of iterations on the coarse level and we do $k_1 = 3$ iterations on the intermediate level. Then the three-level method converges to a

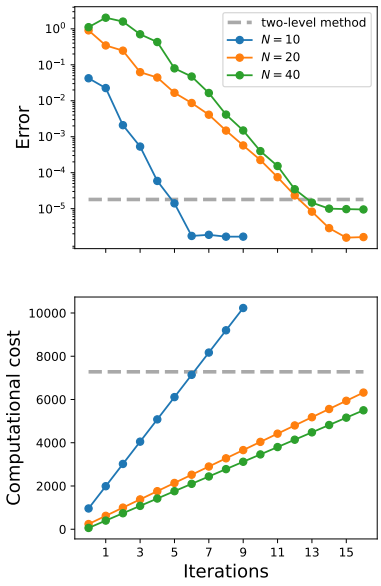

FIG. 5. *Error and computational cost of the numerical solution to the RSWE with a three-level Parareal method with averaging, parameter choice:* $\epsilon = 0.1, F = 1$. *The dashed gray lines show the error and computational cost of a two-level method with a fixed number of iterations.*

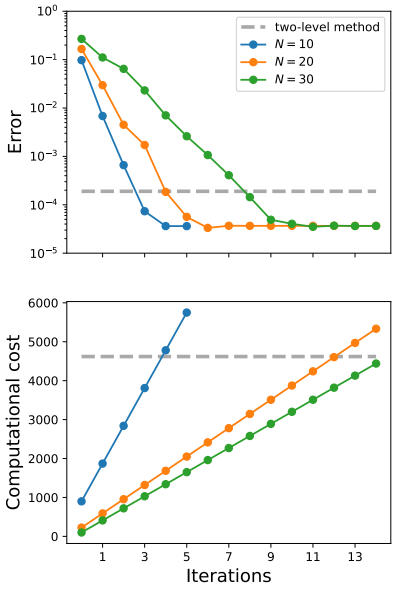

FIG. 6. *Error and computational cost of the numerical solution to the RSWE with a three-level Parareal method with averaging, parameter choice:* $\epsilon = 0.1, F = 1/100$. *The dashed gray lines show the error and computational cost of a two-level method with a fixed number of iterations.*

two-level method with three iterations, which guarantees that the three-level method converges to a two-level method, which is more exact than the two-level method with two iterations. The computational cost can be computed with (4.2) and is given by $C^3 = 240(k_2 + 1) + 140k_2$, where k_2 denotes the number of iterations on level 2. When $k < 28$, we do fewer serial steps with the three-level method than with two-level method with two iterations. With the three-level method we reach the accuracy of the two-level method with two iterations after 13 iterations already. Therefore, we can find a three-level method which is more efficient than the two-level method with two iterations; for a comparison see Figure 5.

Now we consider the case $F = 1/100$. We choose a two-level method with a coarsening factor of $N = 60$ and do two correction iterations. Thus, according to (4.1) $C^2 = 4620$ serial steps must be done. We compare this method to the three-level method with a coarsening factor of $N = 30$ and $k_1 = 3$ iterations on level 1. Thus, depending on the number of iterations k_2 we do $C^3 = 310k_2 + 100$ serial steps with the three-level method. The computational cost C^3 is computed with (4.2). If we do no more that 14 iterations with the three-level method, the computational cost of the three-level method is less than the computational cost of the two-level method. The three-level method is more accurate than the two-level method after eight iterations already, as the two-level method gives an error of $1.90e-4$; see Figure 6.

6. Discussion and conclusion. In the present study, a multilevel Parareal method with and without averaging is proposed and investigated, with a special focus on oscillatory problems. The averaging technique and transformation play a central role when oscillatory problems are solved because they mitigate the oscillatory

stiffness. Therefore, big time-steps can be used on the coarse levels, which is demonstrated in the numerical examples.

One important result is the derivation of error bounds for the methods building on results in the literature and underpinning theoretically the convergence of the method. In particular, the basic time-stepping methods applied on the levels cause an error, denoted as $E_{\bar{l}}$ and δ_0. The errors are amplified as a solution from one level is passed to coarser grids. In the error bound the amplification is given by the amplification factors A_j. Increasing the number of iterations on a level $\bar{l}$ changes $E_{\bar{l}}$. Additionally, the computational cost of the multilevel Parareal method is discussed.

Finally, several numerical examples are studied. The investigation of the numerical examples includes a discussion when multiple levels can be more efficient than the Parareal method with only two levels. If the algorithms without averaging and mapping are used, the multilevel approach can lead to algorithms which are not beneficial in terms of efficiency; see the example in subsection 5.1. However, combined with averaging and the transformation, methods with multiple levels are promising for oscillatory problems, as they allow us to take big time-steps on the coarse levels and we do not have to satisfy severe stability constraints imposed by fast oscillations; see the other examples. The example in subsection 5.3 also shows that a good choice of the averaging windows is crucial for the accuracy. Moreover, the computations in subsection 5.4 show that multilevel methods can be more efficient than a given two-level method with a fixed number of iterations.

For the subsections 5.3 and 5.4 different strategies for increasing the number of levels and thus the parallelism were examined. In subsection 5.3 a three-level method was compared to a two-level method. The coarsest and finest time-steps are the same for the two- and three-level methods, but for the three-level method an intermediate level was introduced. In subsection 5.4 the time-steps on the finest level and the second finest level are comparable for the two- and three-level methods, but for the three-level method an even coarser level was introduced as the coarsest level. The investigations concerning the choice of the time-steps on the levels are not exhaustive and can be a topic for future studies.

The examinations on the example in subsection 5.3 also inspire an interesting variant of the multilevel Parareal method. When introducing an intermediate level, which bridges the scales for the numerical computations and does not represent the model behavior on an intermediate scale, we might use a constant function as an initial guess on the grids of level 1. In particular, the constant functions on level 1 would admit the initial values provided by level 2. Possibly, this would not increase the error significantly, because the level 1 solutions only have to capture the coarse dynamics which is already resolved by the solution from level 2. Additionally, we would save the serial steps needed to compute the initial guess on level 1. The investigations concerning computational cost and efficiency shall be continued in the future on more complex examples including many scales or a continuous range of scales.

Appendix A. Proof of the theorem.

A.1. Theorem 3.3.

Proof. The proof has two building blocks. First, we build upon the two-level proof in [18], but additionally assume that the fine propagator is not exact. Second, an inductive argument is applied to obtain a multilevel result.

One Parareal correction at time T_n is given by

$$\begin{aligned} u(T_n) - U_n^{k+1} = {} & E(u(T_{n-1})) - \bar{G}_1(u(T_{n-1})) \; + \bar{G}_1(U_{n-1}^k) - E(U_{n-1}^k) \\ & + \bar{G}_1(u(T_{n-1})) - \bar{G}_1(U_{n-1}^{k+1}) + E(U_{n-1}^k) - P^0(U_{n-1}^k). \end{aligned} \tag{A.1}$$

The estimate for the last line is given through the accuracy estimate of the fine propagator

$$\|E(U_{n-1}^k) - P^0(U_{n-1}^k)\| \le c\Delta T_1 \Delta T_0^{p_0}. \tag{A.2}$$

The first two lines can be rewritten as

$$\begin{aligned} & E(u(T_{n-1})) - \bar{G}_1(u(T_{n-1})) + \bar{G}_1(U_{n-1}^k) - E(U_{n-1}^k) \\ & \quad = E(u(T_{n-1})) - \bar{E}^1(u(T_{n-1})) + \bar{E}^1(u(T_{n-1})) - \bar{G}_1(u(T_{n-1})) + \cdots \\ & \quad \quad + \bar{G}_1(U_{n-1}^k) - \bar{E}^1(U_{n-1}^k) + \bar{E}^1(U_{n-1}^k) - E(U_{n-1}^k) \\ & \quad = \mathcal{M}_{1,1}(u(T_{n-1}), \epsilon, \eta_1) + \mathcal{M}_{0,1}(u(T_{n-1}), \epsilon, \eta_1, \Delta T) \dots \\ & \quad \quad - \mathcal{M}_{0,1}(U_{n-1}^k, \epsilon, \eta_1, \Delta T) - \mathcal{M}_{1,1}(U_{n-1}^k, \epsilon, \eta_1). \end{aligned} \tag{A.3}$$

The function $\mathcal{M}_{0,l}(v, \epsilon, \eta_l, \Delta T)$, where the index l refers to the level, has a representation of the form

$$\mathcal{M}_{0,l}(v, \epsilon, \eta_l, \Delta T_l) \le C\Delta T^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_l, \omega) \right\} \tilde{\mathcal{M}}_0(v). \tag{A.4}$$

Moreover, $\mathcal{M}_{0,l}(v, \epsilon, \eta_l, \Delta T)$ is Lipschitz continuous in the first component.

In addition, for the function $\mathcal{M}_{1,l}(v, \epsilon, \eta)$ a relation of the form

$$\mathcal{M}_{1,l}(v, \epsilon, \eta_l) \le C\eta_l \epsilon \tilde{\mathcal{M}}_1(v) \tag{A.5}$$

can be found where $\mathcal{M}_1(v, \epsilon, \eta)$ is Lipschitz continuous in v. The aforementioned results about $\mathcal{M}_{0,l}$ and $\mathcal{M}_{1,l}$ can be found in [18]. We take $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_0$ as the maximum of $\tilde{\mathcal{M}}_{1,l}$ and $\tilde{\mathcal{M}}_{0,l}$ over all levels l.

We use the subsequent relations which are given in [18],

$$\|\bar{G}_l(u(T_{n-1})) - \bar{G}_l(U_{n-1}^{k+1})\| \le (1 + C\Delta T_l)\|u(T_{n-1}) - U_{n-1}^{k+1}\|, \tag{A.6}$$

$$\|\mathcal{M}_{1,l}(u(T_{n-1}), \epsilon, \eta_l) - \mathcal{M}_{1,l}(U_{n-1}^k, \epsilon, \eta_l)\| \le C\eta_l \epsilon \|u(T_{n-1}) - U_{n-1}^k\|, \tag{A.7}$$

$$\begin{aligned} & \|\mathcal{M}_{0,l}(u(T_{n-1}), \epsilon, \eta_l, \Delta T) - \mathcal{M}_{0,l}(U_{n-1}^k, \epsilon, \eta_l, \Delta T_l)\| \\ & \quad \le C\Delta T_l^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_l, \omega) \right\} \|u(T_{n-1}) - U_{n-1}^k\|, \end{aligned} \tag{A.8}$$

where U_n^k denote Parareal solutions of level l.

Thus, the following estimate can be obtained for the two-level case:

$$\begin{aligned} \|u(T_n) - U_n^{k+1}\| \le {} & C\eta_1 \epsilon \|u(T_{n-1}) - U_{n-1}^k\| \\ & + C\Delta T_1^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_1, \omega) \right\} \|u(T_{n-1}) - U_{n-1}^k\| \\ & + (1 + C\Delta T_1)\|u(T_{n-1}) - U_{n-1}^{k+1}\| + c\Delta T_1 \Delta T_0^{p_0} \\ = {} & \left(C\eta_1 \epsilon + C\Delta T_1^{p+1} \max_{\omega_0 \le \omega} \left\{ \left| \frac{\epsilon}{\omega} \right| \kappa(\epsilon, \eta_1, \omega) \right\} \right) \|u(T_{n-1}) - U_{n-1}^k\| \\ & + (1 + C\Delta T_1)\|u(T_{n-1}) - U_{n-1}^{k+1}\| + c\Delta T_1 \Delta T_0^{p_0} \\ = {} & \bar{\alpha}_1 e_{n-1}^k + \beta_1 e_{n-1}^{k+1} + \delta_0; \end{aligned} \tag{A.9}$$

this gives us a recurrence relation, where k counts the Parareal iterations and n counts the steps through the time grid.

For $k=0$ we have

(A.10)
$$\begin{aligned}\|u(T_n)-U_n^0\| \le e_n^0 &= \|E(u(T_{n-1}))-G^1(U_{n-1}^0)\| \\ &\le \|E(u(T_{n-1}))-G^1(u(T_{n-1}))\| + \|G^1(u(T_{n-1}))-G^1(U_{n-1}^0)\| \\ &\le \|E(u(T_{n-1}))-\bar{E}^1(u(T_{n-1}))\| + \|\bar{E}^1(u(T_{n-1}))-G^1(u(T_{n-1}))\| \\ &\quad + \|G^1(u(T_{n-1}))-G^1(U_{n-1}^0)\| \\ &\le C\eta_1\epsilon\|\tilde{\mathcal{M}}_1\| + C\Delta T_1^{p+1}\max_{\omega_0\le\omega}\left\{\left|\frac{\epsilon}{\omega}\right|\kappa(\epsilon,\eta_1,\omega)\right\}\|\tilde{\mathcal{M}}_0\| \\ &\quad + (1+C\Delta T_1)\|u(T_{n-1})-U_{n-1}^0\| = \bar{\gamma}_1+\beta_1 e_{n-1}^0\ .\end{aligned}$$

We have used the following notation further above:

$$\bar{\alpha}_l = C\eta_l\epsilon + C\Delta T_l^{p+1}\max_{\omega_0\le\omega}\left\{\left|\frac{\epsilon}{\omega}\right|\kappa(\epsilon,\eta_l,\omega)\right\}, \tag{A.11}$$

$$\beta_l = 1+C\Delta T_l, \qquad \delta_0 = c\Delta T_1\Delta t^{p_0}, \tag{A.12}$$

$$\bar{\gamma}_l = C\eta_l\epsilon\|\tilde{\mathcal{M}}_1\| + C\Delta T_l^{p+1}\max_{\omega_0\le\omega}\left\{\left|\frac{\epsilon}{\omega}\right|\kappa(\epsilon,\eta_l,\omega)\right\}\|\tilde{\mathcal{M}}_0\|. \tag{A.13}$$

Thus, we find the recurrence relation from Lemma B.1. For a two-level result with nonexact fine propagator Lemma B.1 can be applied. The inductive step for the multilevel result can be found in Lemma B.4. ∎

Appendix B. Useful results for the convergence proofs.

LEMMA B.1. *Suppose we are given the following recurrence relation:*

$$e_n^{k+1} = \alpha e_{n-1}^k + \beta e_{n-1}^{k+1} + \delta_n, \qquad e_n^0 = \gamma + \beta e_{n-1}^0, \tag{B.1}$$

with $\delta_0 = 0$ *and* $\delta_n = \delta$ *for* $n \ge 1$. *Then the* e_n^k *can be written in nonrecursive form as*

$$e_n^k \le \binom{n}{k+1}\gamma\alpha^k\beta^{n-k-1} + n\delta\beta^{n-1}(1+\alpha_{0n})^{n-1} \qquad \text{for } n>k. \tag{B.2}$$

Proof. The proof uses generating functions $\rho_k = \sum_{n\ge1} e_n^k\zeta^n$ and arguments very similar to the ones given in [9]. In particular, we find

$$\rho_0 = \frac{1}{1-\beta\zeta}\frac{\gamma\zeta}{1-\zeta} \qquad \text{and} \qquad \rho_{k+1} = \alpha\zeta\rho_k + \beta\zeta\rho_{k+1} + \delta\frac{\zeta}{1-\zeta}. \tag{B.3}$$

The rest follows by induction and the results in Lemmas B.2 and B.3. ∎

LEMMA B.2. *Suppose* $\beta \ge 1$ *and* $0<\zeta<1/\beta$. *Then the following inequality holds:*

$$\gamma\alpha^k\frac{\zeta^{k+1}}{1-\zeta}\frac{1}{(1-\beta\zeta)^{k+1}} \le \sum_{j\ge k+1}\binom{j}{k+1}\gamma\alpha^k\beta^{j-k-1}\zeta^j. \tag{B.4}$$

Proof. Assuming that $0<\zeta<1/\beta$ and $\beta\ge1$, we have $1/(1-\zeta) < 1/(1-\beta\zeta)$. Applying the binomial theorem, we can derive the following relation:

$$\gamma\alpha^k\frac{\zeta^{k+1}}{(1-\beta\zeta)^{k+2}} = \sum_{j\ge0}\binom{k+1+j}{j}\gamma\alpha^k\beta^j\zeta^{k+1+j} = \sum_{j\ge k+1}\binom{j}{k+1}\gamma\alpha^k\beta^{j-k-1}\zeta^j. \tag{B.5}$$

∎

LEMMA B.3. *Suppose we are given $\beta > 1$. Choose ζ such that $\beta\zeta < 1$ and $\frac{\alpha\zeta}{1-\beta\zeta} < 1$. Then, we can find the following estimate:*

$$\frac{\delta\zeta}{1-\zeta}\sum_{r=0}^{k-1}\frac{(\alpha\zeta)^r}{(1-\beta\zeta)^{r+1}} \le \sum_{n=1}^{\infty} n\delta\beta^{n-1}(1+\alpha_{0n})^{n-1}\zeta^n, \text{ where } \alpha_{0n}\in[0,\alpha]. \tag{B.6}$$

Proof. Newton's generalized binomial theorem is applied and we find

$$\frac{\delta\zeta}{1-\zeta}\sum_{r=0}^{k-1}\frac{(\alpha\zeta)^r}{(1-\beta\zeta)^{r+1}} \le \sum_{r=0}^{k-1}\frac{\delta\alpha^r\zeta^{r+1}}{(1-\beta\zeta)^{r+2}} = \sum_{r=0}^{k-1}\delta\alpha^r\zeta^{r+1}\sum_{j\ge 0}\binom{r+1+j}{j}\beta^j\zeta^j \tag{B.7}$$

$$= \sum_{r=0}^{k-1}\sum_{j\ge r+1}\binom{j}{r+1}\delta\alpha^r\beta^{j-r-1}\zeta^j. \tag{B.8}$$

For the summands we define

$$S_r = \alpha^r\delta\sum_{j\ge r+1}\binom{j}{r+1}\beta^{j-r-1}\zeta^j. \tag{B.9}$$

Ordering relation (B.8) with respect to orders of ζ we find

$$\begin{aligned} O(\zeta^n):\ &\delta\sum_{m=1}^{\min(n,k)}\binom{n}{m}\beta^{n-m}\alpha^{m-1} \le \delta\beta^{n-1}\alpha^{-1}\sum_{m=1}^{\min(n,k)}\binom{n}{m}\alpha^m \\ &\le \delta\beta^{n-1}\alpha^{-1}\left((1+\alpha)^n-1\right) \le \delta\beta^{n-1}\alpha^{-1}\alpha n(1+\alpha_{0n})^{n-1} = n\delta\beta^{n-1}(1+\alpha_{0n})^{n-1}, \end{aligned}$$

where $1+\alpha_{0n}\in[1,1+\alpha]$. The value of α_{0n} can be different for different n. The third inequality follows from the intermediate value theorem. Thus, we have

$$\frac{\delta\zeta}{1-\zeta}\sum_{r=0}^{k-1}\left(\frac{(\alpha\zeta)^r}{(1-\beta\zeta)^{r+1}} \le \sum_{r=0}^{k-1}S_r \le \sum_{n=1}^{\infty} n\delta\beta^{n-1}(1+\alpha_{0n})^{n-1}\zeta^n\right). \tag{B.10}$$

Remark. In order to reorder the terms in the last inequality we need unconditional convergence of the sum over the S_r. Unconditional convergence is equivalent to absolute convergence in $\mathbb{R}^n$. As a finite sum over absolutely convergent series, the sum over the S_r is absolutely convergent too. ∎

For the remaining results we use the subsequent notational conventions:

1. To avoid cumbersome notation the constants are not indexed with the level l. Instead we assume that $C_i = \max_l\{C_{i,l}\}$, where l counts the levels.
2. The propagators on the levels $1,\dots,L-1$ have at least an accuracy order of p, i.e., $p = \min_{l=1,\dots,L-1}\{p_l\}$.

In Lemmas B.5, B.6, B.7, and B.8, we assume that we have a coarsening factor N which relates the levels. Then the time-steps on the different levels obey the following relations:

$$\Delta T_l = N^l\Delta T_0, \tag{B.11}$$

$$\Delta T_l = N^{l-(L-1)}\Delta T_{L-1}, \tag{B.12}$$

where ΔT_0 is the time-step on the finest level, level 0, and ΔT_{L-1} is the time-step on the coarsest level, level $L-1$.

LEMMA B.4. *We consider the recursively formulated inequality*

$$e_{n,l}^{k_l} \le E_l + A_l \delta_{l-1}$$

with $\delta_{l-1} = e_{N,l-1}^{k_{l-1}}$ *and* $e_{n,l}^{k_l} \le e_{N,l}^{k_l}$ *for all* l, $n \le N$. *Then we can show the inequality*

$$e_{n,l}^{k_l} \le \sum_{\bar{l}=1}^{l} E_{\bar{l}} \prod_{j=\bar{l}+1}^{l} A_j + \delta_0 \prod_{\bar{l}=1}^{l} A_{\bar{l}}, \tag{B.13}$$

which depends on δ_0 *and is independent of* δ_{l-1} *(if* $l-1 \ge 1$*).*

Proof. For the proof induction can be applied. ∎

LEMMA B.5. *Suppose* δ_0 *and* A_l *satisfy the following relations:*

$$\delta_0 = c \frac{\Delta T_{L-1}^{p_0+1}}{N^{(L-1)(p_0+1)-1}}, \quad A_l \le N \exp\left(C_2 \frac{T}{N^{L-1-l}} + C_1 T \frac{\Delta T_{L-1}^{p}}{N^{(L-1-l)(p+1)}} \right) \tag{B.14}$$

(see Lemma B.8 *for* A_l*). Then, we can find the following bound:*

$$\delta_0 \prod_{\bar{l}=1}^{L-1} A_{\bar{l}} \le cT \Delta T_0^{p_0} \exp\left(C_2 T \frac{1-1/N^L}{1-1/N} + C_1 T \Delta T_{L-1}^{p} \frac{1-1/N^{L(p+1)}}{1-1/N^{p+1}} \right). \tag{B.15}$$

Proof. We use the estimate for A_l from Lemma B.8. In addition, we apply the finite geometric series and the relation $N \Delta T_{L-1} = T$.

$$\begin{aligned} \delta_0 \prod_{\bar{l}=1}^{L-1} A_{\bar{l}} &\le c \frac{\Delta T_{L-1}^{p_0+1}}{N^{(L-1)(p_0+1)-1}} \prod_{l=1}^{L-1} N \exp\left(\frac{C_2 T}{N^{L-1-l}} + \frac{C_1 T \Delta T_{L-1}^{p}}{N^{(p+1)(L-1-l)}} \right) \\ &= cT \left(\frac{\Delta T_{L-1}}{N^{L-1}} \right)^{p_0} \exp\left(C_2 T \frac{1-1/N^L}{1-1/N} + C_1 T \Delta T_{L-1}^{p} \frac{1-1/N^{L(p+1)}}{1-1/N^{p+1}} \right). \end{aligned}$$ ∎

LEMMA B.6. *Suppose* E_l *and* A_l *can be bounded by the relations* (B.26) *and* (B.27)*. Then the following estimate can be shown:*

$$\begin{aligned} \sum_{l=1}^{L-1} E_l \prod_{j=l+1}^{L-1} A_j \le \exp&\left(\frac{C_2 T}{1-1/N} + \frac{C_1 T \Delta T_{L-1}}{1-1/N^{p+1}} \right) \\ &\times \max_l \left(\binom{N}{k_l+1} C_3 C_1^{k_l} \right) \frac{\Delta T_{L-1}^{kp+k+p}}{1-(1/N)^{kp+k+p}}. \end{aligned} \tag{B.16}$$

Proof. We use the result from Lemma B.7.

$$\begin{aligned} \sum_{l=1}^{L-1} E_l \prod_{j=l+1}^{L-1} A_j \le \sum_{l=1}^{L-1} &\binom{N}{k_l+1} C_3 C_1^{k_l} \Delta T_l^{k_l p + k_l + p} \Delta T_{L-1} \\ &\exp\left(\frac{C_2 T}{1-1/N} + C_1 \frac{T \Delta T_{L-1}^{p}}{1-1/N^{p+1}} \right) \end{aligned} \tag{B.17}$$

$$\leq \exp\left(\frac{C_2 T}{1-1/N} + C_1 \frac{T\Delta T_{L-1}^p}{1-1/N^{p+1}}\right) \Delta T_{L-1} \tag{B.18}$$
$$\times \max_l \left(\binom{N}{k_l+1} C_3 C_1^{k_l}\right) \sum_{l=1}^{L-1} \Delta T_l^{k_l p + k_l + p}$$
$$\leq \exp\left(\frac{C_2 T}{1-1/N} + C_1 \frac{T\Delta T_{L-1}^p}{1-1/N^{p+1}}\right) \tag{B.19}$$
$$\times \max_l \left(\binom{N}{k_l+1} C_3 C_1^{k_l}\right) \frac{\Delta T_{L-1}^{kp+k+p}}{1-(1/N)^{kp+k+p}},$$

where we used $k = \min_{l=1,\ldots,L-1}\{k_l\}$ and the geometric series in the last inequality. ∎

LEMMA B.7. *Let E_l and A_l be bounded by the relations* (B.26) *and* (B.27). *For the expression $E_l \prod_{j=l+1}^{L-1} A_j$ we find the following estimate:*

(B.20)
$$E_l \prod_{j=l+1}^{L-1} A_j \leq \binom{N}{k_l+1} C_1^{k_l} C_3 (\Delta T_l)^{(k_l p + k_l + p)}$$
$$\times \Delta T_{L-1} \exp\left(C_2 T \frac{1-1/N^{(L-1-l)}}{1-1/N} + C_1 T \Delta T_{L-1}^p \frac{1-1/N^{(p+1)(L-1-l)}}{1-1/N^{(p+1)}}\right).$$

Proof. The bounds (B.26) and (B.27) from Lemma B.8 are used:

(B.21)
$$E_l \prod_{j=l+1}^{L-1} A_j = \binom{N}{k_l+1} C_1^{k_l} C_3 (\Delta T_l)^{(k_l+1)(p+1)} \exp\left(C_2 \frac{T}{N^{L-1-l}}\right)$$
$$\times N^{L-1-l} \exp\Big(C_2 T (1 + 1/N + \ldots 1/N^{L-1-l-1}) \tag{B.22}$$
$$+ C_1 T \Delta T_{L-1}^p (1 + 1/N^{p+1} + \ldots 1/N^{(p+1)(L-1-l-1)}) \Big) \tag{B.23}$$
$$\leq \binom{N}{k_l+1} C_1^{k_l} C_3 (\Delta T_l)^{(k_l p + k_l + p)} \frac{\Delta T_{L-1}}{N^{L-1-l}} N^{L-1-l} \tag{B.24}$$
$$\times \exp\left(C_2 T \frac{1-1/N^{(L-1-l)}}{1-1/N} + C_1 T \Delta T_{L-1}^p \frac{1-1/N^{(p+1)(L-1-l)}}{1-1/N^{(p+1)}}\right). \tag{B.25}$$ ∎

LEMMA B.8. *Using the relations* (3.4) *and* (3.6) *to define E_l and A_l we can show the following bounds:*

$$E_l \leq \binom{N}{k_l+1} C_1^{k_l} C_3 (\Delta T_l)^{(k+1)(p+1)} \exp\left(C_2 \frac{T}{N^{L-1-l}}\right), \tag{B.26}$$
$$A_l \leq N \exp\left(C_2 \frac{T}{N^{L-1-l}} + C_1 T \frac{\Delta T_{L-1}^p}{N^{(L-1-l)(p+1)}}\right), \tag{B.27}$$

where the propagator on level l for $l \geq 1$ has an accuracy order of at least p.

Proof. According to (3.4) and (3.6) we have

$$E_l = \binom{n}{k_l+1} \gamma_l \alpha_l^{k_l} \beta_l^{N-k_l-1}, \qquad A_l = N \beta_l^{N-1} (1+\alpha_{0N,l})^{N-1}. \tag{B.28}$$

Then the following relations hold:

$$\alpha_l = C_1 \Delta T_l^{p+1} = C_1 \left(\frac{\Delta T}{N^{L-1-l}} \right)^{p+1}, \qquad \beta_l = 1 + C_2 \Delta T_l = \left(1 + C_2 \frac{\Delta T_{L-1}}{N^{L-1-l}} \right),$$

$$\gamma_l = C_3 \Delta T_l^{p+1} = C_3 \left(\frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{p+1}, \qquad \delta_0 = c \Delta T_1 \Delta T_0^{p_0} = c \frac{\Delta T_{L-1}^{p_0+1}}{N^{(L-1)(p_0+1)-1}}.$$

Thus we get for E_l

$$E_l = \binom{N}{k_l+1} C_3 \left(\frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{p+1} \left(C_1 \left(\frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{p+1} \right)^{k_l} \left(1 + C_2 \frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{N-k_l-1}$$

$$\leq \binom{N}{k_l+1} C_1^{k_l} C_3 (\Delta T_l)^{(k+1)(p+1)} \exp\left(C_2 \frac{T}{N^{L-1-l}} \right),$$

where $T_0, T_1, \ldots, T_N$ denote the grid points on the coarsest level.

In addition, for A_l we find

$$A_l = N \beta_l^{N-1} (1 + \alpha_{0N,l})^{N-1} \leq N \beta_l^{N-1} (1 + \alpha_l)^{N-1} \tag{B.29}$$

$$= N \left(1 + C_2 \frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{N-1} \left(1 + C_1 \left(\frac{\Delta T_{L-1}}{N^{L-1-l}} \right)^{p+1} \right)^{N-1} \tag{B.30}$$

$$\leq N \exp\left(C_2 \frac{T}{N^{L-1-l}} + C_1 T \left(\frac{\Delta T_{L-1}^p}{N^{(L-1-l)(p+1)}} \right) \right). \tag{B.31}$$

In the last inequality we exploit that $N \Delta T_{L-1} = T$, where T is the length of the integration interval (on the coarsest level). ∎

Acknowledgments. The authors thank Rupert Klein and the reviewers, whose input improved the paper.

REFERENCES

[1] G. Bal, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in Domain Decomposition Methods in Science and Engineering, Springer, New York, 2005, pp. 425–432.

[2] H. De Sterck, S. Friedhoff, A. J. M. Howse, and S. P. MacLachlan, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, Numer. Linear Algebra Appl., 27 (2020), e2271.

[3] W. E and B. Engquist, *Multiscale modeling and computation*, Notices Amer. Math. Soc., 50 (2003), pp. 1062–1070.

[4] P. F. Embid and A. J. Majda, *Averaging over fast gravity waves for geophysical flows with arbitrary potential vorticity*, Comm. Partial Differential Equations, 21 (1996), pp. 619–658.

[5] B. Engquist and R. Tsai, *Heterogeneous multiscale methods for stiff ordinary differential equations*, Math. Comp., 74 (2005), pp. 1707–1742.

[6] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661.

[7] S. Friedhoff and B. S. Southworth, *On "optimal" h-independent convergence of parareal and multigrid-reduction-in-time using Runge-Kutta time integration*, Numer. Linear Algebra Appl., 28 (2021), e2301.

[8] M. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Comput. Vis. Sci., 19 (2018), pp. 59–74.

[9] M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, in Domain Decomposition Methods in Science and Engineering XVII, Springer, Berlin, 2008, pp. 45–56.

[10] M. J. Gander, T. Lunet, D. Ruprecht, and R. Speck, *A unified analysis framework for iterative parallel-in-time algorithms*, SIAM J. Sci. Comput., 45 (2023), pp. A2275–A2303.
[11] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578.
[12] T. Haut and B. Wingate, *An asymptotic parallel-in-time method for highly oscillatory PDEs*, SIAM J. Sci. Comput., 36 (2014), pp. A693–A713.
[13] A. Hessenthaler, B. S. Southworth, D. Nordsletten, O. Röhrle, R. D. Falgout, and J. B. Schroder, *Multilevel convergence analysis of multigrid-reduction-in-time*, SIAM J. Sci. Comput., 42 (2020), pp. A771–A796.
[14] D. D. Holm and P. Lynch, *Stepwise precession of the resonant swinging spring*, SIAM J. Appl. Dyn. Syst., 1 (2002), pp. 44–64.
[15] B. N. Lawrence, M. Rezny, R. Budich, P. Bauer, J. Behrens, M. Carter, W. Deconinck, R. Ford, C. Maynard, S. Mullerworth, C. Osuna, A. Porter, K. Serradell, S. Valcke, N. Wedi, and S. Wilson, *Crossing the chasm: How to develop weather and climate models for next generation computers?*, Geosci. Model Dev., 11 (2018), pp. 1799–1821.
[16] F. Legoll, T. Lelièvre, and G. Samaey, *A micro-macro parareal algorithm: Application to singularly perturbed ordinary differential equations*, SIAM J. Sci. Comput., 35 (2013), pp. A1951–A1986.
[17] J.-L. Lions, Y. Maday, and G. Turinici, *A "parareal" in time discretization of PDE's*, C. R. Acad. Sci. Ser. I Math., 332 (2001), pp. 661–668.
[18] A. G. Peddle, T. Haut, and B. Wingate, *Parareal convergence for oscillatory PDEs with finite time-scale separation*, SIAM J. Sci. Comput., 41 (2019), pp. A3476–A3497.
[19] D. Ruprecht, *Wave propagation characteristics of parareal*, Comput. Vis. Sci., 19 (2018), pp. 1–17.
[20] F. V. Sanders and J. Murdock, *Averaging Methods in Nonlinear Dynamical Systems*, 2nd ed., Springer, New York, 2007.
[21] S. Schochet, *Fast singular limits of hyperbolic PDEs*, J. Differential Equations, 114 (1994), pp. 476–512.
[22] B. Southworth, W. Mitchell, A. Hessenthaler, and F. Danieli, *Tight two-level convergence of linear parareal and MGRIT: Extensions and implications in practice*, in Parallel-in-Time Integration Methods, PinT 2020, Springer Proc. Math. Stat. 356, B. Ong, J. Schroder, J. Shipton, and S. Friedhoff, eds., Springer, Cham, 2021.
[23] B. S. Southworth, *Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 564–608.
[24] J. Steiner, D. Ruprecht, R. Speck, and R. Krause, *Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number*, in Numerical Mathematics and Advanced Applications, Lect. Notes Comput. Sci. Eng. 103, A. Abdulle, S. Deparis, D. Kressner, F. Nobile, and M. Picasso, eds., Springer, New York, 2015, pp. 195–202.
[25] H. D. Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan, *Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection*, Numer. Linear Algebra Appl., 28 (2021), e2367.