

A PARAREAL ALGORITHM WITH LOW-RANK COARSE SOLVERS*

MARTIN J. GANDER[†], MARIO OHLBERGER[‡], AND STEPHAN RAVE[‡]

Abstract. We consider a new class of Parareal algorithms, which use ideas from localized reduced basis methods to construct the coarse solver from truncated SVD approximations of the transfer operators mapping initial values for a given time interval to the solution at the end of the interval. By leveraging randomized singular value decompositions, these low-rank approximations are obtained embarrassingly parallel by computing local fine solutions for random initial values. We show a priori and a posteriori error bounds in terms of the computed singular values of the transfer operators. Our numerical experiments demonstrate that our approach can significantly outperform Parareal with single-step coarse solvers. At the same time, it permits to further increase parallelism in Parareal by trading global iterations for a larger number of independent local solves.

Key words. Time parallel time integration, Parareal, low-rank coarse solver, approximate transfer operators, reduced basis method.

MSC codes. 65M20, 65M55, 65Y05, 65Y20

1. Introduction. Solving time-dependent partial differential equations (PDEs) efficiently is a crucial challenge in many scientific and engineering applications. Traditional time integration methods can be computationally expensive, particularly for complex models and long simulation times. The Parareal algorithm [25] offers a compelling approach to parallelizing the solution of time-dependent PDEs. By combining coarse-grained and fine-grained parallelism, it provides significant speedups compared to traditional serial or spatially-parallel methods, particularly for problems with many time steps. The core principle of Parareal involves decomposing the time domain into a coarse temporal grid, solving with a coarse solver on that grid, and then iteratively refining the solution using local fine-scale computations that can be computed in parallel without communication. The efficiency and accuracy of Parareal are critically dependent on the performance of the coarse solver used to advance the solution on the coarse grid, see, e.g., [17, 12, 33, 23].

We present and analyze a new approach to improve the performance of Parareal by using localized-in-time reduced basis techniques for the definition of the coarse solver. More precisely, we consider as coarse solvers low-rank approximations of the transfer operators mapping arbitrary initial values for a given time interval to the corresponding solution at the end of the interval. Using randomized singular value decompositions (SVDs), we can efficiently construct approximations of these transfer operators in an embarrassingly parallel manner. Our approach is mainly targeted at linear (systems of) PDEs of parabolic type, for which the transfer operators exhibit a rapid singular value decay, which means that small truncation ranks will be sufficient to obtain a good approximation. This includes diffusion-advection-reaction equations with a sufficiently large diffusion term. Other applications might include time-domain simulations of large, asymptotically stable ODE systems. As the construction of the

*Submitted to the editors 08/12/2025.

Funding: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044 –390685587, Mathematics Münster: Dynamics–Geometry–Structure and the Swiss National Science Foundation. This work has been supported by the Ministry of Culture and Science NRW as part of the Lamarr Fellow Network.

[†]Université de Genève, Switzerland (martin.gander@unige.ch).

[‡]Mathematics Münster, Germany (mario.ohlberge@uni-muenster.de, stephan.rave@uni-muenster.de).

coarse solvers is independent of the inhomogeneous part of the PDE, the same coarse solvers can be reused for new source terms without relevant additional computational work (see Remark 2.2). Our method can be easily implemented on top of existing PDE solvers, as long as the solver allows evaluating the adjoint transfer operator for the given PDE.

We refer to [2, 22, 28] for a review on reduced basis methods and to [3] concerning localized reduced basis constructions. Concepts based on the approximation of transfer operators using randomized SVDs have been studied both with respect to localization in space [4, 30] and localization in time [31].

The Parareal algorithm typically employs a coarser time discretization as its coarse solver, leaving the spatial component unchanged from the fine-scale integrator. In that case, the convergence of the Parareal algorithm is well understood, showing superlinear rates with rapid convergence for parabolic problems, and convergence difficulties for hyperbolic problems [17, 12, 13, 1].

Our approach diverges from this standard practice by using coarse solvers that rely on low-dimensional, problem-adapted approximations in space, but have the same accuracy in time as the fine solver. A first Parareal variant using reduced basis models for the coarse solver was introduced in [21]. However, no practically feasible algorithm for constructing the reduced basis was provided. In [6], using ideas from the Krylov subspace Parareal approach [9, 10], a global reduced basis was constructed on the fly from the linear span of the fine solver values obtained during the iteration of the Parareal algorithm. Our approach differs from [6] as it computes near-best approximations of the fine solver for arbitrary initial values instead of only considering the convergence history. Further, in contrast to [6], our coarse solvers are constructed in an embarrassingly parallel manner before any Parareal iterations take place. Our construction is entirely local and does not require any communication between nodes associated with different Parareal time intervals.

Regarding further Parareal methods with coarse solvers that use reduced approximations in space, see [18], where reduced models obtained from asymptotic expansions for highly oscillatory differential equations were considered. For problems with matrix-valued solutions that can be approximated by low-rank matrices, a Parareal method was introduced in [5], which uses dynamical low-rank approximations of different ranks for the fine and coarse solvers. We also refer to our recent discussion of the Parareal algorithm without coarse solver, which includes a convergence analysis based on separation of variables and Fourier analysis [16].

This paper is organized as follows: In section 2 we define our new Parareal algorithm with low-rank coarse solvers and discuss the algorithmic aspects of the construction of the coarse solvers via randomized SVDs. We provide a complete convergence analysis for our algorithm in section 2, demonstrating superlinear and linear convergence rates depending on the singular value spectra of the transfer operators. We also provide an easily computable a posteriori error bound. In section 4, we finally illustrate our theoretical findings with a series of numerical experiments on one-, two-, and three-dimensional parabolic problems, demonstrating the practical efficiency and robustness of the proposed approach.

2. Definition of the Algorithm. Consider a partition $0 = t_0 < t_1 < \dots < t_N = T$ of the time interval $[0, T]$ and a Hilbert space V . Let $u_0 \in V$. For each $1 \leq n \leq N$, let an operator $F_n : V \rightarrow V$ be given. We assume that $F_n = F(\cdot, t_{n-1}, t_n)$ is a fine solver for a given linear system of differential equations with solution $u(t) \in V$, $t \in [0, T]$. In other words, we assume that $u_n \approx u(t_n)$, where the sequence $u_n \in V$ is

89 given by

$$90 \quad u_{n+1} := F_{n+1}u_n, \quad 0 \leq n < N.$$

91 We further assume the fine solver to be sufficiently good so that our only goal is
 92 to accurately approximate the sequence u_n . To that end, we define coarse solvers
 93 $G_n : V \rightarrow V$, $1 \leq n \leq N$, and consider Parareal algorithms of the form

$$94 \quad (2.1) \quad u_{n+1}^{k+1} := F_{n+1}u_n^k + G_{n+1}u_n^{k+1} - G_{n+1}u_n^k, \quad 0 \leq n < N, k \in \mathbb{N}_0,$$

95 where $u_0^k := u_0$ for all $k \in \mathbb{N}_0$ and

$$96 \quad u_{n+1}^0 := G_{n+1}u_n^0, \quad 0 \leq n < N.$$

97 Our goal is to find suitable coarse solvers G_n such that u_n^k quickly converges to u_n as
 98 the number of Parareal iterations k increases. At the same time, each G_n should be
 99 fast to evaluate, since for each Parareal iteration (2.1), only the fine solvers F_n can
 100 be evaluated in parallel, whereas the G_n have to be evaluated sequentially. A typical
 101 choice for G_n would be a time stepping scheme with a coarse time-step size.

102 In this work, we consider G_n which are obtained from a low-rank approximation
 103 of the corresponding fine solver F_n . To do so, we will always assume that each F_n is
 104 a continuous, affine linear operator. In particular, we can write each F_n as

$$105 \quad (2.2) \quad F_n v = F'_n v + b_n, \quad b_n := F_n 0,$$

106 where $F'_n : V \rightarrow V$ is a continuous linear operator. Finally, we assume that each
 107 F'_n is a compact operator, which is always the case in a discrete setting with finite-
 108 dimensional V . In particular, this means that F'_n has a singular value decomposition
 109 of the form

$$110 \quad (2.3) \quad F'_n v = \sum_{r=1}^{\text{rank } F'_n} \psi_{n,r} \cdot \sigma_{n,r} \cdot (\varphi_{n,r}, v)_V.$$

111 Here, $(\cdot, \cdot)_V$ denotes the inner product on V , $\sigma_{n,1} > \sigma_{n,2} > \dots > 0$ are the uniquely
 112 defined singular values of F'_n , $\varphi_{n,r} \in V$, $\|\varphi_{n,r}\| = 1$ are the corresponding right- and
 113 $\psi_{n,r} \in V$, $\|\psi_{n,r}\| = 1$ the corresponding left-singular vectors. We note that when V is
 114 finite dimensional, (2.3) is the matrix-free formulation of the generalized SVD of F'_n
 115 that takes the inner product on V into account.

116 We now define the coarse solver G_n by considering a truncated singular value
 117 decomposition of F'_n and adding the same affine part b_n :

$$118 \quad (2.4) \quad G_n v := \sum_{r=1}^{R_n} \psi_{n,r} \cdot \sigma_{n,r} \cdot (\varphi_{n,r}, v) + b_n,$$

119 where $0 \leq R_n \leq \text{rank } F'_n$ is the freely choosable truncation rank.

120 Note that

$$121 \quad (F_n - G_n)v = \sum_{r=R_n+1}^{\text{rank } F'_n} \sigma_{n,r} \cdot (\varphi_{n,r}, v) \cdot \psi_{n,r}$$

122 is linear with

$$123 \quad (2.5) \quad \|F_n - G_n\| = \sigma_{n,R_n+1}.$$

In particular, G_n can be made an arbitrarily good approximation of F_n by increasing R_n . We are mostly interested in the case of linear parabolic PDEs, where the analytical solution operators for the initial-value problems on each time interval can be expected to have rapidly decaying singular values. We expect F'_n to be a sufficiently good approximation of the corresponding analytical solution operator in the sense that it inherits its singular value decay (see subsection 4.3.1 for further discussion), so small values of R_n will be sufficient. Indeed, in [16] we have shown that for the heat equation with homogeneous Dirichlet conditions in one spatial dimension, the Parareal iteration (2.1) even converges for $G_n = 0$. We have further shown the exponential decay of the singular values of F'_n when F_n is chosen to be the exact solution of the heat equation (see also experiment 1 below). The argument can be easily extended to higher spatial dimensions. In the context of model order reduction, the maps F'_n are called transfer operators in time. In [31] the compactness of F'_n was shown for a large class of parabolic PDEs.

Remark 2.1. While assuming that the fine solvers F_n are affine restricts the class of possible time integrators, we note that this condition is satisfied by all Runge-Kutta schemes, as long as the underlying system of differential equations is linear. Further, even when the time discretization leads to non-affine F_n , these F_n still approximate the affine exact solution operator of the underlying system of differential equations, and we can interpret the randomized SVD algorithm discussed below as a numerical approximation of the truncated SVD of the exact solution operator. Thus, the assumption that the F_n are affine is mainly a simplification for the theoretical analysis. In particular, the error bounds in section 3 could be extended to also take the error between the fine solver and the exact solution operator into account.

Remark 2.2. Note that the Parareal iteration (2.1) can also be written as $u_{n+1}^{k+1} = F_{n+1}u_n^k + G'_{n+1}u_n^{k+1} - G'_{n+1}u_n^k$, where G'_n is the linear part of G_n . Thus, adding b_n in the definition of G_n only has an effect in the initialization phase where the u_n^0 are computed. We choose to include b_n in the definition of G_n as this will give us a much better initial approximation of the solution without any additional sequential computations.

2.1. Computation of G_n . Even when V is finite dimensional, computing a truncated SVD of F'_n using direct matrix-based algorithms is not an option: determining the matrix of F'_n would mean to locally solve the PDE for every degree of freedom of the solution space. Instead, we consider the following basic randomized SVD algorithm to compute G_n :

1. Draw $R_n + p$ random vectors $\omega_i \in V$ according to some appropriate probability distribution on V , and compute $F'_n \omega_i$.
If we draw enough random vectors, $W := \text{span}\{F'_n \omega_1, \dots, F'_n \omega_{R_n+p}\}$ will be a good approximation of the image of F'_n with high probability. In particular, we can approximate the SVD of F'_n by computing the SVD of $P_W F'_n$, where P_W is the V -orthogonal projection onto W .
2. Compute an orthonormal basis $w_1, \dots, w_{R_n+p}$ of W , e.g., by using the modified Gram-Schmidt process.
3. Compute $F_n'^* w_i$, $1 \leq i \leq R_n + p$, where $F_n'^* : V \rightarrow V$ is the adjoint of F'_n given by $(F_n'^* w, v)_V = (w, F'_n v)_V$ for all $v, w \in V$.
4. Compute an orthonormal basis $v_1, \dots, v_{R_n+p}$ for $\text{span}\{F_n'^* w_1, \dots, F_n'^* w_{R_n+p}\}$.

Then we have:

$$P_W F'_n v = \sum_{i=1}^{R_n+p} w_i \cdot (w_i, F'_n v)_V = \sum_{i,j=1}^{R_n+p} w_i \cdot (F_n'^* w_i, v_j)_V \cdot (v_j, v)_V.$$

5. Compute a singular value decomposition of the small-dimensional matrix $M \in R^{(R_n+p) \times (R_n+p)}$, $M_{i,j} := (F_n'^* w_i, v_j)_V$.

Since the orthonormal bases w_i and v_j span $\text{img } F'_n$ and $(\ker F'_n)^\perp$, it is easy to show that the singular values of M agree with the singular values of $P_W F'_n$. Further, if $\underline{\psi}_r, \underline{\varphi}_r \in \mathbb{R}^{R_n+p}$ are the r -th left and right-singular vectors of M , then

$$\psi_r := \sum_{i=1}^{R_n+p} \underline{\psi}_{r,i} \cdot v_i, \quad \varphi_r := \sum_{i=1}^{R_n+p} \underline{\varphi}_{r,i} \cdot w_i$$

are the r -th left and right-singular vectors of $P_W F'_n$.

6. Return the first R_n singular values and vectors $\sigma_r, \varphi_r, \psi_r$, $1 \leq r \leq R_n$.

The p additionally drawn random vectors are called oversampling vectors, and p has to be chosen large enough to ensure that the first R_n singular values and vectors of $P_W F'_n$ are close to the first R_n singular values and vectors of F'_n . In the case of $V = \mathbb{R}^m$, extensive analysis is available that permits controlling the approximation error with arbitrarily high probability [20]. In particular, adaptive algorithms are available that choose p a posteriori, based on a prescribed (typically small) failure probability. An analysis for general finite-dimensional Hilbert spaces can be found in [4].

Randomized SVD algorithms perform well when the singular values decay rapidly. For operators with slow singular value decay, algorithms with power iterations are considered where $W := \text{span}\{(F'_n F_n'^*)^q F'_n \omega_i \mid 1 \leq i \leq R_n + p\}$ with $q \geq 1$. For parabolic problems, the singular value decay is generally fast enough such that power iterations are not necessary. In fact, for the numerical examples considered in section 4, already a single oversampling vector ($p = 1$) turns out to be sufficient.

2.2. Computational costs. Compared to classical coarse solvers, using low-rank coarse solvers involves an additional setup phase where the truncated singular value decompositions of the F'_n as well as the b_n have to be computed. To compute a G'_n of rank R_n , R_n evaluations of F_n are required, followed by R_n evaluations of $F_n'^*$. The costs for computing the randomized SVD from these evaluations are negligible compared to evaluating the fine solvers. Additionally, a single evaluation of F_n is required to obtain b_n . Overall, after k Parareal iterations, which each require a further fine solver evaluation,

$$(2.6) \quad K_{F,n} := 2(R_n + p) + k + 1$$

fine solver evaluations per time interval are required, compared to only k evaluations for Parareal with classical coarse solvers. However, since our coarse solvers lead to faster converging methods, a smaller number k of sequential Parareal iterations is required, whereas all $2(R_n + p) + 1$ fine solver evaluations for computing G'_n can be carried out in parallel for all time intervals $[t_{n-1}, t_n]$ without communication. Furthermore, the individual evaluations of F_n , F'_n and $F_n'^*$ can be carried out in parallel as well. Hence, choosing R_n allows us to freely balance the amount of parallel work with the number of required communication events. Moreover, since the same initial

value problem is solved locally for $R_n + p$ initial values at the same time, our approach can benefit immensely from using direct linear solvers or SIMD vectorization (see discussion in subsection 4.4). Parallelism can be further increased at the expense of numerical accuracy by evaluating F_n^* on an additional set of random vectors instead of evaluating it on $w_1, \dots, w_{R_n+p}$ (e.g., [26, section 15]). Finally, we will see in subsection 4.4 that, in some cases, the convergence of our method can be so much faster that $K_{F,n}$ can actually be smaller than the number of Parareal iterations required using classical coarse solvers.

After the setup phase, an evaluation of G_n only requires the computation of R_n inner products and the linear combination of $R_n + 1$ vectors. Thus, evaluating G_n is typically much cheaper than using a single backward Euler step as coarse solver.

3. Convergence analysis. We now show a priori and a posteriori error bounds for our algorithm. For the a priori estimates, we follow the analysis approach in [12] under the assumption that we are solving a linear system of differential equations, see also [15] and the recent book [14] for more details. Similar a priori error bounds for time-independent, jointly diagonalizable $F' = F'_n$, $G' = G'_n$ were obtained in [8] for the MGRIT method. Our analysis is based on the following abstract error estimate which holds for arbitrary Parareal algorithms with affine F_n and G_n . As before, we denote the linear parts of F_n and G_n by F'_n and G'_n .

LEMMA 3.1. *Assume that $\|F'_n - G'_n\| \leq \varepsilon$ and $\|G'_n\| \leq \delta$ for all $1 \leq n \leq N$. Let $e_n^k := u_n^k - u_n$ denote the Parareal error at time index n and iteration k . Then we have*

$$(3.1) \quad \|e_n^k\| \leq \varepsilon \|e_{n-1}^{k-1}\| + \delta \|e_{n-1}^k\|$$

for all $n, k \geq 1$. Further, we have

$$(3.2) \quad \|e_n^k\| \leq \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|e_m^{k-1}\|.$$

Proof. We have

$$(3.3) \quad \begin{aligned} e_n^k &= (F_n u_{n-1}^{k-1} + G_n u_{n-1}^k - G_n u_{n-1}^{k-1}) - F_n u_{n-1} \\ &= F'_n e_{n-1}^{k-1} + G'_n e_{n-1}^k - G'_n e_{n-1}^{k-1} \\ &= (F'_n - G'_n) e_{n-1}^{k-1} + G'_n e_{n-1}^k. \end{aligned}$$

Hence, (3.1) follows from the triangle inequality and the bounds assumed on the norms of $F'_n - G'_n$ and G'_n . We show (3.2) by induction on n . For $n = 1$, (3.1) yields $\|e_1^k\| = 0$, since $\|e_0^{k-1}\| = \|e_0^k\| = 0$. For the induction step, we use (3.1) again to obtain

$$\begin{aligned} \|e_n^k\| &\leq \varepsilon \|e_{n-1}^{k-1}\| + \delta \|e_{n-1}^k\| \\ &\leq \varepsilon \|e_{n-1}^{k-1}\| + \delta \sum_{m=1}^{n-1} \delta^{n-1-m-1} \|e_m^{k-1}\| = \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|e_m^{k-1}\|. \quad \square \end{aligned}$$

Using the inequalities from Lemma 3.1, we obtain the following superlinear and linear convergence results:

THEOREM 3.2. *Let*

$$(3.4) \quad \delta := \max_{1 \leq n \leq N} \sigma_{n,1}, \quad \varepsilon := \max_{1 \leq n \leq N} \sigma_{n,R_n+1}.$$

Then for any $1 \leq n \leq N$ we have

$$(3.5) \quad \|e_n^0\| \leq \sum_{m=0}^{n-1} \min(2\delta, (n-m)\varepsilon) \delta^{n-m-1} \|b_m\|,$$

where $b_0 := u_0$. Furthermore, for any $1 \leq n \leq N$ and $k \in \mathbb{N}$ we have

$$(3.6) \quad \|e_n^k\| \leq \varepsilon^k \sum_{m=1}^{n-k} \binom{n-m}{k-1} \delta^{n-m-k} \|e_m^0\|$$

$$(3.7) \quad \leq 2\varepsilon^k \sum_{m=0}^{n-k-1} \binom{n-m}{k} \delta^{n-m-k} \|b_m\|.$$

If $\delta \leq 1$, we have for $k \in \mathbb{N}$ that

$$(3.8) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \binom{N}{k} \varepsilon^k \max_{1 \leq m \leq N-k} \|e_m^0\|.$$

If $\delta < 1$, we have for $k \in \mathbb{N}$ that

$$(3.9) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \left(\frac{\varepsilon}{1-\delta} \right)^k \max_{1 \leq n \leq N-k} \|e_n^0\|.$$

Proof. We have $\|G'_n\| = \|F'_n\| = \sigma_{n,1}$ and (2.5). In particular, the assumptions of Lemma 3.1 are satisfied.

To show (3.5), note that

$$\begin{aligned} u_n^0 - u_n &= \left(\sum_{m=0}^{n-1} G'_n \cdots G'_{m+1} b_m + b_n \right) - \left(\sum_{m=0}^{n-1} F'_n \cdots F'_{m+1} b_m + b_n \right) \\ &= \sum_{m=0}^{n-1} G'_n \cdots G'_{m+1} b_m - F'_n \cdots F'_{m+1} b_m. \end{aligned}$$

Each of the terms in the sum can be bounded by $2\delta^{n-m}\|b_m\|$ using the triangle inequality and that $\|G'_n\|, \|F'_n\| \leq \delta$. Alternatively, we get the bound $(n-m)\varepsilon\delta^{n-m-1}\|b_m\|$ by induction on n ,

$$\begin{aligned} &\|G'_n \cdots G'_{m+1} b_m - F'_n \cdots F'_{m+1} b_m\| \\ &\leq \|(G'_n - F'_n)G'_{n-1} \cdots G'_{m+1} b_m\| \\ &\quad + \|F'_n(G'_{n-1} \cdots G'_{m+1} - F'_{n-1} \cdots F'_{m+1})b_m\| \\ &\leq \varepsilon\delta^{n-m-1}\|b_m\| + \delta \cdot (n-1-m)\varepsilon\delta^{n-1-m-1}\|b_m\| \\ &= (n-m)\varepsilon\delta^{n-m-1}\|b_m\|. \end{aligned}$$

Next, we show (3.6) by induction on k . For $k = 1$, (3.6) is precisely (3.2). For the

induction step, we use (3.2) again to obtain

$$\begin{aligned}
\|e_n^{k+1}\| &\leq \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|e_m^k\| \\
&\leq \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \varepsilon^k \sum_{l=1}^{m-k} \binom{m-l}{k-1} \delta^{m-l-k} \|e_l^0\| \\
&= \varepsilon^{k+1} \sum_{l=1}^{n-1-k} \underbrace{\sum_{m=l+k}^{n-1} \binom{m-l}{k-1} \delta^{m-l-k-1}}_{\leq \binom{n-l}{k}} \|e_l^0\|.
\end{aligned}$$

The bound on the sum of binomial coefficients follows from the hockey-stick identity.
The bound (3.7) follows by combining (3.6) with (3.5),

$$\begin{aligned}
\|e_n^k\| &\leq \varepsilon^k \sum_{m=1}^{n-k} \binom{n-m}{k-1} \delta^{n-m-k} \sum_{l=0}^{m-1} 2\delta^{m-l} \|b_l\| \\
&\leq 2\varepsilon^k \sum_{l=0}^{n-k-1} \underbrace{\sum_{m=l+1}^{n-k} \binom{n-m}{k-1} \delta^{n-k-l}}_{\leq \binom{n-l}{k}} \|b_l\|.
\end{aligned}$$

If $\delta \leq 1$, the bound (3.8) follows directly from (3.6),

$$\|e_n^k\| \leq \varepsilon^k \max_{1 \leq m \leq n-k} \|e_m^0\| \sum_{m=1}^{n-k} \binom{n-m}{k-1} \leq \varepsilon^k \max_{1 \leq m \leq n-k} \|e_m^0\| \cdot \binom{n}{k},$$

for each $1 \leq n \leq N$. Thus, taking the maximum w.r.t. n and using $\binom{n}{k} \leq \binom{N}{k}$ yields the claim.

Finally, to show (3.9), note that for $\delta < 1$, (3.2) can be further bounded by

$$\|e_n^k\| \leq \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|e_m^{k-1}\| < \varepsilon \max_{0 \leq m \leq n-1} \|e_m^{k-1}\| \frac{1}{1-\delta}.$$

Applying this inequality k times yields the claim. $\square$

We see that for stable fine solvers (in the sense that $\delta \leq 1$), we obtain from Theorem 3.2 similar superlinear and linear convergence results as in [17]. These assumptions are satisfied, for instance, for the heat equation or finite-element discretizations thereof, when F is the exact solution operator in time or the result of an appropriate stable time-discretization. More precisely, we have $\delta < 1$ when (homogeneous) Dirichlet boundaries are present and $\delta = 1$ in the case of pure Neumann boundaries. For $\delta < 1$, (3.6) and (3.7) show that the influence of the error or source term contributions associated with time index m on the error at time index n decreases exponentially as $n - m$ increases. We further note that the bound (3.7) does not depend on T or the number of time intervals N . Since the construction of the coarse solvers G_n is entirely local, we obtain weak scalability in the sense that the truncation rank R does not need to be increased to maintain a constant convergence rate as N grows proportionally with T .

As an important special case, we consider the case of a time-independent system with fixed fine solver F' , where G' is obtained by projecting F' onto an invariant subspace. In that case, we have the following exact representation of the error:

THEOREM 3.3. *Assume that $F' = F'_n$, $G' = G'_n$ for all $n \in \mathbb{N}$ and that P is a projection (continuous, linear, idempotent operator) with $PF' = F'P$ and $G' = PF'P$. Then for all $n, k \geq 0$ we have*

$$(3.10) \quad e_n^k = \begin{cases} (F' - G')^k e_{n-k}^0 & n > k, \\ 0 & \text{otherwise.} \end{cases}$$

Assuming $\|F' - G'\| \leq \varepsilon$, we obtain the improved estimate

$$(3.11) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \varepsilon^k \max_{1 \leq n \leq N-k} \|e_n^0\|.$$

Proof. The proof is the same as for the heat equation in [16]. Multiplying the error identity (3.3) from the left with P yields

$$Pe_n^k = P(F' - PF'P)e_{n-1}^{k-1} + P(PF'P)e_{n-1}^k = G'Pe_{n-1}^k.$$

Hence, $Pe_n^k = (G')^n Pe_0^k = 0$. Thus, multiplying (3.3) by $I - P$ and using $(I - P)G' = (I - P)(PF'P) = PF'P - PF'P = 0$ yields

$$e_n^k = (I - P)e_n^k = (I - P)(F' - G')e_{n-1}^{k-1} = (F' - G')(I - P)e_{n-1}^{k-1} = (F' - G')e_{n-1}^{k-1}.$$

Recursively applying this identity yields 0 when the time index reaches 0 before the iteration count. Otherwise, we get $e_n^k = (F' - G')^k e_{n-k}^0$. $\square$

When F' is a normal operator on a Hilbert space and P an orthogonal projection, $\|F' - G'\|$ is given by the maximum absolute value of the neglected spectrum. In the general (possibly) infinite-dimensional case, we can formulate this as follows:

COROLLARY 3.4. *Assume that $F' = F'_n$, $G' = G'_n$ for all $n \in \mathbb{N}$. Assume that F' is normal with spectral decomposition*

$$F' = \int_{\sigma(F')} \lambda d\mu_{F'}(\lambda), \quad \text{and let} \quad G' = \int_{\Gamma} \lambda d\mu_{F'}(\lambda),$$

for some Borel measurable $\Gamma \subseteq \sigma(F')$. Then we have

$$(3.12) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \left(\sup_{\lambda \in \sigma(F') \setminus \Gamma} |\lambda| \right)^k \cdot \max_{1 \leq n \leq N-k} \|e_n^0\|.$$

In particular, when F' is self-adjoint, the SVD of F' agrees with its spectral decomposition, and we obtain:

COROLLARY 3.5. *Assume that $F' = F'_n$, $n \in \mathbb{N}$, is self-adjoint and G the low-rank coarse solver for truncation rank R . Then*

$$(3.13) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \sigma_{R+1}^k \cdot \max_{1 \leq n \leq N-k} \|e_n^0\|.$$

To calculate the error bound (3.7), only the quantities (3.4) and $\|b_m\|$ are required, which are all known after the setup phase of the algorithm. However, while (3.7) provides some insight into the expected convergence speed of the Parareal algorithm, it significantly overestimates the error in practice. Thus, we further derive an a posteriori error bound, which takes the current Parareal updates $u_n^k - u_n^{k-1}$ into account:

THEOREM 3.6. *With the assumptions of Theorem 3.2, for any $1 \leq n \leq N$ and $k \in \mathbb{N}$ we have the a posteriori error bound*

$$(3.14) \quad \|e_n^k\| \leq \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|u_m^k - u_m^{k-1}\|.$$

If $\delta < 1$, we additionally have

$$(3.15) \quad \max_{1 \leq n \leq N} \|e_n^k\| \leq \frac{\varepsilon}{1 - \delta} \cdot \max_{1 \leq n \leq N} \|u_n^k - u_n^{k-1}\|.$$

Proof. We use the error identity

$$\begin{aligned} e_n^k &= u_n^k - F_n u_{n-1}^k + (F_n u_{n-1}^k - F_n u_{n-1}) \\ &= F_n u_{n-1}^{k-1} + G_n u_{n-1}^k - G_n u_{n-1}^{k-1} - F_n u_{n-1}^k + F_n' e_{n-1}^k \\ &= (F_n' - G_n')(u_{n-1}^{k-1} - u_{n-1}^k) + F_n' e_{n-1}^k. \end{aligned}$$

Taking norms then leads to

$$\|e_n^k\| \leq \varepsilon \|u_{n-1}^k - u_{n-1}^{k-1}\| + \delta \|e_{n-1}^k\|.$$

Applying this inequality repeatedly to its own right-hand side and noting that $e_0^k = u_0^k - u_0^{k-1} = 0$, we obtain (3.14). For $\delta < 1$, we obtain (3.15) from (3.14) using $\sum_{m=1}^{n-1} \delta^{n-m-1} \leq 1/(1 - \delta)$. $\square$

As for the a priori bound (3.7), the quantities δ and ε are known after the setup phase. Thus, to evaluate (3.14) and (3.15) while the algorithm converges, only the norms of the Parareal updates have to be computed.

4. Numerical experiments. We study the efficiency of the proposed low-rank coarse solvers in several experiments of increasing complexity. In all cases, we use linear finite elements for the spatial discretization and backward Euler time-stepping with equidistant time-step size as the fine solver F . The error between the fully discrete solution and its Parareal approximation is measured in the Euclidean norm of the solution degrees of freedom vectors (denoted as ℓ^2 -error). Within each Parareal time-interval $[t_{n-1}, t_n)$, we approximate the solution by considering the trajectory obtained from evaluating the fine solver F_n on u_{n-1}^k .

Our software implementation is based on the `pyMOR`¹ model order reduction library [27]. In particular, we use `pyMOR`'s randomized SVD algorithm with no power iterations and a variable number p of oversampling vectors to compute the low-rank coarse solvers. We will compare the proposed randomized SVDs with “exact” SVDs computed using `SciPy`'s² [32] `ARPACK` [24] wrapper. In all cases, we use a fixed truncation rank R for all G_n' . For the one- and two-dimensional test-cases, we use `pyMOR`'s built-in discretization toolkit, whereas experiment 4 uses `scikit-fem`³ [19] for the three-dimensional finite-element discretization. The Parareal algorithm is implemented with distributed-memory parallelization using `mpi4py`⁴ [7]. All code and data needed to conduct the numerical experiments is available on GitHub⁵ and Zenodo [29].

¹<https://pymor.org>

²<https://scipy.org>

³<https://scikit-fem.readthedocs.io>

⁴<https://mpi4py.readthedocs.io/>

⁵<https://github.com/sdrave/LowRankParareal/tree/v2>

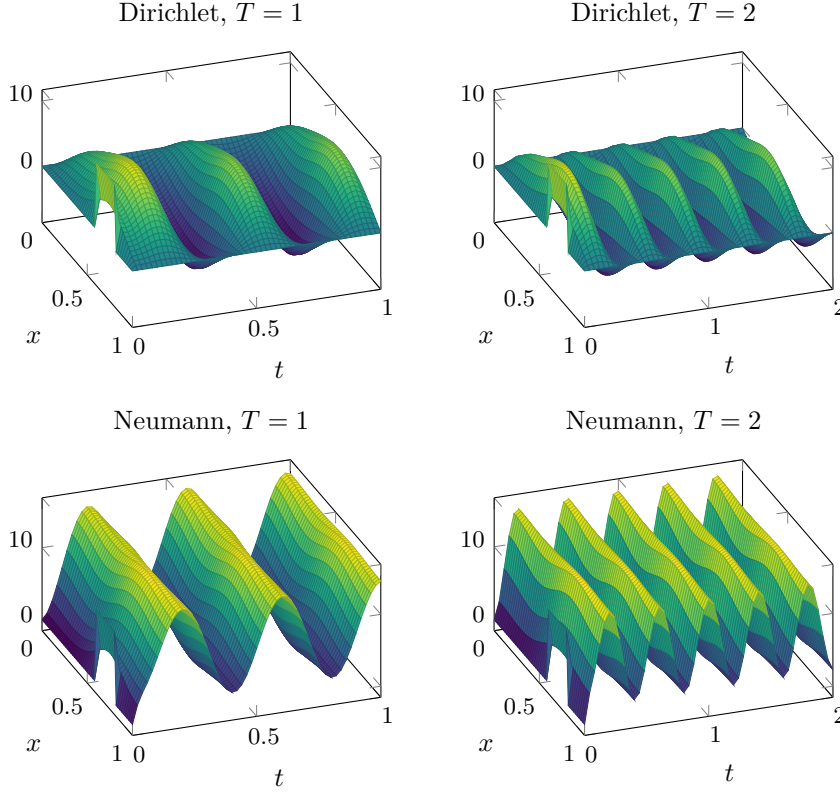


FIG. 1. *Experiment 1: Solutions of test problem (4.1) with different final times T and boundary conditions.*

380 All experiments were executed on a dual-socket server equipped with two Intel
 381 Xeon Gold 6254 CPUs with 18 physical cores each. A single thread per MPI rank was
 382 used for all computations. The server was running under Ubuntu 22.04, using CPython
 383 3.11.10 from the `python-build-standalone`⁶ project. All Python packages were ob-
 384 tained from PyPI⁷ in the exact versions specified by the `uv.lock`⁸ file contained in
 385 [29].

386 **4.1. Experiment 1.** As the most basic test case, we consider a one-dimensional
 387 heat equation on the unit interval,

$$\begin{aligned}
 (4.1) \quad & u_t(x, t) - u_{xx}(x, t) = f(x, t) := 100 \cdot \sin(5\pi t) \cdot (1 + \cos(3\pi x)), \\
 & u(x, 0) = u_0(x) = 10\chi_{[0.6, 0.8]},
 \end{aligned}$$

389 $x \in (0, 1)$, $t \in [0, T]$ for varying final time T . If we apply homogeneous Dirichlet
 390 boundary conditions

$$391 \quad u(0, t) = 0, \quad u(1, t) = 0,$$

⁶<https://github.com/astral-sh/python-build-standalone>

⁷<https://pypi.org>

⁸<https://docs.astral.sh/uv/>

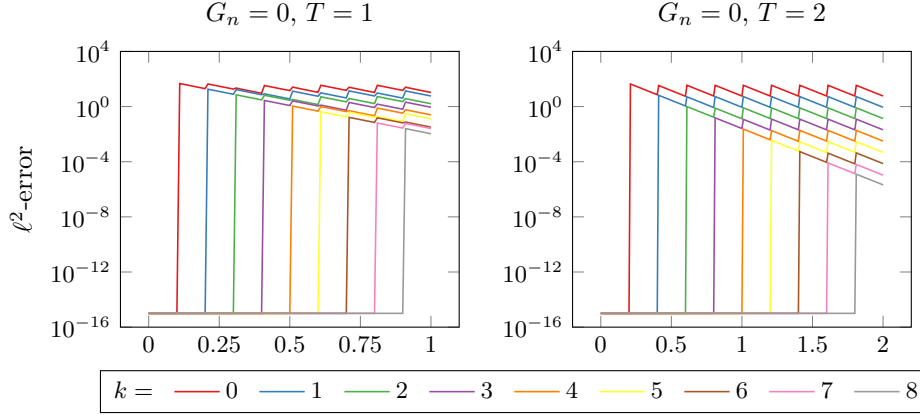


FIG. 2. Experiment 1 with homogeneous Dirichlet conditions: ℓ^2 -errors of Parareal approximations over time for an increasing number of Parareal iterations; $G_n = 0$, 10 Parareal subintervals of different length by varying the final time T .

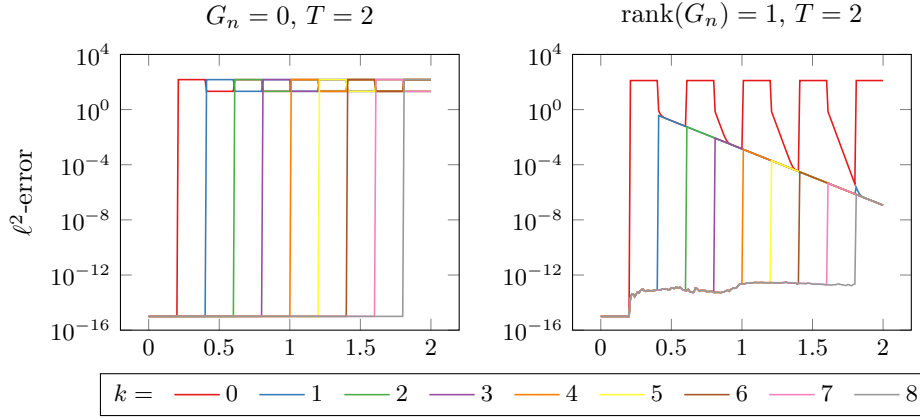


FIG. 3. Experiment 1 with homogeneous Neumann conditions: ℓ^2 -errors of Parareal approximations over time for an increasing number of Parareal iterations; left: $G_n = 0$, right: rank-1 G_n with constant Fourier mode.

then from Fourier analysis we know that the analytical solution of (4.1) is given by

$$\begin{aligned}
 u(x, t) &= \sum_{m=1}^{\infty} \hat{u}_m(t) \sqrt{2} \sin(m\pi x) \\
 \hat{u}_m(t) &= \hat{u}_{0,m} e^{-m^2 \pi^2 t} + \int_0^t \hat{f}_m(\tau) e^{-m^2 \pi^2 (t-\tau)} d\tau,
 \end{aligned}
 \tag{4.2}$$

where $\hat{f}_m$ and $\hat{u}_{0,m}$ denote the coefficients of the Fourier sine series for f and u_0 . Discretizing using a uniform mesh with 100 elements in space and $100 \cdot T$ time steps, we obtain the solutions shown in the top row of Figure 1.

Now, for each T we partition the time domain $[0, T]$ into 10 subintervals of the same length, resulting in $10 \cdot T$ time steps per interval. From (4.2), we can expect that the linear part F' of the fine solver has an exponentially decaying spectrum and

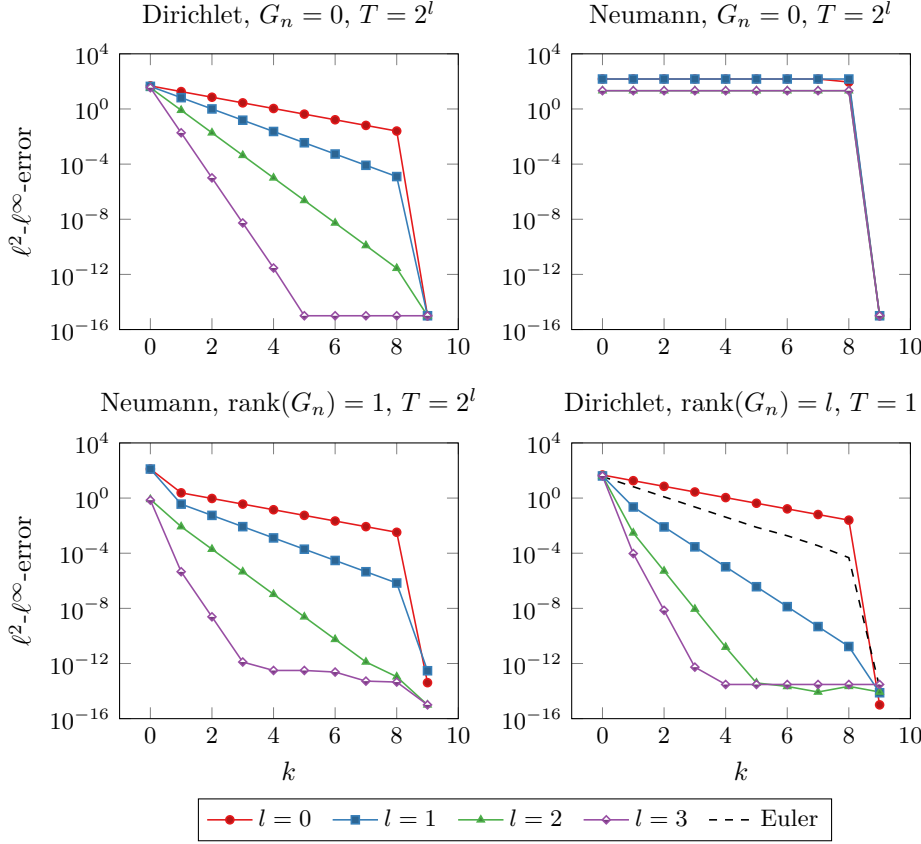


FIG. 4. Experiment 1 with different boundary conditions and coarse solvers G_n . Maximum ℓ^2 -errors over time vs. number of Parareal iterations for different final times $T = 2^l$ (top left/right, bottom left) and ranks R of G_n (bottom right). The dashed line in the bottom-right plot indicates the Parareal errors when the G_n are given by a single backward Euler step.

that the decay rate increases with the length of the Parareal time intervals. As all Fourier modes decay exponentially over time, we expect convergence even for $G = 0$, similar to the results in [16]. Indeed, in Figure 2, which shows the error of the Parareal iterates over time, we observe that the Parareal algorithm converges with a rate that rapidly accelerates with increasing interval length. The ℓ^2 -in-space ℓ^∞ -in-time error is shown in the top-left plot of Figure 4.

If we replace the Dirichlet boundary conditions by Neumann boundary conditions,

$$u_x(0, t) = 0 \quad u_x(1, t) = 0,$$

we obtain a constant Fourier mode, which is not damped over time. Hence, an error in the mean of the solution is propagated over the entire time interval $[0, T]$, and simply choosing $G = 0$ fails miserably (Figure 3, left and Figure 4, top right). To account for the constant mode, we now consider our first low-rank coarse solvers of the form (2.4) with the a priori choices $R_n = 1$, $\sigma_{n,1} = 1$ and $\varphi_{n,1} = \psi_{n,1}$ being the finite-element coefficient vector corresponding to the constant function with value 1. In order to correctly project onto the constant Fourier mode, we choose the L^2 -inner product in the definition of the G_n (2.4). To make the results more comparable with

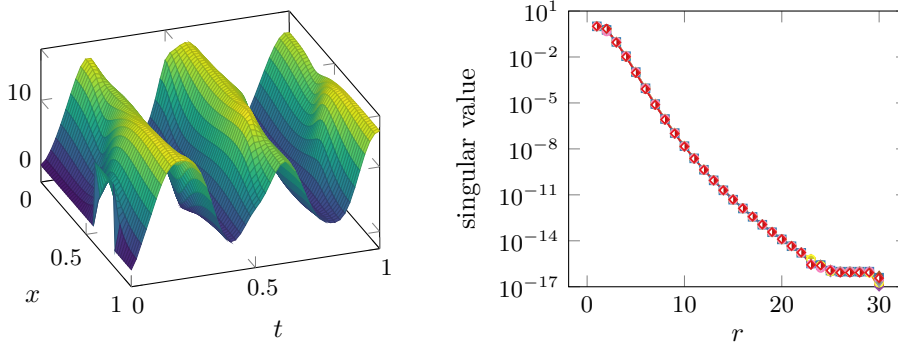


FIG. 5. *Experiment 2: Solution of test problem (4.3) (left) and singular values of the fine solvers F'_n (right). The singular values only differ slightly with respect to the chosen time interval.*

the case $G_n = 0$, we set $b_n = 0$ for all n .

As shown in the right plot of Figure 3 and the bottom-left plot of Figure 4, we successfully recover the convergence of the Parareal algorithm as observed in the Dirichlet case.

Returning to Dirichlet conditions, we finally study the effect of including higher-order Fourier modes $\sqrt{2}\sin(m\pi x)$ into the definition of G (again setting $b_n = 0$). In the bottom-right plot of Figure 4, we see that already two Fourier modes suffice to let Parareal converge to machine precision after only five iterations. This observed exponential convergence of our low-rank Parareal algorithm w.r.t. the rank of G_n agrees with Corollary 3.5 if we take into account that by (4.2), we have

$$\sigma_{R+1} = e^{-(R+1)^2 \pi^2 \Delta T}.$$

A direct analysis of this case is contained in [16]. We stress that already with a single Fourier mode, our algorithm converges much faster than a classical Parareal method using a single backward Euler step as coarse solver (dashed line in Figure 4).

4.2. Experiment 2. In experiment 1, we considered the special case where all fine solvers F_n are equal and known analytically (up to discretization). We now lift both these assumptions by considering a one-dimensional heat equation with a heat conductivity that varies both in space and time:

$$(4.3) \quad \begin{aligned} u_t(x, t) - \partial_x[(1 + 0.9 \sin(7\pi t + 2\pi x))u_x(x, t)] &= f(x, t), \\ u(x, 0) &= u_0(x), \quad u_x(0, t) = 0, \quad u_x(1, t) = 0. \end{aligned}$$

Here, the source term f is again given by (4.1), $x \in (0, 1)$ and $t \in [0, 1]$. As before, we discretize the problem with a uniform mesh with 100 elements in space and 100 time steps. The solution is shown in Figure 5 on the left.

We partition the time domain $[0, 1]$ into 10 equally-sized subintervals of 10 time steps each. Due to the time-dependence of the heat conductivity, the resulting fine solvers F'_n are no longer normal. We plot the decay of the singular values of each F'_n in Figure 5 on the right. We observe that the singular values decay rapidly and almost identically for each time-interval $[t_{n-1}, t_n]$. Thus, we expect good performance of the low-rank coarse solvers already for a small truncation rank R . In Figure 6, we show the maximum ℓ^2 -error over time with respect to the number of Parareal iterations k and truncation ranks R for different choices of coarse solvers. The top-left plot shows the

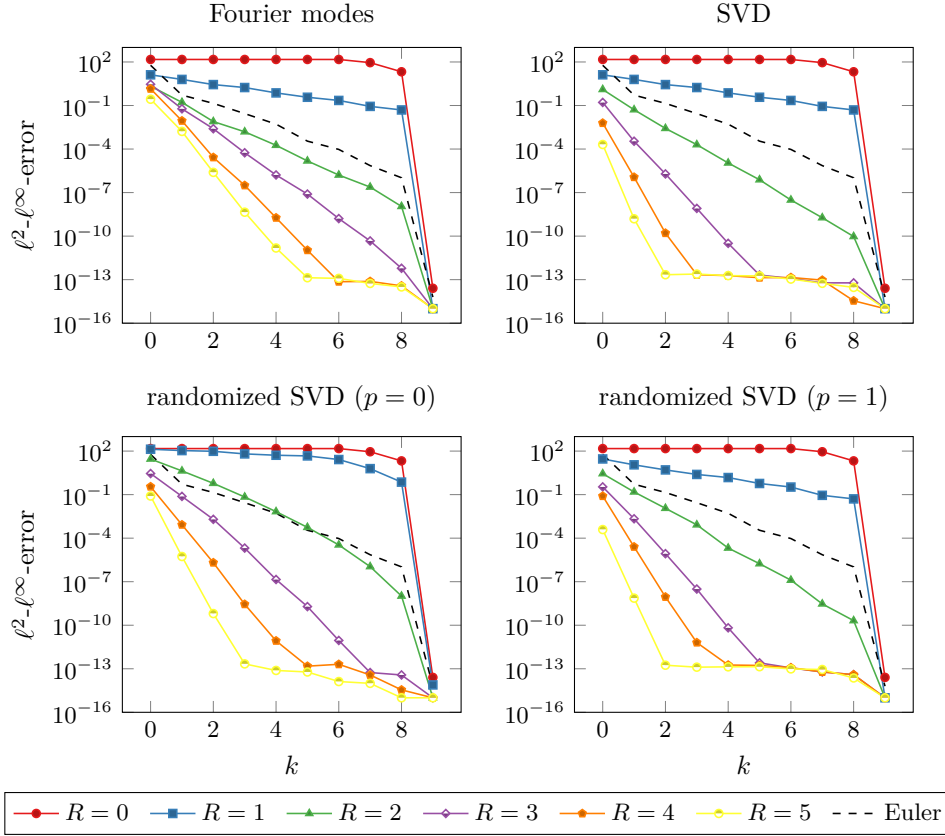
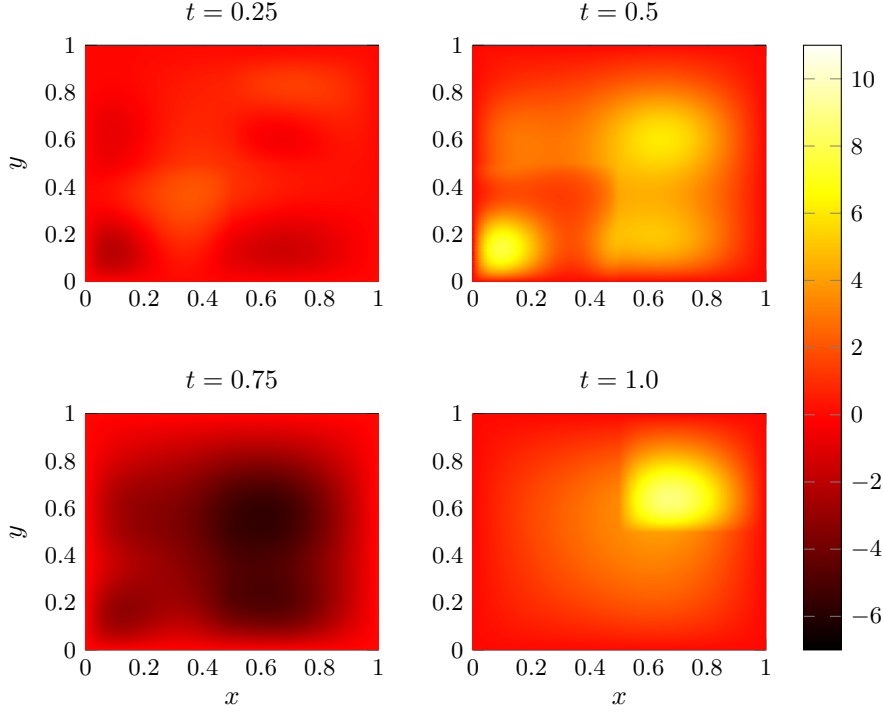


FIG. 6. Experiment 2: Maximum ℓ^2 -errors over time vs. number of Parareal iterations for different choices and ranks of G_n . The dashed lines indicate the Parareal errors when the G_n are given by a single backward Euler step.

error for a naive a priori choice of the G_n using the Fourier modes $\sqrt{2}\cos(m\pi x)$. We compare this choice to SVD-based coarse solvers, where the SVD is either computed exactly (top-right plot), or using a randomized SVD with no ($p = 0$) or a single ($p = 1$) oversampling vector (bottom plots). We now include b_n in the definitions of both the Fourier- and SVD-based coarse solvers.

In all cases, for $R \geq 3$, Parareal with these coarse solvers converges significantly faster compared to Parareal with G_n given by a single Euler step. We also observe that using an SVD to define G_n instead of using Fourier modes greatly improves the convergence speed when higher-order modes ($R \geq 2$) are used. In particular, for $R = 5$ the SVD-based Parareal reaches an error of 10^{-13} already for $k = 2$, whereas with Fourier modes, this error is only reached at $k = 5$. As computing an exact SVD of the fine solvers F'_n using standard methods is prohibitively expensive, it is important to note that using a randomized SVD with only a single oversampling vector already performs almost as well as using an exact SVD. Using no oversampling at all, however, the randomized SVD cannot improve the convergence speed for $R \leq 2$ compared to using Fourier modes.

FIG. 7. *Experiment 3: Solution of test problem (4.4) at different times t .*

4.3. Experiment 3. We proceed with a computationally more challenging problem by considering the two-dimensional heat equation given by

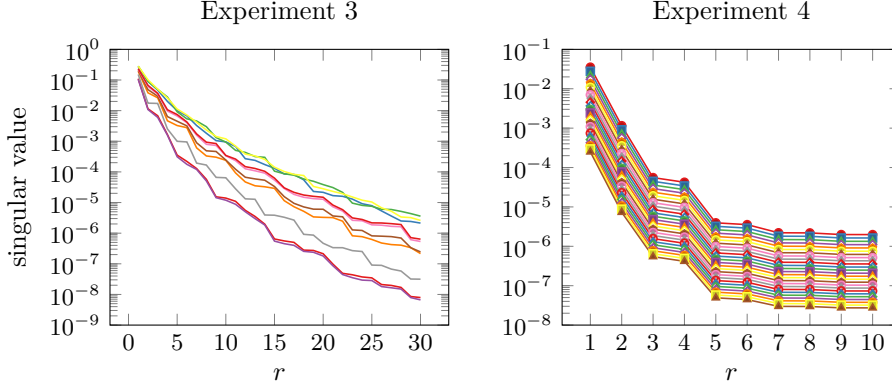
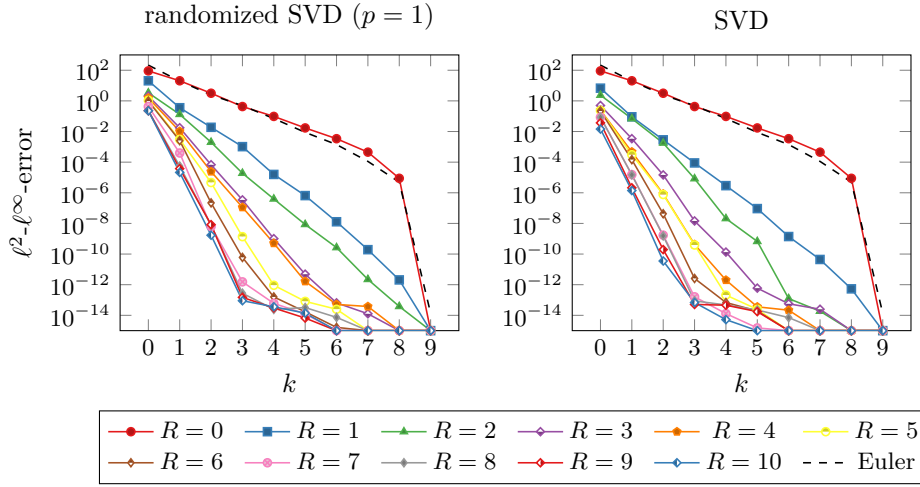
$$\begin{aligned}
 (4.4) \quad & u_t(x, y, t) - \nabla_{x,y} \cdot [d(x, y, t) \nabla_{x,y} u(x, y)] = f(x, y, t) \quad (x, y) \in \Omega, \\
 & u(x, y, 0) = u_0(x, y) \quad (x, y) \in \Omega, \\
 & u(x, y, t) = 0 \quad (x, y) \in \partial\Omega,
 \end{aligned}$$

with $\Omega := (0, 1)^2$, $t \in [0, 1]$ and

$$\begin{aligned}
 d(x, y, t) &= 1 + 0.9 \sin(7\pi t) \chi_{[0, 1/2]^2}(x, y) + 0.9 \cos(5\pi t) \chi_{[1/2, 1]^2}(x, y), \\
 f(x, y, t) &= 100 \sin(5\pi t) (1 + \cos(3\pi x)) (1 + \sin(4\pi y)), \\
 u_0(x, y) &= 10 \cdot \chi_{[0.6, 0.8]^2}(x, y).
 \end{aligned}$$

For the discretization, we use a triangular mesh with $4 \cdot 100 \cdot 100$ elements and 100 time steps ($\Delta t = 0.01$). The solution at times $t = 0.25, 0.5, 0.75, 1.0$ is visualized in Figure 7.

As before, we consider 10 equally-sized time intervals of 10 time steps each. Compared to experiment 2, we observe a slower decay of the singular values of the fine solvers in Figure 8 (left), and the decay rate now visibly depends on the selected time interval. Still, the decay is exponential and SVD-based coarse solvers of rank 1 are already sufficient to outperform single Euler-step coarse solvers (Figure 9). For 5 Parareal iterations, which require at least half the time to solve the PDE sequentially in time, Parareal with single Euler-step coarse solvers only reduces the error to 10^{-2} . Using an exact SVD with truncation rank $R = 1$, an error of 10^{-7} is attained for the

FIG. 8. Singular values of the fine solvers F'_n (left: experiment 3, right: experiment 4).FIG. 9. Experiment 3: Maximum ℓ^2 -errors over time vs. number of Parareal iterations for different ranks R of G_n . The dashed lines indicate the Parareal errors when the G_n are given by a single backward Euler step.

same number of iterations. With truncation rank $R = 5$, an error of 10^{-6} is reached already for $k = 2$, whereas with single Euler-step solvers, this error is not attained until the fine solvers have propagated through the entire time domain. When a randomized SVD with a single oversampling vector ($p = 1$) is used instead of an exact SVD, convergence is slightly slower. $R = 1$ is still sufficient to obtain a significantly faster error decay compared to single Euler-step coarse solvers. To reach an error of 10^{-6} at $k = 2$, the rank of the coarse solvers has to be incremented by one to $R = 6$.

4.3.1. Effect of time discretization. The effectiveness of our method depends on the rapid singular value decay of the fine solvers F'_n . While such a decay is theoretically guaranteed for problems of parabolic type when F'_n is the analytical solution operator, the chosen time-stepping method might have a negative effect on the singular value decay. To study such a potential effect, we repeat experiment 3 for two different second-order Runge-Kutta methods (SDIRK2, SDIRK2-L) with the

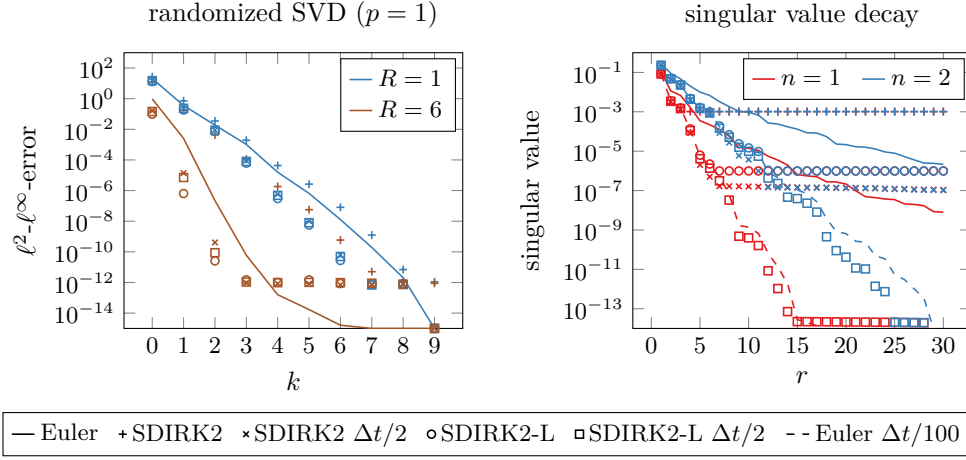


FIG. 10. *Experiment 3: Effect of the choice of time-stepping method on Parareal convergence; left: Maximum ℓ^2 -errors over time vs. number of Parareal iterations for different ranks R of G_n , right: singular values of the fine solvers F'_n for different time intervals $[t_{n-1}, t_n]$. The time-stepping method used is indicated by the line style and marker type. “ $\Delta t/2$ ” and “ $\Delta t/100$ ” indicate that the time-step size of the fine solver has been scaled by a factor of $1/2$ or $1/100$ compared to the standard time-step size for this experiment.*

following Butcher tableaux:

$$\text{SDIRK2: } \begin{array}{c|cc} 1 & 1 & \\ 0 & -1 & 1 \\ \hline & 1/2 & 1/2 \end{array} \quad \text{SDIRK2-L: } \begin{array}{c|cc} \gamma & \gamma & \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array} \quad \gamma = \frac{2-\sqrt{2}}{2}.$$

Both methods belong to the class of singly-diagonally implicit Runge-Kutta (SDIRK) methods. When using direct linear solvers, SDIRK-2 is attractive for solving linear ODE systems with time-dependent operators, as it does not require any additional matrix factorizations compared to the backward Euler method. SDIRK-2 is not L-stable, however. In contrast, the SDIRK2-L method is L-stable, but requires twice as many matrix factorizations.

For both time steppers, we consider the corresponding Parareal method with randomized-SVD-based coarse solvers. The convergence of these schemes is compared against the backward Euler method in the left plot of Figure 10. In all cases, the error is computed with respect to the time-discrete solution obtained with the same time-stepping method.

We observe that for truncation rank $R = 1$, all three time-stepping methods yield comparable convergence rates, with SDIRK2-L showing the fastest convergence. For $R = 6$, however, the convergence rate for SDIRK2 stagnates compared to the other methods. Indeed, for SDIRK2, the decay of the singular values of the corresponding fine solvers F'_n (right plot of Figure 10), stagnates already at a value of 10^{-3} . This effect can be explained by the lack of L-stability, which causes a bad approximation of solution components associated with small singular values of the analytical solution operator. For SDIRK2-L, we observe a much faster singular value decay, although a plateau with slow decay at 10^{-7} is still present. We attribute this plateau to the fact that, while being L-stable, the stability function $R(z)$ of the SDIRK2-L method is not monotonically decreasing for $z \rightarrow -\infty$. In contrast, we observe no such plateau for the backward Euler method, which has a monotonically decreasing

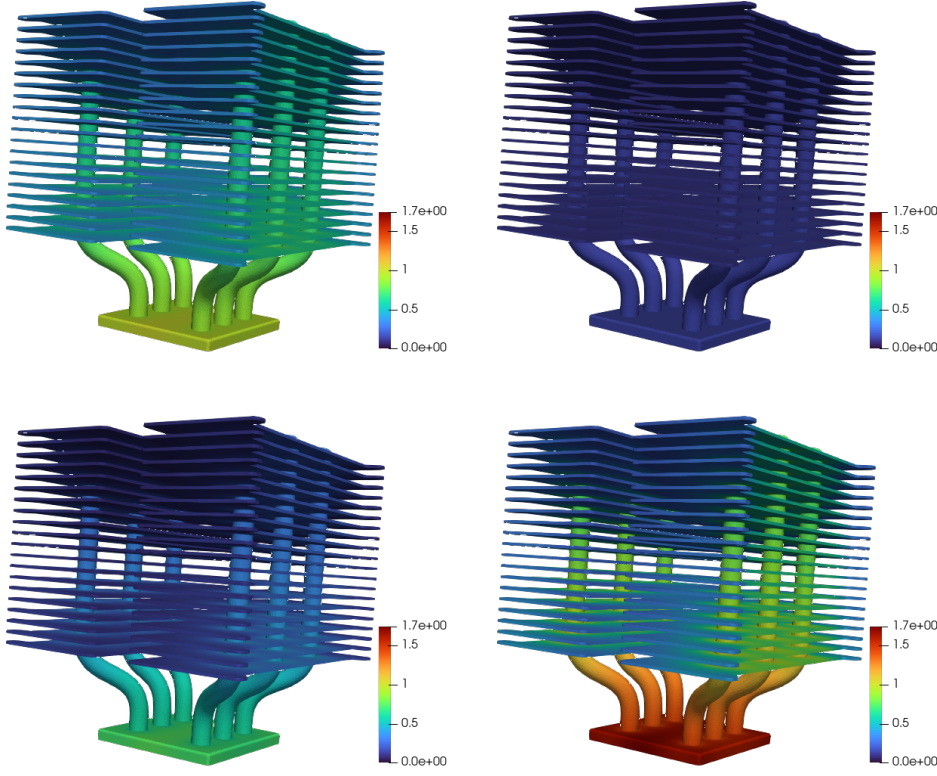


FIG. 11. *Experiment 4: Solution of (4.5) at times $t = 0.25$ (top left), $t = 0.5$ (top right), $t = 0.75$ (bottom left) and $t = 1$ (bottom right).*

stability function. However, being a first-order method, the overall approximation of the analytical solution operator is not very good: When considering a significantly smaller time-step size ($\Delta t/100$), we observe a much faster singular value decay of the F'_n , which is well-matched by SDIRK2-L before reaching the plateau.

Overall, we see that a bad approximation of the analytical solution operator by the chosen time-stepping method can indeed affect the performance of our method. In the case of experiment 3, halving the time step size (SDIRK2/SDIRK2-L with $\Delta t/2$ in Figure 10) is already sufficient for obtaining similar convergence rates for all considered time-stepping methods.

4.4. Experiment 4. Finally, we consider a larger three-dimensional test problem on the heat-sink geometry shown in Figure 11. The geometry is scaled to fit into a cube with side length 1. We partition the domain Ω into subdomains corresponding to the fins (Ω_{fin}), heat pipes (Ω_{pipe}) and the baseplate (Ω_{base}) of the heat sink. Further, we let Γ_{fin} denote the boundary of the fins and Γ_{bot} the bottom boundary of

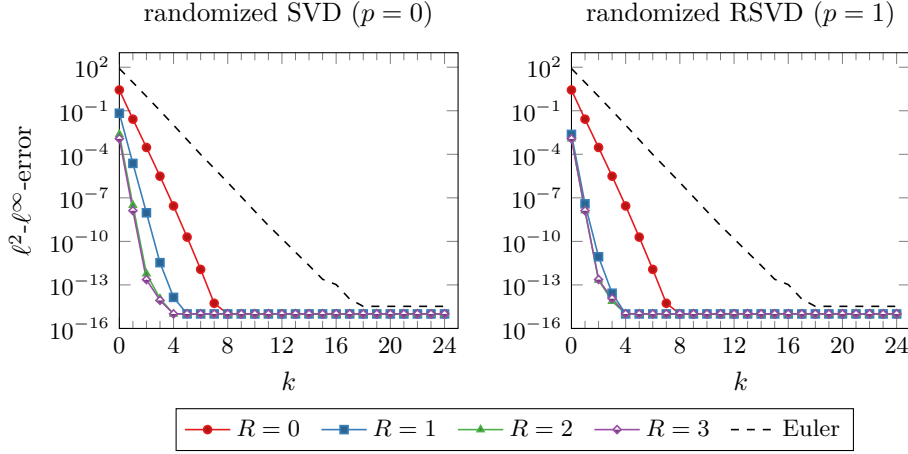


FIG. 12. *Experiment 4: Maximum ℓ^2 -errors over time vs. number of Parareal iterations for different ranks R of G_n . The dashed lines indicate the Parareal errors when the G_n are given by a single backward Euler step.*

the baseplate. We solve the following heat equation with mixed boundary conditions:

$$\begin{aligned}
 u_t(\mathbf{x}, t) - \nabla \cdot [d(\mathbf{x}) \nabla_{\mathbf{x}} u(\mathbf{x})] &= 0 & \mathbf{x} \in \Omega, \quad t \in [0, 1], \\
 -d(\mathbf{x}) \nabla_{\mathbf{x}} u(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) &= \left(\frac{1}{2} + t\right) \cdot u(\mathbf{x}, t) & \mathbf{x} \in \Gamma_{\text{fin}}, \\
 -d(\mathbf{x}) \nabla_{\mathbf{x}} u(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) &= -g(t) & \mathbf{x} \in \Gamma_{\text{bot}}, \\
 -d(\mathbf{x}) \nabla_{\mathbf{x}} u(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) &= 0 & \mathbf{x} \in \Gamma \setminus (\Gamma_{\text{fin}} \cup \Gamma_{\text{bot}}), \\
 u(\mathbf{x}, 0) &= 0.
 \end{aligned}
 \tag{4.5}$$

Here, $d(\mathbf{x})$ is given by

$$d(\mathbf{x}) = \begin{cases} 10 & \mathbf{x} \in \Omega_{\text{fin}}, \\ 100 & \mathbf{x} \in \Omega_{\text{base}}, \\ 1000 & \mathbf{x} \in \Omega_{\text{pipe}}, \end{cases}$$

accounting for different heat conductivities of the materials that make up the heat-sink components. We assume that the heat inflow at the baseplate $g(t)$ is given by different load patterns of the form

$$g(t) = \begin{cases} 50 \cdot \frac{t}{0.3} & t \leq 0.3, \\ 50 \cdot \left(1 + \text{sign} \left(\sin \left(\frac{t-0.3}{0.3} \cdot 8 \cdot \pi \right) \right) \right) & 0.3 < t \leq 0.6, \\ 50 \cdot \left(1 + \cos \left(\frac{t-0.6}{0.4} \cdot 20 \cdot \pi \right) \right) & 0.6 < t. \end{cases}$$

The time-dependent Robin parameter $(1/2 + t)$ roughly models the cooling of the heat sink by a fan that speeds up over time.

For the discretization in space, we use a tetrahedral mesh with 1,405,317 elements, yielding a finite-element space with 444,693 degrees of freedom. For the time discretization, we use 500 equidistant backward Euler steps. The solution at times $t = 0.25, 0.5, 0.75, 1.0$ is visualized in Figure 11.

For the Parareal algorithm, we partition the time domain into 25 intervals of equal size with 20 time steps each. The singular values of the fine solvers F_n decay quickly,

TABLE 4.1

Experiment 4. Total runtimes in seconds (total number of fine solver and adjoint fine solver evaluations per time interval) for different coarse solvers. “ G_n ” refers to the computation of the rank- R coarse solvers using randomized SVDs with oversampling parameter p . The required number of fine solver evaluations per time interval is given by $K_{F,n} + 1$, where $K_{F,n}$ is given by (2.6). One additional fine solver evaluation is needed to obtain an approximation at intermediate time points. The total time for a sequential solution is 3,290 seconds.

	G_n	u^0	u^1	u^2	u^3	u^4
$p = 0 \ r = 0$	169 (1)	336 (2)	503 (3)	671 (4)	838 (5)	1,005 (6)
$p = 0 \ r = 1$	337 (3)	506 (4)	674 (5)	843 (6)	1,013 (7)	1,182 (8)
$p = 0 \ r = 2$	339 (5)	507 (6)	676 (7)	844 (8)	1,013 (9)	1,182 (10)
$p = 0 \ r = 3$	342 (7)	510 (8)	678 (9)	846 (10)	1,014 (11)	1,182 (12)
$p = 1 \ r = 0$	335 (3)	502 (4)	669 (5)	837 (6)	1,005 (7)	1,174 (8)
$p = 1 \ r = 1$	339 (5)	509 (6)	678 (7)	847 (8)	1,017 (9)	1,186 (10)
$p = 1 \ r = 2$	342 (7)	509 (8)	677 (9)	845 (10)	1,014 (11)	1,183 (12)
$p = 1 \ r = 3$	345 (9)	513 (10)	682 (11)	851 (12)	1,020 (13)	1,189 (14)

with the third singular value being almost three orders of magnitude smaller than the first (Figure 8, right). The singular values show a very similar decay throughout all time intervals, up to a factor that decreases as the time index n and, hence, the Robin parameter $(1/2 + t)$, increase.

We consider low-rank coarse solvers G_n that are computed using randomized SVDs with no ($p = 0$) or one ($p = 1$) oversampling vector and compare their performance with single Euler-step coarse solvers in Figure 12. We observe that the Euler coarse solvers perform very badly for this example, reaching machine precision only at $k = 18$, whereas using constant G_n ($R = 0$), machine precision is reached already after half as many iterations. Using low-rank coarse solvers, we can further halve the number of required iterations when we set the truncation rank R to 2 or higher. When we are willing to accept a slightly larger error of around 10^{-13} , $k = 2$ iterations suffice, whereas for $k = 1$, we can still reach an error of 10^{-7} . The use of oversampling in the SVD computation only has a relevant impact for $R = 1$, where the error can improve several orders of magnitude. For $R > 1$, there is no significant difference.

For this experiment, we also report algorithm runtimes along with the number of required (adjoint) fine solver evaluations in Table 4.1. We see that the required time for the randomized SVDs does not significantly increase for larger R . This is due to the fact that we have chosen a direct linear solver (**SuperLU**) for the time stepping: in each time interval, we need to evaluate F_n for $R+p+1$ initial values. As all initial values are known at the same time, we can perform the time stepping simultaneously by solving the same linear system for $R+p+1$ right-hand sides in each time step. Since for direct solvers, the bulk of the computational time lies within the LU decomposition of the system matrix, solving for additional right-hand sides comes at only a very small additional cost. After having evaluated F_n , we can proceed in the same way with the $R+p$ evaluations of F_n^* .

While using a direct linear solver might not be realistic in many typical Parareal applications, we remark that also with iterative solvers significant optimizations are possible, for instance by using block Krylov methods or vectorized software implementations. Additionally, all individual initial and terminal value problems are completely independent, so their solution can be trivially parallelized.

Finally, the number of required fine solver evaluations when using low-rank coarse

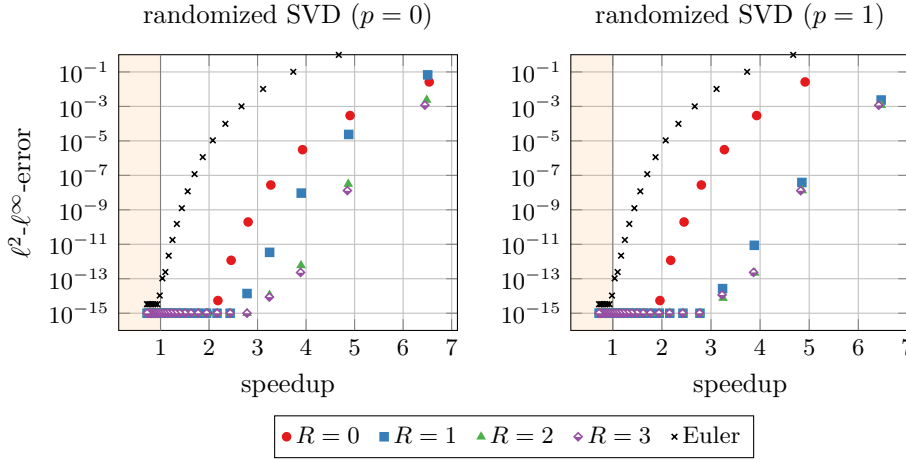


FIG. 13. Experiment 4: Maximum ℓ^2 -errors over time vs. speedup w.r.t. a sequential solution for different ranks R of G_n and different numbers of Parareal iterations. The marker $\times$ indicates the Parareal speedup when the G_n are given by a single backward Euler step.

solvers is actually smaller than for the Euler coarse solver due to the much higher Parareal convergence speed. For instance, to reach an error below 10^{-7} , no more than 7 evaluations of F_n and F_n^* per time interval are required, whereas with Euler coarse solvers, 10 evaluations of F_n are needed.

We plot the overall speedup factor achieved by the different Parareal variants in comparison to a completely sequential solution of (4.5) in Figure 13. To achieve an error of 10^{-7} , an optimal speedup of 5 is achieved with $R \geq 1$ whereas with Euler coarse solvers, only a speedup of around 1.8 is possible. For an error of 10^{-13} , there is almost no speedup with the Euler coarse solvers, whereas with low-rank coarse solvers, we can still reach a speedup of around 4 for $R \geq 2$.

4.5. Error estimator efficiency. For experiments 3 and 4, we evaluate the efficiencies η^k of the a posteriori error estimator (3.14) given by

$$(4.6) \quad \eta^k := \frac{\max_{t \in [0, T]} \|u^k(t) - u(t)\|}{\max_{1 \leq n < N} \varepsilon \sum_{m=1}^{n-1} \delta^{n-m-1} \|u_m^k - u_m^{k-1}\|},$$

where, as in the previous experiments, we consider the L^∞ -in-time error by extending u_{n-1}^k into $[t_{n-1}, t_n)$ via F_n . The constants δ and ε are obtained from the computed randomized SVDs, where we use the additional oversampling vector ($p = 1$) to approximate $\sigma_{n, R+1}$. We plot η^k for different values of R in Figure 14. We observe that in some cases, η^k can become larger than 1, which means that the error is underestimated. However, this only happens when the actual error is below 10^{-13} (dashed segments), so we attribute these cases to limited floating point accuracy. For both experiments, the efficiency of the error estimator is quite good with $\eta^k > 0.1$ in most cases. In general, we cannot expect to be able to bound η^k from below: except for the special case in Theorem 3.3, the actual convergence rate might be better than what is predicted by the SVD truncation error, since dominant left-singular vectors of F'_n might be projected onto less significant right-singular vectors of F'_{n+1} . Thus, it is not surprising that we observe cases where $\eta_k \ll 0.1$. However, since $\|u_n^{k+1} - u_n^k\| \leq \|e_n^{k+1}\| + \|e_n^k\| \leq 2\|e_n^k\|$ when $\|e_n^{k+1}\| < \|e_n^k\|$, we expect no more than

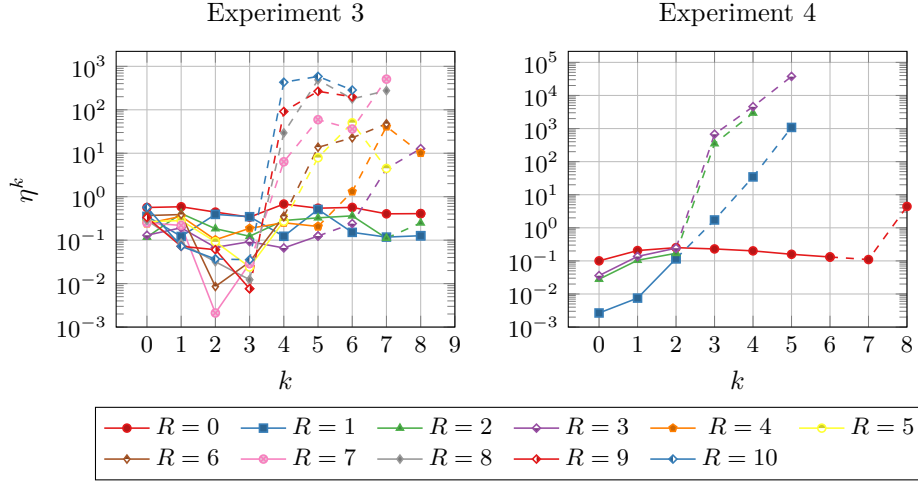


FIG. 14. Efficiencies η^k (4.6) for the a posteriori error bound (3.14) vs. number of Parareal iterations for different ranks R of G_n , $p = 1$. The plots are drawn dashed when the maximum error falls below 10^{-13} .

one unnecessary Parareal iteration in general.

5. Conclusions. We have shown that using low-rank coarse solvers can dramatically accelerate the convergence of Parareal for problems of parabolic type. Thus, low-rank coarse solvers can increase the overall parallelism of Parareal by trading global iterations for embarrassingly parallel fine solver evaluations in the algorithm's setup phase. After the setup phase, low-rank coarse solvers are very fast to evaluate, which reduces the runtime of the sequential coarse corrections and, hence, increases parallelism even further. At the same time, as we have seen in experiment 4, low-rank coarse solvers can even lead to a reduction of the overall computational work.

In this work we have focused on the solution of a single linear PDE. While this approach cannot be expected to perform well for purely hyperbolic problems [11], an extension of the approach to parametric or non-linear problems by applying further techniques from the reduced basis methodology seems promising. For large-scale problems, the approximation of space- and time-localized transfer operators could accelerate the setup phase of the algorithm by replacing expensive globally-coupled fine solver evaluations with small independent time-evolution problems that could be solved on a single compute node.

REFERENCES

- [1] K. AIT-AMEUR AND Y. MADAY, *Multi-step variant of the parareal algorithm: convergence analysis and numerics*, ESAIM Math. Model. Numer. Anal., 58 (2024), pp. 673–694, <https://doi.org/10.1051/m2an/2024014>.
- [2] P. BENNER, M. OHLBERGER, A. PATERA, G. ROZZA, AND K. URBAN, eds., *Model reduction of parametrized systems*, vol. 17 of MS&A. Modeling, Simulation and Applications, Springer, Cham, 2017, <https://doi.org/10.1007/978-3-319-58786-8>. Selected papers from the 3rd MoRePaS Conference held at the International School for Advanced Studies (SISSA), Trieste, October 13–16, 2015.
- [3] A. BUHR, L. IAPICHINO, M. OHLBERGER, S. RAVE, F. SCHINDLER, AND K. SMETANA, *Localized model reduction for parameterized problems*, in Model Order Reduction (Volume 2), P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira,

- eds., De Gruyter, Berlin, Boston, 2021, <https://doi.org/10.1515/9783110671490-006>.
- [4] A. BUHR AND K. SMETANA, *Randomized local model order reduction*, SIAM J. Sci. Comput., 40 (2018), pp. A2120–A2151, <https://doi.org/10.1137/17M1138480>.
- [5] B. CARREL, M. J. GANDER, AND B. VANDEREYCKEN, *Low-rank parareal: a low-rank parallel-in-time integrator*, BIT Numerical Mathematics, 63 (2023), p. 13, <https://doi.org/10.1007/s10543-023-00953-3>.
- [6] F. CHEN, J. S. HESTHAVEN, AND X. ZHU, *On the use of reduced basis methods to accelerate and stabilize the parareal method*, Reduced Order Methods for modeling and computational reduction, (2014), pp. 187–214, https://doi.org/10.1007/978-3-319-02090-7_7.
- [7] L. DALCIN AND Y.-L. L. FANG, *Mpi4py: Status Update After 12 Years of Development*, Computing in Science & Engineering, 23 (2021), pp. 47–54, <https://doi.org/10.1109/MCSE.2021.3083216>.
- [8] V. A. DOBREV, TZ. KOLEV, N. A. PETERSSON, AND J. B. SCHRODER, *Two-Level Convergence Theory for Multigrid Reduction in Time (MGRIT)*, SIAM Journal on Scientific Computing, 39 (2017), pp. S501–S527, <https://doi.org/10.1137/16M1074096>.
- [9] C. FARHAT, J. CORTIAL, C. DASTILLUNG, AND H. BAVESTRELLO, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, International Journal for Numerical Methods in Engineering, 67 (2006), pp. 697–724, <https://doi.org/10.1002/nme.1653>.
- [10] M. GANDER AND M. PETCU, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, ESAIM: Proceedings, 25 (2008), pp. 114–129, <https://doi.org/10.1051/proc:082508>.
- [11] M. J. GANDER, *Parareal for Hyperbolic Problems Just Does not Work – Or Does it?*, in Proceedings of the 29th International Conference on Domain Decomposition Methods, 2026, https://www.ddm.org/DD29/proceedings/ID35_pages.pdf. To appear.
- [12] M. J. GANDER AND E. HAIRER, *Nonlinear convergence analysis for the parareal algorithm*, in Domain decomposition methods in science and engineering XVII, Springer, 2008, pp. 45–56, https://doi.org/10.1007/978-3-540-75199-1_4.
- [13] M. J. GANDER AND E. HAIRER, *Analysis for parareal algorithms applied to Hamiltonian differential equations*, Journal of Computational and Applied Mathematics, 259 (2014), pp. 2–13, <https://doi.org/10.1016/j.cam.2013.01.011>.
- [14] M. J. GANDER AND T. LUNET, *Time Parallel Time Integration*, SIAM, Philadelphia, PA, 2024, <https://doi.org/10.1137/1.9781611978025>.
- [15] 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, <https://doi.org/10.1137/22M1487163>.
- [16] M. J. GANDER, M. OHLBERGER, AND S. RAVE, *A Parareal algorithm without Coarse Propagator?*, 2024, <https://arxiv.org/abs/2409.02673>.
- [17] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578, <https://doi.org/10.1137/05064607X>.
- [18] L. GRIGORI, S. A. HIRSTOAGA, V.-T. NGUYEN, AND J. SALOMON, *Reduced model-based parareal simulations of oscillatory singularly perturbed ordinary differential equations*, J. Comput. Phys., 436 (2021), pp. Paper No. 110282, 18, <https://doi.org/10.1016/j.jcp.2021.110282>.
- [19] T. GUSTAFSSON AND G. D. MCBAIN, *Scikit-fem: A Python package for finite element assembly*, Journal of Open Source Software, 5 (2020), p. 2369, <https://doi.org/10.21105/joss.02369>.
- [20] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, <https://doi.org/10.1137/090771806>.
- [21] L. HE, *The reduced basis technique as a coarse solver for parareal in time simulations*, Journal of Computational Mathematics, (2010), pp. 676–692, <https://www.jstor.org/stable/43693608>.
- [22] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics, Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016, <https://doi.org/10.1007/978-3-319-22470-1>. BCAM SpringerBriefs.
- [23] B. JIN, Q. LIN, AND Z. ZHOU, *Optimizing coarse propagators in parareal algorithms*, SIAM J. Sci. Comput., 47 (2025), pp. A735–A761, <https://doi.org/10.1137/23M1619733>.
- [24] R. B. LEHOUCQ, D. C. SORENSSEN, AND C. YANG, *ARPACK Users’ Guide*, Software, Environments, and Tools, Society for Industrial and Applied Mathematics, 1998, <https://doi.org/10.1137/1.9780898719628>.
- [25] J.-L. LIONS, Y. MADAY, AND G. TURINICI, *Résolution d’edp par un schéma en temps pararéel*,

- Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332 (2001), pp. 661–668, [https://doi.org/10.1016/S0764-4442\(00\)01793-6](https://doi.org/10.1016/S0764-4442(00)01793-6).
- [26] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numerica, 29 (2020), pp. 403–572, <https://doi.org/10.1017/S0962492920000021>.
- [27] R. MILK, S. RAVE, AND F. SCHINDLER, *pyMOR – Generic Algorithms and Interfaces for Model Order Reduction*, SIAM Journal of Scientific Computing, 38 (2016), pp. S194–S216, <https://doi.org/10.1137/15M1026614>.
- [28] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced basis methods for partial differential equations*, vol. 92 of Unitext, Springer, Cham, 2016, <https://doi.org/10.1007/978-3-319-15431-2>.
- [29] S. RAVE, *A parareal algorithm with low-rank coarse solvers (supplement)*. Zenodo, 2026, <https://doi.org/10.5281/zenodo.20442179>.
- [30] J. SCHLEUSS AND K. SMETANA, *Optimal local approximation spaces for parabolic problems*, Multiscale Model. Simul., 20 (2022), pp. 551–582, <https://doi.org/10.1137/20M1384294>.
- [31] J. SCHLEUSS, K. SMETANA, AND L. TER MAAT, *Randomized quasi-optimal local approximation spaces in time*, SIAM J. Sci. Comput., 45 (2023), pp. A1066–A1096, <https://doi.org/10.1137/22M1481002>.
- [32] P. VIRTANEN, R. GOMMERS, T. E. OLIPHANT, M. HABERLAND, T. REDDY, D. COURNAPEAU, E. BUROVSKI, P. PETERSON, W. WECKESSER, J. BRIGHT, S. J. VAN DER WALT, M. BRETT, J. WILSON, K. J. MILLMAN, N. MAYOROV, A. R. J. NELSON, E. JONES, R. KERN, E. LARSON, C. J. CAREY, Í. POLAT, Y. FENG, E. W. MOORE, J. VANDERPLAS, D. LAXALDE, J. PERKTOLD, R. CIMRMAN, I. HENRIKSEN, E. A. QUINTERO, C. R. HARRIS, A. M. ARCHIBALD, A. H. RIBEIRO, F. PEDREGOSA, P. VAN MULBREGT, AND SCI-PY 1.0 CONTRIBUTORS, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 17 (2020), pp. 261–272, <https://doi.org/10.1038/s41592-019-0686-2>.
- [33] S.-L. WU, *Toward parallel coarse grid correction for the parareal algorithm*, SIAM J. Sci. Comput., 40 (2018), pp. A1446–A1472, <https://doi.org/10.1137/17M1141102>.