

# Lossy data compression reduces communication time in hybrid time-parallel integrators

Lisa Fischer · Sebastian Götschel · Martin Weiser

To appear in *Computing and Visualization in Science*, DOI 10.1007/s00791-018-0293-2

**Abstract** Parallel-in-time methods for solving initial value problems are a means to increase the parallelism of numerical simulations. Hybrid parareal schemes interleaving the parallel-in-time iteration with an iterative solution of the individual time steps are among the most efficient methods for general nonlinear problems. Despite the hiding of communication time behind computation, communication has in certain situations a significant impact on the total runtime. Here we present strict, yet not sharp, error bounds for hybrid parareal methods with inexact communication due to lossy data compression, and derive theoretical estimates of the impact of compression on parallel efficiency of the algorithms. These and some computational experiments suggest that compression is a viable method to make hybrid parareal schemes robust with respect to low bandwidth setups.

## 1 Introduction

Nowadays, the computing speed of single CPU cores barely increases, such that performance gains are mostly due to increasing parallelism: number of compute nodes, number of CPU cores per node, graphics cards, and vectorization. Correspondingly, parallelization of algorithms such as PDE solvers for initial value problems is constantly gaining importance. In addition to established spatial domain decomposition methods [33], the surprisingly old idea of parallelization in time has seen growing interest in the last decade [12].

One of the prototypical parallel-in-time algorithms for initial value problems is the parareal scheme [23],

computing  $N$  subtrajectories in parallel by an exact or fine propagator and transporting the resulting jumps over the whole time horizon by means of a suitable, sequential fast or coarse propagator, the latter being indispensable for efficiency. Explicit error estimates guarantee  $r$ -linear [4, 8, 13] or  $q$ -linear [15, 36] convergence rates.

As after  $N$  steps the exact solution has been computed, both in sequential and in parareal schemes, the parallel efficiency is bounded by  $1/J_h$ , the inverse of the number of iterations needed. A fast convergence within a small number of iterations independent of  $N$  usually requires a rather accurate and hence expensive coarse propagator, which in turn limits parallel efficiency.

A step towards higher parallel efficiency are hybrid parareal methods computing the fine propagator in an iterative way. This allows to interleave the parareal iteration with each fine propagator iteration and hence to perform the global transport of corrections more often. Examples for iterative fine propagator schemes used in hybrid parareal methods are multigrid solvers for implicit Runge-Kutta methods [25] or spectral deferred correction methods [9, 34] for solving higher order collocation systems [7, 10, 26]. Convergence theory for hybrid parareal schemes is less well covered, see, e.g., [6, 24, 25].

Apart from convergence speed, the repeated communication necessary in parareal schemes for large scale problems such as discretized partial differential equations affects wall clock time and therefore parallel efficiency. Its impact has been reduced by adaptive work scheduling [3, 27] and pipelining computations, such that communication occurs mostly parallel to computation and both transmission time and latency are hidden to some extent [7, 10], even though this appears to be difficult to achieve in practice with current message passing interface (MPI) implementations. Hiding communica-

tion behind computation is also difficult because what is called computation might be a spatially parallelized domain decomposition solver, performing inter-node communication and possibly saturating the communication network on its own. The impact of communication is moreover growing, not only because of increasing parallelism, but also because computing power tends to grow faster than communication bandwidth [20].

Approaches for reducing communication include Nievergelt type parallel-in-time integration [5] mainly for linear equations, and the concept of guided simulations in case several similar equations are to be solved [32], e.g., for parameter studies. Especially for hyperbolic equations, the so called swept rule [1, 2] can be used to reduce not the amount of data transmitted, but the number of required messages, which is of interest in systems with high latency.

Besides methods for such specific problem settings, compression of MPI messages has gained interest in the last decade, aiming at methods which can be used regardless of the actual application. Mostly, lossless compression algorithms are considered in order not to spoil the results of computations, such that only a small reduction in size can be achieved. Nevertheless, reductions of computation times are still achieved [11, 19, 28]. Let us also note that data compression may also reduce the energy spent on communication – an interesting option for HPC systems approaching the power wall [22, 29, 31].

Here, we explore the reduction of communication in hybrid parareal schemes by lossy data compression. On the one hand, lossy compression, if applicable, achieves higher compression factors than lossless compression, e.g., for audio (MP3) and images (JPEG). On the other hand, it affects the convergence of the parallel-in-time algorithm. We estimate the influence of the quantization error onto convergence, such that a suitable tolerance with only minor impact can be chosen. In the course of estimating the effect of inexact communication, we also derive a rigorous, though not sharp, error bound for hybrid parareal methods. For estimating compression efficiency and in the numerical tests we employ transform coding tailored towards finite element coefficient vectors [16, 35].

In Section 2 we specify the abstract problem setting and state the general assumptions. For comparison purposes, we give a sequential iterative time stepping scheme as a reference in Section 3, using the same stationary iteration as will be employed for the hybrid parareal method in Section 4. Besides an a priori error bound that takes inexact communication into account, parallel efficiency is estimated. Section 5 briefly recalls lossy compression before investigating its expected im-

act on parallel efficiency in several situations. Finally, actual computation results are presented in Section 6.

## 2 Problem setting

Let us begin with stating the abstract problem to be solved and introducing the associated notation. We are interested in numerically solving the initial value problem

$$\dot{u} = f(u), \quad u(0) = v_*^0$$

for given starting value  $v_*^0$  on the global time interval  $[0, T]$ , where we assume that the exact solution  $u \in C([0, T], V)$  exists and assumes values in some Hilbert space  $V$ . Without loss of generality, we restrict the attention to autonomous problems.

For a time-parallel solution, the global interval is subdivided into  $N$  equidistant local time intervals  $I^n = [nh, (n+1)h]$ ,  $n = 0, \dots, N-1$ , with length  $h = T/N$ . On each local interval  $I^n$ , the solution subtrajectory  $u^n = u|_{I^n}$  is approximated by an element of a subspace  $U^n \subset C(I^n, V)$  equipped with the max-norm  $\|u^n\| := \max_{t \in I^n} \|u^n(t)\|$ . Furthermore, assume that at least affine functions of time are contained in  $U^n$ , i.e.  $\mathbb{P}_1(I^n, V) \subset U^n$ , with  $\mathbb{P}_k$  denoting the space of polynomials of order up to  $k$ .

Abstracting from the details of time stepping, we assume the availability of convergent fixed point iterations in terms of  $F^n(u; v)$ ,  $F^n : U^n \times V \rightarrow U^n$  for solving the initial value problems  $\dot{u} = f(u)$ ,  $u(nh) = v$  on  $I^n$  for initial value  $v \in V$ . We accept the fixed point  $u_*^n(v) := F^n(u_*^n(v); v)$  as a function of the initial value  $v$  as the “exact” solution on  $I^n$ , even if it is only a particular approximation, e.g., a collocation solution. Concrete examples of such iterative solvers are spectral deferred correction methods converging towards the collocation solution, where  $U^n = \mathbb{P}_k(I^n, V)$ , or multigrid V-cycles converging towards an implicit Euler step, where  $V$  is a finite element space and  $U^n(I^n, V) = \mathbb{P}_1(I^n, V)$  is affine in time. The global trajectory  $u_*$  over the global interval  $[0, T]$  is then defined by chaining the local fixed points, i.e.  $u_*|_{I^n} = u_*^n(v_*^n)$  via the continuity condition  $v_*^{n+1} = u_*^n((n+1)h)$ ,  $n = 0, \dots, N-1$ . We simply write  $u_*^n$  for  $u_*^n(v_*)$  if no ambiguity arises.

In other words,  $u_j^n$  denotes the approximate subtrajectory on the  $n$ -th time interval at iteration  $j$ , defined in terms of the initial value  $v_j^n$  and the previous iterate  $u_{j-1}^n$ . The subscript  $*$  always indicates the limit point for  $j \rightarrow \infty$ .

**Assumption 1** Let the exact evolution have a Lipschitz constant of  $L \geq 1$ , i.e.  $\|u_*^n(\hat{v}) - u_*^n(v)\| \leq L\|\hat{v} - v\|$  for arbitrary  $\hat{v}, v \in V$ , the fixed point iterations  $F^n$

have a uniform contraction factor  $0 < \rho < 1$ , and the right hand side be bounded such that  $\|u_*^n(v)((n+1)h) - v\| \leq ch$ . Moreover, assume there is  $K < \infty$  such that  $\|u_*^n(v) - u_*^n(\hat{v}) - (v - \hat{v})\| \leq \eta\|v - \hat{v}\|$  with  $\eta := hK \exp(hK)$  holds for any  $v$  and  $\hat{v}$ .

*Remark 1* Already from the choice of the max-norm on  $U^n$ , the bound  $L \geq 1$  is implied. This excludes the proper treatment of purely dissipative problems, which, however, are anyways not particularly interesting in the autonomous setting.

Note that usually both  $L$  and  $\rho$  depend on  $h$ , which is important to keep in mind when looking at strong scalability with  $T = \text{const}$  and growing  $N$ . The last assumption is natural as it is a direct result for any reasonable discretization defining  $u_*^n$  if the right hand side  $f$  is Lipschitz continuous due to the following lemma.

**Lemma 1** *Assume that  $u(v) \in C([0, T], V)$  satisfies*

$$\dot{u}(v)(t) = f(u(v)(t)) \quad \text{on } [0, h]$$

*for any initial value  $v$  and that the right hand side  $f$  is Lipschitz continuous with constant  $K$ . Then,*

$$\|u(v) - u(\hat{v}) - (v - \hat{v})\| \leq \eta\|v - \hat{v}\|, \quad \eta := hK \exp(hK)$$

*Proof* Let  $\xi(t) := (u(v) - u(\hat{v}))(t) - (v - \hat{v})$ . Then,  $\xi$  satisfies

$$\begin{aligned} \|\xi(t)\| &= \left\| \int_{\tau=0}^t (f(u(v)(\tau)) - f(u(\hat{v})(\tau))) d\tau \right\| \\ &\leq \int_{\tau=0}^t K \|u(v)(\tau) - u(\hat{v})(\tau)\| d\tau \\ &\leq \int_{\tau=0}^t K (\|\xi(\tau)\| + \|v - \hat{v}\|) d\tau \\ &= tK \|v - \hat{v}\| + \int_{\tau=0}^t K \|\xi(\tau)\| d\tau. \end{aligned}$$

Gronwall's inequality now yields

$$\|\xi(t)\| \leq tK \|v - \hat{v}\| \exp(tK) \leq hK \exp(hK) \|v - \hat{v}\|$$

and thus the claim.  $\square$

### 3 Sequential reference

Judging the performance and efficiency bounds of hybrid parareal schemes requires the comparison with a suitable reference. This will be provided by the obvious method to compute a complete trajectory within the same problem setting. That is the sequential approach of stepping through the intervals one after the other, performing a constant number  $J$  of fixed point iterations on each interval:

Algorithm 1 Sequential iteration

(i) given starting value

$$v_j^0 = v_*^0 \quad \text{for } j = 0, \dots, J$$

(ii) constant initialization

$$u_0^n(t) = v_0^n \quad \text{for } n = 0, \dots, N-1, t \in I^n$$

(iii) approximate integration

$$u_j^n = F^n(u_{j-1}^n; v_j^n)$$

$$\text{for } n = 0, \dots, N-1, j = 1, \dots, J$$

(iv) continuity

$$v_j^n = u_j^{n-1}(nh)$$

$$\text{for } n = 1, \dots, N-1, j = 0, \dots, J$$

Let us introduce the function  $\nu : [1, \infty[ \rightarrow \mathbb{R}$  by

$$\nu_n(x) := \sum_{i=0}^{n-1} x^i \tag{1}$$

and gather some of its elemental properties for later use.

**Lemma 2**  *$\nu_n(x)$  is monotonically increasing in  $x$  and bounded by*

$$x^n \leq \nu_{n+1}(x) \leq (n+1)x^n.$$

*Moreover,  $1 + x\nu_n(x) = \nu_{n+1}(x)$  holds.*

With that, we derive an error estimate for the sequential Algorithm 1 defined above.

**Theorem 1** *Let Assumption 1 hold. Then the total error  $e_j^n$  of the sequential iterative Algorithm 1 with  $J$  iterations in each interval  $I^n$  is bounded by*

$$\|u_j^n - u_*^n\| \leq e_j^n := ch\rho^j \nu_{i+1}(L). \tag{2}$$

*Proof* By the triangle inequality and Assumption 1 we obtain Lady Windermere's fan as

$$\begin{aligned} \|u_j^n - u_*^n\| &\leq \|u_j^n - u_*^n(v_j^n)\| + \|u_*^n(v_j^n) - u_*^n\| \\ &\leq \rho^J \|u_0^n - u_*^n(v_j^n)\| + L \|v_j^n - v_*^n\| \\ &\leq \rho^J ch + L \|u_j^{n-1} - u_*^{n-1}\| \\ &\leq ch\rho^J + Le_j^{n-1} \\ &= ch\rho^J (1 + L\nu_n(L)). \end{aligned}$$

Applying Lemma 2 and induction over  $n$  yields the claim.  $\square$

Assume we ask for an accuracy of TOL relative to the condition of the problem, i.e.

$$\|u_j^n - u_*^n\| \leq \text{TOL } ch\nu_{i+1}(L) \quad \text{for } n = 0, \dots, N-1. \tag{3}$$

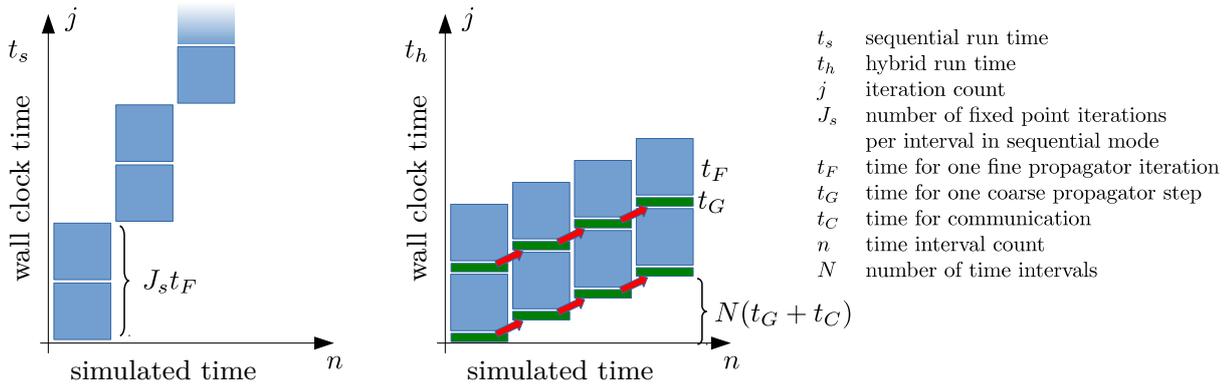


Fig. 1: Schematic representation of time stepping with iterative schemes. The number  $J_s = 2$  of iterations is deliberately chosen for illustration purposes only. *Left*: Sequential method. *Right*: Hybrid parareal scheme.

Then the number of iterations required by Algorithm 1 to reach this accuracy is at most

$$J_s \leq \left\lceil \frac{\log \text{TOL}}{\log \rho} \right\rceil. \quad (4)$$

Let  $t_F$  denote the computing time for a single evaluation of  $F^n$ , e.g., for one SDC sweep or a single V-cycle, assumed to be the same for all  $n$ . The total time for computing the final value  $u_{J_s}^{N-1}$  is bounded by  $t_s \leq N J_s t_F$ , see Fig. 1, left.

#### 4 An inexact hybrid parareal algorithm

The hybrid parareal scheme interleaves the time stepping with the stationary iteration on each local interval  $I^n$ , propagating the result of each iteration on to the next interval. As passing the information on by only one interval per iteration leads to slow convergence if  $N$  is large, sequential but fast coarse propagators  $G^n : V \rightarrow V$  are used to distribute the correction quickly through the remaining time intervals, see Fig. 1, right.

The analysis follows the line of [13], but includes the hybrid parareal fixed point iteration and considers the communication of corrections of the initial value from one time interval to the next to be subject to errors due to lossy compression. The modification of the initial value is represented by communication operators  $C_j^n : V \rightarrow V$ .

#### Algorithm 2 Hybrid parallel iteration

(i) given starting value

$$v_j^0 = v_*^0 \quad \text{for } j = 0, \dots, J$$

(ii) coarse propagation

$$v_0^n = G^{n-1}(v_0^{n-1}) \quad \text{for } n = 1, \dots, N$$

(iii) piecewise linear initialization

$$u_0^n(t) = v_0^n + (t/h - n)(v_0^{n+1} - v_0^n) \\ \text{for } n = 0, \dots, N - 1$$

(iv) approximate integration

$$u_j^n = F^n(u_{j-1}^n + v_j^n - v_{j-1}^n; v_j^n) \\ \text{for } n = 0, \dots, N - 1, j = 1, \dots, J$$

(v) continuity

$$v_j^n = v_{j-1}^n + C_j^n \left( u_{j-1}^{n-1}(nh) - v_{j-1}^n \right. \\ \left. + G^{n-1}(v_{j-1}^{n-1}) - G^{n-1}(v_{j-1}^{n-1}) \right) \\ \text{for } n = 1, \dots, N - 1, j = 1, \dots, J$$

In contrast to [24] we do not assume the fixed point iterations on each interval to converge unaffected by changing initial values, but treat this perturbation explicitly and consider inexact communication as well.

**Theorem 2** *In addition to Assumption 1, let the coarse propagators  $G^n$  satisfy*

$$\|u_0^n - u_*^n(v_0^n)\| \leq ch\gamma_0, \quad (5)$$

$$\|(u_*^n(v) - u_*^n(\hat{v}))((n+1)h) - (G^n(v) - G^n(\hat{v}))\| \\ \leq \gamma \|v - \hat{v}\|, \quad (6)$$

and

$$\|G^n(v) - G^n(\hat{v})\| \leq L \|v - \hat{v}\| \quad (7)$$

for arbitrary  $v, \hat{v} \in V$  and some  $\gamma_0, \gamma < \infty$ . Let the communication error be bounded by

$$\|v - C_j^n(v)\| \leq \Delta_C \|v\|$$

for some  $\Delta_C < \infty$ . Then the error estimate

$$\|v_j^n - v_*^n\| \leq \delta_j^n := ch\gamma_0\alpha\rho^{j-1}\nu_{n+1}(\beta_j) \quad (8)$$

holds with  $\alpha = \frac{1+\Delta_C}{1-\Delta_C/\rho}$  and

$$\beta_j = \alpha \left( \frac{\gamma}{\rho} + L + \eta(1 + \rho^{-1})j \right).$$

Before proving the theorem, let us stress that in general  $\gamma$  in (6) is of order one unless the coarse propagator has a consistency order of at least 0, which is sufficient for  $\gamma = \mathcal{O}(h)$ . This is, however, *not* satisfied by the common realization of coarse propagators acting on coarser grids in PDE problems. But it is necessary to guarantee fast convergence for large  $N$ , since otherwise correction components that cannot be represented on the coarse grid and hence not be propagated by  $G$  are handed on by just one interval per iteration. The related impact of insufficient coarse propagator accuracy in time has been investigated in [30] for advection-diffusion equations. The practical success of coarse grid propagators is due to the usually small amplitude of such error components and a quick reduction of high frequent error components in dissipative systems.

*Proof* In addition to (8) we will prove

$$\|u_j^n - u_*^n(v_j^n)\| \leq \epsilon_j^n := ch\gamma_0\rho^j(1 + \alpha\eta(1 + \rho^{-1})j\nu_{n+1}(\beta_j)).$$

Starting the induction at  $n = 0$  yields

$$\|v_j^0 - v_*^0\| = 0 \leq \delta_j^0 \quad \text{and} \quad \|u_j^0 - u_*^0\| \leq ch\gamma_0\rho^j \leq \epsilon_j^0$$

due to linear fixed point contraction with contraction factor  $0 < \rho < 1$ . On the other hand,  $j = 0$  provides  $\|u_0^n - u_*^n(v_0^n)\| \leq ch\gamma_0 \leq \epsilon_0^n$  by (5) and

$$\begin{aligned} \|v_0^{n+1} - v_*^{n+1}\| &\leq \|u_0^n - u_*^n\| \\ &\leq \|u_0^n - u_*^n(v_0^n)\| + \|u_*^n(v_0^n) - u_*^n\| \\ &\leq ch\gamma_0 + L\|v_0^n - v_*^n\| \leq ch\gamma_0\nu_{n+1}(L) \leq \delta_0^{n+1}. \end{aligned}$$

For general  $n, j > 0$  we let  $\tilde{v}_{j+1}^{n+1} := u_j^n((n+1)h) + G^n(v_{j+1}^n) - G^n(v_j^n)$  and obtain

$$\begin{aligned} \|v_{j+1}^{n+1} - v_*^{n+1}\| &\leq \|v_j^{n+1} + C_{j+1}^{n+1}(\tilde{v}_{j+1}^{n+1} - v_j^{n+1}) - \tilde{v}_{j+1}^{n+1}\| \\ &\quad + \|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| \\ &\leq \Delta_C \|\tilde{v}_{j+1}^{n+1} - v_j^{n+1}\| + \|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| \\ &\leq \Delta_C (\|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| + \|v_*^{n+1} - v_j^{n+1}\|) \\ &\quad + \|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| \\ &\leq \Delta_C \|v_*^{n+1} - v_j^{n+1}\| \\ &\quad + (1 + \Delta_C)\|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\|. \end{aligned} \quad (9)$$

With the definitions of  $\tilde{v}_{j+1}^{n+1}$  and  $\epsilon_j^n$ , the last term in (9) can be bounded by

$$\begin{aligned} &\|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| \\ &= \|u_j^n((n+1)h) + G^n(v_{j+1}^n) \\ &\quad - G^n(v_j^n) - u_*^n(v_*^n)((n+1)h)\| \\ &\leq \|u_j^n - u_*^n(v_j^n)\| \\ &\quad + \|(u_*^n(v_j^n) - u_*^n(v_*^n))((n+1)h) \\ &\quad - (G^n(v_j^n) - G^n(v_*^n))\| + \|G^n(v_{j+1}^n) - G^n(v_*^n)\| \\ &\leq \epsilon_j^n + \gamma\|v_j^n - v_*^n\| + L\|v_{j+1}^n - v_*^n\|. \end{aligned}$$

Using the monotonicity of  $\nu_{n+1}$  to bound

$$\delta_j^n = \frac{\nu_{n+1}(\beta_j)}{\rho\nu_{n+1}(\beta_{j+1})}\delta_{j+1}^n \leq \frac{1}{\rho}\delta_{j+1}^n, \quad (10)$$

which is only asymptotically sharp, we can further estimate

$$\begin{aligned} &\epsilon_j^n + \gamma\|v_j^n - v_*^n\| + L\|v_{j+1}^n - v_*^n\| \\ &\leq \epsilon_j^n + \gamma\delta_j^n + L\delta_{j+1}^n \\ &\leq \epsilon_j^n + \left(\frac{\gamma}{\rho} + L\right)\delta_{j+1}^n \\ &\leq ch\gamma_0\rho^j \left(1 + \alpha\eta(1 + \rho^{-1})j\nu_{n+1}(\beta_j) \right. \\ &\quad \left. + \left(\frac{\gamma}{\rho} + L\right)\alpha\nu_{n+1}(\beta_{j+1})\right) \\ &\leq ch\gamma_0\rho^j \left(1 + \alpha \left(\eta(1 + \rho^{-1})j + \frac{\gamma}{\rho} + L\right)\nu_{n+1}(\beta_{j+1})\right) \\ &\leq ch\gamma_0\rho^j(1 + \beta_{j+1}\nu_{n+1}(\beta_{j+1})) \\ &= ch\gamma_0\rho^j\nu_{n+2}(\beta_{j+1}) \\ &\leq \frac{1}{\alpha}\delta_{j+1}^{n+1}. \end{aligned}$$

Inserting this into (9) yields

$$\begin{aligned} \|v_{j+1}^{n+1} - v_*^{n+1}\| &\leq \Delta_C\delta_j^{n+1} + (1 + \Delta_C)\|\tilde{v}_{j+1}^{n+1} - v_*^{n+1}\| \\ &\leq \frac{\Delta_C}{\rho}\delta_{j+1}^{n+1} + (1 + \Delta_C)\frac{1 - \Delta_C/\rho}{1 + \Delta_C}\delta_{j+1}^{n+1} \\ &\leq \delta_{j+1}^{n+1}. \end{aligned}$$

Moreover, due to the approximate integration by  $F^n$  in step (iv) of Algorithm 2, we have

$$\begin{aligned} &\|u_j^n - u(v_j^n)\| \\ &\leq \rho\|u_{j-1}^n + v_j^n - v_{j-1}^n - u_*^n(v_j^n)\| \\ &\leq \rho(\|u_{j-1}^n - u_*^n(v_{j-1}^n)\| \\ &\quad + \|u_*^n(v_{j-1}^n) - u_*^n(v_j^n) - (v_{j-1}^n - v_j^n)\|). \end{aligned}$$

Using  $\eta$  as given in Assumption 1 we get

$$\begin{aligned}
& \rho(\|u_{j-1}^n - u_*^n(v_{j-1}^n)\| \\
& \quad + \|u_*^n(v_{j-1}^n) - u_*^n(v_j^n) - (v_{j-1}^n - v_j^n)\|) \\
& \leq \rho(\epsilon_{j-1}^n + \eta\|v_j^n - v_{j-1}^n\|) \\
& \leq \rho(\epsilon_{j-1}^n + \eta(\|v_j^n - v_*^n\| + \|v_{j-1}^n - v_*^n\|)) \\
& \leq \rho(\epsilon_{j-1}^n + \eta(\delta_j^n + \delta_{j-1}^n)) \\
& \leq \rho(\epsilon_{j-1}^n + \eta(1 + \rho^{-1})\delta_j^n) \\
& \leq ch\gamma_0\rho^j(1 + \alpha\eta(1 + \rho^{-1})(j-1)\nu_{n+1}(\beta_{j-1}) \\
& \quad + \alpha\eta(1 + \rho^{-1})\nu_{n+1}(\beta_j)) \\
& \leq ch\gamma_0\rho^j(1 + \alpha\eta(1 + \rho^{-1})j\nu_{n+1}(\beta_j)) = \epsilon_j^n,
\end{aligned}$$

which completes the induction.  $\square$

The asymptotic convergence rate is  $\rho$ , as in the sequential SDC algorithm, independent of communication accuracy and fast propagator.

Let us investigate the error bound in a straightforward but nontrivial example situation.

*Example 1* We consider the harmonic oscillator

$$\dot{x} = y, \quad \dot{y} = -x$$

with initial value  $x = 0, y = 1$  and time horizon  $T = 3\pi$ , subdivided into  $N = 16$  intervals.

As a stationary fine propagator iteration we use SDC on a two-point Gauss collocation grid with an explicit Euler base method, converging towards a fourth order energy preserving collocation scheme. Consequently,  $L = 1$  and  $\eta = 0.58$  hold. As fast propagator we consider (i) the zero propagator  $G^n(v) = 0$  yielding just a parallel SDC method [18] and (ii) the explicit Euler propagator  $G^n(v) = v + hf(v)$ .

In that setting, the iterates and hence right hand sides are bounded in Assumption 1 with (i)  $c = 1186$  and (ii)  $c = 18$ , respectively, and the fast propagation errors can be bounded by (i)  $\gamma_0 = \gamma = 0.58$  and (ii)  $\gamma_0 = \gamma = 0.17$ . The observed SDC contraction rate is  $\rho = 0.20$ .

Resulting iteration errors and error bounds are depicted in Fig. 2. While the global error behavior is reproduced rather well by the bound (8), the bound is far from being sharp, in particular for larger  $n$ . On one hand this is due to the rather general assumptions on the stationary iteration, with the observed average case being much better than the bounded worst case. On the other hand, the estimates in the proof are not sharp, in particular (10). For that reason we provide a fitted error estimate as well, where the constants in (8) are obtained by a least squares fit to the actual errors. Of course, this reproduces the actual errors much better.

The impact of communication errors is simulated by adding normally distributed noise of relative standard deviation  $\Delta_C = 0.1$ , which amounts to less than four bits per coefficient. Comparing the results in the last row of Fig. 2 with the second row corresponding to  $\Delta_C = 0$  suggests that the impact of even such a large relative communication error on convergence is minor.

*Parallel efficiency.* Next we estimate the parallel efficiency of the hybrid parareal method based on the error bound (8). As this is overestimating the error, the efficiency estimates will be rather pessimistic.

For a relative accuracy of TOL as specified in (3), the number  $J_h$  of required iterations is bounded by

$$J_h \leq 1 + \frac{\log\left(\frac{\text{TOL}\nu_{N+1}(L)}{\alpha\gamma_0\nu_{N+1}(\beta_{J_h})}\right)}{\log\rho} \leq 1 + \frac{\log\left(\frac{\text{TOL}L^N}{\alpha\gamma_0(N+1)\beta_{J_h}^N}\right)}{\log\rho}.$$

Assuming  $L > 1$  and  $N$  rather large, this can be approximated by

$$\begin{aligned}
J_h & \approx 1 + \frac{\log\text{TOL}}{\log\rho} - \frac{N\log(\beta_{J_h}/L) + \log(\alpha\gamma_0(N+1))}{\log\rho} \\
& \approx 1 + J_s \\
& \quad - \frac{N\log\left(\alpha\left(\frac{\gamma}{L\rho} + 1 + \frac{\eta}{L}(1 + \rho^{-1})J_h\right)\right)}{\log\rho} \\
& \quad - \frac{\log(\alpha\gamma_0(N+1))}{\log\rho}.
\end{aligned}$$

With  $\alpha = (1 + \Delta_C)/(1 - \Delta_C/\rho)$  and  $\Delta_C \ll 1$ , we obtain  $\log\alpha \approx (1 + \rho^{-1})\Delta_C$  and thus

$$\begin{aligned}
J_h & \approx 1 + J_s - \frac{N\log\left(\alpha\left(\frac{\gamma}{L\rho} + 1 + \frac{\eta}{L}(1 + \rho^{-1})J_h\right)\right)}{\log\rho} \\
& \quad - \frac{(1 + \rho^{-1})\Delta_C + \log(\gamma_0(N+1))}{\log\rho} \\
& \leq 1 + J_s - \frac{N\left(\frac{\gamma}{L\rho} + \frac{\eta}{L}(1 + \rho^{-1})J_h\right)}{\log\rho} \\
& \quad - \frac{(N+1)(1 + \rho^{-1})\Delta_C + \log(\gamma_0(N+1))}{\log\rho}.
\end{aligned}$$

Thus,  $J_h$  grows only logarithmically with  $N$ , if both  $\Delta_C = \mathcal{O}(N^{-1})$  and

$$\frac{\gamma}{L\rho} + \frac{\eta}{L}(1 + \rho^{-1})J_h = \mathcal{O}(N^{-1})$$

hold. The former is rather easy to satisfy with a small, i.e. logarithmic, increase in communication time. The latter condition, however, requires in particular  $\gamma$  and  $\eta$  to be of order  $N^{-1}$ . As  $\gamma$  denotes the consistency error of the coarse propagator, e.g., an Euler step with  $\mathcal{O}(h^2)$ , and  $\eta$  is of order  $\mathcal{O}(h)$ , this means  $Nh = \mathcal{O}(1)$ . Actually,

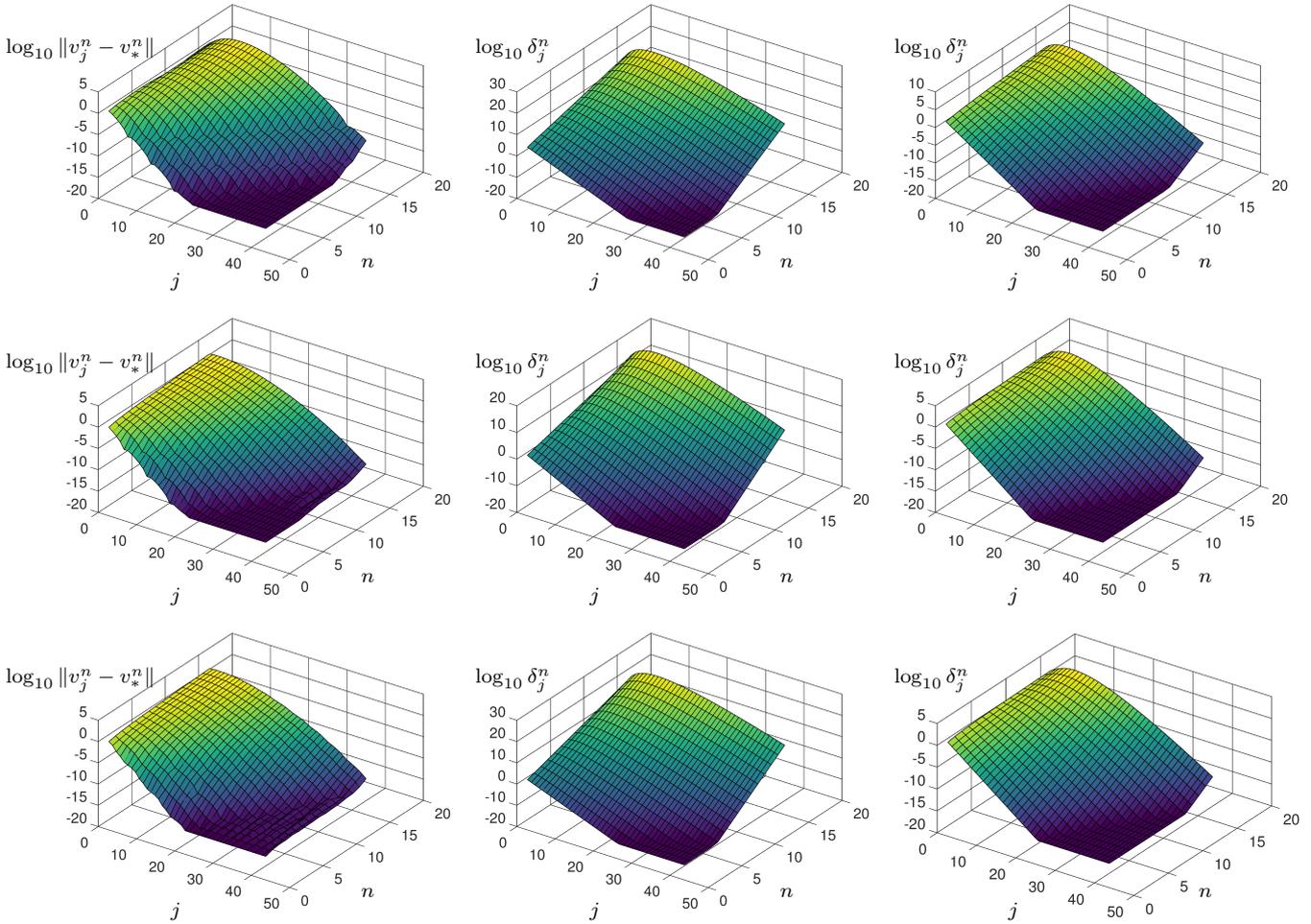


Fig. 2: Errors and bounds for Example 1 in logarithmic scale. *Left*: errors  $\|v_j^n - v_*^n\|$ . *Mid*: error bound (8). *Right*: error bound with constants estimated from actual errors. *Top*: zero fast propagator. *Mid*: Euler fast propagator. *Bottom*: Euler fast propagator with communication error  $\Delta_C = 0.1$ .

this restriction is to be expected, since the theory covers ill-conditioned and chaotic systems, where an extension of the integration interval cannot be expected to keep the iteration number constant.

As is apparent from Fig. 1, the total computation time is  $t_h = J_h(t_F + t_G) + (N - 1)(t_G + t_C)$ , which yields a parallel efficiency of

$$E_h = \frac{t_s}{N t_h} = \frac{J_s t_F}{J_h(t_F + t_G) + (N - 1)(t_G + t_C)}. \quad (11)$$

Obviously, a large efficiency close to one requires  $J_h \approx J_s$ , which in turn requires TOL very small, or both  $\Delta_C$  and  $h$  small. Additionally, efficiency requires  $t_G \ll t_F$ , and  $N(t_G + t_C) \ll J_h t_F$ . The former condition is rather easy to satisfy, in fact, using the same fixed point iteration for both fine and coarse propagators would push the efficiency only down to 0.5 due to the  $J_h(t_F + t_G)$  term in the denominator. The second condition is

much harder to satisfy, in particular for highly parallel computations with large  $N$ .

The relation of  $\Delta_c$  and  $t_C$  and their impact on parallel efficiency  $E_h$  is discussed in the next section.

## 5 Lossy data compression

First we recall lossy compression methods for coefficient vectors  $y \in \mathbb{R}^m$  of spatial discretizations, in particular of finite element approaches. The results are applicable as well to similar techniques like finite difference and finite volume schemes.

The simplest compression (rounding of coefficients to the required accuracy) provides a compression ratio

$$S = -\frac{\log_2 \Delta_C}{64} \quad (12)$$

relative to uncompressed double precision binary representation. Note that transform coding with hierarchical basis or wavelet transform and entropy coding yields even better results [35]. For the sake of simplicity, however, we will stick with (12).

For very small  $\Delta_C$ , the message size is usually bounded from above by the size of the raw binary representation of double accuracy floating point values, i.e.  $S \leq 1$ . Given a time  $t_T$  required to transmit the uncompressed data set and a latency  $t_L$  independent of  $\Delta_C$  that includes the time for encoding and decoding, if performed, as well as usual communication latencies, we arrive at a communication time  $t_C = t_L + St_T$ . Inserting this into (11) yields the parallel efficiency model

$$E_h = \frac{J_s t_F}{J_h(t_F + t_G) + (N - 1)(t_G + t_L - t_T \log_2 \Delta_C / 64)}, \quad (13)$$

which allows to investigate the impact of data compression. This impact and the optimal value of  $\Delta_C$  depends on the values of  $N, J_s, t_F, J_h, t_G, t_L$ , and  $t_T$  characterizing the problem to be solved, the coarse and fine propagators, and the computing system used.

In the following we will explore several scenarios in order to get an idea in which settings data compression may have a significant impact on parallel efficiency. Starting from a nominal setting with SDC of order six as stationary iteration, and first order Euler as fast propagator, other scenarios modify the parameters in a particular direction as outlined below.

**Nominal scenario.** This is a basic setting characterized by moderate parallelism, iteration counts that are typically observed in SDC methods for engineering tolerances, a coarse propagator time that could come from a single Euler step compared to a full SDC sweep, and communication times that are plausible for current standard hardware. Times are given relative to the fine propagator time  $t_F$ . Values for the nominal scenario are indicated by  $\hat{\cdot}$ :  $\hat{N} = 16$ ,  $\hat{L} = 1.5$ ,  $\widehat{\text{TOL}} = 10^{-6}$ ,  $\hat{\rho} = 0.2$ ,  $\hat{\gamma} = 0.05$ ,  $\hat{\gamma}_0 = 0.1$ ,  $\hat{\eta} = 0.05$ ,  $\hat{t}_F = 1$ ,  $\hat{t}_G = 0.1$ ,  $\hat{t}_T = 0.1$ . For the latency  $\hat{t}_L = t_{L,e/d} + t_{L,c}$ , we assume a contribution of  $t_{L,e/d} = 0.005$  from encoding/decoding if performed and the same amount  $t_{L,c} = 0.005$  from the communication system. With these parameters, (4), (8) and (11) yield estimates of  $J_s = 9$ ,  $J_h = 28$ , and  $E_h = 0.28$ , respectively.

**Different tolerances.** Aiming at a different tolerance  $\text{TOL} = s\widehat{\text{TOL}}$  for  $s \in [10^{-4}, 10^2]$  usually goes along with choosing a different order of collocation discretization and, at the same time, a different time step size. Smaller time steps tend to decrease  $\rho$ ,

whereas higher order SDC tends to increase  $\rho$ . For simplicity, we assume the contraction  $\rho = \hat{\rho}$  to be unaffected by  $s$ . In contrast, for a nominal order six we assume the time step  $h$  to scale like  $s^{1/6}$  and hence obtain  $L = 1 + (\hat{L} - 1)s^{1/6}$  and  $\eta = s^{1/6}\hat{\eta}$ . Moreover, the Euler consistency error depends on the time step as well, such that we assume  $\gamma_0 = s^{1/3}\hat{\gamma}_0$  and  $\gamma = s^{1/3}\hat{\gamma}$ .

**Scalability.** The natural mode for parallel-in-time integration is to have only time interval per processor, and hence scaling means in general weak scaling with growing time horizon  $T = s\hat{T}$  and  $N = s\hat{N}$ . Thousands of intervals have been used for time-parallel computations [14], such that we consider  $s \in [0.25, 512]$ . Except for  $N$ , no parameter entering the parallel efficiency estimate is assumed to change.

**Commodity network hardware.** Here we assume slower communication links, due to cheaper network infrastructure, which might be encountered in workstation clusters, cloud computing, or even DSL or WiFi connections. Slower communication affects in particular the bandwidth, i.e.  $t_C = s\hat{t}_C$ , but to some extent also latency  $t_L = \sqrt{s}t_{L,c} + t_{L,e/d}$ , with  $s \in [10^{-2}, 10]$ .

In each scenario, and for each value of the scaling parameter  $s$ , an optimal compression accuracy  $\Delta_C$  has been computed numerically, maximizing the parallel efficiency based on (8). Figs. 3 and 4 show the parallel efficiency of both the uncompressed ( $\Delta_C = 0$  and  $S = 1$ ) method and the lossy compression ( $\Delta_C = \text{opt}$ ,  $S = -\log_2 \Delta_C / 64$ ).

In every setup, the optimal quantization error is mostly in the range of  $10^{-3}$  to  $10^{-2}$ , and improves parallel efficiency compared to the uncompressed version. The effect, however, is rather small, predicted by theory to be less than 5%. For decreasing tolerance, the parallel efficiency grows as expected, see Fig. 3 left. However, the benefit of compression, that could be assumed to grow with larger tolerances, decreases. This is due to an actually increasing number of iterations in the parallel scheme due to larger step sizes affecting the effectivity of the fast propagator, see Fig. 3 right. The number of iterations in the sequential scheme, on the other hand, decreases as expected, and is virtually unaffected by the compression.

The theoretically predicted scalability is rather unsatisfactory, see Fig. 4 left, due to the exponential increase of  $\nu$  in  $N$  and the corresponding growth of  $J_h$  for growing number of time intervals. Correspondingly, the benefit of compression that is assumed to improve only the single sequential startup-phase becomes relatively negligible, even though in absolute numbers it increases linearly with  $N$ .

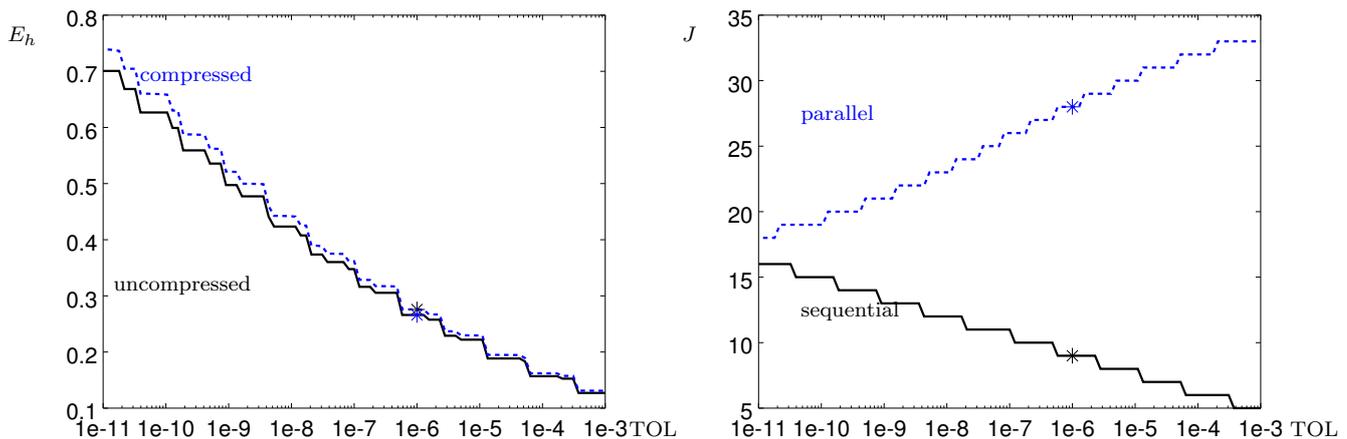


Fig. 3: Impact of lossy compression for varying tolerance as described in the *Different tolerances* scenario. The nominal setting is marked in both plots. *Left*: Parallel efficiency  $E_h$  as estimated in (11) and (13) is improved slightly by compression. *Right*: Iteration numbers  $J_h$  and  $J_s$  grow with the tolerance for the parallel scheme due to larger step sizes decreasing the accuracy of the fast propagator. In contrast, faster SDC contraction leads to fewer iterations in the sequential scheme.

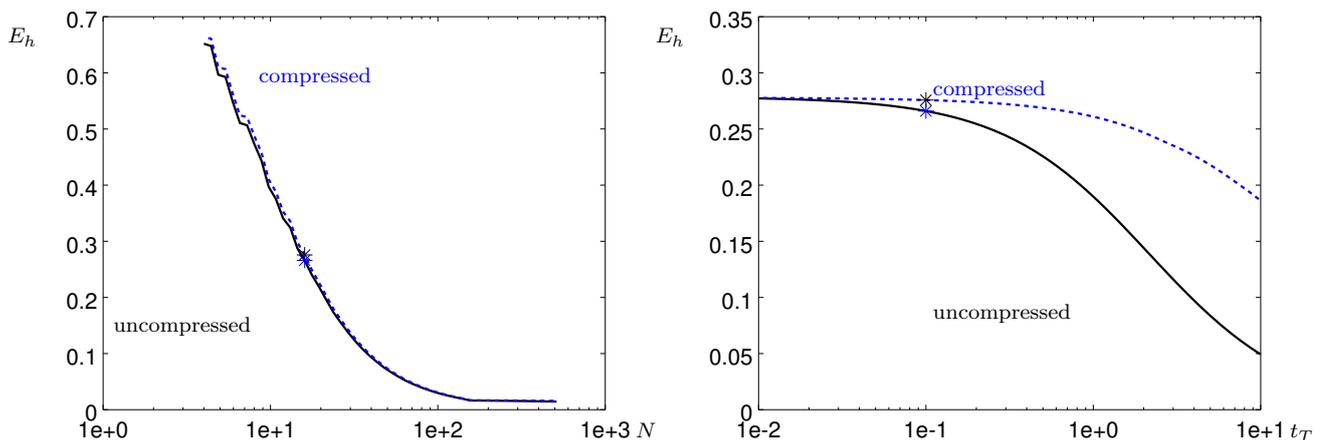


Fig. 4: Impact of lossy compression on parallel efficiency  $E_h$ . *Left*: Varying number  $N$  of time intervals. *Right*: Varying communication time  $t_T$ . The nominal setting is marked in both plots.

As expected, a varying bandwidth has a significantly larger effect on the compression benefit, see Fig. 4 right. For transmission times  $t_T$  that are at least three times as high as assumed in the nominal scenario, the parallel efficiency of the uncompressed scheme decreases rather quickly, while compression allows to maintain the efficiency for much longer communication times.

## 6 Computational Examples

Here we consider the inhomogeneous heat equation

$$\begin{aligned} u_t - \Delta u &= f && \text{in } \Omega \times (0, T) \\ \partial_\nu u &= 0 && \text{on } \partial\Omega \times (0, T) \\ u(\cdot, 0) &= 0 && \text{in } \Omega, \end{aligned} \quad (14)$$

with  $\Omega = (0, 1)^2$  and source term

$$\begin{aligned} f(x, t) &= (8\pi^2 + \exp(-t) \\ &\quad - 8\pi^2 \exp(-t)) \cos(2\pi x_1) \cos(2\pi x_2). \end{aligned}$$

The analytical solution to this equation is given by

$$u(x, t) = (1 - \exp(-t)) \cos(2\pi x_1) \cos(2\pi x_2).$$

For the numerical solution we discretize in space first, using linear finite elements on a uniformly refined triangular mesh. The arising system of ODEs is solved using the Parallel Full Approximation Scheme in Space and Time (PFASST) [10] as an instance of hybrid parareal methods. For implementation we use PFASST++ [21] in combination with the finite element toolbox Kas-kade 7 [17].

## 6.1 Two-rank setup

First we consider a minimal setup with two time intervals distributed on two workstations connected by Ethernet, in order to measure communication times with as few disturbances as possible. We use a time interval size of 0.125, so  $T = 0.25$ . For the coarse propagator, we use SDC with 3 Gauss-Lobatto collocation nodes; the fine propagator is SDC on 5 Gauss-Lobatto nodes. In each case, there are 263 169 spatial degrees of freedom. Overall we perform 10 iterations. This setting allows to investigate the effect of compression on the runtimes of the method as well as on the accuracy, without being influenced too much by factors like shared access to network and compute resources.

In Table 1 we report the wall-clock times (averaged over 5 runs) for uncompressed and compressed communication with a prescribed relative tolerance of  $10^{-8}$ , leading to an overall compression factor of 3.7.

Note that in the PFASST algorithm, as well as in the PFASST++ library used here, communication is performed in the fine and coarse propagator separately, where the fine propagator can use interleaving of computation and communication, leading to very small communication times. The coarse propagator in contrast has to wait for the send/receive to finish before computations can continue. As it uses the same spatial discretization, communication times are more significant there. With the time required for compression being larger than the send/receive times of the fine propagator, this suggests to use compression only for the coarse propagators communication.

The achieved accuracy for this setting, using a quantization tolerance of  $10^{-8}$ , is shown in Table 2. Using compression only for the coarse propagator leads to similar accuracy as in the uncompressed case, using compression in fine and coarse propagators leads to a slightly larger residual. The overall communication times were reduced to 83% and 81%, respectively.

Despite the simplified setup, computation times vary between different runs, making them hard to compare. Besides the shared access to resources, there are other factors influencing the results. E.g., switching off compression for the fine propagator leads to longer wait times for the coarse propagators communication, thus increasing the communication time there.

## 6.2 More than two ranks

Here we consider  $T = 1$  and use  $N = 16$  time intervals and processors, distributed across different workstations equipped with Intel Xeon E3-1245 v5 CPUs clocked at 3.5GHz and connected by gigabit Ethernet.

On each macro time interval of size 0.0625 we again use SDC with 3 and 5 Gauss-Lobatto quadrature nodes as coarse and fine propagator. For each, 66 049 spatial degrees of freedom were used.

We stop the iterations on time interval  $n$  when the maximum norm of the correction on this interval is below  $10^{-6}$ , and the previous time step  $n - 1$  is already converged. This leads to 12 iterations on the last time step. Using compression with relative quantization tolerance of  $10^{-8}$  (leading to a rather small compression factor of about 2) takes the same number of iterations, but reduces the overall communication time (for all processors) from 160.1s to 73.9s (time per sweep: 0.46s and 0.21s respectively), a reduction by 53.9%. In this test, we did not use compression for sending the very first update to the initial value, as the quantization error there had larger impact on convergence and communication times than the following incremental updates. The overall runtime of the PFASST algorithm is reduced by 10%, from 103.9s to 93.2s. In both cases we get a relative  $L^2([0, T] \times \Omega)$ -error of  $2.1 \cdot 10^{-4}$  in the final iterate compared to the analytical solution. Note that the stopping criterion in this experiment differs from the standard stopping criterion of the PFASST method, where usually the norm of the residual of the collocation formulation

$$r(t) := u(0) - \left( u(t) - \int_0^t f(u(\tau)) d\tau \right)$$

is used. This is not feasible here: Using lossy compression for sending updates to initial values typically leads to spatially oscillating compression errors in the approximate solution, which are amplified by the Laplace operator in the original equation. In the two-rank test case above, the computation was stopped after a fixed number of sweeps and not iterated until convergence; as the corrections and residuals are still large enough, this error amplification is not an issue there.

Increasing the compression factor did not yield any further improvements for the run times. One reason for this might be that communication and computation is not as well interleaved as before, leading to longer wait times in the communication. Note that not only computation times but also communication times strongly depend on algorithmic parameters like the number of quadrature nodes for the coarse and fine propagators. The interplay of compression, algorithmic options and other ingredients of PFASST, like the FAS-correction, requires further investigation to facilitate an optimal choice of these parameters.

Moreover, as the tests are run on a standard TCP/IP over Ethernet connection not optimized for high performance computing, there are various sources influencing

rank	task	uncompressed	compress fine and coarse	compress only coarse
0	sweeps fine	104.7	104.6	105.4
	sweeps coarse	52.3	52.2	52.7
	comm fine	0.002	1.3	0.002
	comm coarse	11.1	8.9	9.1
1	sweeps fine	105.9	106.3	106.1
	sweeps coarse	53.0	53.1	52.9
	comm fine	0.002	0.5	0.002
	comm coarse	3.6	1.5	2.8

Table 1: Wall clock times in seconds for the two-rank setup for the individual steps.

rank	uncompressed	compress fine and coarse	compress only coarse
0	$9.1 \cdot 10^{-6}$	$9.1 \cdot 10^{-6}$	$9.1 \cdot 10^{-6}$
1	$1.7 \cdot 10^{-4}$	$2.5 \cdot 10^{-5}$	$1.8 \cdot 10^{-5}$

Table 2: Final relative residuals.

communication times besides shared access to the network, for example configuration of switches or firewalls, varying Ethernet adapters, or other, unknown ones. In our test runs, communication times varied from workstation to workstation, despite having the same processors, memory, and software. While the available network bandwidth can be measured to some extent, in the actual computations it was drastically lower than predicted. In settings like this, or even more severely in cloud computing, robustness with respect to the bandwidth is important; in this respect using lossy compression for communication can be quite beneficial.

Considering scaling experiments on a state of the art compute cluster, we fixed a final time  $T = 2$  and macro time step 0.0625 and used PFASST (as described above) to solve the equation on 1, 8, 16 and 32 processors on the HLRN supercomputer (Cray XC30/40 with Aries interconnect of the North-German Supercomputing Alliance, [www.hlrn.de](http://www.hlrn.de)). The sequential version required a total of 348 iterations, an average of 10.875 per time interval. In the parallel case, on the last rank an overall 50, 25, 12 iterations were required for 8, 16, 32 processors (each computing 4, 2, 1 macro time intervals), respectively. Using compressed communication with a quantization tolerance of  $10^{-8}$  did not influence the number of iterations; also the overall computation time was only influenced insignificantly. This indicates that in this experiment communication bandwidth and latency are not an issue. In realistic, large-scale applications, using domain decomposition in space as well as time-parallel integration, we expect this to be different, as memory bandwidth would already be saturated due to communication required by the spatial domain decomposition.

## Conclusions

A complete and rather general convergence theory for hybrid parareal schemes with inexact communication has been presented. While the error estimates are not in the least sharp, the qualitative error behavior is captured quite well and allows to estimate the impact of inexact communication due to lossy compression on the iteration count and hence on the parallel efficiency. The theoretical results, supported by computational experiments, indicate that lossy compression improves the efficiency, but the amount of improvement depends on the setup. Compression is particularly effective if the available communication bandwidth is small. This includes commodity systems, clusters and cloud computing, where bandwidth is often low or varying, but also high performance systems where the communication network is already saturated due to concurrent communication going on.

*Acknowledgement.* Partial funding by BMBF Project SOAK and DFG Project WE 2937/6-1 is gratefully acknowledged. The authors would also like to thank Th. Steinke, F. Wende, A. Kammeyer and the North-German Supercomputing Alliance (HLRN) for supporting the numerical experiments.

## References

1. M. Alhubail, Q. Wang, and J. Williams. The swept rule for breaking the latency barrier in time advancing two-dimensional PDEs. Preprint arXiv:1602.07558, 2016.
2. Maitham Alhubail and Qiqi Wang. The swept rule for breaking the latency barrier in time advancing PDEs. *J. Comput. Phys.*, 307:110–121, 2016.
3. E. Aubanel. Scheduling of tasks in the parareal algorithm. *Parallel Computing*, 37:172–182, 2011.

4. G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Proceedings of DD15*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 425–432. Springer, 2004.
5. A.T. Barker. A minimal communication approach to parallel time integration. *Int. J. Comp. Math.*, 91(3):601–615, 2014.
6. M. Bolten, D. Moser, and R. Speck. Asymptotic convergence of the parallel full approximation scheme in space and time for linear problems. Preprint arXiv:1703.07120, 2017.
7. A.J. Christlieb, C.B. Macdonald, and B.W. Ong. Parallel high-order integrators. *SIAM J. Sci. Comput.*, 32(2):818–835, 2010.
8. M. Duarte, M. Massot, and S. Descombes. Parareal operator splitting techniques for multi-scale reaction waves: Numerical analysis and strategies. *M2AN*, 45(5):825–852, 2011.
9. A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT*, 40(2):241–266, 2000.
10. M. Emmett and M.L. Minion. Toward an efficient parallel in time method for partial differential equations. *Comm. Appl. Math. Comp. Sci.*, 7(1):105–132, 2012.
11. R. Filgueira, D.E. Singh, J. Carretero, A. Calderón, and F. García. Adaptive-compi: Enhancing MPI-based applications’ performance and scalability by using adaptive compression. *The International Journal of High Performance Computing Applications*, 25(1):93–114, 2011.
12. M. Gander. 50 years of time parallel time integration. In T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, editors, *Multiple Shooting and Time Domain Decomposition Methods*, volume 9 of *Contributions in Mathematical and Computational Sciences*, pages 69–113. Springer, 2015.
13. M.J. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. In U. Langer, M. Discacciati, D.E. Keyes, O.B. Widlund, and W. Zulehner, editors, *Domain Decomposition Methods in Science and Engineering XVII*, pages 45–56. Springer Berlin Heidelberg, 2008.
14. M.J. Gander and M. Neumüller. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM J. Sci. Comput.*, 38(4):A2173–A2208, 2016.
15. M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.
16. S. Götschel and Weiser. M. Lossy compression for PDE-constrained optimization: Adaptive error control. *Comput. Optim. Appl.*, 62:131–155, 2015.
17. S. Götschel, M. Weiser, and A. Schiela. Solving optimal control problems with the Kaskade 7 finite element toolbox. In A. Dedner, B. Flemisch, and R. Klöforn, editors, *Advances in DUNE*, pages 101–112. Springer, 2012.
18. D. Guibert and D. Tromeur-Dervout. Parallel deferred correction method for CFD problems. In J.-H. Kwon, J. Périaux, P. Fox, N. Satofuka, and A. Ecer, editors, *Parallel Computational Fluid Dynamics 2006: Parallel Computing and its Applications*, pages 131–136, 2007.
19. J. Ke, M. Burtscher, and E. Speight. Runtime compression of MPI messages to improve the performance and scalability of parallel applications. In *Supercomputing, 2004. Proceedings of the ACM/IEEE SC2004 Conference*, page 59, 2004.
20. S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco. Gpus and the future of parallel computing. *IEEE Micro*, 31(5):7–17, 2011.
21. T. Klatt, M. Emmett, D. Ruprecht, R. Speck, and S. Terzi. PFAST+. <http://www.parallelintime.org/codes/pfast.html>, 2015. Retrieved: 16 50, May 04, 2017 (GMT).
22. S. Leyffer, S.M. Wild, M. Fagan, M. Snir, K. Palem, K. Yoshii, and H. Finkel. Doing Moore with less - Leapfrogging Moore’s law with inexactness for supercomputing. *CoRR*, abs/1610.02606, 2016.
23. J.-L. Lions, Y. Maday, and G. Turinici. A parareal in time discretization of pdes. *C.R. Acad. Sci. Paris, Serie I*, 332:661–668, 2001.
24. J. Liu, Y. Wang, and R. Li. A hybrid algorithm based on optimal quadratic spline collocation and parareal deferred correction for parabolic PDEs. *Math. Probl. Eng.*, 2016. Article ID 6943079.
25. E. McDonald and A.J. Wathen. A simple proposal for parallel computation over time of an evolutionary process with implicit time stepping. In *Proceedings of the ENUMATH 2015*, Lecture Notes in Computational Science and Engineering. Springer, to appear.
26. M.L. Minion. A hybrid parareal spectral deferred corrections method. *Comm. Appl. Math. Comp. Sci.*, 5(2):265–301, 2010.
27. A.S. Nielsen, G. Brunner, and J.S. Hesthaven. Communication-aware adaptive parareal with application to a nonlinear hyperbolic system of partial differential equations. EPFL-ARTICLE 228189, EPFL, 2017.
28. A. Núñez, R. Filgueira, and M.G. Merayo. SANComSim: A scalable, adaptive and non-intrusive framework to optimize performance in computational science applications. *Procedia Computer Science*, 18:230 – 239, 2013.
29. T. Palmer. Build imprecise supercomputers. *Nature*, 526:32–33, 2015.
30. D. Ruprecht. Wave propagation characteristics of parareal. Preprint arXiv:1701.01359v1 [math.NA], 2017.
31. K.P. Saravanan, P.M. Carpenter, and A. Ramirez. A performance perspective on energy efficient hpc links. In *ICS ’14 Proceedings of the 28th ACM international conference on Supercomputing*, pages 313–322, 2014.
32. A. Srinivasan and N. Chandra. Latency tolerance through parallelization of time in scientific applications. *Parallel Computing*, 31:777–796, 2005.
33. A. Toselli and O.B. Widlund. *Domain Decomposition Methods – Algorithms and Theory*, volume 34 of *Computational Mathematics*. Springer, 2005.
34. M. Weiser. Faster SDC convergence on non-equidistant grids by DIRK sweeps. *BIT Numerical Mathematics*, 55(4):1219–1241, 2015.
35. M. Weiser and S. Götschel. State trajectory compression for optimal control with parabolic PDEs. *SIAM J. Sci. Comput.*, 34(1):A161–A184, 2012.
36. S.-L. Wu and T. Zhou. Convergence analysis for three parareal solvers. *SIAM J. Sci. Comput.*, 37(2):A970–A992, 2015.