A Nonlinear ParaExp Algorithm

Martin J. Gander, Stefan Güttel, and Madalina Petcu

1 Derivation of the Nonlinear ParaExp Algorithm

Time parallelization has a long history, see [1] and references therein. The parallel speedup obtained is in general not as good as with space parallelization, especially for hyperbolic problems. A notable exception are waveform relaxation-type methods [3, 4], which in the hyperbolic case are related to the more recent tent-pitching approach [6], and the ParaExp algorithm [7, 9] based on Krylov methods, which is however restricted to linear problems. For an application in a nonlinear context, see [10], and for a different approach using Krylov information, see [8]. Here we propose and analyze a variant of the ParaExp algorithm for the nonlinear initial value problem

$$\mathbf{u}'(t) = A\mathbf{u}(t) + B\big(\mathbf{u}(t)\big) + \mathbf{g}(t), \quad t \in [0,T], \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{1.1}$$

with $A \in \mathbb{C}^{m\times m}$, $B : \mathbb{C}^m \to \mathbb{C}^m$ a nonlinear operator, $\mathbf{g} : [0,T] \to \mathbb{C}^m$ a source function, and $\mathbf{u} : [0,T] \to \mathbb{C}^m$ the sought solution. Throughout this note we assume that all stated initial value problems have unique solutions. For the ParaExp algorithm, the

M. J. Gander
Université de Genève, 2-4 rue du Lièvre, 1211 Genève, Switzerland,
e-mail: Martin.Gander@unige.ch

S. Güttel
School of Mathematics, The University of Manchester, United Kingdom,
e-mail: stefan.guettel@manchester.ac.uk

M. Petcu
Laboratoire de Mathématiques, Université de Poitiers, France,
The Institute of Statistics and Applied Mathematics of the Romanian Academy,
The Institute of Mathematics of the Romanian Academy, Bucharest, Romania,
e-mail: Madalina.Petcu@math.univ-poitiers.fr

time interval $[0,T]$ is partitioned into N subintervals $[T_{n-1},T_n]$ with $n=1,\ldots,N$, and a direct application of this algorithm to the nonlinear problem (1.1) gives

Step 1: Solve for $n \geq 1$ in parallel the nonlinear problems with zero initial data

$$\begin{aligned} \mathbf{v}_n'(t) &= A\mathbf{v}_n(t) + B\big(\mathbf{v}_n(t)\big) + \mathbf{g}(t), \quad t \in [T_{n-1},T_n], \\ \mathbf{v}_n(T_{n-1}) &= \mathbf{0}. \end{aligned}$$

Step 2: Solve for $n \geq 1$ in parallel the linear non-homogeneous problems

$$\begin{aligned} \mathbf{w}_n'(t) &= A\mathbf{w}_n(t), \qquad t \in [T_{n-1},T], \\ \mathbf{w}_n(T_{n-1}) &= \mathbf{v}_{n-1}(T_{n-1}), \quad \mathbf{v}_0(T_0) = \mathbf{u}_0. \end{aligned}$$

ParaExp then forms the linear combination $\mathbf{u}(t) = \mathbf{v}_n(t) + \sum_{j=1}^n \mathbf{w}_j(t)$, $t \in [T_{n-1},T_n)$, which still satisfies the initial condition, but not equation (1.1) since $\mathbf{u}'(t) = A\mathbf{u}(t) + B\big(\mathbf{v}_n(t)\big) + \mathbf{g}(t)$, $t \in [T_{n-1},T_n]$, except when B is not present. One can however naturally separate the solution into $\mathbf{u}(t) = \mathbf{v}(t) + \mathbf{w}(t)$, with $\mathbf{w}$ solving the linear problem $\mathbf{w}'(t) = A\mathbf{w}(t)$, $\mathbf{w}(t) = \mathbf{u}_0$, and $\mathbf{v}$ solving the nonlinear remaining part $\mathbf{v}'(t) = A\mathbf{v}(t) + B\big(\mathbf{v}(t) + \mathbf{w}(t)\big) + \mathbf{g}(t)$, $\mathbf{v}(0) = \mathbf{0}$. To apply this splitting on multiple time intervals $[T_{n-1},T_n]$ we need to iterate. Using the initialization $\mathbf{v}_n^0(T_n) = \mathbf{0}$ for $n = 1,\ldots,N$ (or some other approximation), we perform for $k = 1,2,\ldots$

Step 1: Solve for $n \geq 1$ in parallel the linear problems

$$\begin{aligned} \big(\mathbf{w}_n^k\big)'(t) &= A\mathbf{w}_n^k(t), \qquad t \in [T_{n-1},T], \\ \mathbf{w}_n^k(T_{n-1}) &= \mathbf{v}_{n-1}^{k-1}(T_{n-1}), \quad \mathbf{w}_1^k(T_0) = \mathbf{u}_0. \end{aligned} \tag{1.2}$$

Step 2: Solve for $n \geq 1$ in parallel the nonlinear problems

$$\begin{aligned} \big(\mathbf{v}_n^k\big)'(t) &= A\mathbf{v}_n^k(t) + B\Big(\mathbf{v}_n^k(t) + \sum_{j=1}^n \mathbf{w}_j^k(t)\Big) + \mathbf{g}(t), \quad t \in [T_{n-1},T_n], \\ \mathbf{v}_n^k(T_{n-1}) &= \mathbf{0}. \end{aligned} \tag{1.3}$$

The new approximate solution is then defined by $\mathbf{u}^k(t) = \mathbf{v}_n^k(t) + \sum_{j=1}^n \mathbf{w}_j^k(t)$, $t \in [T_{n-1},T_n)$, which now satisfies equation (1.1) on each time interval $[T_{n-1},T_n)$, and $\mathbf{u}^k(0) = \mathbf{u}_0$. The solution of the linear part (1.2) can still be computed efficiently as in the ParaExp algorithm using Krylov techniques, but (1.3) requires the computation of $\sum_{j=1}^n \mathbf{w}_j^k$ on $[T_{n-1},T_n]$, and thus would need the Krylov approximation of $\mathbf{w}_j^k$ on the entire interval $[T_{n-1},T_n]$. To avoid this, we rewrite the algorithm in terms of $\mathbf{u}_n^k$ instead of $\mathbf{v}_n^k$, where $\mathbf{u}_n^k$ approximates $\mathbf{u}$: starting with $\mathbf{u}_n^0(T_n) = \mathbf{w}_j^0(T_n) = \mathbf{0}$ for all j and n, the nonlinear ParaExp algorithm performs for $k = 1,2,\ldots$

Step 1: Solve for $n \geq 1$ in parallel the linear problems

$$\begin{aligned}\big(\mathbf{w}_n^k\big)'(t) &= A\mathbf{w}_n^k(t), && t\in[T_{n-1},T],\\ \mathbf{w}_n^k(T_{n-1}) &= \mathbf{u}_{n-1}^{k-1}(T_{n-1})-\sum_{j=1}^{n-1}\mathbf{w}_j^{k-1}(T_{n-1}), && \mathbf{w}_1^k(T_0)=\mathbf{u}_0.\end{aligned}\tag{1.4}$$

Step 2: Solve for $n\ge 1$ in parallel the nonlinear problems

$$\begin{aligned}\big(\mathbf{u}_n^k\big)'(t) &= A\mathbf{u}_n^k(t)+B\big(\mathbf{u}_n^k(t)\big)+\mathbf{g}(t), \quad t\in[T_{n-1},T_n],\\ \mathbf{u}_n^k(T_{n-1}) &= \sum_{j=1}^{n}\mathbf{w}_j^k(T_{n-1}),\end{aligned}\tag{1.5}$$

and form the new approximate solution as

$$\mathbf{u}^k(t)=\mathbf{u}_n^k(t),\quad t\in[T_{n-1},T_n).\tag{1.6}$$

Remark 1. To avoid the computation of $\mathbf{u}_n^k$ as the solution of a nonlinear problem, one could linearize (1.5) by using in the nonlinear term $B(\mathbf{u}_n^{k-1})$ instead of $B(\mathbf{u}_n^k)$, where $\mathbf{u}_n^0=\mathbf{0}$ or some other approximation of the solution. However, in what follows we focus on the fully nonlinear version, since then $\mathbf{u}^k$ is the solution of the nonlinear problem (1.1) on each time interval.

2 Analysis of the Nonlinear ParaExp Algorithm

We first show that the nonlinear ParaExp algorithm introduced in the previous section converges in a finite number of steps.

Theorem 1. *The approximate solution $\mathbf{u}^k$ obtained at iteration k and defined by* (1.6) *coincides with the exact solution $\mathbf{u}$ on the time interval* $[T_0,T_k)$.

Proof. Since $\mathbf{w}_1^k(T_0)=\mathbf{u}_0$ for all $k=1,2,\ldots$, $\mathbf{w}_1^k=\mathbf{w}_1^{k-1}$ on the time interval $[T_0,T]$ for all $k=2,3,\ldots$. Next, for $k=1$ we have $\mathbf{u}^1(t)=\mathbf{u}_1^1(t)$ on $[T_0,T_1]$, and since $\mathbf{u}_1^1(T_0)=\mathbf{w}_1^1(T_0)=\mathbf{u}_0$ we get by the uniqueness of the solution of (1.5) that $\mathbf{u}_1^1$ coincides with the exact solution $\mathbf{u}$ on the time interval $[T_0,T_1]$.

We now prove by induction that for all $k=2,3\ldots$ we have

$$\mathbf{u}_n^k=\mathbf{u}\ \text{ on }\ [T_{n-1},T_n],\ \forall n\le k,\qquad \mathbf{w}_n^k=\mathbf{w}_n^{k-1}\ \text{ on }\ [T_{n-1},T],\ \forall n\le k-1.\tag{2.1}$$

For $k=2$, we only need to prove property (2.1) for $\mathbf{u}^2$, since for $\mathbf{w}_1^2$ it is ensured by the fact that $\mathbf{w}_1^k=\mathbf{w}_1^{k-1}$ for all $k\ge 2$. The initial condition for $\mathbf{u}_2^2$ is

$$\mathbf{u}_2^2(T_1)=\mathbf{w}_1^2(T_1)+\mathbf{w}_2^2(T_1)=\mathbf{w}_1^2(T_1)+\mathbf{u}_1^1(T_1)-\mathbf{w}_1^1(T_1)=\mathbf{u}_1^1(T_1)=\mathbf{u}(T_1),$$

where we used the fact that $\mathbf{w}_1^2=\mathbf{w}_1^1$ and that $\mathbf{u}_1^1$ is the exact solution on the time interval $[T_0,T_1]$. Since $\mathbf{u}_2^2$ satisfies the same equation as $\mathbf{u}$ on the time interval $[T_1,T_2]$ and $\mathbf{u}_2^2(T_1)=\mathbf{u}(T_1)$, $\mathbf{u}_2^2$ must coincide with $\mathbf{u}$ on $[T_1,T_2]$. But we also know that

$\mathbf{u}_1^2(T_0) = \mathbf{w}_1^2(T_0) = \mathbf{u}_0$ and that $\mathbf{u}_1^2$ satisfies (1.5), which implies $\mathbf{u}_1^2 = \mathbf{u}$ on $[T_0, T_1]$, and hence $\mathbf{u}^2$ coincides with the exact solution of (1.1) on the time interval $[T_0, T_2)$.

We now suppose that (2.1) holds for all iterations up to an arbitrarily fixed index k and we prove (2.1) for $k+1$. To first check that $\mathbf{w}_n^{k+1} = \mathbf{w}_n^k$ on $[T_{n-1}, T]$ for all $n = 2, 3, \dots, k$, we compute

$$\begin{aligned}\mathbf{w}_n^{k+1}(T_{n-1}) &= \mathbf{u}_{n-1}^k(T_{n-1}) - \sum_{j=1}^{n-1} \mathbf{w}_j^k(T_{n-1}) = \mathbf{u}(T_{n-1}) - \sum_{j=1}^{n-1} \mathbf{w}_j^{k-1}(T_{n-1}) \\ &= \mathbf{u}_{n-1}^{k-1}(T_{n-1}) - \sum_{j=1}^{n-1} \mathbf{w}_j^{k-1}(T_{n-1}) = \mathbf{w}_n^k(T_{n-1}),\end{aligned}$$

where we have used the recurrence hypothesis (2.1). Since $\mathbf{w}_n^{k+1}$ and $\mathbf{w}_n^k$ satisfy the same equation and have the same initial condition, the result follows. We next prove that $\mathbf{u}_n^{k+1} = \mathbf{u}$ on $[T_{n-1}, T_n]$ for all $n \le k+1$. Since we already know that $\mathbf{u}_n^{k+1}$ and $\mathbf{u}$ satisfy the same equation on the time interval $[T_{n-1}, T_n]$, we only need to check that the initial condition satisfied by $\mathbf{u}_n^{k+1}$,

$$\begin{aligned}\mathbf{u}_n^{k+1}(T_{n-1}) &= \sum_{j=1}^{n} \mathbf{w}_j^{k+1}(T_{n-1}) = \sum_{j=1}^{n-1} \mathbf{w}_j^{k+1}(T_{n-1}) + \mathbf{u}_{n-1}^k(T_{n-1}) - \sum_{j=1}^{n-1} \mathbf{w}_j^k(T_{n-1}) \\ &= \mathbf{u}_{n-1}^k(T_{n-1}),\end{aligned}$$

where we used the first result we just proved for $\mathbf{w}_n^{k+1}$ and that $\mathbf{w}_1^{k+1} = \mathbf{w}_1^k$ for all k. Now, using the recurrence hypothesis (2.1), we know that $\mathbf{u}_{n-1}^k$ coincides with the exact solution of (1.1) on $[T_{n-2}, T_{n-1}]$, which implies that $\mathbf{u}_n^{k+1}(T_{n-1}) = \mathbf{u}(T_{n-1})$. □

We now show that the nonlinear ParaExp algorithm can be interpreted in the context of the Parareal algorithm if written as a multiple shooting method (see [5, 2]). We will need the following result.

Lemma 1. *Let* $(\mathbf{u}_n^k)_{k,n}$ *be the sequence defined by the nonlinear ParaExp algorithm* (1.4)–(1.6). *Defining* $\widetilde{\mathbf{u}}_n^0(T_n) = \mathbf{0}$ *and* $\mathbf{C}_n^0(T_n) = \mathbf{0}$ *for all* $n \ge 0$, *let* $\left(\mathbf{C}_n^k\right)_{k,n}$ *for all* $k \ge 1$ *and* $n \ge 1$ *be the solutions of the linear problems*

$$\begin{aligned}&\left(\mathbf{C}_n^k\right)'(t) = A\mathbf{C}_n^k(t), && t \in [T_{n-1}, T_n], \\ &\mathbf{C}_n^k(T_{n-1}) = \mathbf{C}_{n-1}^k(T_{n-1}) + \widetilde{\mathbf{u}}_{n-1}^{k-1}(T_{n-1}) - \mathbf{C}_{n-1}^{k-1}(T_{n-1}), && \mathbf{C}_1^k(T_0) = \mathbf{u}_0,\end{aligned}$$

and let $\left(\widetilde{\mathbf{u}}_n^k\right)_{k,n}$ *be the solutions of the nonlinear problems*

$$\begin{aligned}&\left(\widetilde{\mathbf{u}}_n^k\right)'(t) = A\widetilde{\mathbf{u}}_n^k(t) + B\left(\widetilde{\mathbf{u}}_n^k(t)\right) + \mathbf{g}(t), \quad t \in [T_{n-1}, T_n], \\ &\widetilde{\mathbf{u}}_n^k(T_{n-1}) = \mathbf{C}_n^k(T_{n-1}).\end{aligned}$$

Then $\mathbf{u}_n^k = \widetilde{\mathbf{u}}_n^k$ *on* $[T_{n-1}, T_n]$ *for all* $n \ge 0$ *and* $k \ge 1$.

Proof. At step $k = 1$ and for all $n \ge 1$, $\mathbf{C}_n^1$ is the solution of the linear problem

$$(\mathbf{C}_n^1)'(t) = A\mathbf{C}_n^1(t), \qquad t \in [T_{n-1}, T_n],$$
$$\mathbf{C}_n^1(T_{n-1}) = \mathbf{C}_{n-1}^1(T_{n-1}), \quad \mathbf{C}_1^1(T_0) = \mathbf{u}_0.$$

Hence $\mathbf{C}_n^1$ is the restriction of the solution of $\mathbf{u}' = A\mathbf{u}$, $\mathbf{u}(0) = \mathbf{u}_0$ on $[T_0, T]$ to the time interval $[T_{n-1}, T_n]$. Taking into account the definition (1.4) of $\mathbf{w}_n^1$, we notice that $\mathbf{w}_n^1 = \mathbf{0}$ for $n > 1$ and $\mathbf{w}_1^1$ is the solution of the linear problem $\mathbf{u}' = A\mathbf{u}$, $\mathbf{u}(0) = \mathbf{u}_0$ on $[T_0, T]$. Thus, $\mathbf{C}_n^1(t) = \sum_{j=1}^n \mathbf{w}_j^1(t)$ on $[T_{n-1}, T_n]$, and $\widetilde{\mathbf{u}}_n^1$ satisfies for $n \geq 1$

$$\big(\widetilde{\mathbf{u}}_n^1\big)'(t) = A\widetilde{\mathbf{u}}_n^1(t) + B\big(\widetilde{\mathbf{u}}_n^1(t)\big) + \mathbf{g}(t), \quad t \in [T_{n-1}, T_n],$$
$$\mathbf{u}_n^1(T_{n-1}) = \mathbf{C}_n^1(T_{n-1}) = \sum_{j=1}^n \mathbf{w}_j^1(T_{n-1}).$$

Comparing this with (1.5) and using the uniqueness of the solution for the nonlinear problem, we deduce that $\mathbf{u}_n^1(t) = \widetilde{\mathbf{u}}_n^1(t)$ on $[T_{n-1}, T_n]$ for all $n \geq 1$.

Assuming now that for all $n \geq 1$ and a given k we have $\mathbf{C}_n^k(t) = \sum_{j=1}^n \mathbf{w}_j^k(t)$, $\mathbf{u}_n^k(t) = \widetilde{\mathbf{u}}_n^k(t)$ on $[T_{n-1}, T_n]$,we need to show that this also holds for $k+1$. To do so, we prove by recurrence with respect to n that $\mathbf{C}_n^{k+1}(t) = \sum_{j=1}^n \mathbf{w}_j^{k+1}(t)$ on $[T_{n-1}, T_n]$. For $n = 1$, we have that $\mathbf{C}_1^{k+1}(T_0) = \mathbf{u}_0 = \mathbf{w}_1^{k+1}(T_0)$ and, since $\mathbf{C}_1^{k+1}$ and $\mathbf{w}_1^{k+1}$ satisfy the same equation and the same initial condition, we conclude that $\mathbf{C}_1^{k+1} = \mathbf{w}_1^{k+1}$ on $[T_0, T_1]$. Next, we suppose that $\mathbf{C}_n^{k+1}(t) = \sum_{j=1}^n \mathbf{w}_j^{k+1}(t)$ on $[T_{n-1}, T_n]$ and prove that $\mathbf{C}_{n+1}^{k+1}(t) = \sum_{j=1}^{n+1} \mathbf{w}_j^{k+1}(t)$ on $[T_n, T_{n+1}]$. By checking the initial condition of $\mathbf{C}_{n+1}^{k+1}$ at T_n and using the recurrence hypothesis, we find

$$\mathbf{C}_{n+1}^{k+1}(T_n) = \mathbf{C}_n^{k+1}(T_n) + \mathbf{u}_n^k(T_n) - \sum_{j=1}^n \mathbf{w}_j^k(T_n) = \mathbf{C}_n^{k+1}(T_n) + \mathbf{w}_{n+1}^{k+1}(T_n) = \sum_{j=1}^{n+1} \mathbf{w}_j^{k+1}(T_n).$$

Since $\mathbf{C}_{n+1}^{k+1}$ and $\sum_{j=1}^{n+1} \mathbf{w}_j^{k+1}$ solve the same linear problem on $[T_n, T_{n+1}]$ and satisfy the same initial condition at T_n, we obtain $\mathbf{C}_{n+1}^{k+1} = \sum_{j=1}^{n+1} \mathbf{w}_j^{k+1}$ on $[T_n, T_{n+1}]$. Further, for $n \geq 1$ we have

$$\big(\widetilde{\mathbf{u}}_n^{k+1}\big)'(t) = A\widetilde{\mathbf{u}}_n^{k+1}(t) + B\big(\widetilde{\mathbf{u}}_n^{k+1}(t)\big) + \mathbf{g}(t), \quad t \in [T_{n-1}, T_n],$$
$$\widetilde{\mathbf{u}}_n^{k+1}(T_{n-1}) = \mathbf{C}_n^{k+1}(T_{n-1}) = \sum_{j=1}^n \mathbf{w}_j^{k+1}(T_{n-1}).$$

Thus, $\widetilde{\mathbf{u}}_n^{k+1}$ and $\mathbf{u}_n^{k+1}$ solve the same equation with identical initial condition on $[T_{n-1}, T_n]$ and hence $\widetilde{\mathbf{u}}_n^{k+1} = \mathbf{u}_n^{k+1}$ on $[T_{n-1}, T_n]$. □

The following theorem is essentially a reformulation of Lemma 1 in the usual notation of the parareal algorithm in terms of a coarse and a fine integrator [11].

Theorem 2. *Let the coarse propagator $G(T_n, T_{n-1}, \mathbf{U})$ solve the linear problem*

$$\mathbf{u}'(t) = A\mathbf{u}(t) \;\; \text{on} \;\; [T_{n-1}, T_n], \quad \mathbf{u}(T_{n-1}) = \mathbf{U},$$

and let the fine propagator $F(T_n, T_{n-1}, \mathbf{U})$ solve the nonlinear problem

$$\mathbf{u}'(t) = A\mathbf{u}(t) + B\big(\mathbf{u}(t)\big) + \mathbf{g}(t) \;\; on \;\; [T_{n-1}, T_n], \quad \mathbf{u}(T_{n-1}) = \mathbf{U}.$$

Then the solution $\mathbf{u}^k$ computed by the nonlinear ParaExp algorithm (1.4)–(1.6) *coincides at each time point T_n with the solution $\mathbf{U}_n^k$ computed by the parareal algorithm*

$$\mathbf{U}_n^k = F(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}) + G(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) - G(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k-1}). \tag{2.2}$$

Proof. Using the definition of $\mathbf{u}^k$ in (1.6) and the notation of Lemma 1, we have

$$\begin{aligned}
&\mathbf{u}^k(T_n) = \mathbf{u}_{n+1}^k(T_n) = \mathbf{C}_{n+1}^k(T_n) = \mathbf{C}_n^k(T_n) + \mathbf{u}_n^{k-1}(T_n) - \mathbf{C}_n^{k-1}(T_n) \\
&= G\big(T_n, T_{n-1}, \mathbf{C}_n^k(T_{n-1})\big) - G\big(T_n, T_{n-1}, \mathbf{C}_n^{k-1}(T_{n-1})\big) + \widetilde{\mathbf{u}}_n^{k-1}(T_n) \\
&= G\big(T_n, T_{n-1}, \mathbf{C}_n^k(T_{n-1})\big) - G\big(T_n, T_{n-1}, \mathbf{C}_n^{k-1}(T_{n-1})\big) + F\big(T_n, T_{n-1}, \mathbf{C}_n^{k-1}(T_{n-1})\big).
\end{aligned}$$

Thus $\mathbf{u}^k(T_n) = \mathbf{U}_n^k$ with $\mathbf{U}_n^k = \mathbf{C}_{n+1}^k(T_n)$. □

Theorem 2 shows that the nonlinear ParaExp algorithm is mathematically equivalent to the parareal algorithm (2.2) where the coarse integrator G is an exponential integrator for $\mathbf{w}' = A\mathbf{w}$. There is however an important computational difference: due to the linearity of G we can write

$$\begin{aligned}
&G(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k+1}) \\
&= G\big(T_n, T_{n-1}, F(T_{n-1}, T_{n-2}, \mathbf{U}_{n-2}^k) - G(T_{n-1}, T_{n-2}, \mathbf{U}_{n-2}^k) + G(T_{n-1}, T_{n-2}, \mathbf{U}_{n-2}^{k+1})\big) \\
&= G\big(T_n, T_{n-1}, F(T_{n-1}, T_{n-2}, \mathbf{U}_{n-2}^k) - G(T_{n-1}, T_{n-2}, \mathbf{U}_{n-2}^k)\big) + G(T_n, T_{n-2}, \mathbf{U}_{n-2}^{k+1}),
\end{aligned}$$

which corresponds to the coarse propagation of a jump over $[T_{n-1}, T_n]$ plus the coarse propagation of $\mathbf{U}_{n-2}^{k+1}$ over a longer time interval $[T_{n-2}, T_n]$. Repeating a similar calculation for $G(T_n, T_{n-2}, \mathbf{U}_{n-2}^{k+1})$, we derive

$$\begin{aligned}
G(T_n, T_{n-2}, \mathbf{U}_{n-2}^{k+1}) \;=\; & G\big(T_n, T_{n-2}, F(T_{n-2}, T_{n-3}, \mathbf{U}_{n-3}^k) - G(T_{n-2}, T_{n-3}, \mathbf{U}_{n-3}^k)\big) \\
& + G(T_n, T_{n-3}, \mathbf{U}_{n-3}^{k+1}),
\end{aligned}$$

which again corresponds to the coarse propagation of a jump (over two intervals) plus a coarse propagation of $\mathbf{U}_{n-3}^{k+1}$ (over three intervals). This recursion can be repeated, and it will terminate as $\mathbf{U}_{n-n}^{k+1} = \mathbf{U}_0$ is known, leading to an alternative, more compact formulation of the nonlinear ParaExp algorithm:

$$\text{initialize} \quad \mathbf{U}_n^0 = G(T_n, T_0, \mathbf{U}_0) \quad \text{for } n = 0, 1, \ldots, N,$$

$$\mathbf{U}_n^{k+1} = G(T_n, T_0, \mathbf{U}_0) + \sum_{j=1}^{n} G\big(T_n, T_j, F(T_j, T_{j-1}, \mathbf{U}_{j-1}^k) - G(T_j, T_{j-1}, \mathbf{U}_{j-1}^k)\big).$$

Here the coarse integrator is applied in parallel, which is different from parareal. The price to pay is that the coarse integrations now span multiple overlapping time intervals $[T_j, T_n]$. As in the original ParaExp algorithm, these linear homogeneous problems can be solved very efficiently using Krylov methods.

3 Numerical Illustration

We now investigate the nonlinear ParaExp algorithm numerically. We solve the nonlinear wave equation $u_{tt} = u_{xx} + \alpha u^2$ on the time-space domain $[0,4] \times [-1,1]$ with homogeneous Dirichlet boundary conditions and $u(0,x) = e^{-100x^2}$, $u'(0,x) = 0$, where the parameter $\alpha \geq 0$ controls the nonlinear character of the problem. The problem is discretized in space using finite differences with $m = 200$ equispaced interior grid points on $[-1,1]$. This gives rise to the ODE

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}' = \begin{bmatrix} O & I \\ L & O \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \alpha \mathbf{u}^2 \end{bmatrix},$$

where $L = \text{tridiag}(1,-2,1)/h^2$, $h = 2/(m+1)$, and the operation $\mathbf{u}^2$ has to be understood component-wise. We partition the time interval $[0,4]$ into $n = 20$ slices of equal length and use as fine integrator MATLAB's `ode15s` routine with a relative error tolerance of 10^{-6}. For the linear coarse integration we use MATLAB's `expm`.

Table 1 lists, for varying $\alpha \in \{0,2,4,6,8.2\}$, the number of iterations required by our nonlinear ParaExp algorithm to achieve an error of order $\approx 1e-6$ over all time slices. Figure 1 shows, again for varying α, the reference solutions $u(t,x)$ on the left, and on the right the error of the ParaExp solution at each time point t_j after $k = 1,2,\ldots$ iterations. Here a number of $k = 0$ iterations corresponds to the error of the ParaExp initialization with the coarse integrator.

The parameter $\alpha = 0$ gives rise to a linear problem. Note that for this case the error of the initialization is of order $\approx 10^{-6}$, and not of order machine precision as one would expect from the exponential integration using `expm`. This is because our reference solution has been computed via `ode15s` and is of lower accuracy.

For increasing values of α the nonlinear character of the wave equation becomes more pronounced and typically more ParaExp iterations are required. It depends on the efficiency of the coarse propagator (in this case `expm`) if any speed-up would be obtained in a parallel implementation. For large-scale problems the use of (rational) Krylov techniques as in [7] is recommended. The nonlinear ParaExp method becomes inefficient for highly nonlinear problems, with 14 iterations required for $\alpha = 8.2$. This is expected and we note that for $\alpha \approx 9$ the solution $u(t,x)$ even appears to have a singularity in the time-space domain of interest.

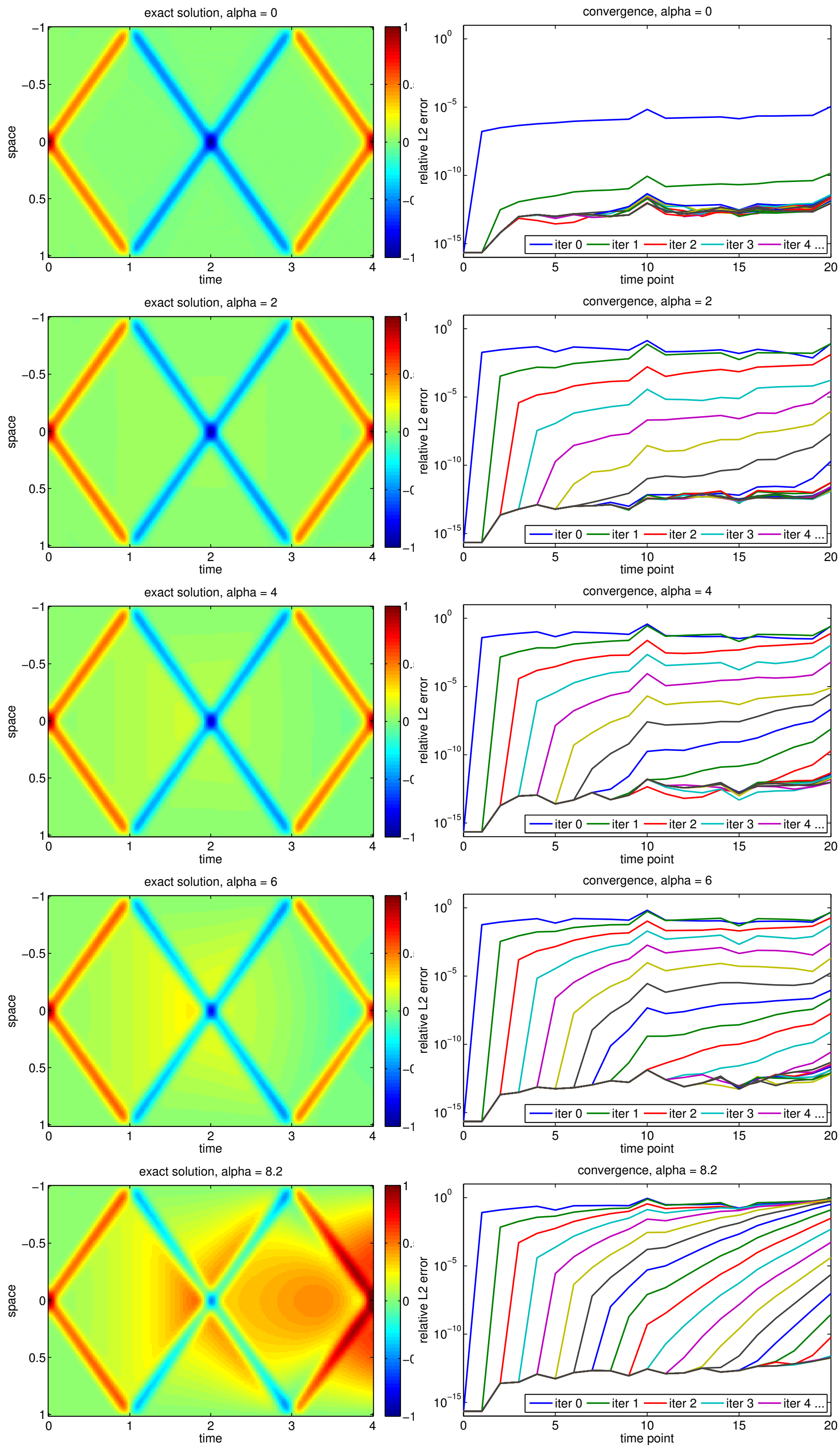

Fig. 1 Exact solutions (left) and convergence (right) of the nonlinear ParaExp algorithm applied to a nonlinear wave equation with varying parameter $\alpha \in \{0, 2, 4, 6, 8.2\}$ (top to bottom).

parameter α	0	2	4	6	8.2
# iterations	1	5	7	7	14

Table 1 Number of iterations required by the nonlinear ParaExp algorithm to solve a nonlinear wave equation to fixed accuracy uniformly over a time interval. The parameter α controls the nonlinearity of the problem.

References

1. M. J. GANDER, *50 years of time parallel time integration*, Multiple Shooting and Time Domain Decomposition Methods, Springer, pp. 69–113, 2015.
2. M. J. GANDER AND E. HAIRER, *Nonlinear convergence analysis for the parareal algorithm*, Domain Decomposition Methods in Science and Engineering XVII, Springer, pp. 45–56, 2007.
3. M. J. GANDER, L. HALPERN, AND F. NATAF, *Optimal Schwarz waveform relaxation for the one dimensional wave equation*, SIAM J. Numer. Anal., 41:1643–1681, 2003.
4. M. J. GANDER AND L. HALPERN, *Absorbing boundary conditions for the wave equation and parallel computing*, Math. Comp., 74:153–176, 2004.
5. M.J. GANDER AND S. VANDERWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29(2):556–578, 2007.
6. J. GOPALAKRISHNAN, J. SCHÖBERL, AND C. WINTERSTEIGER, *Mapped tent pitching schemes for hyperbolic systems*, SIAM J. Sci. Comput., 39(6), B1043–B1063, 2017.
7. M.J. GANDER AND S. GÜTTEL, *PARAEXP: A parallel integrator for linear initial-value problems*, SIAM J. Sci. Comput., 35(2):C123–C142, 2013.
8. M. J. GANDER AND M. PETCU, *Analysis of a Krylov subspace enhanced parareal algorithm*, ESAIM Proc., 25:45–56, 2008.
9. S. GÜTTEL, *A parallel overlapping time-domain decomposition method for ODEs*, Domain Decomposition Methods in Science and Engineering XX, Springer, pp. 483–490, 2013.
10. G. KOOIJ, M. BOTCHEV, AND B. GEURTS, *A block Krylov subspace implementation of the time-parallel ParaExp method and its extension for nonlinear partial differential equations*, J. Comput. Appl. Math., 316:229–246, 2017.
11. J.-L. LIONS, Y. MADAY, AND G. TURINICI, *A "parareal" in time discretization of PDE's*, C. R. Acad. Sci. Paris Sér. I Math., 332:661–668, 2001.