A Nonlinear ParaExp Algorithm

Gander, Martin J. and Güttel, Stefan and Petcu, Madalina

2017

MIMS EPrint: 2017.17

Manchester Institute for Mathematical Sciences
School of Mathematics
The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097
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

\[ u'(t) = Au(t) + B(u(t)) + g(t), \quad t \in [0, T], \quad u(0) = u_0, \]  

(1.1)

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

\[ v'_n(t) = Av_n(t) + B(v_n(t)) + g(t), \quad t \in [T_{n-1}, T_n], \]

\[ v_n(T_{n-1}) = 0. \]
Step 2: Solve for \( n \geq 1 \) in parallel the linear non-homogeneous problems

\[
\begin{align*}
\mathbf{w}_n(t) &= A \mathbf{w}_n(t), & t &\in [T_{n-1}, T], \\
\mathbf{w}_n(T_{n-1}) &= \mathbf{v}_{n-1}(T_{n-1}), & \mathbf{v}_0(T_0) &= \mathbf{u}_0.
\end{align*}
\]

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 \( \mathbf{u}'(t) = A \mathbf{u}(t) + B(\mathbf{v}_n(t)) + \mathbf{g}(t), t \in [T_{n-1}, T_n] \), 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 \( \mathbf{v}'(t) = A\mathbf{v}(t) + B(\mathbf{v}(t) + \mathbf{w}(t)) + \mathbf{g}(t), \mathbf{v}(0) = \mathbf{0} \). To apply this splitting on multiple time intervals \([T_{n-1}, T_n]\) we need to iterate. Using the initialization \( \mathbf{v}_n(T_n) = \mathbf{0} \) 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

\[
\begin{align*}
(\mathbf{w}_n^k)'(t) &= A \mathbf{w}_n^k(t), & t &\in [T_{n-1}, T], \\
\mathbf{w}_n^k(T_{n-1}) &= \mathbf{v}_{n-1}^{k-1}(T_{n-1}), & \mathbf{w}_1^0(T_0) &= \mathbf{u}_0.
\end{align*}
\]

(1.2)

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

\[
\begin{align*}
(\mathbf{v}_n^k)'(t) &= A \mathbf{v}_n^k(t) + B(\mathbf{v}_n^k(t) + \sum_{j=1}^n \mathbf{w}_j^k(t)) + \mathbf{g}(t), & t &\in [T_{n-1}, T_n], \\
\mathbf{v}_n^k(T_{n-1}) &= \mathbf{0}.
\end{align*}
\]

(1.3)

The new approximate solution is then defined by \( \mathbf{u}_n^k(t) = \mathbf{v}_n^k(t) + \sum_{j=1}^n \mathbf{w}_j^k(t), t \in [T_{n-1}, T_n] \), which now satisfies equation (1.1) on each time interval \([T_{n-1}, T_n]\), and \( \mathbf{u}_n^k(0) = \mathbf{u}_0 \). 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 \( \mathbf{u}_j^k \) instead of \( \mathbf{v}_j^k \), where \( \mathbf{u}_j^k \) approximates \( \mathbf{u} \): starting with \( \mathbf{u}_j^0(T_n) = \mathbf{w}_j^0(T_n) = \mathbf{0} \) 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

\[
\begin{align*}
(\mathbf{w}_n^k)'(t) &= A \mathbf{w}_n^k(t), & t &\in [T_{n-1}, T], \\
\mathbf{w}_n^k(T_{n-1}) &= \mathbf{u}_{n-1}^{k-1}(T_{n-1}) - \sum_{j=1}^{n-1} \mathbf{w}_j^{k-1}(T_{n-1}), & \mathbf{w}_1^0(T_0) &= \mathbf{u}_0.
\end{align*}
\]

(1.4)

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

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

(1.5)
and form the new approximate solution as
\[ u^k(t) = u_n^k(t), \quad t \in [T_{n-1}, T_n). \] (1.6)

**Remark 1.** To avoid the computation of \( u_n^k \) as the solution of a nonlinear problem, one could linearize (1.5) by using in the nonlinear term \( B(u_n^{k-1}) \) instead of \( B(u_n^k) \), where \( u_n^0 = 0 \) or some other approximation of the solution. However, in what follows we focus on the fully nonlinear version, since then \( u^k \) 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 \( u^k \) obtained at iteration \( k \) and defined by (1.6) coincides with the exact solution \( u \) on the time interval \([T_0, T_k]\).

**Proof.** Since \( w_1^k(T_0) = u_0 \) for all \( k = 1, 2, \ldots \), \( w_1^k = w_1^{k-1} \) on the time interval \([T_0, T]\) for all \( k = 2, 3, \ldots \). Next, for \( k = 1 \) we have \( u_1(t) = u_1(t) \) on \([T_0, T_1]\), and since \( u_1(T_0) = w_1(T_0) = u_0 \) we get by the uniqueness of the solution of (1.5) that \( u_1 \) coincides with the exact solution \( u \) on the time interval \([T_0, T_1]\).

We now prove by induction that for all \( k = 2, 3 \ldots \) we have
\[ u_n^k = u \quad \text{on} \quad [T_{n-1}, T_n], \quad \forall n \leq k, \quad w_n^k = w_n^{k-1} \quad \text{on} \quad [T_{n-1}, T], \quad \forall n \leq k-1. \] (2.1)

For \( k = 2 \), we only need to prove property (2.1) for \( u^2 \), since for \( w_1^2 \) it is ensured by the fact that \( w_1^2 = w_1^1 \) for all \( k \geq 2 \). The initial condition for \( u_2^0 \) is
\[ u_2^0(T_1) = w_2^0(T_1) + w_2^1(T_1) = w_2^1(T_1) + u_1^1(T_1) - u_1^1(T_1) = u(T_1), \]
where we used the fact that \( w_2^1 = w_2^1 \) and that \( u_1 \) is the exact solution on the time interval \([T_0, T_1]\). Since \( u_2^0 \) satisfies the same equation as \( u \) on the time interval \([T_0, T_2]\) and \( u_2^0(T_1) = u(T_1) \), \( u_2^0 \) must coincide with \( u \) on \([T_1, T_2]\). But we also know that \( u_2^1(T_0) = w_2^1(T_0) = u_0 \) and that \( u_2^1 \) satisfies (1.5), which implies \( u_2^1 = u \) on \([T_0, T_1]\), and hence \( u^2 \) 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 \( w_n^{k+1} = w_n^k \) on \([T_{n-1}, T]\) for all \( n = 2, 3, \ldots, k \), we compute
\[
w_n^{k+1}(T_{n-1}) = u_{n-1}^k(T_{n-1}) - \sum_{j=1}^{n-1} w_j^k(T_{n-1}) = u(T_{n-1}) - \sum_{j=1}^{n-1} w_j^{k-1}(T_{n-1}) = u_{n-1}^{k-1}(T_{n-1}) - \sum_{j=1}^{n-1} w_j^{k-1}(T_{n-1}) = w_n^k(T_{n-1}),
\]
where we have used the recurrence hypothesis (2.1). Since \( w_n^{k+1} \) and \( w_n^k \) satisfy the same equation and have the same initial condition, the result follows. We next prove that \( u_n^{k+1} = u \) on \([T_{n-1}, T_n]\) for all \( n \leq k+1 \). Since we already know that \( u_n^{k+1} \) and \( u \) satisfy the same equation on the time interval \([T_{n-1}, T_n]\), we only need to check the initial condition satisfied by \( u_n^{k+1} \),

\[
\begin{align*}
    u_n^{k+1}(T_{n-1}) &= \sum_{j=1}^{n} w_j^{k+1}(T_{n-1}) = \sum_{j=1}^{n-1} w_j^{k+1}(T_{n-1}) + u_{n-1}^k(T_{n-1}) - \sum_{j=1}^{n-1} w_j^k(T_{n-1}) \\
    &= u_{n-1}^k(T_{n-1}),
\end{align*}
\]

where we used the first result we just proved for \( w_n^{k+1} \) and that \( w_n^k = w_1^k \) for all \( k \).

Now, using the recurrence hypothesis (2.1), we know that \( u_n^{k+1} \) coincides with the exact solution of (1.1) on \([T_{n-2}, T_{n-1}]\), which implies that \( u_n^{k+1}(T_{n-1}) = u(T_{n-1}) \). \( \square \)

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

**Lemma 1.** Let \((u_n^k)_{k,n}\) be the sequence defined by the nonlinear ParaExp algorithm (1.4)-(1.6). Defining \( \tilde{u}_n^0(T_n) = 0 \) and \( C_n^0(T_n) = 0 \) for all \( n \geq 0 \), let \((C_n^k)_{k,n}\) for all \( k \geq 1 \) and \( n \geq 1 \) be the solutions of the linear problems

\[
\begin{align*}
    (C_n^k)'(t) &= AC_n^k(t), & t &\in [T_{n-1}, T_n], \\
    C_n^k(T_{n-1}) &= C_{n-1}^k(T_{n-1}) + \tilde{u}_{n-1}^{k-1}(T_{n-1}) - C_{n-1}^{k-1}(T_{n-1}), & C_1^1(T_0) &= u_0,
\end{align*}
\]

and let \((\tilde{u}_n^k)_{k,n}\) be the solutions of the nonlinear problems

\[
\begin{align*}
    (\tilde{u}_n^k)'(t) &= A\tilde{u}_n^k(t) + B(\tilde{u}_n^k(t)) + g(t), & t &\in [T_{n-1}, T_n], \\
    \tilde{u}_n^k(T_{n-1}) &= C_n^k(T_{n-1}).
\end{align*}
\]

Then \( u_n^k = \tilde{u}_n^k \) on \([T_{n-1}, T_n]\) for all \( n \geq 0 \) and \( k \geq 1 \).

**Proof.** At step \( k = 1 \) and for all \( n \geq 1 \), \( C_n^1 \) is the solution of the linear problem

\[
\begin{align*}
    (C_n^1)'(t) &= AC_n^1(t), & t &\in [T_{n-1}, T_n], \\
    C_n^1(T_{n-1}) &= C_{n-1}^1(T_{n-1}), & C_1^1(T_0) &= u_0.
\end{align*}
\]

Hence \( C_n^1 \) is the restriction of the solution of \( u' = Au, u(0) = u_0 \) on \([T_0, T]\) to the time interval \([T_{n-1}, T_n]\), Taking into account the definition (1.4) of \( w_n^1 \), we notice that \( w_n^1 = 0 \) for \( n > 1 \) and \( w_1^1 \) is the solution of the linear problem \( u' = Au, u(0) = u_0 \) on \([T_0, T]\). Thus, \( C_n^1(t) = \sum_{j=1}^{n} w_j^1(t) \) on \([T_{n-1}, T_n]\), and \( \tilde{u}_n^1 \) satisfies for \( n \geq 1 \)}
Thus, we deduce that $u_n^{1}(t)$ is the solution to the problem over $[T_{n-1}, T_n]$, for all $n \geq 1$.

Assuming now that for all $n \geq 1$ and a given $k$ we have $C_n^k(t) = \sum_{j=1}^n w_j^k(t)$, $u_n^{k}(t) = \tilde{u}_n^k(t)$ 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 $C_{n+1}^k(t) = \sum_{j=1}^n w_j^{k+1}(t)$ on $[T_{n-1}, T_n]$.

For $n = 1$, we have that $C_{1+1}^k(T_0) = u_0 = w_1^{k+1}(T_0)$ and, since $C_1^{k+1}$ and $w_1^{k+1}$ satisfy the same equation and the same initial condition, we conclude that $C_1^{k+1} = w_1^{k+1}$ on $[T_0, T_1]$. Next, we suppose that $C_{n+1}^k(t) = \sum_{j=1}^n w_j^{k+1}(t)$ on $[T_{n-1}, T_n]$ and prove that $C_{n+1}^k(t) = \sum_{j=1}^{n+1} w_j^{k+1}(t)$ on $[T_{n+1}, T_n]$. By checking the initial condition of $C_{n+1}^{k+1}$ at $T_n$ and using the recurrence hypothesis, we find

$$C_{n+1}^{k+1}(T_n) = C_{n}^{k+1}(T_{n}) + u_n^{k}(T_n) - \sum_{j=1}^n w_j^{k}(T_n) = C_{n}^{k+1}(T_{n}) + w_{n+1}^{k+1}(T_n) = \sum_{j=1}^{n+1} w_j^{k+1}(T_n).$$

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

$$(u_n^{k+1})'(t) = A u_n^{k+1}(t) + B(u_n^{k+1}(t)) + g(t), \quad t \in [T_{n-1}, T_n],$$

Thus, $u_n^{k+1}$ and $u_n^{k+1}$ solve the same equation with identical initial condition on $[T_{n-1}, T_n]$ and hence $u_n^{k+1} = u_n^{k+1}$ on $[T_{n-1}, T_n]$. \hfill \Box

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 $G(T_n, T_{n-1}, U)$ solve the linear problem

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

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

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

Then the solution $u_n^k$ computed by the nonlinear ParaExp algorithm (1.4)–(1.6) coincides at each time point $T_n$ with the solution $U_n^k$ computed by the parareal algorithm

$$U_n^k = F(T_n, T_{n-1}, U_{n-1}^k) + G(T_n, T_{n-1}, U_n^k) - G(T_n, T_{n-1}, U_{n-1}^k).$$

(2.2)
Proof. Using the definition of $u^k$ in (1.6) and the notation of Lemma 1, we have

\[ u^k(T_n) = u^k_{n+1}(T_n) = C^k_{n+1}(T_n) = C^k_n(T_n) + u^k_{n-1}(T_n) - C^k_{n-1}(T_n) \]

\[ = G(T_n, T_{n-1}, C^k_n(T_{n-1})) - G(T_n, T_{n-1}, C^k_{n-1}(T_{n-1})) + u^k_{n-1}(T_n) \]

\[ = G(T_n, T_{n-1}, C^k_n(T_{n-1})) - G(T_n, T_{n-1}, C^k_{n-1}(T_{n-1})) + F(T_n, T_{n-1}, C^k_{n-1}(T_{n-1})). \]

Thus $u^k(T_n) = U^k_n$ with $U^k_n = C^k_{n+1}(T_n)$. □

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 $w' = Aw$. There is however an important computational difference: due to the linearity of $G$ we can write

\[ G(T_n, T_{n-1}, U^k_{n-1}) \]

\[ = G(T_n, T_{n-1}, F(T_{n-1}, T_{n-2}, U^k_{n-2}) - G(T_{n-1}, T_{n-2}, U^k_{n-2}) + G(T_{n-1}, T_{n-2}, U^k_{n-2})) \]

\[ = G(T_n, T_{n-1}, F(T_{n-1}, T_{n-2}, U^k_{n-2}) - G(T_{n-1}, T_{n-2}, U^k_{n-2}) + G(T_{n-1}, T_{n-2}, U^k_{n-2})), \]

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

\[ G(T_n, T_{n-2}, U^k_{n-2}) = G(T_n, T_{n-2}, F(T_{n-2}, T_{n-3}, U^k_{n-3}) - G(T_{n-2}, T_{n-3}, U^k_{n-3})) \]

\[ + G(T_n, T_{n-3}, U^k_{n-3}), \]

which again corresponds to the coarse propagation of a jump (over two intervals) plus a coarse propagation of $U^k_{n-3}$ (over three intervals). This recursion can be repeated, and it will terminate as $U^k_{n-n} = U_0$ is known, leading to an alternative, more compact formulation of the nonlinear ParaExp algorithm:

\[ \text{initialize} \quad U^0_n = G(T_n, T_0, U_0) \quad \text{for} \quad n = 0, 1, \ldots, N, \]

\[ U^k_{n+1} = G(T_n, T_0, U_0) + \sum_{j=1}^{n} G(T_n, T_j, F(T_j, T_{j-1}, U^k_{j-1}) - G(T_j, T_{j-1}, U^k_{j})) \].

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.

We finally 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 Dirchlet 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...
\[
\begin{bmatrix}
u \\
v' \\
\end{bmatrix} = \begin{bmatrix} O & I \\ L & O \end{bmatrix} \begin{bmatrix} u \\
v \end{bmatrix} + \begin{bmatrix} 0 \\
\alpha u^2 \end{bmatrix},
\]

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

In Figure 1 we show the reference solutions \(u(t, x)\) for varying \(\alpha \in \{0, 2, 4, 6, 8, 2\}\) 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. We 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 \texttt{expm}. This is because our reference solution has been computing via \texttt{ode15s} and is of lower accuracy.

For \(\alpha = 2\) we solve a mildly nonlinear problem and our nonlinear ParaExp algorithm achieves an error of order \(\approx 1e-6\) over all time slices already after 5 iterations. For \(\alpha = 4\) it requires 7 iterations to achieve this. As the parameter \(\alpha\) increases further, the nonlinear character of the wave equation becomes more pronounced and the nonlinear ParaExp algorithm becomes less efficient. For \(\alpha \approx 9\) the solution \(u(t, x)\) appears to have a singularity in the time-space domain of interest.

References

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).