A scalable adaptive parareal in time algorithm with online stopping criterion

Yvon Maday and Olga Mula

VII Workshop on Parallel in Time Methods
Roscoff 02 – 05/05/2018
The classical parareal in time algorithm

Let $G$ and $F$ be the coarse and fine propagators of an evolution problem. If $k = 0$,

$$
\begin{align*}
y_0^N &= G(T_N, \Delta T, y_{0}^{N-1}), \quad 1 \leq N \leq N. \\
y_0^0 &= u(0).
\end{align*}
$$

If $k \geq 1$,

$$
\begin{align*}
y_k^N &= G(T_{N-1}, \Delta T, y_k^{N-1}) + F(T_{N-1}, \Delta T, y_k^{N-1}) - G(T_{N-1}, \Delta T, y_{k-1}^{N-1}), \\
y_k^0 &= u(0).
\end{align*}
$$

Two major obstructions

1. Parallel efficiency:
   - $\text{eff} \approx 1/K$
   - Problem: repeated use of $F$

2. No online stopping criteria $\rightarrow$ Need for a posteriori estimators
Motivations: scalability and online stopping criteria

Our approach

- Reformulate rigorously the algorithm in an infinite dimensional setting.
- Derive implementable versions where time dependent subproblems are solved at increasing accuracy across the parareal iterations.

Natural by-products

- Errors measured w.r.t. exact solution and not a finely discretized one.
- Online stopping criteria with a posteriori error estimators
- Cost to achieve a certain final accuracy is designed to be near-minimal.
Setting and notations

Let $\mathbb{U}$ be a Banach space over a domain $\Omega \subset \mathbb{R}^d$,

**Problem:** find $u \in C^1([0, T], \mathbb{U})$ solution to

$$u'(t) + A(t, u(t)) = 0, \quad t \in [0, T],$$

$$u(0) = u_0 \in \mathbb{U}$$

**Propagator:**

- $\mathcal{E}(t, s, w) = \mathcal{E}$ (initial time, step, initial condition in $\mathbb{U}$)
  
  $\mathcal{E}(0, t, u_0) = u(t)$

- For any $\zeta > 0$, $[\mathcal{E}(t, s, w); \zeta]$ is an element of $\mathbb{U}$ satisfying

  $$\|\mathcal{E}(t, s, w) - [\mathcal{E}(t, s, w); \zeta]\| \leq \zeta s (1 + \|w\|).$$

**Discretization in time:** $T_0 = 0 < T_1 < \cdots < T_N = T$.

**Goal:** For a given target accuracy $\eta$, build $\tilde{u}(T_N)$ such that

$$\max_{0 \leq N \leq N} \|u(T_N) - \tilde{u}(T_N)\| \leq \eta.$$
Reformulation of the parareal in time algorithm

**Coarse propagator** $G$: For any $t \in [0, T]$ and $s \in [0, T - t]$,

$$G(t, s, x) = [\mathcal{E}(t, s, x), \varepsilon_G] \iff \|\delta G(t, s, x)\| \leq s(1 + \|x\|)\varepsilon_G$$

$$\|G(t, s, x) - G(t, s, y)\| \leq (1 + C_c s)\|x - y\|,$$

$$\|\delta G(t, s, x) - \delta G(t, s, y)\| \leq C_d s\varepsilon_G \|x - y\|$$

where $\delta G := \mathcal{E} - G$.

**Ideal parareal iterations:** We build a sequence $(y^N_k)_k$ of approximations of $u(T_N)$ for $0 \leq N \leq N$ following the recursive formula

$$\begin{cases}
y_0^{N+1} = G(T_N, \Delta T, y_0^N), & 0 \leq N \leq N - 1 \\
y_{k+1}^{N+1} = G(T_N, \Delta T, y_{k+1}^N) \\
+ \mathcal{E}(T_N, \Delta T, y_k^N) - G(T_N, \Delta T, y_k^N), & 0 \leq N \leq N - 1, \ k \geq 0, \\
y_0^0 = u(0).
\end{cases}$$
Convergence analysis

We introduce the quantities

\[ \mu := \frac{e^{C_c T}}{C_d} \max_{0 \leq N \leq N} (1 + \|u(T_N)\|), \quad \text{and} \quad \tau := C_d T e^{-C_c \Delta T \varepsilon G}. \]

Theorem (Convergence of the ideal iteration (see [GH08]))

If \( G \) and \( \delta G \) satisfy the previous hypothesis, then,

\[ \max_{0 \leq N \leq N} \|u(T_N) - y_k^N\| \leq \mu \frac{\tau^{k+1}}{(k+1)!}, \quad \forall k \geq 0. \]

Sufficient condition to converge:

\[ \tau < 1 \iff \varepsilon_G < \frac{1}{C_d T e^{C_c \Delta T}} \quad \text{(Coarse solver cannot be too coarse)} \]
Ideal parareal iterations: We build a sequence \((y_k^N)\) of approximations of \(u(T_N)\) for \(0 \leq N \leq N\) following the recursive formula

\[
\begin{align*}
    y_0^{N+1} &= G(T_N, \Delta T, y_0^N), & 0 \leq N \leq N - 1 \\
    y_{k+1}^{N+1} &= G(T_N, \Delta T, y_k^N) \\
    &+ \mathcal{E}(T_N, \Delta T, y_k^N) - G(T_N, \Delta T, y_k^N), & 0 \leq N \leq N - 1, \ k \geq 0, \\
    y_0^0 &= u(0).
\end{align*}
\]

Feasible parareal iterations: We build a sequence \((\tilde{y}_k^N)\) of approximations of \(u(T_N)\) for \(0 \leq N \leq N\) following the recursive formula

\[
\begin{align*}
    \tilde{y}_0^{N+1} &= G(T_N, \Delta T, \tilde{y}_0^N), & 0 \leq N \leq N - 1 \\
    \tilde{y}_{k+1}^{N+1} &= G(T_N, \Delta T, \tilde{y}_k^N) \\
    &+ \mathcal{E}(T_N, \Delta T, y_k^N), \zeta_k^N \right] - G(T_N, \Delta T, \tilde{y}_k^N), & 0 \leq N \leq N - 1, \ k \geq 0, \\
    \tilde{y}_0^0 &= u(0).
\end{align*}
\]

Question: minimal accuracy \(\zeta_k^N\) to preserve the convergence rate of ideal scheme?
We keep the same notations

\[ \mu := \frac{e^{C_c T}}{C_d} \max_{0 \leq N \leq N} (1 + \|u(T_N)\|), \quad \text{and} \quad \tau := C_d T e^{-C_c \Delta T} \varepsilon_G. \]

**Theorem (Convergence of the feasible iteration [MM18])**

Let \( G \) and \( \delta G \) satisfy the previous hypothesis.

Let \( k \geq 0 \) be any given positive integer.

If for all \( 0 \leq p < k \) and all \( 0 \leq N < N \), the approximation \([E(T_N, \Delta T, \zeta^N_p)]\) has accuracy

\[ \zeta^N_p \leq \zeta_p := \frac{\varepsilon^{p+2}}{(p + 1)!}, \]

then

\[ \max_{N \in \{0, \ldots, N\}} \|u(T_N) - \tilde{y}_k^N\| \leq \mu \frac{(\varepsilon_G + \tau)^{k+1}}{(k + 1)!}. \]
Parallel efficiency

**Assumption 1:** The numerical cost to realize $[E(T_N, \Delta T, y_k^N), \zeta_k]$ is

$$\text{cost}(\zeta_k, \Delta T) \simeq \Delta T \zeta_k^{-1/\alpha}$$

with $\alpha > 0$ being linked to the order of the numerical scheme.

**Assumption 2:** The numerical cost of the coarse solver is negligible.

**Assumption 3:** $\tilde{\tau} := \varepsilon_G + \tau = \varepsilon_G + C_d T e^{-C_c \Delta T} \varepsilon_G < 1$.

**Lemma (see [MM18])**

$$\text{eff}(\eta, [0, T]) = \frac{\text{cost}_{AP}(\eta, [0, T])}{\text{cost}_{seq}(\eta, [0, T])} = \frac{1 - \tau^{1/\alpha}}{1 - \tau^{K(\eta)/\alpha}} \sim \frac{1}{1 + \varepsilon_G^{1/\alpha}}.$$

and

$$\text{speed-up}(\eta, [0, T]) = N \text{eff}(\eta, [0, T]) \sim N \frac{1}{1 + \varepsilon_G^{1/\alpha}}.$$
Connection to other works/approaches

Classical formulation of parareal: We can interpret the fine solver as

$$\mathcal{F}(T_N, \Delta T, w) = [\mathcal{E}(T_N, \Delta T, w), \zeta_F],$$

where $\zeta_F$ is small and kept constant across the parareal iterations.

Improvement of speed-up with info from previous iterations:

- Coupling of the parareal algorithm with spatial domain decomposition (see [MT05, Gue12, ABGM17]).
- Combination of the parareal algorithm with iterative high order methods in time like spectral deferred corrections (see [MW08, Min10, MSB⁺15]).
- Solution of internal fixed points initialized with solutions at previous parareal iterations (work in progress, see [Mul14]).
- In a similar spirit, applications of the parareal algorithm to solve optimal control problems (see [MT05, MST07]).
An example with obstructions

The brusselator system: We consider the system

\[
\begin{align*}
    x' &= 1 + x^2 y - 4x \\
    y' &= 3x - x^2 y,
\end{align*}
\]

for \( t \in [0, 18] \) and with initial condition \( x(0) = 0 \) and \( y(0) = 1 \).

We set

\[ \eta = 7.10^{-5} \]

and implement the algorithm with

\[ N = 60, \quad \Delta T = \frac{T}{N} = 3.10^{-1}. \]

Coarse solver \( G \): Explicit RK4 of step \( \Delta T \rightarrow \varepsilon_G = 5.10^{-1} \).

Propagations \( [\mathcal{E}(T_N, \Delta T, y_N^k); \zeta_k] \): Explicit RK4 with time step \( \delta t \) dyadically refined until accuracy \( \zeta_k \) is reached.
**Results:** Convergence in 7 parareal iterations, so $k = 0, 1, \ldots, 6$.

**Figure:** Left: Trajectory of the brusselator system over $[0, 12]$. Right: Convergence history of the adaptive parareal algorithm in the whole interval $[0, 18]$. 
Refinements in $\delta t$ to build $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k]$:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\delta t = \Delta T$</th>
<th>$\delta t = \Delta T/2$</th>
<th>$\delta t = \Delta T/2^2$</th>
<th>$\delta t = \Delta T/2^3$</th>
<th>$\delta t = \Delta T/2^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>54</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>54</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>52</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>46</td>
<td>8</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>30</td>
<td>19</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>30</td>
<td>18</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Table: Number of time steps of different sizes $\delta t$ at each iteration $k$.

Task imbalance:

- Some intervals are more refined than others and take longer to compute.
- Need for a rebalancing strategy.
An example with obstructions

**Speed-up adaptive parareal:**

\[
\text{speed-up}_{\text{SA}} = \frac{T_{\text{seq}}(\eta)}{T_{\text{SA}}(\eta)} = 1.96, \quad \text{eff}_{\text{SA}} = \frac{\text{speed-up}_{\text{SA}}}{N} = 3.28 \cdot 10^{-2}.
\]

**Remark:** The sequential solver for the comparison has accuracy \(\eta\) with the largest possible \(\delta t\) when we search among dyadic refinements.

**Speed-up plain parareal:**

\[
\text{speed-up}_{\text{PP}} = \frac{T_{\text{seq}}(\eta)}{T_{\text{AP}}(\eta)} \approx 1.62, \quad \text{eff}_{\text{PP}} = \frac{\text{speed-up}_{\text{PP}}}{N} \approx 2.7 \cdot 10^{-2}.
\]
A trivial example with good efficiency

The circular trajectory: We consider the system

\[
\begin{aligned}
    x'(t) &= -y(t), \\
    y'(t) &= x(t),
\end{aligned}
\]

for \( t \in [0, 3] \) and with initial condition \( x(0) = 0 \) and \( y(0) = 1 \).

We set

\[ \eta = 10^{-3} \]

and implement the algorithm with

\[ N = 8, \quad \Delta T = \frac{T}{N} = 3.75 \cdot 10^{-1}. \]

Coarse solver \( G \): Explicit Euler of step \( \Delta T \longrightarrow \varepsilon_G = 7.12 \cdot 10^{-1} \).

Propagations \( [E(T_N, \Delta T, y_k^N); \zeta_k] \): Explicit Euler with time step \( \delta t \) dyadically refined until accuracy \( \zeta_k \) is reached.
**A trivial example with good efficiency**

**Results:** Convergence in 6 parareal iterations, so \( k = 0, 1, \ldots, 5 \).

**Figure:** Trajectories and convergence history of the adaptive parareal algorithm
A trivial example with good efficiency

Refinements in $\delta t$ to build $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k]$:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\delta t$ to compute $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k^N]$</th>
<th>cost([\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k^N])]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\Delta T/2 \approx 1.9.10^{-1}$</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>$\Delta T/2^2 \approx 9.4.10^{-2}$</td>
<td>$2^2$</td>
</tr>
<tr>
<td>2</td>
<td>$\Delta T/2^4 \approx 2.3.10^{-2}$</td>
<td>$2^4$</td>
</tr>
<tr>
<td>3</td>
<td>$\Delta T/2^7 \approx 2.9.10^{-3}$</td>
<td>$2^7$</td>
</tr>
<tr>
<td>4</td>
<td>$\Delta T/2^9 \approx 7.3.10^{-4}$</td>
<td>$2^9$</td>
</tr>
</tbody>
</table>

**Table:** Time steps $\delta t$ and cost to compute $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k^N]$ at each iteration $k$. 
A trivial example with good efficiency

**Speed-up adaptive parareal:**

\[
\text{speed-up}_{SA} = \frac{T_{\text{seq}}(\eta)}{T_{SA}(\eta)} = 5, \quad \text{eff}_{SA} = \frac{\text{speed-up}_{SA}}{N} = 0.65.
\]

**Remark:** The sequential solver for the comparison has accuracy \( \eta \) with the largest possible \( \delta t \) when we search among dyadic refinements.

**Speed-up plain parareal:**

\[
\text{speed-up}_{PP} = \frac{T_{\text{seq}}(\eta)}{T_{AP}(\eta)} \approx 1.96, \quad \text{eff}_{PP} = \frac{\text{speed-up}_{PP}}{N} \approx 0.25.
\]
Conclusions and future works

The adaptive parareal algorithm:
- Promising approach to significantly improve scalability
- Measures errors measured w.r.t. exact solution and not a finely discretized one.
- Gives naturally an online stopping criterion
- Is designed to converge near-optimally and limit numerical costs

Future works:
- Implement and analyze rebalancing scheme
- Use a posteriori error estimators with space-time fem
- Analyze advantages to re-use previous informations (first results in [Mul14]).


