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

Motivations: scalability and online stoping criteria

The classical parareal in time algorithm

Let $\mathcal{G}$ and $\mathcal{F}$ be the coarse and fine propagators of an evolution problem.

If $k = 0$,

$$\begin{cases} y_0^N &= \mathcal{G}(T_N, \Delta T, y_0^{N-1}),\ 1 \leq N \leq \underline{N}. \\ y_0^0 &= u(0). \end{cases}$$

If $k \geq 1$,

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

Two major obstructions

1. Parallel efficiency:
   - **eff** $\approx 1/K$
   - Problem: repeated use of $\mathcal{F}$
2. No online stopping criteria $\longrightarrow$ Need for a posteriori estimators

Motivations: scalability and online stoping 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 $u \in \mathcal{C}^1([0,T],\mathbb{U})$ solution to

$$u'(t) + \mathcal{A}(t, u(t)) = 0, \quad t \in [0,T],$$
$$u(0) = u_0 \in \mathbb{U}$$

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: $T_0 = 0 < T_1 < \cdots < T_{\underline{N}} = T$.

Goal: For a given taget accuracy η, build $\tilde{u}(T_N)$ such that

$$\max_{0 \leq N \leq \underline{N}} \|u(T_N) - \tilde{u}(T_N)\| \leq \eta.$$

Reformulation of the parareal in time algorithm

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

$$\mathcal{G}(t,s,x) = [\mathcal{E}(t,s,x), \varepsilon_{\mathcal{G}}] \quad \Leftrightarrow \quad \|\delta\mathcal{G}(t,s,x)\| \leq s(1+\|x\|)\varepsilon_{\mathcal{G}}$$
$$\|\mathcal{G}(t,s,x) - \mathcal{G}(t,s,y)\| \leq (1 + C_c s)\|x-y\|,$$
$$\|\delta\mathcal{G}(t,s,x) - \delta\mathcal{G}(t,s,y)\| \leq C_d s \varepsilon_{\mathcal{G}} \|x-y\|$$

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

Ideal parareal iterations: We build a sequence $(y_k^N)_k$ of approximations of $u(T_N)$ for $0 \leq N \leq \underline{N}$ following the recursive formula

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

Convergence analysis

We introduce the quantities

$$\mu := \frac{e^{C_c T}}{C_d} \max_{0 \le N \le \underline{N}} (1 + \|u(T_N)\|), \quad \text{and} \quad \tau := C_d T e^{-C_c \Delta T} \varepsilon_{\mathcal{G}}.$$

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

If $\mathcal{G}$ and $\delta\mathcal{G}$ satisfy the previous hypothesis, then,

$$\max_{0 \le N \le \underline{N}} \|u(T_N) - y_k^N\| \le \mu \frac{\tau^{k+1}}{(k+1)!}, \quad \forall k \ge 0.$$

Sufficient condition to converge:

$$\tau < 1 \quad \Longleftrightarrow \quad \varepsilon_{\mathcal{G}} < \frac{1}{C_d T e^{C_c \Delta T}}$$ (Coarse solver cannot be too coarse)

Implementable version of algorithm

Ideal parareal iterations: We build a sequence $(y_k^N)_k$ of approximations of $u(T_N)$ for $0 \leq N \leq \underline{N}$ following the recursive formula

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

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

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

Question: minimal accuracy ζ_k^N to preserve the convergence rate of ideal scheme?

Convergence analysis

We keep the same notations

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

Theorem (Convergence of the feasible iteration [MM18])

Let $\mathcal{G}$ and $\delta\mathcal{G}$ satisfy the previous hypothesis.

Let $k \geq 0$ be any given positive integer.

If for all $0 \leq p < k$ and all $0 \leq N < \underline{N}$, the approximation $[\mathcal{E}(T_N, \Delta T, \zeta_p^N)]$ has accuracy

$$\zeta_p^N \leq \zeta_p := \frac{\varepsilon_{\delta\mathcal{G}}^{p+2}}{(p+1)!},$$

then

$$\max_{N \in \{0, \ldots \underline{N}\}} \|u(T_N) - \tilde{y}_k^N\| \leq \mu \frac{(\varepsilon_{\mathcal{G}} + \tau)^{k+1}}{(k+1)!}.$$

Parallel efficiency

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

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

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: $\tilde{\tau} := \varepsilon_{\mathcal{G}} + \tau = \varepsilon_{\mathcal{G}} + C_d T e^{-C_c \Delta T} \varepsilon_{\mathcal{G}} < 1$.

Lemma (see [MM18])

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

and

$$\textbf{speed-up}(\eta, [0, T]) = \underline{N}\, \textbf{eff}(\eta, [0, T]) \sim \underline{N} \frac{1}{(1 + \varepsilon_{\mathcal{G}}^{1/\alpha})}.$$

Connection to other works/approaches

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

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

where $\zeta_{\mathcal{F}}$ 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 progrees, 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

$$\begin{cases} x' = 1 + x^2 y - 4x \\ y' = 3x - x^2 y, \end{cases}$$

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

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

Coarse solver $\mathcal{G}$: Explicit RK4 of step $\Delta T \longrightarrow \varepsilon_{\mathcal{G}} = 5.10^{-1}$.
Propagations $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k]$: Explicit RK4 with time step δt dyadically refined until accuracy ζ_k is reached.

An example with obstructions

Results: Convergence in 7 parareal iterations, so $k = 0, 1, \ldots, 6$.

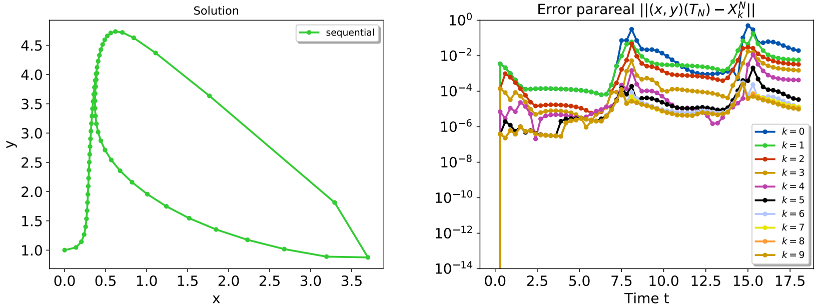

Figure: Left: Trajectory of the brusselator system over $[0, 12]$. Right: Convergence history of the adaptive parareal algorithm in the whole interval $[0, 18]$.

An example with obstructions

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

	Number of subintervals $[T_N, T_{N+1}]$ with time step δt				
k	$\delta t = \Delta T$	$\delta t = \Delta T/2$	$\delta t = \Delta T/2^2$	$\delta t = \Delta T/2^3$	$\delta t = \Delta T/2^4$
0	0	54	6	0	0
1	0	54	4	2	0
2	0	52	4	4	0
3	0	46	8	2	4
4	0	30	19	5	6
5	0	30	18	6	6

Table: Number ot time steps of different sizes δt 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:

$$\textbf{speed-up}_{\mathsf{SA}} = \frac{T_{\mathsf{seq}}(\eta)}{T_{\mathsf{SA}}(\eta)} = 1.96, \quad \textbf{eff}_{\mathsf{SA}} = \frac{\textbf{speed-up}_{\mathsf{SA}}}{\underline{N}} = 3.28.10^{-2}.$$

Remark: The sequential solver for the comparison has accuracy η with the largest possible δt when we search among dyadic refinements.

Speed-up plain parareal:

$$\textbf{speed-up}_{\mathsf{PP}} = \frac{T_{\mathsf{seq}}(\eta)}{T_{AP}(\eta)} \approx 1.62, \quad \textbf{eff}_{\mathsf{PP}} = \frac{\textbf{speed-up}_{\mathsf{PP}}}{\underline{N}} \approx 2.7.10^{-2}.$$

A trivial example with good efficiency

The circular trajectory: We consider the system

$$\begin{cases} x'(t) = -y(t), \\ y'(t) = x(t), \end{cases}$$

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

$$\underline{N} = 8, \quad \Delta T = \frac{T}{\underline{N}} = 3.75.10^{-1}.$$

Coarse solver $\mathcal{G}$: Explicit Euler of step $\Delta T \longrightarrow \varepsilon_{\mathcal{G}} = 7.12.10^{-1}$.
Propagations $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k]$: Explicit Euler with time step δt dyadically refined until accuracy ζ_k is reached.

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

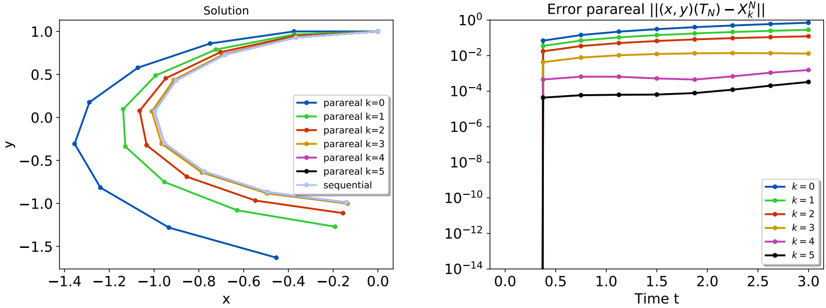

Figure: Trajectories and convergence history of the adaptive parareal algorithm

A trivial example with good efficiency

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

k	δt to compute $[\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k^N]$	**cost**$([\mathcal{E}(T_N, \Delta T, y_k^N); \zeta_k^N])$
0	$\Delta T/2 \approx 1.9.10^{-1}$	2
1	$\Delta T/2^2 \approx 9.4.10^{-2}$	2^2
2	$\Delta T/2^4 \approx 2.3.10^{-2}$	2^4
3	$\Delta T/2^7 \approx 2.9.10^{-3}$	2^7
4	$\Delta T/2^9 \approx 7.3.10^{-4}$	2^9

Table: Time steps δt 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:

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

Remark: The sequential solver for the comparison has accuracy η with the largest possible δt when we search among dyadic refinements.

Speed-up plain parareal:

$$\textbf{speed-up}_{\text{PP}} = \frac{T_{\text{seq}}(\eta)}{T_{AP}(\eta)} \approx 1.96, \quad \textbf{eff}_{\text{PP}} = \frac{\textbf{speed-up}_{\text{PP}}}{\underline{N}} \approx 0.25.$$

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]).

References I

- S. Aouadi, D. Q. Bui, R. Guetat, and Y. Maday, *Convergence analysis of the coupled parareal-schwarz waveform relaxation method*, 2017, in preparation.

- M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, Domain decomposition methods in science and engineering XVII, Springer, 2008, pp. 45–56.

- R. Guetat, *Méthode de parallélisation en temps: Application aux méthodes de décomposition de domaine*, Ph.D. thesis, Paris VI, 2012.

- M. Minion, *A hybrid parareal spectral deferred corrections method*, Comm. App. Math. and Comp. Sci. **5** (2010), no. 2.

- Y. Maday and O. Mula, *A scalable adaptive parareal algorithm with online stopping criterion*, https://hal.archives-ouvertes.fr/hal-01781257 (2018).

References II

- M. L. Minion, R. Speck, M. Bolten, M. Emmett, and D. Ruprecht, *Interweaving PFASST and parallel multigrid*, SIAM Journal on Scientific Computing **37** (2015), no. 5, S244–S263.

- Y. Maday, J. Salomon, and G. Turinici, *Monotonic parareal control for quantum systems*, SIAM Journal on Numerical Analysis **45** (2007), no. 6, 2468–2482.

- Y. Maday and G. Turinici, *The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation*, Domain Decomposition Methods in Science and Engineering, Springer Berlin Heidelberg, 2005, pp. 441–448.

- O. Mula, *Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time*, Ph.D. thesis, Université Pierre et Marie Curie-Paris VI, 2014.

References III

M. L. Minion and S. A. Williams, *Parareal and spectral deferred corrections*, AIP Conference Proceedings **1048** (2008), no. 1, 388–391.