



HAL
open science

Multi-step variant of parareal algorithm

Katia Ait-Ameur, Yvon Maday, Marc Tajchman

► **To cite this version:**

Katia Ait-Ameur, Yvon Maday, Marc Tajchman. Multi-step variant of parareal algorithm. 2019. hal-02321882v2

HAL Id: hal-02321882

<https://hal.science/hal-02321882v2>

Preprint submitted on 3 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-step variant of parareal algorithm

Katia Ait-Ameur, Yvon Maday and Marc Tajchman

Abstract In this paper, we propose a variant of the parareal algorithm for time-dependent problems involving a multi-step time scheme in the coarse and/or fine propagators. This choice can potentially bring higher approximation orders than plain one-step methods but the initialization of each time window needs to be appropriately chosen. In this work, we explore some possible initializations and demonstrate their relevance on a Dahlquist test equation, an advection-diffusion equation and an industrial application.

1 Introduction

In the field of nuclear energy, computations of complex two-phase flows are required for the design and safety studies of nuclear reactors. System codes are dedicated to the thermal-hydraulic analysis of nuclear reactors at system scale by simulating the whole reactor. We are here interested in the Cathare code developed by CEA, [5]. Typical cases involve up to a million of numerical time iterations, computing the approximate solution during long physical simulation times. A space domain decomposition method has already been implemented. To improve the response time,

Katia Ait-Ameur

Laboratoire Jacques Louis Lions (LJLL), Sorbonne Université, 75005 Paris, France
C.E.A, CEA Saclay - DEN/DANS/DM2S/STMF/LMES - 91191 Gif-Sur-Yvette Cedex, France
e-mail: aitameur.katia@gmail.com

Yvon Maday

Laboratoire Jacques Louis Lions (LJLL), Sorbonne Université, 75005 Paris, France
Institut Universitaire de France
e-mail: maday@ann.jussieu.fr

Marc Tajchman

C.E.A, CEA Saclay - DEN/DANS/DM2S/STMF/LMES - 91191 Gif-Sur-Yvette Cedex, France
e-mail: marc.tajchman@cea.fr

we will consider a strategy of time domain decomposition, based on the *parareal method* [11]. The Cathare time discretization is based on a multi-step time scheme (see [8]). In this paper, we derive a strategy to adapt the parareal algorithm to multi-step schemes that is not implemented in the code. The paper is organized as follows. In Section 2, we recall the classical version of the parareal algorithm, and then detail the variant that allows us to use multi-step time schemes for the fine solvers. A couple of remarks on the algorithm will be discussed. The numerical convergence for a simplified test case is shown in Section 3 on a Dahlquist test equation followed by numerical results on an advection-diffusion equation and on an industrial test case with an application on the Cathare code.

2 Parareal algorithm and the multi-step variant

After the discretization of a PDE in space, we obtain an ODE system of the form:

$$\frac{\partial u}{\partial t} + A(t, u) = 0, \quad t \in [0, T], \quad u(t = 0) = u_0 \quad (1)$$

where $A : \mathbb{R} \times \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}^{\mathcal{N}}$, and \mathcal{N} denotes the number of degrees of freedom.

We here recall the classical parareal algorithm as initially proposed in [11], [2], [4]. Let G and F be two propagators such that, for any given $t \in [0, T]$, $s \in [0, T - t]$ and any function w in a Banach space, $G(t, s, w)$ (respectively $F(t, s, w)$) takes w as an initial value at time t and propagates it at time $t + s$. The full time interval is divided into N^c sub-intervals $[T^n, T^{n+1}]$ of size ΔT that will each be assigned to a processor. The algorithm is defined using two propagation operators:

- $G(T^n, \Delta T, u^n)$ computes a coarse approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (low computational cost)
- $F(T^n, \Delta T, u^n)$ computes a more accurate approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (high computational cost)

Starting from a coarse approximation u_0^n at times T^0, T^1, \dots, T^{N^c} , obtained using G , the parareal algorithm performs for $k = 0, 1, \dots$ the following iteration:

$$u_{k+1}^{n+1} = G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^n) - G(T^n, \Delta T, u_k^n)$$

In the parareal algorithm, the value $u(T^n)$ is approximated by u_k^n at each iteration k with an accuracy that tends rapidly to the one achieved by the fine solver, when k increases. The coarse approximation G can be chosen much less expensive than the fine solver F by the use of a scheme with a much larger time step (even $\delta T = \Delta T$) $\delta T \gg \delta t$ (time step of the fine solver) or by using a reduced model. All the fine propagations are made in parallel over the time windows and the coarse propagations are computed in a sequential way but have a low computational cost. We refer to [12] about the parallel efficiency of parareal and a recent work offering a new formulation of the algorithm to improve the parallel efficiency of the original one. The main

convergence properties were studied in [7] and stability analysis was made in [14], [3].

In the sequel, we will consider the case that the coarse solver is based on a one-step time scheme and the fine solver on a two-step time scheme. Hence we will use the following notation for the fine solver that takes two initial values: $F(t, s, x, y)$, for $t \in [0, T]$, $s \in [0, T - t[$ and x, y in a Banach space.

Example

If one solves (1) with a multi-step time scheme as fine propagator F like the second-order BDF method:

$$\frac{3}{2}u^{j+1} - 2u^j + \frac{1}{2}u^{j-1} = -\delta t A(u^{j+1}, t^{j+1}), \quad j = 1, \dots, N^f, t^{j+1} - t^j = \delta t$$

Here the fine solver reads as: $u^{j+1} = F(t^j, \delta t, u^{j-1}, u^j)$. Now, we apply the parareal algorithm with a coarse grid: T^0, \dots, T^{N^c} where:

$$T^{n+1} - T^n = \Delta T = N^f \delta t.$$

Then we can write: $u(T^n + j\delta t) \simeq u^{n,j}$, $j = 1, \dots, N^f$, $n = 1, \dots, N^c$.

In order to perform the fine propagation, in a given time window $[T^n, T^{n+1}]$, we only need the local initial condition u_k^n and a consistent approximation of $u(T^n - \delta t)$.

In [1], the authors propose a consistent approximation in the context of the simulation of molecular dynamics. The proposed method was linked to the nature of the model and the symplectic character of their algorithm is shown, which is an important property to verify for molecular dynamics.

In the context of our application to the thermohydraulic code Cathare, we want to derive a multi-step variant of parareal that will not be intrusive in the software. We seek a consistent approximation of $u(T^n - \delta t)$. The only fine trajectory at our disposal is $F(T^{n-1}, \Delta T, u_k^{n-2, N^f-1}, u_k^{n-1})$. Its final value at T^n is:

$F(T^{n-1}, \Delta T, u_k^{n-2, N^f-1}, u_k^{n-1})(T^n)$ from which we compute u_{k+1}^n by the parareal correction. Hence, we translate the solution:

$F(T^{n-1}, \Delta T - \delta t, u_k^{n-2, N^f-1}, u_k^{n-1})(T^n - \delta t)$ by the same correction:

$u_{k+1}^n - F(T^{n-1}, \Delta T, u_k^{n-2, N^f-1}, u_k^{n-1})$ and obtain the so called consistent approximation u_{k+1}^{n-1, N^f-1} to initialize the fine propagation in $[T^n, T^{n+1}]$. We now detail our algorithm:

$$\begin{cases} u_0^{n+1} &= G(T^n, \Delta T, u_0^n), \quad 0 \leq n \leq N-1 \\ u_{k+1}^{n+1} &= G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^{n-1, N^f-1}, u_k^n) \\ &\quad - G(T^n, \Delta T, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \\ u_{k+1}^{n, N^f-1} &= F(T^n, \Delta T - \delta t, u_k^{n-1, N^f-1}, u_k^n) + u_{k+1}^{n+1} \\ &\quad - F(T^n, \Delta T, u_k^{n-1, N^f-1}, u_k^n), \quad 0 \leq n \leq N-1, \quad k \geq 0 \end{cases} \quad (2)$$

Another option to treat this issue is to use a one-step time scheme to initialize the fine computation or to make one iteration with a second-order Runge Kutta method. We will see from the numerical results that these choices modify the fine scheme and prevent the parareal algorithm to converge to the fine solution: even after N^c iterations (where N^c is the number of time windows), the parareal algorithm does not converge to the fine solution with the machine precision but the parareal error stagnates around 10^{-6} .

This method adds consistency with the fine scheme. Also, this strategy can be applied to multi-step time schemes involving several fine time steps preceding the time T^n by applying the same correction to terms taking the form: $u_{k+1}^{n, N^f - i}$, $i = 1, \dots, I$. The convergence analysis will be shown in a forthcoming work.

3 Numerical results

We now show some numerical experiments, first for an ordinary differential equation, the test equation of Dahlquist, then for a partial differential equation, namely the advection-diffusion equation and finally on an industrial application with the Cathare code.

3.1 Dahlquist equation

We first use the Dahlquist test equation :

$$u'(t) = \lambda u(t), \quad t \in (0, T), \quad \text{with } u(0) = 1,$$

discretized by a second-order BDF method. With $\lambda = -1$, $T = 5$, $\Delta T = T/50$, which correspond to 50 processors, and $\delta t = T/5000$, we obtain the convergence curves shown in 3.1. Here, the fine solver is based on a two-step time scheme where the computation of the solution $u^{n, j+1}$ at time $T^n + (j + 1)\delta t$ needs the knowledge of the solutions $u^{n, j}$ and $u^{n, j-1}$ at times $T^n + j\delta t$ and $T^n + (j - 1)\delta t$, respectively. We use the multi-step variant of parareal (2) to initialize the fine solver in each time window, starting from the parareal iteration $k \geq 2$. At the parareal iteration $k = 1$, we use a Backward Euler method to initialize the fine solver since we don't have the fine solution yet. The coarse solver is based on a one-step time scheme, namely the Backward Euler method. We plot the relative error in $L^\infty(0, T)$ between the fine solution computed in a sequential way and the parareal solution as a function of iteration number for the classical parareal algorithm where the Backward Euler method is used at each iteration for the initialization of the fine solver (circles), and for the multi-step variant of the parareal algorithm that we introduced in the previous section (squares). We see in the Fig. 1 (without multi-step) that starting from the fourth parareal iteration the error stagnates around 10^{-6} without recovering the fine

solution at the machine precision, even after 50 iterations. On the other hand, we see in Fig. 1 (with multi-step) that the error continues to decrease after the fourth parareal iteration until reaching the machine precision at the eleventh iteration. In this case, if we don't use the multi-step variant of parareal, we lose one of the well known property of parareal: to recover the fine solution at the machine precision after N^c iterations.

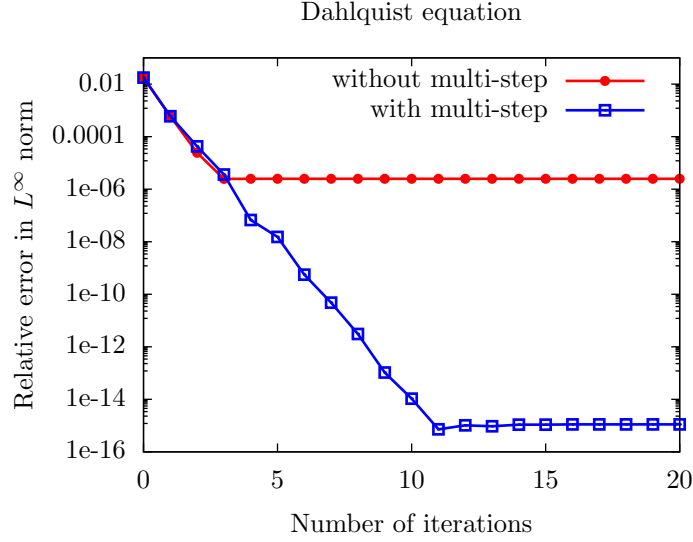


Fig. 1 Convergence of the multi-step parareal for the Dahlquist test equation

3.2 Advection-diffusion equation

We now study the behavior of the multi-step parareal (2) applied to the advection-diffusion equation:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} = 0, & (x, t) \in (0, 2\pi) \times (0, T) \\ u(x, 0) = u_0(x), & x \in (0, 2\pi) \\ u(x, t) = u(x + 2\pi, t), & t \in (0, T) \end{cases} \quad (3)$$

We have chosen a spectral Fourier approximation in space (truncated series with $N = 16$) and a second-order BDF method in time for a propagation over $[0, 2]$. The parareal in time algorithm is implemented with $\Delta T = 0.1$ and $\delta t = 10^{-3}$. We have

chosen the following initial condition:

$$u_0(x) = \sum_{l=-N/2+1}^{l=N/2} \hat{u}_l e^{ilx}, \text{ with } \hat{u}_l = \frac{\text{sgn}(l)}{|l|^p}.$$

We choose $p = 4$, hence the initial condition is sufficiently regular.

The coarse and fine solvers are the same as in the previous numerical example and we use the same initialization of the fine solver at the first parareal iteration. We plot the following error as a function of iteration number: $E_k^n = \frac{\max_n \|U_{seq}(T^n) - U_k^n\|_{L^2((0,2\pi))}}{\max_n \|U_{seq}(T^n)\|_{L^2((0,2\pi))}}$, where U_{seq} is the fine solution computed in a sequential way. In Fig. 2 (circles), we observe a similar behavior as in the previous case, the error stagnates around 10^{-6} after the fifth parareal iteration without reaching the machine precision when the number of iterations is equal to the number of time windows (20 in this case). In Fig. 2 (squares), using the multi-step variant, the error continues to decrease after the fifth iteration until reaching the machine precision around 10^{-16} at iteration number 20.

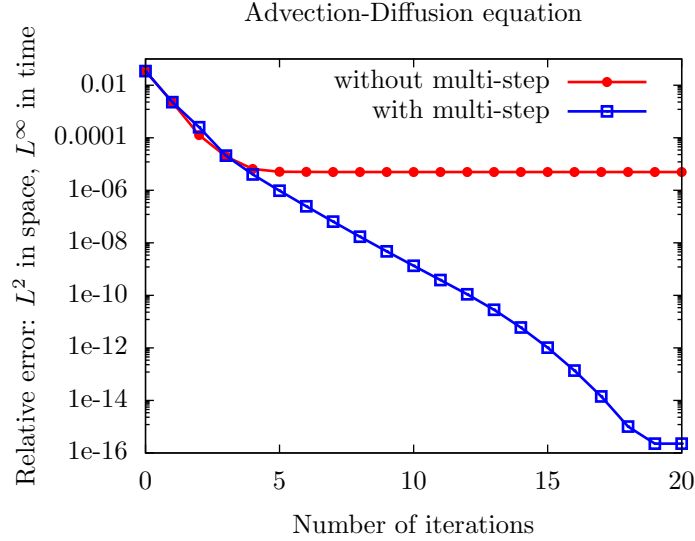


Fig. 2 Convergence of the multi-step parareal for the advection-diffusion equation

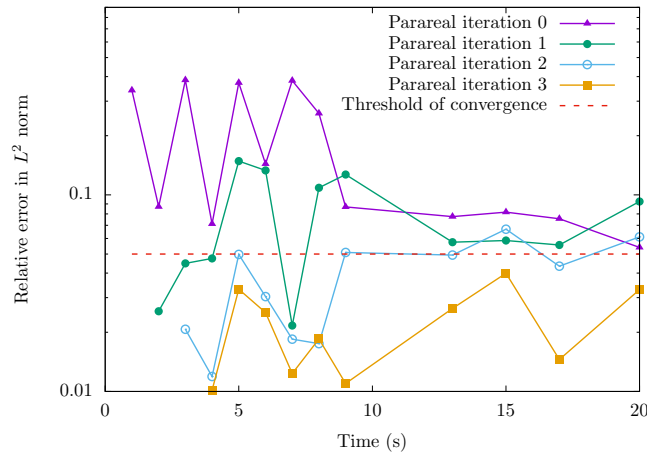


Fig. 3 Multi-step parareal for an industrial application with Cathare code

this study show that this variant converges numerically on different examples: the two simple test equations allow to see clearly the advantage of our strategy. The application on an industrial code shows its efficiency on a more realistic test case without being intrusive in the software.

The convergence analysis of this algorithm will be the subject of a forthcoming paper. The extension of this method to the use of a multi-step scheme in the coarse solver will be also investigated. This point was treated in [1] and a similar strategy will be studied.

Acknowledgements This research is partially supported by ANR project CINE-PARA (ANR-15-CE23-0019).

References

1. C. Audouze, M. Massot, and S. Volz. Symplectic multi-time step parareal algorithms applied to molecular dynamics. <http://hal.archives-ouvertes.fr/hal-00358459/fr/>, 2009.
2. L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66:p. 057701, 2002.
3. G. Bal. Parallelization in time of (stochastic) ordinary differential equations. <http://www.columbia.edu/~gb2030/PAPERS/parallelttime.pdf>, 2003.
4. G. Bal and Y. Maday. A "parareal" time discretization for non-linear pde's with application to the pricing of an american put. *Recent developments in domain decomposition methods*, 23:pp. 189–202, 2002.
5. D. Bestion. The physical closure laws in the cathare code. *Nuclear Engineering and Design*, vol. 124:pp 229–245, 1990.
6. D.A. Drew and S.L. Passman. *Theory of multicomponent fluids*. Springer-Verlag, New-York, 1999.

7. M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):pp. 556–578, 2007.
8. E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, Second Revised Edition, 1993.
9. G.F. Hewitt, J.M. Delhay, and N. Zuber. *Multiphase science and technology*, volume vol. 6. 1991.
10. M. Ishii. *Thermo-fluid dynamic theory of two-phase flow*. Eyrolles, Paris, 1975.
11. J.-L. Lions, Y. Maday, and G. Turinici. Résolution par un schéma en temps "pararéel". *C. R. Acad. Sci. Paris*, 332(7):pp. 661–668, 2001.
12. Y. Maday and O. Mula. An adaptive parareal algorithm. Submitted, <https://arxiv.org/pdf/1909.08333.pdf>, 2019.
13. M. Ndjinga. Influence of interfacial pressure on the hyperbolicity of the two-fluid model. *C. R. Acad. Sci. Paris, Ser. I* 344:pp. 407–412, 2007.
14. G.A. Staff and E.M. Ronquist. Stability of the parareal algorithm. *Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering*, vol. 40:pp. 449–456, 2005.