Multi-step variant of the parareal algorithm: convergence analysis and numerics

Katia Ait-Ameur, Yvon Maday

ESAIM: M2AN 58 (2024) 673–694
https://doi.org/10.1051/m2an/2024014

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

MULTI-STEP VARIANT OF THE PARAREAL ALGORITHM: CONVERGENCE ANALYSIS AND NUMERICS

Katia Ait-Ameur[1,*] and Yvon Maday[2,3]

Abstract. In this paper, we consider the problem of accelerating the numerical simulation of time dependent problems involving a multi-step time scheme by the parareal algorithm. The parareal method is based on combining predictions made by a coarse and cheap propagator, with corrections computed with two propagators: the previous coarse and a precise and expensive one used in a parallel way over the time windows. A multi-step time scheme can potentially bring higher approximation orders than plain one-step methods but the initialisation of each time window needs to be appropriately chosen. Our main contribution is the design and analysis of an algorithm adapted to this type of discretisation without being too much intrusive in the coarse or fine propagators. At convergence, the parareal algorithm provides a solution that coincides with the solution of the fine solver. In the classical version of parareal, the local initial condition of each time window is corrected at every iteration. When the fine and/or coarse propagators is a multi-step time scheme, we need to choose a consistent approximation of the solutions involved in the initialisation of the fine solver at each time windows. Otherwise, the initialisation error will prevent the parareal algorithm to converge towards the solution with fine solver's accuracy. In this paper, we develop a variant of the algorithm that overcome this obstacle. Thanks to this, the parareal algorithm is more coherent with the underlying time scheme and we recover the properties of the original version. We show both theoretically and numerically that the accuracy and convergence of the multi-step variant of parareal algorithm are preserved when we choose carefully the initialisation of each time window.

Mathematics Subject Classification. 65M12, 65N55, 65Y05, 65Y20, 65L06.

Received February 26, 2023. Accepted March 1, 2024.

1. Introduction

Solving complex models with high accuracy and within a reasonable computing time has motivated the search for numerical schemes that exploit efficiently parallel computing architectures. In this work, the model consists of a Partial Differential Equation (PDE) set on a space time domain Ω. In this context, one of the main ideas to parallelize a simulation is to break the problem into subproblems defined over subdomains of a partition of Ω. The domain can potentially have higher dimensionality and be composed of additional variables

Keywords and phrases. Time domain decomposition, multi-step time scheme, parareal algorithm.

[1] Inria, team LEMON / IMAG, Univ. Montpellier, CNRS, 34095 Montpellier, France.
[2] Sorbonne Université, CNRS, Université Paris Cité, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France.
[3] Institut Universitaire de France, Paris, France.
*Corresponding author: katia.ait-ameur@inria.fr

to space and time like velocity or even more specific variables for some problems. There exist algorithms with very good scalability properties for the decomposition of the spatial variable (see [33] or [35] for an overview). Time domain decomposition is more and more considered to complement this strategy when the speed up performance stagnates despite remaining computing resources. This strategy of parallelization can be very efficient especially for long time simulations where the final time T and characteristic time step δt are such that $\frac{T}{\delta t}$ is huge. Research on time parallel algorithms is currently very active and has by now a history of at least 50 years (back to at least [31]) during which several algorithms have been explored (see [19,32] for an overview). Four iterative algorithms have received significant attention, namely Parareal [28], PFASST [14], MGRIT [15] and a specific form of Space-Time Multi-Grid (STMG) [22]. Other algorithms have been proposed, *e.g.* the parallel implicit time-integrator PITA [18] which is very similar to Parareal, the diagonalization technique [30], RIDC [8], ParaExp [20] or REXI [34].

Much work has been done on developing efficient time stepping methods. This work has resulted in very efficient variable step and, in the case of linear multi-step methods, variable order adaptive methods. In particular, the linear multi-step methods have been shown to be very efficient in a number of application areas [3,27]. In contrast to one-step methods, where the numerical solution is obtained solely from the differential equation and the initial value, the algorithm of multi-step time schemes consists of two parts: firstly, a starting procedure which provides $u^1, \ldots, u^{n-1}$ (approximations to the exact solution at the points $t^0 + \delta t, \ldots, t^0 + (n-1)\delta t$) and, secondly, a multi-step formula to obtain an approximation to the exact solution $u(t^0 + n\delta t)$. This is then applied recursively, based on the numerical approximations of n successive steps, to compute $u(t^0 + (n+1)\delta t)$. There are several possibilities for obtaining the missing starting values. In Adams methods [5], they are actually computed using the Taylor series expansion of the exact solution. Another possibility is the use of any one-step method, *e.g.*, a Runge–Kutta method. Other multi-step formulas are based on the numerical differentiation of a given function and are known as backward differentiation formulas (or BDF-methods). These methods are, since the work of [24], widely used for the integration of stiff differential equations. A general theory of multi-step methods was started by the work of Dahlquist [9,10]. As the numerical solution of a multi-step method does not depend only on the initial value problem but also on the choice of the starting values, the definition of the local error is not as straightforward as for one-step methods. A challenge with parallel-in-time methods has been developing strategies that accommodate these highly efficient adaptive methods.

In this work, we report our recent effort to adapt one particular time-parallel algorithm: the parareal in time algorithm, to multi-step time schemes. The parareal method was first introduced in [28] and has been well accepted by the community because it is easily applicable to a relatively large spectrum of problems (some specific difficulties are nevertheless encountered on certain types of PDEs as reported in [11, 18] for hyperbolic systems or [12] for hamiltonian problems). Some limitations persist for the classical version of the parareal algorithm like the parallel efficiency that decreases with the final number of iterations K as $1/K$. This limitation is addressed in [29] that proposes an adaptive variant of the parareal method where the accuracy of the fine solver is dynamically increasing across the parareal iterations (see [6, 25] for the coupling between space domain decomposition methods and the parareal algorithm). Without entering into very specific details of the algorithm at this stage, we can summarize the procedure by saying that we build iteratively a sequence to approximate the exact solution of the problem by a predictor-corrector algorithm. At every iteration, predictions are made by a solver which has to be as numerically inexpensive as possible since it is run on the full time interval. It usually involves coarse physics and/or coarse solution. Corrections involve a solver with high-fidelity physics and high accuracy (and thus expensive) solution which is propagated independently in parallel over small non overlapping time subdomains. In the classical version of the parareal algorithm, the fine solver has a fixed high accuracy across all iterations. It is set to the one that we would use to solve the dynamics at the desired accuracy with a purely sequential solver. One classical property of the parareal algorithm is the so called "consistency" that states that the solution of the parareal algorithm is exact after a number of iterations equal to the number of time windows (of course it is expected that the iterative process converges to a given threshold more rapidly). We would like to preserve this notion of consistency in the context of multi-step times schemes. At each iteration, the local initial conditions are corrected for every time windows until convergence.

Multi-step time schemes require several previous steps for the initialization of the time propagation on each time window. We thus need to choose an initialization that preserves the consistency of the parareal scheme. Otherwise, the initialisation error will prevent the parareal algorithm to converge towards the solution with the fine solver's accuracy. This point was addressed in the context of multigrid in time method in [16, 17]. Here, the authors adapt the MGRIT algorithm framework to the use of multi-step time schemes, the BDF methods. In our work, we propose a variant of the parareal algorithm that overcome this obstacle. Thanks to this, the parareal algorithm is more coherent with the underlying time scheme and we recover the properties of the original version: consistency with the sequential fine solver and same convergence rate.

We present in Section 2 the variant of the parareal algorithm adapted to multi-step time schemes. This method includes additional corrections at previous steps involved in the initialisation of the fine and/or coarse solver at each time window. This choice has the benefit to be non intrusive into the code we seek to parallelise by a time domain decomposition. In Section 2.4, we discuss how the new paradigm can be generalised to multi-step time schemes, not only two-step times schemes, used in the fine and/or the coarse solver. In the last section, we illustrate the performance of the algorithm on numerical examples: the Brusselator and the Van der Pol oscillator. We show that this variant allows the parareal algorithm to converge towards the solution with fine solver's accuracy.

2. A multi-step variant of the parareal algorithm

In this section, after introducing some preliminary notations in Section 2.1, we formulate the new variant of the parareal algorithm adapted to multi-step time schemes (Sect. 2.3). We then present the hypothesis we consider in this article and restrict ourselves to two-step time schemes for the convergence analysis (Sect. 2.2). We prove that the multi-step variant converges with a convergence rate similar to that of the classical parareal algorithm.

2.1. Notations and preliminaries

Let $\mathbb{U}$ be a Banach space of functions defined over a domain $\Omega \subset \mathbb{R}^d$ ($d \geq 1$). Let

$$E : [0,T] \times [0,T] \times \mathbb{U} \to \mathbb{U}, \tag{1}$$

be a propagator, that is, an operator such that, for any given time $t \in [0,T]$, $s \in [0, T-t]$ and any function $w \in \mathbb{U}$, $E(t,s,w)$ takes w as an initial value at time t and propagates it at time $t+s$. We assume that E satisfies the semi group property

$$E(r, t-r, w) = E(s, t-s, E(r, s-r, w)), \quad \forall w \in \mathbb{U}, \quad \forall (r,s,t) \in [0,T]^3, \quad r < s < t. \tag{2}$$

We further assume that E is implicitly defined through the solution $u \in \mathcal{C}^1([0,T], \mathbb{U})$ of the time-dependent problem

$$\begin{cases} u'(t) + \mathcal{A}(t, u(t)) = 0, & t \in [0,T], \\ u(0) \in \mathbb{U}, \end{cases} \tag{3}$$

where $\mathcal{A}$ is an operator from $[0,T] \times \mathbb{U}$ into $\mathbb{U}$ with adequate regularity we shall detail later. Then, given $w \in \mathbb{U}$, $E(t,s,w)$ denotes the solution to (3) at time $t+s$ with initial condition w at time $t \geq 0$. Let us note that s can be negative if it is small enough. The propagator E could also be associated to a discretized version of the evolution equation.

Since, in general, the problem (3) does not have an explicit solution, we seek to approximate the solution of problem (3) at a given target accuracy $\eta > 0$ by a solver S. Given a time discretisation of the time interval $[0,T]$, we denote S the time propagator such that, for any discrete time t and any function $w \in \mathbb{U}$ takes an initial value at time t and propagates it at time $t+s$. S is a generic time propagator that can be a coarse propagator G or a fine time propagator F.

$$\|E(t,s,w) - S(t,s,w)\| \leq \eta s (1 + \|w\|), \tag{4}$$

where $\|\cdot\|$ denotes the norm in $\mathbb{U}$. Thus, $S(t,s,w)$ approximates $E(t,s,w)$ with an accuracy η. We consider a given decomposition of the time interval $[0,T]$ into N subdomains $[T^n,T^{n+1}]$, $n=0,\ldots,N-1$. Without loss of generality, we will take them of uniform size $\Delta T = T/N$ which means that $T^n = n\Delta T$ for $n=0,\ldots,N$. For a given target accuracy $\eta > 0$, the goal of the parareal algorithm is to accelerate the computation of an approximation $\tilde{u}(T^n)$ of $u(T^n)$ such that:

$$\max_{1\leq n\leq N} \|u(T^n) - \tilde{u}(T^n)\| \leq \eta.$$

The classical way to compute such an approximation is to set $\tilde{u}(T^n) = S(0,T^n,u(0)), 1 \leq n \leq N$, where S is some sequential solver in $[0,T]$. On the other hand, the strategy of the parareal algorithm follows the following steps, using two propagation operators:

- $G(t,s,w)$: provides a coarse approximation of the solution of (3), $u(r), r \in [t,t+s]$ with initial condition $u(t) \simeq w$. The coarse propagation is sequential but has a low computational cost.
- $F(t,s,w)$: provides a more accurate approximation of the solution of (3), $u(r), r \in [t,t+s]$ with initial condition $u(t) \simeq w$. The action of F is distributed over N time windows and N processors solve over each interval $[T^n,T^{n+1}]$ of size ΔT instead of solving over $[0,T]$. The fine time step is chosen in order to reach the target approximation of the exact solution with accuracy η defined in (4). This choice has also to respect the stability conditions of the fine solver F. The fine time step size is not necessarily the same for all the coarse intervals and an adaptive time stepping scheme can be used.

The parareal algorithm starts with an initial approximation $u_0^n, n=0,\ldots,N$, at time $T^0,\ldots,T^N$ given by the coarse computation of $u_0^{n+1} = G(T^n,\Delta T,u_0^n)$, with $u_0^0 = u(0)$, and then performs for $k \geq 0$ the correction iteration:

$$\begin{cases} u_0^{n+1} = G(T^n,\Delta T,u_0^n), & 0\leq n\leq N-1, \\ u_{k+1}^{n+1} = G\big(T^n,\Delta T,u_{k+1}^n\big) + F(T^n,\Delta T,u_k^n) - G(T^n,\Delta T,u_k^n), & 0\leq n\leq N-1, \quad k\geq 0. \end{cases} \tag{5}$$

The parareal algorithm consists in building iteratively a series u_k^n of approximations of $u(T_n)$ for $0 \leq n \leq N$ following the recursive formula (5). We initialise $u_{k+1}^0 = u^0$ and then calculate $u_{k+1}^1, u_{k+1}^2, \ldots, u_{k+1}^N$. The second equation in (5) is recursive in k. For a given iteration k, N fine propagations of size ΔT are required, each of them over distinct intervals $[T^n,T^{n+1}]$, each of them with independent initial conditions u_k^n. Since they are independent from each other, they can be computed over N parallel processors and the original computation over $[0,T]$ is decomposed into parallel computations over N subintervals of size ΔT.

2.2. Necessary assumptions

The choice of the solver F determines the quality of the approximation and the computational cost of its implementation. One can potentially bring higher approximation orders than plain one-step methods by using a multi-step time discretisation method or Runge–Kutta time schemes [26]. Multi-step time schemes require several previous steps to compute the solution at a new point in time. For a example with a two-step time scheme, the computation of the solution u^{n+1} at time T^{n+1} depends on the solutions u^n and u^{n-1} at times T^n and T^{n-1}, respectively. At the initial step T^0, a common choice is to set $u^{n-1} = u^n$. In the parareal algorithm, we need to initialise the two-step time scheme for each time window $[T^n,T^{n+1}]$ in a way to recover the fine solution with the target accuracy η. A first option is to make a one-step time scheme iteration to initialise the fine propagations in each time window $[T^n,T^{n+1}]$, as in [4], when the authors propose a consistent approximation in the context of the simulation of molecular dynamics. The proposed method is based on second-order approximations of the solution $u(T^n - \delta t)$ at each time window allowing to initialise the two-step time scheme, the Verlet integrator. The consistency of their algorithm is shown up to the second order. We will see later in the numerical results that making an initialisation error for a multi-step fine solver will prevent the parareal algorithm to obtain the approximation of the exact solution with the desired accuracy. In this work, we propose a new variant of the

parareal algorithm that takes into account a consistent approximation of $u(T^n - \delta t)$ and allows to recover a full consistency and accuracy according to multi-time schemes. In order to define this new algorithm, we introduce a different notation for the fine time propagators that are based on two-step time schemes:

$$F : [0,T] \times [0,T] \times \mathbb{U} \times \mathbb{U} \to \mathbb{U}, \tag{6}$$

such that $F(t,s,w^1,w^2)$ for any given time $t \in [0,T]$, $s \in [0,T-t]$ and any functions $w^1, w^2 \in \mathbb{U}$ takes two initial values at times t and $t-\delta t$ and propagates them at time $t+s$, where δt is the time step of the two-step time scheme F.

In what follows, we analyse the convergence rate of the multi-step variant of parareal algorithm when the coarse solver is a one-step time scheme and the fine one is a two-step time scheme. The algorithm relies on the use of a solvers G and F with the following properties involving the operators:

$$\delta G = E - G, \quad \delta F = E - F.$$

In the following, we assume the exact propagator E is such that $E(t,-\delta t,y))$ is defined for $y \in \mathbb{U}$ and for any $t \in [0,T]$.

Assumptions (H). There exists $\varepsilon_G, C_d, C > 0$ such that for any functions $x, y \in \mathbb{U}$ and for any $t \in [0,T]$ and s such that $s \in [\Delta T, T-t]$, $\frac{s}{\delta t} \gg 1$,

$$\|E(t,s,x) - G(t,s,x)\| \le s(1+\|x\|)\varepsilon_G \Leftrightarrow \|\delta G(t,s,x)\| \le s\varepsilon_G(1+\|x\|), \tag{7}$$
$$\|G(t,s,x) - G(t,s,y)\| \le (1+Cs)\|x-y\|, \tag{8}$$
$$\|F(t,s,x_1,y_1) - F(t,s,x_2,y_2)\| \le (1+Cs)(\|x_1-x_2\| + \|y_1-y_2\|), \tag{9}$$
$$\|\delta G(t,s,x) - \delta G(t,s,y)\| \le C_d s\varepsilon_G\|x-y\|, \tag{10}$$
$$\|(F(t,s,E(t,-\delta t,y_1),y_1) - E(t,s,y_1)) - (F(t,s,E(t,-\delta t,y_2),y_2) - E(t,s,y_2))\| \le Cs\delta t\|y_1-y_2\|, \tag{11}$$
$$\begin{aligned}&\|(F(t,s-\delta t,y_1-\delta_1,y_1) - F(t,s-\delta t,y_2-\delta_2,y_2)) - (F(t,s,y_1-\delta_1,y_1) - F(t,s,y_2-\delta_2,y_2))\| \\ &\quad \le (1+Cs)\delta t(\|\delta_1-\delta_2\| + \|y_1-y_2\|),\end{aligned} \tag{12}$$
$$\|F(t,s,E(t,-\delta t,y),y) - E(t,s,y)\| \le s\varepsilon_F(1+\|y\|). \tag{13}$$

2.3. The multi-step variant of the parareal method

We now detail our algorithm:

$$\begin{cases} u_0^{n+1} &= G(T^n,\Delta T,u_0^n), & 0 \le n \le N-1, \\ u_0^{n,N^f-1} &= u_0^{n+1}, & 0 \le n \le N-1, \\ u_{k+1}^{n+1} &= G\big(T^n,\Delta T,u_{k+1}^n\big) + F\Big(T^n,\Delta T,u_k^{n-1,N^f-1},u_k^n\Big) & \\ &\quad - G(T^n,\Delta T,u_k^n), & 0 \le n \le N-1, \quad k \ge 0, \\ u_{k+1}^{n,N^f-1} &= F\Big(T^n,\Delta T-\delta t,u_k^{n-1,N^f-1},u_k^n\Big) + u_{k+1}^{n+1} & \\ &\quad - F\Big(T^n,\Delta T,u_k^{n-1,N^f-1},u_k^n\Big), & 0 \le n \le N-1, \quad k \ge 0. \end{cases} \tag{14}$$

At this point, several comments are in order. To derive a consistent approximation of $u(T^n - \delta t)$, we use the last fine trajectory at our disposal which 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 $[T^n,T^{n+1}]$.

2.4. Comments and extensions

An important feature of this algorithm is to preserve a well known property of the parareal algorithm:

$$u_k^n = F\big(T^0, T^n - T^0, u^0\big), \quad \text{for} \quad n \le k, \quad n = 0, \dots, N. \tag{15}$$

In our case, the multi-step variant of the parareal algorithm verifies (15) and the additional correction of the solution at time $T^n - \delta t$ leads to:

$$\begin{aligned} u_{k+1}^{n,N^f-1} &= F\Big(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n\Big) + u_{k+1}^{n+1} \\ &\quad - F\Big(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n\Big), && 0 \le n \le N-1, \quad k \ge 0, \\ &= F\Big(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n\Big) + G\big(T^n, \Delta T, u_{k+1}^n\big) \\ &\quad - G(T^n, \Delta T, u_k^n), && 0 \le n \le N-1, \quad k \ge 0. \end{aligned}$$

Hence, the multi-step parareal method satisfies the same property (15) at time $T^n - \delta t$:

$$u_k^{n,N^f-1} = F\big(T^0, T^n - \delta t - T^0, u^0\big), \quad \text{for} \quad n \le k, \quad n = 0, \dots, N. \tag{16}$$

This new variant of the parareal algorithm (14) proposes a consistent approximation of the solution at time $T^n - \delta t$ in a non intrusive way. The initialisation of the fine propagation in each time window has to be appropriately chosen because an initialisation error would be propagated over the whole time interval and would prevent the parareal algorithm to converge towards the target solution. Another option to treat this issue is to use a one-step time scheme or a multi-stage Runge–Kutta method to initialize the fine computation. This option is intrusive since we have to implement new time schemes for the initialisation. Moreover, we will see in Section 4 that this strategy prevents the parareal to converge to the numerical solution with the target accuracy since the first-order scheme error will dominate.

The multi-step parareal adds consistency with the fine scheme. This strategy can be extended to the case of a general multi-step time scheme involving several fine time steps preceding the time T^n. The same correction can be applied to the terms taking the form: $u_{k+1}^{n,N^f-i}, i = 1, \dots, I$.

We detail the algorithm for a multi-step time scheme involving more than one fine time step preceding time T^n.

$$\begin{cases} u_0^{n+1} = G(T^n, \Delta T, u_0^n), & 0 \le n \le N-1, \\ u_{k+1}^{n+1} = G\big(T^n, \Delta T, u_{k+1}^n\big) + F\Big(T^n, \Delta T, u_k^{n-1,N^f-I}, u_k^{n-1,N^f-I+1}, \dots, u_k^n\Big) \\ \qquad - G(T^n, \Delta T, u_k^n), & 0 \le n \le N-1, \quad k \ge 0, \\ u_{k+1}^{n,N^f-i} = F\Big(T^n, \Delta T - i\delta t, u_k^{n-1,N^f-I}, u_k^{n-1,N^f-I+1}, \dots, u_k^n\Big) + u_{k+1}^{n+1} \\ \qquad - F\Big(T^n, \Delta T, u_k^{n-1,N^f-I}, u_k^{n-1,N^f-I+1}, \dots, u_k^n\Big), & i = 0, \dots, I, \\ & 0 \le n \le N-1, \quad k \ge 0, \end{cases} \tag{17}$$

where we denote $F(t, s, w^1, w^2, \dots, w^I)$, the multi-step propagator such that, for any given time $t \in [0, T]$, $s \in [0, T - t]$ and any function $w^1, \dots, w^I \in \mathbb{U}$, F takes I initial values at times $t - i\delta t, i = 0, \dots, I-1$ and propagates it at time $t + s$, where δt is the fine time step. We illustrate the good convergence properties in the next section by applying the parareal algorithm to an ODE system solved by a coarse solver based on a one-step time scheme and a fine solver based on a third-order BDF method.

When the coarse solver is a multi-step time scheme, there exists several options to initialise it on each time window:

– If the coarse time step δT is equal to the size of the time window ΔT, there is no additional correction in the parareal algorithm since the solution at every coarse time steps are updated.

– If $\delta T < \Delta T$, there are intermediate coarse time iterations in each time window. In [4], the initialisation of the coarse solver is addressed and the authors propose a parareal-type correction at time $T^n - \delta T$:

$$u_{k+1}^{n+1-N_{\text{int}}^c} = G\Big(T^n, \Delta T - \delta T, u_{k+1}^{n-N_{\text{int}}^c}, u_{k+1}^n\Big) + F\Big(T^n, \Delta T - \delta T, u_k^{n-1,N^f-1}, u_k^n\Big) - G\Big(T^n, \Delta T - \delta T, u_k^{n-N_{\text{int}}^c}, u_k^n\Big), \quad N_{\text{int}}^c = \frac{\delta T}{\delta t} \quad 0 \le n \le N-1, \quad k \ge 0. \tag{18}$$

3. Convergence analysis of the multi-step parareal method

3.1. Comments on the assumptions

Assumptions (7)–(10) are the classical properties of numerical schemes related to stability and accuracy. The parameters ϵ_G depends on the coarse time step ΔT and on the order accuracy α of the coarse propagator G. Thus $\epsilon_G \simeq \Delta T^\alpha$. The parameters C_d and C depend on the regularity of the operator $\mathcal{A}$ and of the solution u and on final time T^f. Assumptions (8) and (9) are Lipschitz conditions and the quantity ε_G is a small constant which, in the case of the explicit Euler scheme, would be proportional to the time step size. Assumptions (11) and (12) are specific to two-step time schemes in the context of time domain decomposition. There are two sources of error for two-step time schemes:

– the error from the discretisation of the time derivative, common to one-step time schemes;
– the error from the inconsistency between the two initial conditions x_1 and y_1 in $F(t, s, x_1, y_1)$.

In hypothesis (11), there is no inconsistency between the two initial values since x_1 is computed with the exact propagator starting from y_1. Hence, the only remaining errors are:

– the difference between y_1 and y_2;
– the error from the time propagation over a time window of size s;
– the error between the fine and the exact propagators that is proportional to the fine time step δt.

On the other hand, we assume hypothesis (12) holds for $s \ge \Delta T$, the time window size. This hypothesis includes the inconsistency between the two initial values and is denoted δ_1 and δ_2. Hence, we describe here the errors coming from:

– the inconsistency δ_i between the two initial values x_i and y_i of the fine solver;
– the difference between the principal initial values y_1 and y_2;
– the time propagation over time windows of size s.

Example 3.1. Here, we illustrate the validity of hypothesis (12) on a simple linear ODE.

$$\begin{cases} y'(t) = y(t), & t \in [0, T], \\ y^{0a}, & y^{0b} \in \mathbb{R} \text{ given}, \end{cases} \tag{19}$$

where y^{0a}, y^{0b} are the two seed values to initialise the time propagation with the second-order BDF method:

$$\frac{3}{2}y^{n+1} - 2y^n + \frac{1}{2}y^{n-1} = \delta t y^{n+1}. \tag{20}$$

The parameters involved in hypothesis (12) are: s, y_1, y_2, δ_1 and δ_2. In this example, we have:

$$y^{0a} = (y_1 - \delta_1) - (y_2 - \delta_2), \quad y^{0b} = y_1 - y_2.$$

We solve the ODE (19) by the second-order BDF method (20). We obtain the expression of the numerical solution y^n for $n = 0, \ldots, N^f$ with $N^f = \frac{T}{\delta t}$, the number of fine time steps per time window:

$$y^n = \alpha r_1^n + \beta r_2^n,$$

such that:

– $r_1 = \frac{2+\sqrt{1+2\delta t}}{3-2\delta t} = 1 + \delta t + \mathcal{O}(\delta t^2)$.
– $r_2 = \frac{2-\sqrt{1+2\delta t}}{3-2\delta t} = \frac{1}{3} - \frac{\delta t}{9} + \mathcal{O}(\delta t^2)$.
In (20), the term r_2^n tends rapidly to zero when n goes to infinity. Here, we are interested in illustrating Hypothesis (12) on a simple case for s such that the ratio $\frac{s}{\delta t}$ is large. This corresponds to the study of the sequence y^n for large n. In this context, we can consider that the term r_2^n is negligible. Thus we neglect its contribution.
– $\alpha = \frac{r_2(\delta_1-\delta_2)+(1-r_2)(y_1-y_2)}{r_1-r_2}$.
– $\beta = \frac{(r_1-1)(y_1-y_2)-r_1(\delta_1-\delta_2)}{r_1-r_2}$.

In the linear case, we can write hypothesis (12), where the norm here is the absolute value:

$$\left\|y^{N+1} - y^N\right\| \leq (1+Cs)\delta t\left\|y^{0a} - y^{0b}\right\| + (1+Cs)\delta t\left\|y^{0b}\right\|,$$

where: $y^N = F(0, s-\delta t, y^{0a}, y^{0b})$, $y^{N+1} = F(0, s, y^{0a}, y^{0b})$ and $(N+1)$ is the number of fine time steps in a time window of size s: $N+1 = \frac{s}{\delta t} = \frac{\Delta T}{\delta t}$. From the expression of y^n in (20):

$$y^{N+1} - y^N = \alpha r_1^N(r_1 - 1) + \beta r_2^N(r_2 - 1).$$

Neglecting the term r_2^N, we obtain:

$$y^{N+1} - y^N = \frac{r_2}{r_1 - r_2}(1+\Delta T)\delta t(\delta_1 - \delta_2) + \frac{1-r_2}{r_1-r_2}(1+\Delta T)\delta t(y_1 - y_2) + \mathcal{O}\left(\delta t^2\right).$$

Hence, the second-order BDF method verify hypothesis (12), in the linear case which makes it reasonable to extend it to non linear case under suitable assumptions, *e.g.* uniform Lipschitz continuity, on operator $\mathcal{A}$ in (3).

In the proof of convergence, we apply this hypothesis (12) for $y_1 = u_{k-1}^n$, the parareal solution at iteration $(k-1)$ and time T^n, and $y_2 = u(T^n)$, the exact solution at time T^n, hence these two values are very close. On the other hand, $\delta_1 = u_{k-1}^n - u_{k-1}^{n-1,N^f-1}$, where u_{k-1}^{n-1,N^f-1} is the parareal solution at iteration $(k-1)$ and time $T^n - \delta t$, and $\delta_2 = u(T^n) - u(T^n - \delta t)$, where $u(T^n - \delta t)$ is the exact solution at time $T^n - \delta t$ and s is equal to the time window size ΔT.

3.2. Preliminary results

The convergence result of Theorem 3.4 and its proof are helpful to understand the main mechanisms driving the convergence of the algorithm and explaining its behavior. To present it, we introduce the shorthand notation for the error norm:

$$E_k^n := u_k^n - E(T^0, T^n - T^0, u^0), \quad k \geq 0, \quad 0 \leq n \leq N.$$

We introduce the following quantities:

$$\begin{cases} \alpha := C_d \varepsilon_G \Delta T, \\ \mu := C\Delta T \delta t, \\ \beta := 1 + C\Delta T, \\ \gamma_G := \Delta T \varepsilon_G \max_{0\leq n\leq N}(1 + \|u(T^n)\|), \\ \gamma_F := \Delta T \varepsilon_F \max_{0\leq n\leq N}(1 + \|u(T^n)\|), \end{cases} \tag{21}$$

as shorthand notations for the proof of convergence. We denote $\tilde{e}_k^n$ a perturbation of the error term $\|E_k^n\|$ made by the multi-step parareal algorithm.

Proposition 3.2 (Convergence of the error perturbation sequence)**.** *If the sequence* $\{\tilde{e}_k^n\}_{0\leq n\leq N, 0\leq k\leq K}$ *verifies the following recursive inequalities:*

$$\begin{cases} \tilde{e}_0^n \leq \beta \tilde{e}_0^{n-1} + \tilde{\gamma}_G, \\ \tilde{e}_1^n \leq \beta \tilde{e}_1^{n-1} + \tilde{\alpha}\tilde{e}_0^{n-1} + \tilde{\gamma}_F, \\ \tilde{e}_k^n \leq \beta \tilde{e}_k^{n-1} + \tilde{\alpha}\tilde{e}_{k-1}^{n-1} + C^2\delta t \tilde{e}_{k-2}^{n-2}, \end{cases} \tag{22}$$

with $\tilde{e}_0^0 = 0$, *for some* $\tilde{\alpha}$, $\tilde{\gamma}_G$, $\tilde{\gamma}_F$ *such that:*

$$\frac{C^2\delta t}{(\alpha + 3\mu + C\delta t)^2} < 1,$$

then the error perturbation $\tilde{e}_k^n$ *of the multi-step parareal scheme* (14) *is bounded by:*

$$\begin{aligned} \tilde{e}_0^n &\leq \frac{\tilde{\gamma}_G}{CT} N e^{Cn\Delta T}, && n \geq 1, \\ \tilde{e}_k^n &\leq \tilde{\gamma}_G \tilde{\alpha}^k f_k \binom{n}{k+1} \beta^{n-k-1} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} f_{k-1} \binom{n}{k} \beta^{n-k}, && n \geq k+1, \quad k \geq 1. \end{aligned} \tag{23}$$

Proof. The proof of Proposition 3.2 is given in Appendix A. □

In the context of fine propagators based on second-order multi-step time schemes, we assume that the operator $\mathcal{A}$, from system (3), and its derivatives $\frac{\partial \mathcal{A}}{\partial t}$, $\frac{\partial \mathcal{A}}{\partial u}$ are locally Lipschitz. For higher order multi-step time schemes, additional assumptions have to be made on the regularity of operator $\mathcal{A}$ to satisfy the convergence Theorem 3.4.

Lemma 3.3 (Behavior of the multi-step parareal errors)**.** *We denote* $E_k^{n+1}, E_k^{n,N^f-1}, \delta E_k^{n+1}$, *the error terms made by the multi-step parareal algorithm* (14), *defined by:*

$$\begin{cases} E_k^{n+1} = u_k^{n+1} - E\big(T^0, T^{n+1} - T^0, u^0\big), \\ E_k^{n,N^f-1} = u_k^{n,N^f-1} - E\big(T^0, T^{n+1} - \delta t - T^0, u^0\big), \\ \delta E_k^{n+1} = E_k^{n,N^f-1} - E_k^{n+1}, \end{cases} \tag{24}$$

then, there exists $\delta t_0 > 0$ *such that for any* $\delta t \leq \delta t_0$, *the sequences* $\{E_k^n\}_{0\leq n\leq N, 0\leq k\leq K}$, $\{E_k^{n,N^f-1}\}_{0\leq n\leq N, 0\leq k\leq K}$, $\{\delta E_k^n\}_{0\leq n\leq N, 0\leq k\leq K}$ *verify the following recurrence relations:*

$$\big\|E_k^{n+1}\big\| \leq \beta \|E_k^n\| + \left(\alpha + \mu + C\delta t + \frac{C\delta t^2}{2}\right)\big\|E_{k-1}^n\big\| + C\big\|\delta E_{k-1}^n\big\| + \gamma_F, \tag{25}$$

$$\Big\|E_k^{n,N^f-1}\Big\| \leq \beta \|E_k^n\| + \left(\alpha + \mu + 2C\delta t + \frac{C\delta t^2}{2}\right)\big\|E_{k-1}^n\big\| + C(1+\delta t)\big\|\delta E_{k-1}^n\big\| + 3\gamma_F, \tag{26}$$

$$\big\|\delta E_k^{n+1}\big\| \leq C\delta t\big\|\delta E_{k-1}^n\big\| + C\delta t\big\|E_{k-1}^n\big\| + 2\gamma_F. \tag{27}$$

Proof. The proof is in the spirit of existing results from the literature [7,21,28,29] with similar techniques.

If $k = 0$, using definition (14) for u_0^n, we have for $0 \leq n \leq N-1$,

$$\begin{aligned} E_0^{n+1} &= u_0^{n+1} - E\big(T^0, T^{n+1} - T^0, u^0\big), \\ E_0^{n+1} &= G(T^n, \Delta T, u_0^n) - E(T^n, \Delta T, u(T^n)), \\ \big\|E_0^{n+1}\big\| &\leq \|G(T^n, \Delta T, u_0^n) - G(T^n, \Delta T, u(T^n))\| + \|G(T^n, \Delta T, u(T^n)) - E(T^n, \Delta T, u(T^n))\|, \\ &\leq (1 + C\Delta T)\|E_0^n\| + \Delta T \varepsilon_G (1 + \|u(T^n)\|), \\ &\leq \beta \|E_0^n\| + \gamma_G, \end{aligned}$$

where we have used (7) and (8) to derive the second to last inequality.

For $k \geq 1$, starting from (14), we have

$$\begin{aligned} E_k^{n+1} &= u_k^{n+1} - E\big(T^0, T^{n+1} - T^0, u^0\big), \\ &= G(T^n, \Delta T, u_k^n) + F\Big(T^n, \Delta T, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n\Big) - G\big(T^n, \Delta T, u_{k-1}^n\big) - E(T^n, \Delta T, u(T^n)). \end{aligned}$$

In what follows, we add and substract the following quantities to E_k^{n+1}:

– $G(T^n, \Delta T, u(T^n))$ and $E(T^n, \Delta T, u_{k-1}^n)$,

– $E(T^n, \Delta T, u(T^n))$ and $F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))$,

– $F(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n)$ and $F(T^n, \Delta T, u_{k-1}^n - \hat{\delta x}, u_{k-1}^n)$,

where:

$$\begin{aligned}
\delta x &= u(T^n) - u(T^n - \delta t), \quad \tilde{\delta x} = u_{k-1}^n - u_{k-1}^{n-1,N^f-1}, \quad \hat{\delta x} = u_{k-1}^n - E\big(T^n, -\delta t, u_{k-1}^n\big) \\
E_k^{n+1} &= G(T^n, \Delta T, u_k^n) - G(T^n, \Delta T, u(T^n)) - \delta G(T^n, \Delta T, u(T^n)) + \delta G\big(T^n, \Delta T, u_{k-1}^n\big) \\
&\quad + F\big(T^n, \Delta T, E\big(T^n, -\delta t, u_{k-1}^n\big), u_{k-1}^n\big) - E\big(T^n, \Delta T, u_{k-1}^n\big) \\
&\quad - \big(F(T^n, \Delta T, E(T^n, -\delta t, u(T^n)), u(T^n)) - E(T^n, \Delta T, u(T^n))\big) \\
&\quad + F\big(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n\big) - F\Big(T^n, \Delta T, u_{k-1}^n - \hat{\delta x}, u_{k-1}^n\Big) \\
&\quad + F\Big(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - F\big(T^n, \Delta T, u_{k-1}^n - \delta x, u_{k-1}^n\big) \\
&\quad + F(T^n, \Delta T, u(T^n - \delta t), u(T^n)) - E(T^n, \Delta T, u(T^n)).
\end{aligned}$$

Taking norms and using (8)–(11) and (13), we derive:

$$\begin{aligned}
\big\|E_k^{n+1}\big\| &\le (1 + C\Delta T)\|E_k^n\| + C\Delta T \varepsilon_G \big\|E_{k-1}^n\big\| + C\Delta T \delta t \big\|E_{k-1}^n\big\| + C\Big\|\hat{\delta x} - \delta x\Big\| + C\Big\|\delta x - \tilde{\delta x}\Big\| \\
&\quad + \Delta T \varepsilon_F (1 + \|u(T^n)\|). \qquad (28)
\end{aligned}$$

Let us note that in Assumption (9) we can limit the factor $(1 + C\Delta T)$ by a constant $C(T)$ that only depends on the final time T, since we have $\Delta T \le T$.

On the one hand, the term $\delta x - \tilde{\delta x}$ becomes:

$$\delta x - \tilde{\delta x} = u_{k-1}^{n-1,N^f-1} - u(T^n - \delta t) - \big(u_{k-1}^n - u(T^n)\big) = E_k^{n,N^f-1} - E_k^{n+1} = \delta E_k^{n+1}.$$

On the other hand, we derive a bound for the term $\|\delta x - \hat{\delta x}\|$:

$$\Big\|\delta x - \hat{\delta x}\Big\| = \big\|u(T^n - \delta t) - E\big(T^n, -\delta t, u_{k-1}^n\big) - \big(u(T^n) - u_{k-1}^n\big)\big\|.$$

Writing the Taylor expansions of $u(T^n - \delta t)$ and $E(T^n, -\delta t, u_{k-1}^n)$ around T^n and u_{k-1}^n respectively, we obtain formally:

$$\begin{aligned}
u(T^n - \delta t) - u(T^n) &= \delta t \mathcal{A}(T^n, u(T^n)) + \frac{\delta t^2}{2}\left(\frac{\partial \mathcal{A}}{\partial t} + \frac{\partial \mathcal{A}}{\partial u}\mathcal{A}\right)(T^n, u(T^n)) + \mathcal{O}\big(\delta t^3\big), \\
E\big(T^n, -\delta t, u_{k-1}^n\big) - u_{k-1}^n &= \delta t \mathcal{A}\big(T^n, u_{k-1}^n\big) + \frac{\delta t^2}{2}\left(\frac{\partial \mathcal{A}}{\partial t} + \frac{\partial \mathcal{A}}{\partial u}\mathcal{A}\right)\big(T^n, u_{k-1}^n\big) + \mathcal{O}\big(\delta t^3\big).
\end{aligned}$$

Since the operator $\mathcal{A}$ from system (3) and its derivatives $\frac{\partial \mathcal{A}}{\partial t}$, $\frac{\partial \mathcal{A}}{\partial u}$ are locally Lipschitz:

$$\Big\|\delta x - \hat{\delta x}\Big\| \le \left(C\delta t + \frac{C\delta t^2}{2}\right)\big\|E_{k-1}^n\big\| + C\delta t^3.$$

We recall: $\gamma_F = \Delta T \varepsilon_F \max_{0 \le n \le N}(1 + \|u(T^n)\|)$. Since, the fine solver is based on a two-step time then $\varepsilon_F \approx \delta t^2$. Neglecting the contribution of δt^3 in (28) with respect to γ_F is valid only for $\delta t \le \delta t_0$ with some δt_0 chosen small enough. That leads to:

$$\big\|E_k^{n+1}\big\| \le \beta\|E_k^n\| + \left(\alpha + \mu + C\delta t + \frac{C\delta t^2}{2}\right)\big\|E_{k-1}^n\big\| + C\big\|\delta E_{k-1}^n\big\| + \gamma_F.$$

Now, we derive an upper bound for the error term δE_k^{n+1}.

$$\delta E_k^{n+1} = E_k^{n,N^f-1} - E_k^{n+1},$$
$$\delta E_k^{n+1} = u_k^{n,N^f-1} - E\big(T^0, T^{n+1} - \delta t - T^0, u^0\big) - u_k^{n+1} + E\big(T^0, T^{n+1} - T^0, u^0\big).$$

Using the definition of u_k^{n,N^f-1} in the multi-step parareal algorithm (14), we obtain:

$$\delta E_k^{n+1} = F\Big(T^n, \Delta T - \delta t, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - E(T^n, \Delta T - \delta t, u(T^n)) - \Big(F(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n) - E(T^n, \Delta T, u(T^n))\Big),$$

where:

$$\tilde{\delta x} = u_{k-1}^n - u_{k-1}^{n-1,N^f-1}.$$

In what follows, we add and substract the following quantities to δE_k^{n+1}:

$$F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)), \quad F(T^n, \Delta T, u(T^n) - \delta x, u(T^n)), \tag{29}$$

and we obtain:

$$\begin{aligned}\delta E_k^{n+1} &= F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) - E(T^n, \Delta T - \delta t, u(T^n)) \\ &\quad - (F(T^n, \Delta T, u(T^n) - \delta x, u(T^n)) - E(T^n, \Delta T, u(T^n))) \\ &\quad + F\Big(T^n, \Delta T - \delta t, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) \\ &\quad - \Big(F\Big(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))\Big).\end{aligned}$$

Taking norms and using (12) and (13), we derive:

$$\begin{aligned}\big\|\delta E_k^{n+1}\big\| &\le 2\Delta T \varepsilon_F (1 + \|u(T^n)\|) + C\delta t \Big\| u_{k-1}^{n-1,N^f-1} - u(T^n - \delta t) - (u_{k-1}^n - u(T^n))\Big\| \\ &\quad + C\delta t \big\| u_{k-1}^n - u(T^n)\big\|, \\ \big\|\delta E_k^{n+1}\big\| &\le C\delta t \big\|\delta E_{k-1}^n\big\| + C\delta t \big\| E_{k-1}^n\big\| + 2\gamma_F.\end{aligned}$$

Let us note that in Assumption (12) we can limit the factor $(1 + C\Delta T)$ by a constant $C(T)$ that only depends on the final time T, since we have $\Delta T \le T$.

Now, we derive an upper bound for the error term E_k^{n,N^f-1}.

$$\begin{aligned}E_k^{n,N^f-1} &= u_k^{n,N^f-1} - E(T^0, T^{n+1} - \delta t - T^0, u^0), \\ &= F\Big(T^n, \Delta T - \delta t, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n\Big) + u_k^{n+1} - F\Big(T^n, \Delta T, u_{k-1}^{n-1,N^f-1}, u_{k-1}^n\Big) - E(T^n, \Delta T - \delta t, u(T^n)).\end{aligned}$$

In what follows, we add and substract the same quantities (29) to E_k^{n,N^f-1} as those for the term δE_k^{n+1}.

$$\begin{aligned}E_k^{n,N^f-1} &= u_k^{n+1} - E(T^n, \Delta T, u(T^n)) + F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) - E(T^n, \Delta T - \delta t, u(T^n)) \\ &\quad - (F(T^n, \Delta T, u(T^n) - \delta x, u(T^n)) - E(T^n, \Delta T, u(T^n))) \\ &\quad + F\Big(T^n, \Delta T - \delta t, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - F(T^n, \Delta T - \delta t, u(T^n) - \delta x, u(T^n)) \\ &\quad - \Big(F\Big(T^n, \Delta T, u_{k-1}^n - \tilde{\delta x}, u_{k-1}^n\Big) - F(T^n, \Delta T, u(T^n) - \delta x, u(T^n))\Big).\end{aligned}$$

Taking norms and using (12) and (13), we derive:

$$\left\|E_k^{n,N^f-1}\right\| \leq \beta\|E_k^n\| + (C + C\delta t)\left\|\delta E_{k-1}^n\right\| + \left(\alpha + \mu + 2C\delta t + \frac{C\delta t^2}{2}\right)\left\|E_{k-1}^n\right\| + 3\gamma_F.$$

□

3.3. The main convergence theorem

Theorem 3.4 (Convergence of the multi-step parareal algorithm)**.** *Let G, F and δG satisfy Assumptions (7)–(13). Let $k \geq 0$ be any given positive integer. If the time step δt of the fine solver verifies:*

$$\delta t \leq C\Delta T^2 \varepsilon_G^2, \tag{30}$$

then the sequence $(u_k^n)_n$ defined by the multi-step parareal scheme (14) satisfy:

$$\begin{cases} \max_{0\leq n\leq N}\|u_0^n - u(T^n)\| \leq \dfrac{\tilde{\gamma}_G}{CT} N e^{Cn\Delta T}, & n \geq 1, \\ \max_{0\leq n\leq N}\|u_k^n - u(T^n)\| \leq \lambda \dfrac{\tilde{\tau}^{k+1}}{k+1!}\left(\dfrac{f_k}{2^{k+1}}\dfrac{\tilde{\gamma}_G}{\gamma_G}\left(\dfrac{\tilde{\alpha}}{\alpha}\right)^k + \dfrac{k+1}{\tau}\dfrac{f_{k-1}}{2^{k+1}}\dfrac{\tilde{\gamma}_F}{\gamma_G}\left(\dfrac{\tilde{\alpha}}{\alpha}\right)^{k-1}\right), & n \geq k+1, \quad k \geq 1, \end{cases} \tag{31}$$

where:

$$\lambda = \frac{e^{CT}\max_{0\leq n\leq N}(1 + \|u(T^n)\|)}{C_d}, \quad \tilde{\tau} = 2\tau = 2C_d T e^{-C\Delta T}\varepsilon_G,$$

and $\tilde{\alpha}$, $\tilde{\gamma}_G$ and $\tilde{\gamma}_F$ being given by:

$$\begin{cases} \tilde{\alpha} = \alpha + 3\mu + 3C\delta t + \frac{C\delta t^2}{2}, \\ \tilde{\gamma}_G = \gamma_G + C\delta t + \frac{2C+3+2C^2\delta t}{\beta - 1 + \alpha + 3\mu + C\delta t + C\frac{\delta t^2}{2} + C^2\delta t}\gamma_F, \\ \tilde{\gamma}_F = \left(\frac{2C+3+2C^2\delta t}{\beta - 1 + \alpha + 3\mu + C\delta t + C\frac{\delta t^2}{2} + C^2\delta t} + 3\right)\gamma_F + C\delta t + 2C\delta t^2. \end{cases} \tag{32}$$

Remark 3.5. Let us make a couple of remarks before giving the proof of the theorem. First, we recall that the factors α, μ, $\tilde{\alpha}$, $\tilde{\gamma}_G$ and $\tilde{\gamma}_F$ depend on the time window size ΔT. Then, the convergence rate of the multi-step parareal algorithm is similar to the one of the classical parareal algorithm with the factor $\frac{\tilde{\tau}^{k+1}}{k+1!}$, since in the classical version the convergence rate is $\frac{\tau^{k+1}}{k+1!}$. The remaining factors $\frac{\tilde{\gamma}_G}{\gamma_G}$ and $(\frac{\tilde{\alpha}}{\alpha})^k$ are close to 1 and thus bounded by a constant, independent of n. The factor $\frac{\tilde{\gamma}_F}{\gamma_G}$ is negligible in the sense that we have:

$$\frac{\tilde{\gamma}_F}{\gamma_G} = \mathcal{O}(\Delta T\varepsilon_G) \ll 1.$$

The factors $\tilde{\alpha}$, $\tilde{\gamma}_G$ and $\tilde{\gamma}_F$ are perturbations of the coefficients α, γ_G and γ_F respectively, such that:

$$\frac{\tilde{\gamma}_G}{\gamma_G} = 1 + \mathcal{O}(\Delta T\varepsilon_G), \quad \frac{\tilde{\alpha}}{\alpha} = 1 + \mathcal{O}(\Delta T\varepsilon_G).$$

The term $\frac{f_k}{2^{k+1}}$ is specific to the multi-step variant and tends to zero exponentially fast when k goes to infinity. Indeed, it has the following asymptotic behaviour:

$$f_k = \frac{(1+\sqrt{5})^{k+1} - (1-\sqrt{5})^{k+1}}{2^{k+1}\sqrt{5}}, \quad \frac{f_k}{2^{k+1}} \underset{k\to+\infty}{\sim} \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{4}\right)^{k+1}.$$

In practise, in order to respect assumption (30), we choose the fine time step δt such that the condition $\delta t \leq C\Delta T^{2+2\alpha}$ is satisfied.

Proof. The proof is in the spirit of existing results from the literature [7,21,28,29] with similar techniques based on the use of generating functions (A.4). We also refer to [23] for the convergence study of several parallel in time algorithms with generating functions.

From Lemma 3.3, the error terms made by the multi-step parareal method (14) verify the inequalities (25)–(27). Since the upper bound of the error term $\|E_k^{n,N^f-1}\|$ depends on the error terms $\|E_k^{n+1}\|$ and $\|\delta E_k^{n+1}\|$ we focus on the inequalities (25)–(27). Hence, we can write by induction:

$$\|E_k^n\| \leq \beta\|E_k^{n-1}\| + \left(\alpha+\mu+C\delta t+\frac{C\delta t^2}{2}\right)\|E_{k-1}^{n-1}\| + C^2\delta t\sum_{j=2}^{k}(C\delta t)^{j-2}\|E_{k-j}^{n-j}\| + C(C\delta t)^{k-1}\|\delta E_0^{n-k}\| + \left(1+2C\frac{1-(C\delta t)^{k-1}}{1-(C\delta t)}\right)\gamma_F. \tag{33}$$

The governing term in the sum $C^2\delta t\sum_{j=2}^{k}(C\delta t)^{j-2}\|E_{k-j}^{n-j}\|$ is the term $C^2\delta t\|E_{k-2}^{n-2}\|$. To ensure that it does not dominate the term $(\alpha+\mu+C\delta t+\frac{C\delta t^2}{2})\|E_{k-1}^{n-1}\|$, we suppose that the fine time step verifies: $\delta t \leq C\Delta T^2\varepsilon_G^2$ (see hypothesis (30)).

In what follows, we show that the residual terms δE_0^{n-k}, $(1+2C\frac{1-(C\delta t)^{k-1}}{1-(C\delta t)})\gamma_F$ and all the terms of the sum for $j\geq 3$ can be distributed over the terms: $\tilde{e}_k^n$, $\tilde{e}_k^{n-1}$, $\tilde{e}_{k-1}^{n-1}$ and $\tilde{e}_{k-2}^{n-2}$, where $\tilde{e}_k^n$ is a perturbation of $\|E_k^n\|$. Setting the error perturbation to:

$$\tilde{e}_k^n = \|E_k^n\| + \|\delta E_k^n\| + C\delta t\left(\|E_{k-1}^{n-1}\| + \|\delta E_{k-1}^{n-1}\|\right) + \frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F. \tag{34}$$

We show in the sequel that the error perturbation $\tilde{e}_k^n$ satisfy the inequality (22).

- We start from the inequality satisfied by $\|E_k^n\|$ (25) and replace δE_{k-1}^{n-1} by its upper bound in (27) that depends on δE_{k-3}^{n-3}, E_{k-3}^{n-3} and E_{k-2}^{n-2}.
- Then we add $\|\delta E_k^n\| + C\delta t(\|E_{k-1}^{n-1}\| + \|\delta E_{k-1}^{n-1}\|) + \frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F$ to obtain $\|\tilde{E}_k^n\|$ on the left hand side (34).
- Each term of the sum is bounded by the error perturbation:
  - For the error term $\|\tilde{E}_k^{n-1}\|$, we have:

$$\beta\left(\|E_k^{n-1}\| + \frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F\right) \leq \beta\|\tilde{E}_k^{n-1}\|.$$

  - For the error term $\|\tilde{E}_{k-1}^{n-1}\|$, we have:

$$\left(\alpha+\mu+C\delta t+\frac{C\delta t^2}{2}\right)\|E_{k-1}^{n-1}\| + 2C\delta t\left(\|E_{k-1}^{n-1}\| + \|\delta E_{k-1}^{n-1}\| + \frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F\right) \leq \left(\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}\right)\|\tilde{E}_{k-1}^{n-1}\|.$$

  - For the error term $\|\tilde{E}_{k-2}^{n-2}\|$, we have:

$$C^2\delta t\left(\|E_{k-2}^{n-2}\| + C\delta t\|E_{k-3}^{n-3}\| + C\delta t\|\delta E_{k-3}^{n-3}\| + \frac{2C+3+2C^2\delta t}{\beta-1+\alpha+3\mu+C\delta t+\frac{C\delta t^2}{2}+C^2\delta t}\gamma_F\right) \leq C^2\delta t\|\tilde{E}_{k-2}^{n-2}\|.$$

Hence, the error perturbation $\tilde{e}_k^n$ satisfy the inequality (22). Since $\|E_k^n\| \leq \tilde{e}_k^n$, we derive the upper bound (23) for the error term $\|E_k^n\|$ by applying Proposition 3.2 to the error perturbation $\tilde{e}_k^n$ defined by (34).

$$\begin{aligned}
&\|E_0^n\| \leq \tilde{e}_0^n \leq \frac{\tilde{\gamma}_G}{CT} N e^{Cn\Delta T}, && n \geq 1,\\
&\|E_k^k\| = \mathcal{O}(\varepsilon_F), && k \geq 1,\\
&\|E_k^n\| \leq \tilde{e}_k^n \leq \tilde{\gamma}_G \tilde{\alpha}^k f_k \binom{n}{k+1} \beta^{n-k-1} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} f_{k-1} \binom{n}{k} \beta^{n-k}, && n \geq k+1, \quad k \geq 1,
\end{aligned}$$

which ends the proof. □

We illustrate the behavior of the multi-step parareal algorithm with specific initialisations in the next section in the case of a second and third order time integration method.

4. Numerical tests

We apply our multi-step parareal algorithm to two stiff ODEs, the Brusselator system and the Van der Pol oscillator. Our results illustrate that our approach improves the convergence properties with respect to the classical parareal algorithm. We also show that the generalisation of this approach to third-order time schemes holds and the convergence properties derived in Theorem 3.4 are preserved. Finally, we address the question of the parallel efficiency of multi-step parareal. In the last section, we apply the adaptive parareal algorithm (see [29]) where the accuracy of the fine solver is increased across the iterations.

4.1. Numerical convergence results

4.1.1. The Brusselator system

We consider the Brusselator system where the unknows vector is given by $u = \begin{pmatrix} x \\ y \end{pmatrix}$:

$$\begin{cases} x' = A + x^2 y - (B+1)x, \\ y' = Bx - x^2 y, \end{cases}$$

with initial condition $x(0) = 0$ and $y(0) = 1$. This is a stiff ODE that models a chain of chemical reactions. It was already studied in previous works on the parareal algorithm, [21, 29]. The system has a fixed point at $x = A$ and $y = \frac{B}{A}$ which becomes unstable when $B > 1 + A^2$ and leads to oscillations. We place ourselves in this oscillatory regime by setting $A = 1$ and $B = 3$. The dynamics present large velocity variations in some time sub-intervals, making the use of high order time schemes particularly desirable for an appropriate treatment of the transient. The coarse solver is a Backward Euler method with a coarse time step:

$$\Delta T = 0.1,$$

which corresponds to 180 time windows since $T = 18$. The fine solver is a second-order BDF method with a fine time step $\delta t = 10^{-4}$ (respecting hypothesis (30)). In Figure 1, the fine solver is based on a two-step time scheme where the computation of the solution $u^{n,j+1}$ at time $T^n + (j+1)\delta t$ depends on 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 (14) to initialise 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 initialise the fine solver since we did not use the fine propagator yet.

In this section, we analyse the evolution of two different errors across the parareal iterations:

– the error between the fine solution computed in a sequential way and the parareal solution in $L^\infty(0,T)$ norm,

$$\max_{1 \leq n \leq N} \left\| u_k^n - F(T^0, T^n - T^0, u^0) \right\|, \tag{35}$$

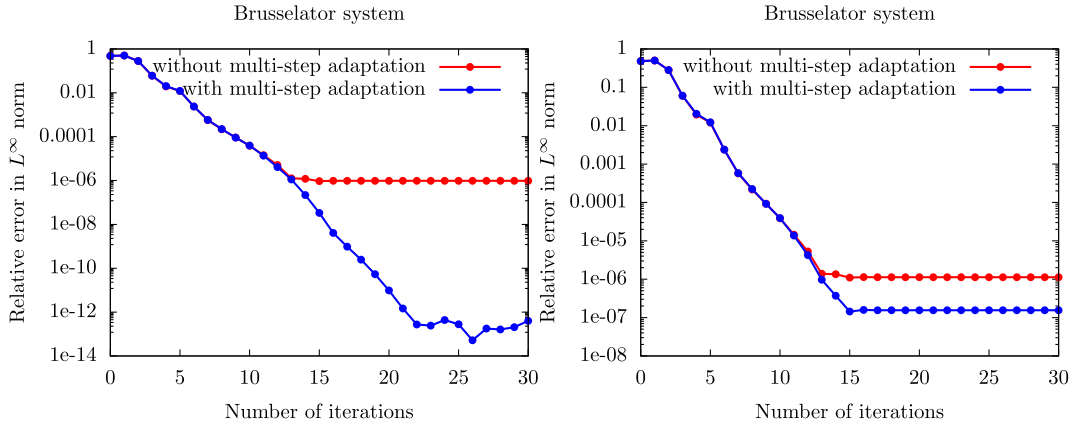

FIGURE 1. Convergence of the multi-step parareal for the second-order BDF method, $\delta t = 10^{-4}$ (*left*: error (35), *right*: error (36)).

– the error between the exact solution and the parareal solution in $L^\infty(0,T)$ norm

$$\max_{1\leq n\leq N} \|u_k^n - u(T^n)\|. \tag{36}$$

In all the figures of this section, we plot the evolution of errors (35) and (36) in the two following cases:

– Without a multi-step adaptation (red curve): the error between the parareal solution where the Backward Euler method is used at each iteration for the initialisation of the fine solver and the fine solution computed in a sequential way for (35) (the exact solution for (36)), on one hand.
– With a multi-step adaptation (blue curve): the error between the solution given by the multi-step parareal algorithm and the fine solution computed in a sequential way for (35) (the exact solution for (36)), on the other hand.

In Figure 1, we see that without the multi-step adaptation the error (35) stagnates around 10^{-6} without recovering the fine solution at the machine precision, even after 180 iterations. On the other hand, using the multi-step parareal algorithm, the error continues to decrease until reaching the machine precision. Moreover, in the right figure, we see that the only way to recover the correct approximation of the exact solution is to use a multi-step adaptation, otherwise, without adaptation, the parareal algorithm will not reach the target accuracy. This result shows that making an initialisation error for a multi-step fine solver will prevent the parareal algorithm to obtain the approximation of the exact solution with the desired accuracy.

The convergence properties are illustrated in Figure 1 on a fine solver based on the second-order BDF method with time step $\delta t = 10^{-4}$.

In Figure 2, we apply the extension of the multi-step parareal algorithm (17) to three-step time schemes by giving a consistent approximation of the solutions $u(T^n - \delta t)$ and $u(T^n - 2\delta t)$. We illustrate the convergence properties of this strategy by applying it on a fine solver based on the third-order BDF method with time steps $\delta t = 10^{-4}$ (see Fig. 2). We observe the same behavior of the errors (35) and (36): without a multi-step adaptation, the fine propagation is initialised by two Backward Euler iterations and does not allow to recover the target approximation of the exact solution while the multi-step parareal converges to the exact solution with the desired accuracy.

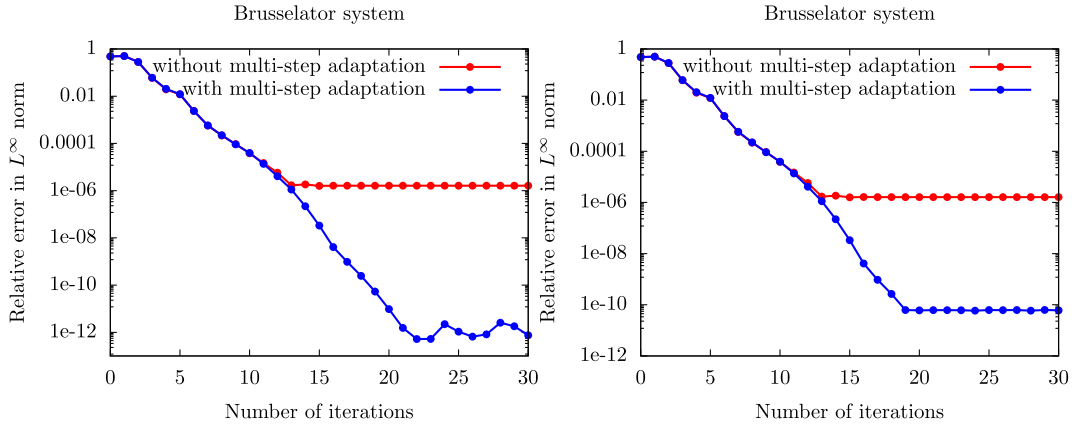

FIGURE 2. Convergence of the multi-step parareal for the third-order BDF method, $\delta t = 10^{-4}$ (*left*: error (35), *right*: error (36)).

4.1.2. The Van der Pol oscillator

We next consider the Van der Pol oscillator

$$\begin{cases} x' = y, \\ y' = \mu(1 - x^2)y - x, \end{cases}$$

with initial condition $x(0) = 2$ and $y(0) = 0$. When $\mu = 0$, this equation is a simple nonstiff harmonic oscillator. When $\mu > 0$, the system has a limit cycle and becomes stiffer and stiffer as its value is increased. For our tests, we set $\mu = 4$ which is a relatively stiff case.

The coarse solver is an explicit Runge–Kutta method of order 3 with an adaptive time stepping (see [13]). The time window size is $\Delta T = 0.1$ which corresponds to 200 time windows since $T = 20$. The fine solver is a third-order BDF method with a fine time step $\delta t = 10^{-4}$ (respecting hypothesis (30)). In Figure 3, the fine solver is based on a three-step time scheme. We apply the extension of the multi-step parareal algorithm (17) to three-step time schemes by giving a consistent approximation of the solutions $u(T^n - \delta t)$ and $u(T^n - 2\delta t)$.

Likewise, we analyse the evolution of the errors (35) and (36) across the parareal iterations. In Figure 3, we see that without the multi-step adaptation the error (35) stagnates around 10^{-5} without recovering the fine solution at the machine precision, even after 200 iterations. On the other hand, using the multi-step parareal algorithm, the error continues to decrease until reaching the machine precision. Moreover, in the right figure, we see that the only way to recover the correct approximation of the exact solution is to use a multi-step adaptation, otherwise, without adaptation, the parareal algorithm will not reach the target accuracy. This result shows that making an initialisation error for a multi-step fine solver will prevent the parareal algorithm to obtain the approximation of the exact solution with the desired accuracy.

The convergence properties are illustrated in Figure 3 on a fine solver based on the third-order BDF method with time step $\delta t = 10^{-4}$.

4.2. Parallel efficiency

We address in this section the question of the speed up performance for the multi-step parareal algorithm. The only additional operations in the multi-step variant compared to the classical parareal are the corrections of solutions involved in the initialisation of the fine solver in each time window (update of u_{k+1}^{n,N^f-1} in (14) for example). Hence, we consider that the computational cost of the multi-step variant is the same as the one of the classical parareal. In [29], the authors propose a new method, the adaptive parareal algorithm, where the

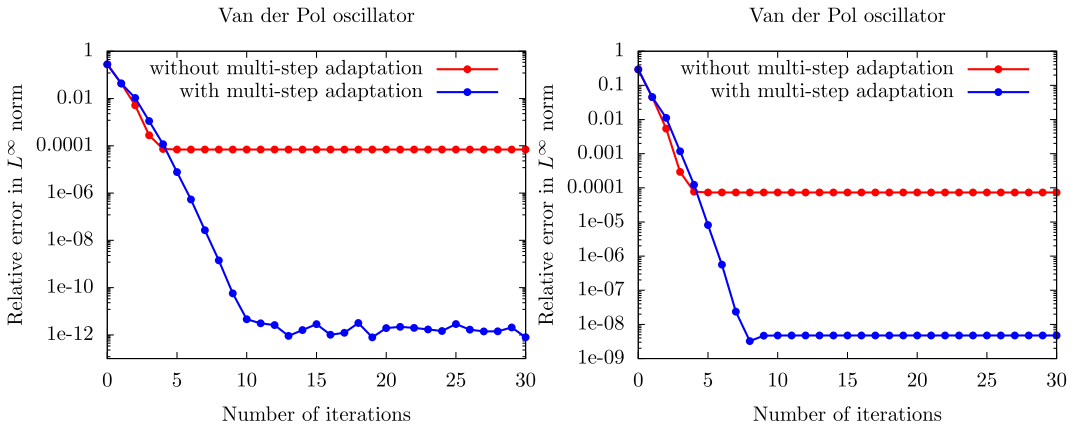

FIGURE 3. Convergence of the multi-step parareal for the third-order BDF method, $\delta t = 10^{-4}$ (*left*: error (35), *right*: error (36)).

accuracy of the fine solver is increased across the iterations. This new point of view improves the speed up performance of the parareal method. In this section, we seek to improve the parallel efficiency of the multi-step parareal method by increasing the accuracy of the fine solver at each iteration. We first recall the parallel efficiency for the classical parareal (CP) and the adaptive parareal (AP) to obtain a solution with accuracy η and a propagation over $[0, T]$:

$$\begin{aligned} &\text{eff}_{CP}(\eta, [0,T]) \sim \tfrac{1}{K(\eta)}, \\ &\text{eff}_{AP}(\eta, [0,T]) \sim \tfrac{1}{1+\varepsilon_G^{1/\alpha}}, \quad \text{under the hypothesis of Proposition 3.1 in [29]}, \end{aligned}$$

where $K(\eta)$ is the number of parareal iterations to obtain the approximation of the exact solution with the target accuracy η and α, the order of the fine time scheme. To apply this approach on the multi-step variant, we need to carefully initialise each time window. If the fine scheme is the second-order BDF method, the computation of u^{n+1} depends on u^n and u^{n-1} and with the adaptive paradigm we have:

$$t^n - t^{n-1} \neq t^{n+1} - t^n.$$

Hence, we initialise the fine solver with one variable step-size BDF method.

We apply this strategy to the Brusselator system with the Backward Euler method as a coarse solver ($\Delta T = 0.1$) and the second-order BDF method as a fine solver with the sequence of time steps indicated in Table 1.

The multi-step parareal algorithm with adaptivity converges to the exact solution with an accuracy obtained by a sequential fine solution with time step $\delta t = 10^{-3}$ after 8 iterations such as the multi-step method without adaptivity (see Tab. 1). With the sequence of fine time steps used in the adaptive parareal method, convergence is reached with the same number of iterations as the multi-step variant. The adaptive algorithm allows to obtain better speed-up performance compared to the nonadaptive version since the fine solver ($\delta t = 10^{-3}$) is used only one time instead of 8 times in the multi-step variant. In Table 2, we give the speed-up and the efficiency of the adaptive and multi-step parareal algorithms applied to the Brusselator system. The speed-up is defined as the ratio:

$$S(\eta, [0,T]) := \frac{T_{\text{seq}}(\eta, [0,T])}{T_{\text{par}}(\eta, [0,T])},$$

between the cost to run a sequential fine solver achieving a target accuracy η with the cost to run a parareal algorithm providing at the end the same target accuracy η. The parallel efficiency of the method is then defined

TABLE 1. Convergence of adaptive parareal and multi-step parareal with a target accuracy $\eta = 10^{-5}$.

	Multi-step parareal	Adaptive parareal		
Iteration	Time step	Error	Time step	Error
$k = 1$	10^{-4}	5×10^{-2}	10^{-2}	5×10^{-2}
$k = 2$	10^{-4}	9×10^{-3}	5×10^{-3}	2×10^{-2}
$k = 3$	10^{-4}	10^{-3}	10^{-3}	3×10^{-3}
$k = 4$	10^{-4}	9×10^{-5}	5×10^{-4}	3×10^{-4}
$k = 5$	10^{-4}	7×10^{-6}	4×10^{-4}	2×10^{-5}
$k = 6$	10^{-4}	4×10^{-7}	2.5×10^{-4}	3×10^{-6}
$k = 7$	10^{-4}	3.8×10^{-8}	2×10^{-4}	3×10^{-7}
$k = 8$	10^{-4}	2×10^{-8}	10^{-4}	2.9×10^{-8}

TABLE 2. Speed up and efficiency with $T = 18$, $\delta t = 10^{-3}$ and $N = 180$.

Speed-up	Multi-step parareal	Adaptive parareal
With cost G	10.9	23.7
Without cost G	12.5	32.2
Efficiency	**Multi-step parareal**	**Adaptive parareal**
With cost G	6%	13%
Without cost G	7%	18%

as the ratio of the above speed up with the number of processors which gives a target of 1 to any parallel solver:

$$\text{eff}(\eta, [0,T]) := \frac{S(\eta, [0,T])}{N}.$$

To compare the speed-up of the multi-step and adaptive parareal algorithms, we use the number of fine and coarse propagations involved in the numerical solution and the computational cost of the coarse and fine propagations (communication delays have not been taken into account). For example, in Table 1, the cost of the multi-step parareal algorithm is equal to the cost of 9 coarse propagations over $[0, T]$ plus 8 fine propagations over $[T^n, T^{n+1}]$ with a fine time step $\delta t = 10^{-3}$. In [29], the authors show that the main element affecting the performance of the adaptive parareal method is no longer the cost of the fine solver but the cost of the coarse solver. Hence, we compare the speed-up and efficiency when we count or do not count the cost of the coarse solver in Table 2. Obviously, when we do not count the cost of the coarse solver, the performance of both algorithms improves.

5. Conclusion

We have built a new variant of the parareal algorithm allowing to overcome the issue of initialising the fine and the coarse solvers when they are based on a multi-step time scheme [1, 2]. The convergence properties of the multi-step parareal are very close to that of the classical parareal algorithm in the case of two-step time schemes. An extension of our approach to generic multi-step time schemes is proposed and validated numerically on a three-step time scheme. In addition, the accuracy of the multi-step parareal algorithm is illustrated on the numerical solution of stiff ODEs such as the Brusselator system and the Van der Pol oscillator. Finally, we

address the question of the parallel efficiency of our strategy by coupling it with the adaptive parareal algorithm proposed in [29]. The new adaptive formulation of the parareal algorithm opens the door to improve significantly the parallel efficiency of the method provided that the cost of the coarse solver is moderate.

Appendix A. Proof of Proposition 3.2

The proof is in the spirit of existing results from the literature [7,21,28,29] with similar techniques based on the use of generating functions (A.4).

The sequence $(\tilde{e}_k^n)_{n\geq 0,k\geq 0}$ is defined recursively as follows. For $k=0$:

$$\tilde{e}_0^n = \begin{cases} 0, & \text{if } n=0,\\ \beta\tilde{e}_0^{n-1} + \tilde{\gamma}_G, & \text{if } n\geq 1. \end{cases} \tag{A.1}$$

For $k=1$:

$$\tilde{e}_1^n = \begin{cases} 0, & \text{if } n=0,1,\\ \beta\tilde{e}_1^{n-1} + \tilde{\alpha}\tilde{e}_0^{n-1} + \tilde{\gamma}_F, & \text{if } n\geq 2. \end{cases} \tag{A.2}$$

For $k\geq 2$:

$$\tilde{e}_k^n = \begin{cases} 0, & \text{if } n=0,1,2,\\ \beta\tilde{e}_k^{n-1} + \tilde{\alpha}\tilde{e}_{k-1}^{n-1} + C^2\delta t\tilde{e}_{k-2}^{n-2}, & \text{if } n\geq 3. \end{cases} \tag{A.3}$$

We analyse the behavior of $(\tilde{e}_k^n)$ to derive a bound for the sequence. For this, we consider the generating function:

$$\tilde{\rho}_k(\xi) = \sum_{n\geq 0} \tilde{e}_k^n \xi^n.$$

From (A.1)–(A.3) we get:

$$\begin{cases} \tilde{\rho}_0(\xi) = \frac{\tilde{\gamma}_G\xi}{(1-\beta\xi)(1-\xi)},\\ \tilde{\rho}_1(\xi) = \frac{\tilde{\alpha}\xi}{1-\beta\xi}\tilde{\rho}_0(\xi) + \frac{\tilde{\gamma}_F\xi}{(1-\beta\xi)(1-\xi)},\\ \tilde{\rho}_k(\xi) = \frac{\tilde{\alpha}\xi}{1-\beta\xi}\tilde{\rho}_{k-1}(\xi) + \frac{C^2\delta t\xi^2}{1-\beta\xi}\tilde{\rho}_{k-2}(\xi), \quad k\geq 2. \end{cases} \tag{A.4}$$

In the convergence analysis of the classical parareal algorithm, the generating function $\rho_k(\xi)$ satisfies a recurrence relation involving only two steps of the sequence ρ_k and ρ_{k-1}. In the convergence analysis of the multi-step parareal algorithm, the error perturbation $\tilde{e}_k^n$, hence the generating function $\tilde{\rho}_k$, verifies a recurrence relation involving the steps k, $k-1$ and $k-2$. We could use the general theory of multi-step recurrence sequences but the resulting expression of $\tilde{\rho}_k$ does not lead to a workable formula. Writing the first terms of the sequence $\tilde{\rho}_k$ yields to the alternative expression for $k\geq 1$ that is proven by induction on odd and even k:

$$\begin{aligned} \tilde{\rho}_k(\xi) = {} & \tilde{\gamma}_G\tilde{\alpha}^k\frac{\xi^{k+1}}{(1-\xi)}\sum_{j=0}^{[k/2]}\frac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}}\binom{k-j}{j}\frac{1}{(1-\beta\xi)^{k+1-j}}\\ & + \tilde{\gamma}_F\tilde{\alpha}^{k-1}\frac{\xi^k}{(1-\xi)}\sum_{j=0}^{[k-1/2]}\frac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}}\binom{k-1-j}{j}\frac{1}{(1-\beta\xi)^{k-j}}. \end{aligned} \tag{A.5}$$

For $k=0$, we have:

$$\tilde{\rho}_0(\xi) = \tilde{\gamma}_G\xi\left(\sum_{p\geq 0}\xi^p\right)\left(\sum_{p\geq 0}\beta^p\xi^p\right) = \tilde{\gamma}_G\sum_{p\geq 0}\left(\sum_{l=0}^{p}\beta^l\right)\xi^{p+1}.$$

By identification of terms, we obtain after a change of variable $p = n-1$:

$$\tilde{e}_0^n = \tilde{\gamma}_G \left(\sum_{l=0}^{n-1} \beta^l \right) \leq \frac{\tilde{\gamma}_G}{CT} N e^{Cn\Delta T}, \quad n \geq 1.$$

For $k \geq 1$, using the binomial expansion in (A.5):

$$\frac{1}{(1-\beta\xi)^{k+1-j}} = \sum_{p \geq 0} \binom{k-j+p}{p} \beta^p \xi^p,$$

and by a change of variable, we obtain:

$$\sum_{n \geq 0} \tilde{e}_k^n \xi^n = \tilde{\gamma}_G \tilde{\alpha}^k \sum_{n \geq k+1} K_{n-k-1} \xi^n + \tilde{\gamma}_F \tilde{\alpha}^{k-1} \sum_{n \geq k} K'_{n-k} \xi^n,$$

where K_p and K'_p are given by:

$$\begin{cases} K_p = \sum_{l=0}^{p} \sum_{j=0}^{[k/2]} \dfrac{(C^2\delta t)^j}{(\alpha + 3\mu + C\delta t)^{2j}} \binom{k-j}{j} \binom{k-j+l}{l} \beta^l, \\ K'_p = \sum_{l=0}^{p} \sum_{j=0}^{[k-1/2]} \dfrac{(C^2\delta t)^j}{\tilde{\alpha}^{2j}} \binom{k-1-j}{j} \binom{k-1-j+l}{l} \beta^l. \end{cases} \tag{A.6}$$

Identifying the term ξ^k in the expansion yields to:

$$\tilde{e}_k^k = \tilde{\gamma}_F \tilde{\alpha}^{k-1} K'_0.$$

This gives an upper bound for the error terms $\tilde{e}_k^k, k \geq 1$. We do not use this estimate since the parareal algorithm ensures $u_k^n = F(T^0, T^n - T^0, u^0)$ for $k \geq n$, which yields:

$$\|E_k^k\| = \mathcal{O}(\varepsilon_F), \quad k \geq 1.$$

In what follows, we identify the terms ξ^n for $n \geq k+1$ in the expansion:

$$\tilde{e}_k^n = \tilde{\gamma}_G \tilde{\alpha}^k K_{n-k-1} + \tilde{\gamma}_F \tilde{\alpha}^{k-1} K'_{n-k}.$$

We now compute the terms K_p and K'_p. From Assumption (30), we have:

$$\frac{C^2\delta t}{(\alpha + 3\mu + C\delta t)^2} \leq 1.$$

Using: $\binom{k-j+l}{l} \leq \binom{k+l}{l}$ for $j = 0, \ldots, [\frac{k-1}{2}]$, we obtain:

$$K_p \leq \sum_{l=0}^{p} \binom{k+l}{l} \beta^l \sum_{j=0}^{[k/2]} \binom{k-j}{j}, \quad K'_p \leq \sum_{l=0}^{p} \binom{k-1+l}{l} \beta^l \sum_{j=0}^{[k-1/2]} \binom{k-1-j}{j}.$$

We recognize here the general term f_k of the Fibonacci sequence:

$$f_k = \sum_{j=0}^{[k/2]} \binom{k-j}{j} = \frac{(1+\sqrt{5})^{k+1} - (1-\sqrt{5})^{k+1}}{2^{k+1}\sqrt{5}}.$$

Using $\sum_{l=0}^{p}\binom{k+l}{l} = \binom{k+1+p}{p}$ and $\beta > 1$, we obtain:

$$K_p \leq f_k\binom{k+1+p}{p}\beta^p, \quad K'_p \leq f_{k-1}\binom{k+p}{p}\beta^p.$$

Hence, we derive the bound:

$$\begin{aligned}
\tilde{e}_0^n &\leq \frac{\tilde{\gamma}_G}{CT}Ne^{Cn\Delta T}, && n \geq 1,\\
\tilde{e}_k^n &\leq \tilde{\gamma}_G\tilde{\alpha}^k f_k\binom{n}{k+1}\beta^{n-k-1} + \tilde{\gamma}_F\tilde{\alpha}^{k-1}f_{k-1}\binom{n}{k}\beta^{n-k}, && n \geq k+1, \quad k \geq 1,\\
\tilde{e}_k^n &\leq \tilde{\gamma}_G\frac{\tilde{\alpha}^k f_k\beta^{n-k-1}}{(k+1)!}\prod_{j=0}^{k}(n-j) + \tilde{\gamma}_F\frac{\tilde{\alpha}^{k-1}f_{k-1}\beta^{n-k}}{k!}\prod_{j=0}^{k-1}(n-j), && n \geq k+1, \quad k \geq 1,
\end{aligned} \tag{A.7}$$

which ends the proof of Proposition 3.2.

Acknowledgements

This work was supported by the ANR project CINE-PARA under grant ANR-15-CE23-0019 and from the European High Performance Computing Joint Undertaking (EuroHPC JU) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 955701 – project TIME-X).

References

[1] K. Ait-Ameur, Y. Maday and M. Tajchman, Multi-step variant of the parareal algorithm, in Domain Decomposition Methods in Science and Engineering XXV, edited by R. Haynes, S. MacLachlan, X.-C. Cai, L. Halpern, H.H. Kim, A. Klawonn and O. Widlund. Springer International Publishing, Cham (2020) 393–400.

[2] K. Ait-Ameur, Y. Maday and M. Tajchman, Time-parallel algorithm for two phase flows simulation, in Numerical Simulation in Physics and Engineering: Trends and Applications; Lecture Notes of the XVIII 'Jacques-Louis Lions' Spanish-French School, edited by D. Greiner, M. Asensio and R. Montenegro (2021) 169–178.

[3] J. Astic, A. Bihain and M. Jerosolimski, The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems. *IEEE Trans. Power Syst.* **9** (1994) 929–935.

[4] C. Audouze, M. Massot and S. Volz, Symplectic multi-time step parareal algorithms applied to molecular dynamics. https://hal.science/hal-00358459 (2009).

[5] F. Bashforth and J.C. Adams, Theories of Capillary Action. Cambridge University Press, Cambridge (1883).

[6] D.Q. Bui, C. Japhet, Y. Maday and P. Omnes, Coupling parareal with optimized Schwarz waveform relaxation for parabolic problems. *SIAM J. Numer. Anal.* **60** (2022) 913–939.

[7] B. Carrel, M. Gander and B. Vandereycken, Low-rank parareal: a low-rank parallel-in-time integrator. *BIT Numer. Math.* **63** (2023).

[8] A.J. Christlieb, C.B. Macdonald and B.W. Ong, Parallel high-order integrators. *SIAM J. Sci. Comput.* **32** (2010) 818–835.

[9] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand.* **4** (1956) 33–53.

[10] G. Dahlquist, Stability and error bounds in the numerical integration of ordinary differential equations. Trans. of the Royal Inst. of Techn., Nr. 130 (1959) 87.

[11] X. Dai and Y. Maday, Stable parareal in time method for first- and second-order hyperbolic systems. *SIAM J. Sci. Comput.* **35** (2013) A52–A78.

[12] X. Dai, C. Le Bris, F. Legoll and Y. Maday, Symmetric parareal algorithms for hamiltonian systems. *ESAIM: M2AN* **47** (2013) 717–742.

[13] J.R. Dormand and P.J. Prince, A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6** (1980) 19–26.

[14] M. Emmett and M.L. Minion, Toward an efficient parallel in time method for partial differential equations. *Commun. Appl. Math. Comput. Sci.* **7** (2012) 105–132.

[15] R.D. Falgout, S. Friedhoff, T.V. Kolev, S.P. MacLachlan and J.B. Schroder, Parallel time integration with multigrid. *SIAM J. Sci. Comput.* **36** (2014) C635–C661.

[16] R. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, J. Schroder and S. Vandewalle, Multigrid methods with space–time concurrency. *Comput. Vis. Sci.* **18** (2017) 1–21.

[17] R.D. Falgout, M. Lecouvez and C.S. Woodward, A parallel-in-time algorithm for variable step multistep methods. *J. Comput. Sci.* **37** (2019) 101029.
[18] C. Farhat and M. Chandesris, Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *Int. J. Numer. Methods Eng.* **58** (2003) 1397–1434.
[19] M.J. Gander, 50 years of time parallel time integration, in Multiple Shooting and Time Domain Decomposition Methods, edited by T. Carraro, M. Geiger, S. Körkel and R. Rannacher. Springer International Publishing, Cham (2015) 69–113.
[20] M.J. Gander and S. Güttel, PARAEXP: a parallel integrator for linear initial-value problems. *SIAM J. Sci. Comput.* **35** (2013) C123–C142.
[21] M.J. Gander and E. Hairer, Nonlinear convergence analysis for the parareal algorithm, in Domain Decomposition Methods in Science and Engineering XVII, edited by U. Langer, M. Discacciati, D.E. Keyes, O.B. Widlund and W. Zulehner. Springer Berlin Heidelberg, Berlin, Heidelberg (2008) 45–56.
[22] M.J. Gander and M. Neumüller, Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM J. Sci. Comput.* **38** (2016) A2173–A2208.
[23] M.J. Gander, T. Lunet, D. Ruprecht and R. Speck, A unified analysis framework for iterative parallel-in-time algorithms. *SIAM J. Sci. Comput.* **45** (2023).
[24] C. Gear, Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, Upper Saddle River (1971).
[25] R. Guetat, *Méthode de parallélisation en temps: application aux méthodes de décomposition de domaine*. Ph.D. thesis, Paris VI. https://www.theses.fr/2011PA066629 (2012).
[26] E. Hairer, S. Nørsett and G. Wanner, Solving Ordinary Differential Equations I Nonstiff Problems, 2nd edition. Springer, Berlin (2000).
[27] H. Jiménez-Pérez and J. Laskar, A time-parallel algorithm for almost integrable Hamiltonian systems. Preprint arXiv:1106.3694 (2011).
[28] J.-L. Lions, Y. Maday and G. Turinici, Résolution d'EDP par un schéma en temps pararéel. *C. R. Acad. Sci. - Ser. I - Math.* **332** (2001) 661–668.
[29] Y. Maday and O. Mula, An adaptive parareal algorithm. *J. Comput. Appl. Math.* **377** (2020) 112915.
[30] Y. Maday and E.M. Rønquist, Parallelization in time through tensor-product space-time solvers. *C. R. Math.* **346** (2008) 113–118.
[31] J. Nievergelt, Parallel methods for integrating ordinary differential equations. *Commun. ACM* **7** (1964) 731–733.
[32] B.W. Ong and J.B. Schroder, Applications of time parallelization. *Comput. Vis. Sci.* **23** (2020).
[33] A. Quarteroni and A. Valli, Domain Decomposition Methods for Partial Differential Equations. Von Karman Institute for Fluid Dynamics (1996).
[34] M. Schreiber, P.S. Peixoto, T. Haut and B. Wingate, Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. *Int. J. High Perform. Comput. Appl.* **32** (2018) 913–933.
[35] A. Toselli and O. Widlund, Domain Decomposition Methods: Algorithms and Theory. *Springer Series in Computational Mathematics*. Springer Berlin, Heidelberg (2005).