HAL open science

An adaptive parareal algorithm: application to the simulation of molecular dynamics trajectories

Frédéric Legoll, Tony Lelièvre, Upanshu Sharma

To cite this version:

Frédéric Legoll, Tony Lelièvre, Upanshu Sharma. An adaptive parareal algorithm: application to the simulation of molecular dynamics trajectories. SIAM Journal on Scientific Computing, 2022, 44 (1), pp.B146-B176. 10.1137/21M1412979 . hal-03189428v2

HAL Id: hal-03189428

https://hal.science/hal-03189428v2

Submitted on 1 Oct 2021

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.

An adaptive parareal algorithm: application to the simulation of molecular dynamics trajectories

Frédéric Legoll, Tony Lelièvre, Upanshu Sharma

October 1, 2021

Abstract

The aim of this article is to design parareal algorithms in the context of thermostated molecular dynamics. In its original setup, the fine and coarse propagators used in the parareal algorithm solve the same dynamics with different time-steps, with the goal of achieving accuracy in the limit of small time-step of the integrators involved. This is typically not useful in molecular dynamics, where one is interested in extremely long trajectories and where the time-step of the fine propagator is in practice chosen as large as possible, that is close to the limit of stability of the numerical scheme. In this article, we consider a version of the parareal algorithm which is better suited to molecular dynamics simulations, and wherein the propagators involved use the same time-step while employing different potential energy landscapes to drive the dynamics.

Although the parareal algorithm always converges, it suffers from various limitations in this context: intermediate blow-up of the trajectory (which makes it impossible to postprocess) may be observed; in certain cases the trajectory encounters undefined values before converging (the way the algorithm handles them might depend on the computer architecture); more critically, the computational gain of the algorithm in terms of wall-clock time (compared to a standard sequential integration) converges to one when the time-horizon increases. We highlight these issues with numerical experiments and provide some elements of theoretical analysis. We then present a modified version of the parareal algorithm wherein the algorithm adaptively divides the entire time-horizon into smaller time-slabs where the aforementioned issues are circumvented. Using numerical experiments on toy examples, we show that the adaptive algorithm overcomes the various limitations of the standard parareal algorithm, thereby allowing for significantly improved gains.

1 Introduction

A central objective in molecular dynamics (MD) is the computation of ensemble averages and dynamical quantities, both of which involve averages over long trajectories of stochastic dynamics. Computations using such dynamics are often difficult in practice since typical quantities of interest evolve on time-scales which are orders of magnitude larger than those accessible by classical numerical methods. Consequently, the development of numerical algorithms which speed up computations is a challenging perspective in MD. More specifically, these issues arise due to extremely high-dimensional non-convex potential energy landscapes, which trap the system in a certain region for extremely long times, before letting it explore other parts of the state space (a phenomenon called metastability). In practice, this issue may be mitigated by using approximate potentials, which are cheaper to compute (and thus make it less challenging to reach the requested long time horizons) while retaining the essential features of the original potential. However this strategy suffers from the obvious drawback that it only captures the original dynamics approximately.

In this article we present a numerical method that aims at efficiently simulating the original dynamics over long trajectories. To this end, we consider an adjusted version of the parareal algorithm, using an approximate potential as a predictor and the original potential as a corrector. We recall that the parareal algorithm has been originally proposed in [LMT01] to efficiently solve initial-value problems using parallel-in-time computations. More precisely, the algorithm exploits the presence of multiple processors to reduce

the wall-clock time needed to obtain a solution on some time interval. It is based on a decomposition of the time interval into subintervals and makes use of a predictor-corrector strategy, in which the calculation of the corrections is performed concurrently, in parallel, on the different processors that are available. The algorithm has been successfully applied to many problems, see e.g. [FC03, FHM05, GEF05]. Several variants of the algorithm have been proposed for specific applications: we refer e.g. to [BBK10, Mad07], [LLS13] and [Eng09] for multiscale-in-time problems, to [DLBLM13, DM13] for Hamiltonian ODEs or PDEs, to [Bal, LLMS20] for stochastic differential equations (with a specific emphasis, in [LLMS20], to high-dimensional SDEs stemming from MD and for which qualitative features, such as positivity and norm conservation of the associated probability distribution, are to be preserved), and also to [FC03, GLFE06].

Let us now briefly introduce the classical version of the parareal algorithm (as first presented in [LMT01] and next reinterpreted in [BM02]). To fix the ideas, we consider the simple evolution problem given by

$$\frac{dx}{dt} = f(x), \qquad x(0) = x_0, \qquad x(t) \in \mathbb{R}^d, \qquad t \in [0, T]. \tag{1.1}$$

Assume that we have at hand two propagators for this system: $\mathcal{F}_{\Delta t}$ is a fine, expensive propagator which accurately approximates the exact flow of (1.1), and $\mathcal{C}_{\Delta t}$ is a coarse, less expensive propagator which is also less accurate. In the classical approach, $\mathcal{F}_{\Delta t}$ and $\mathcal{C}_{\Delta t}$ are often integrators of the same dynamics (1.1) over the time range Δt with a given discretization scheme, using a small time-step for $\mathcal{F}_{\Delta t}$ and a large time-step for $\mathcal{C}_{\Delta t}$. More precisely, $\mathcal{F}_{\Delta t}$ advances the system over the time Δt by performing several time-steps of small length $\delta t_{\mathcal{F}}$, whereas $\mathcal{C}_{\Delta t}$ advances the system over the time Δt by performing a few (or even only one) time-steps of length $\delta t_{\mathcal{C}} > \delta t_{\mathcal{F}}$. The parareal algorithm iteratively builds a sequence of N-tuples $\{x_n^k\}_{1 \leq n \leq N}$ (with $N = T/\Delta t$) such that, at every parareal iteration $k \geq 0$, x_n^k is an approximation of $x(n\Delta t)$. For $k = 0$, the initial approximation is obtained using the coarse propagator $\mathcal{C}_{\Delta t}$ in a sequential manner:

$$x_{n+1}^{k=0} = \mathcal{C}_{\Delta t}(x_n^{k=0}), \qquad x_{n=0}^{k=0} = x_0.$$

The subsequent parareal iterations are given by

$$x_{n+1}^{k+1} = \mathcal{C}_{\Delta t}(x_n^{k+1}) + \mathcal{F}_{\Delta t}(x_n^k) - \mathcal{C}_{\Delta t}(x_n^k), \qquad x_{n=0}^{k+1} = x_0. \tag{1.2}$$

The solution to (1.2) can be computed efficiently in parallel using the following procedure. Once the solution at a parareal iteration k has been computed, i.e. once $\{x_n^k\}_{0 \leq n \leq N}$ is known, the corrections $\mathcal{F}_{\Delta t}(x_n^k) - \mathcal{C}_{\Delta t}(x_n^k)$ are first computed in parallel over each subinterval $[n\Delta t, (n+1)\Delta t]$, $0 \leq n \leq N-1$. These corrections are then propagated sequentially, by adding $\mathcal{C}_{\Delta t}(x_n^{k+1})$ to the stored correction $\mathcal{F}_{\Delta t}(x_n^k) - \mathcal{C}_{\Delta t}(x_n^k)$, which yields the solution at parareal iteration $k+1$. Note that the fine propagator is never used sequentially in the algorithm. A crucial property of the parareal algorithm is that, for sufficiently large k, it always converges to the reference solution (given by the fine propagator), since it can be checked (see e.g. [BBM+02, BM02]) that

$$\forall k \geq n, \qquad x_n^k = \mathcal{F}_{\Delta t}^n(x_0). \tag{1.3}$$

This result is however not very useful to understand the computational gain one can expect from (1.2), that we now discuss.

The computational gain of the parareal algorithm arises from the key observation that, in (1.2), the fine propagator can be computed in parallel over different processors (since it only depends on the previous iteration). Suppose that the cost of a single evaluation of $\mathcal{F}_{\Delta t}$ is considerably larger than the cost of propagating the system according to $\mathcal{C}_{\Delta t}$ over the complete time range $[0, T]$. Assuming that the cost of one call to the fine-scale propagator $\mathcal{F}_{\Delta t}$ is c_0, the cost of K iterations of the parareal algorithm is $K\, c_0$. This cost is to be compared to the cost of computing the reference solution using the fine-scale propagator sequentially, which is equal to $N\, c_0$. Assuming that the algorithm has converged after K parareal iterations, the computational gain is then N/K. This gain is always larger than one since the number of parareal iterations required to obtain convergence is smaller than or equal to N, in view of (1.3). In practice, for many applications, convergence is actually reached for a number of iterations K considerably smaller than N, thus the practical efficiency of the algorithm. A noteworthy exception is the case of hyperbolic

problems, where a larger number K of iterations is often needed to reach convergence, as observed e.g. in [DLBLM13, DM13].

In this article, we are interested in accurately and efficiently computing long trajectories in the context of MD. Specifically, we focus on the so-called *Langevin dynamics* (see (2.1) below), which is a popular dynamics to simulate molecular systems at fixed temperature. In applications, it is observed that the limiting time-step is not determined by the required accuracy of the dynamics, but by its stability: simulations are thus performed with a time-step at the limit of stability of the numerical method. Therefore, the classical approach to parareal algorithms as described above, with different time-steps for the fine and coarse propagators, is not viable. Indeed, a typical choice would consist in choosing the time-step of the fine propagator close to the limit of stability (because this is how a reference computation would be performed), and thus a coarse propagator based on the same numerical scheme for a larger time-step would be unstable. Instead, it is more appropriate in MD to use propagators with the same time-step but different potentials: the fine propagator solves the Langevin dynamics with the original potential V_f, while the coarse propagator uses an approximate potential V_c. Using different evolutions for the propagators is atypical for the parareal literature, with the exception of [BBM$^+$02] (see also [DLBLM13, Section 7]), where the authors employ propagators with different dynamics and different time-steps in the context of short-time computations in ab-initio MD. More precisely, in [BBM$^+$02, Sect. 5], the fine and the coarse propagators both perform ab-initio MD simulations, using pseudo-potentials (for the computation of the electronic configuration) of very different complexity (thus the difference in cost between $\mathcal{C}_{\Delta t}$ and $\mathcal{F}_{\Delta t}$, which both perform a unique time-step to advance the system over the time range Δt). In contrast to our work here, the dynamics considered there is the Hamiltonian dynamics (i.e. the Newton equations of motion, integrated using a Verlet algorithm of time-step Δt) rather than the Langevin dynamics, and it is integrated until a rather short time horizon: the final time corresponds to $T = 30\Delta t$. On this short time range, the parareal algorithm performs nicely, providing accurate trajectories and a very good energy preservation after only 5 parareal iterations.

The article is organised as follows. In Section 2, we present our MD setting, including two examples that are used throughout the article. Subsequently, in Section 3, we discuss various features of the parareal algorithm in the context of MD. We furthermore detail two limitations when dealing with long trajectories:

- The parareal trajectory, as a function of k, typically goes through an intermediate regime wherein the error is extremely large before converging.
- The gain N/K of the parareal algorithm converges to one as the time-horizon over which the algorithm is used gets larger.

Both of these limitations are discussed with numerical examples, and obviously affect the utility of the parareal algorithm in MD. In Section 4, we present an adaptive version of the parareal algorithm, which overcomes these issues by adaptively choosing the size of the time-slab over which the trajectory is computed. On this smaller time-slab (say $[0, \widetilde{T}_1] \subset [0, T]$, to fix the ideas), the parareal trajectory quickly stabilises. Once the computation has converged on $[0, \widetilde{T}_1]$, the algorithm moves to the subsequent part of the time domain (here $[\widetilde{T}_1, T]$) and again adaptively chooses the size of the time-slab (say $[\widetilde{T}_1, \widetilde{T}_2] \subset [\widetilde{T}_1, T]$) so that convergence is quickly reached over that time range. The procedure is repeated until we reach the targeted time horizon T. The numerical results reported in Section 4 show that the gain dramatically improves compared to the standard version of parareal, which consists in running the algorithm at once on the whole time range $[0, T]$. In Section 5 we present conclusions and additional discussions. We next collect in Section 6 some theoretical analysis on a toy example, which sheds light on the limitations of the standard parareal algorithm. From that theoretical viewpoint, studying the Langevin dynamics is challenging, and we therefore prefer to consider a simpler problem, namely the case of (1.1) in dimension $d = 1$, with forces driven by harmonic potentials. Even though simple, this case is useful for a two-fold reason. First, we observe on that case the same limitations of the classical parareal algorithm as those pointed out above and discussed in Section 3 for more realistic problems. Second, it allows to understand the interplay (on the qualitative behaviour of the algorithm along the parareal iterations k) between the time horizon T and the difference between the coarse and fine propagators. The analysis helps to quantify (on admittedly a toy-problem case) the expected fact that the algorithm can be used on longer time windows whenever the coarse propagator

is closer (in a sense made precise) to the fine propagator. Stated otherwise, for a given couple of coarse and fine propagators, there is a maximum time horizon beyond which the parareal algorithm does not behave satisfactorily. Our main result in that section is Proposition 6.5.

2 The molecular dynamics model problem

In this article we consider the Langevin dynamics

$$\begin{aligned} dq_t &= p_t\, dt, \\ dp_t &= -\nabla V(q_t)\, dt - \gamma\, p_t\, dt + \sqrt{2\,\gamma\,\beta^{-1}}\, dW_t, \end{aligned} \tag{2.1}$$

with initial condition (q_0, p_0). Here $(q_t, p_t) \in \mathbb{R}^{2md}$ are the positions and momenta at time t of m particles in dimension d, $V : \mathbb{R}^{md} \to \mathbb{R}$ is the potential energy of the system, $\gamma > 0$ is the friction coefficient, β is the inverse temperature and W_t is a standard Brownian motion in $\mathbb{R}^{md}$. We have set the mass of each particle to unity for simplicity. Throughout this article, we use the following numerical scheme (see [JS12, Eq.(4.2)]) to integrate the Langevin dynamics:

$$\begin{aligned} p^{n+\frac{1}{4}} &= p^n - \frac{\Delta t}{2}\, \nabla V(q^n), \\ q^{n+1} &= q^n + \Delta t\, p^{n+\frac{1}{4}}, \\ p^{n+\frac{1}{2}} &= p^{n+\frac{1}{4}} - \frac{\Delta t}{2}\, \nabla V(q^{n+1}), \\ p^{n+1} &= e^{-\gamma \Delta t}\, p^{n+\frac{1}{2}} + \sqrt{(1 - e^{-2\gamma\Delta t})\, \beta^{-1}}\, G^n, \end{aligned} \tag{2.2}$$

where G_n is a standard Gaussian random variable. This scheme uses a Verlet integration for the Hamiltonian part of (2.1) (which corresponds to the first term in the right hand side of both lines in (2.1)) and an explicit solution for the Ornstein-Uhlenbeck part of (2.1) (which corresponds to the last two terms in the right-hand side of the second line of (2.1)).

As stated in the introduction, in the MD context, it is more appropriate to use propagators with the same time-step but different potentials V, and thereby the propagators solve different dynamics. Henceforth, the fine (resp. coarse) propagator $\mathcal{F}_{\Delta t}$ (resp. $\mathcal{C}_{\Delta t}$) corresponds to (2.2) with the potential V_f (resp. V_c). Note that $\mathcal{F}_{\Delta t}$ and $\mathcal{C}_{\Delta t}$ amount to advancing the system for a time range Δt by performing a single time-step of the integrator (2.2). Of course, for the parareal algorithm to converge, it is crucial that the coarse and fine dynamics are driven by the *same* noise. We thus use Gaussian increments G^n in (2.2) which, at a given time-step n, are the same for both propagators $\mathcal{F}_{\Delta t}$ and $\mathcal{C}_{\Delta t}$ and throughout all parareal iterations.

Remark 2.1 (Overdamped Langevin dynamics)**.** While here we focus on the Langevin dynamics since it is preferred in practice, other choices are also possible. For instance, one can use the overdamped Langevin dynamics

$$dq_t = -\nabla V(q_t)\, dt + \sqrt{2\,\beta^{-1}}\, dW_t, \tag{2.3}$$

which arises as the large-friction limit of (2.1) (see [LRS10, Section 2.2.4]). It turns out that the difficulties we encountered when applying the parareal algorithm described in this article to the Langevin dynamics also appear when working with its overdamped limit (2.3). □

2.1 Two examples

We now present two pairs of fine and coarse potentials that are used throughout this article.

2.1.1 One-dimensional harmonic potentials (Har-1d)

In the first example, we consider V_f and V_c to be two different harmonic potentials in dimension $d = 1$, i.e., for some $\omega > 0$ and for any $q \in \mathbb{R}$,

$$V_f(q) = \frac{q^2}{2}, \qquad V_c(q) = \omega \, \frac{q^2}{2}. \tag{2.4}$$

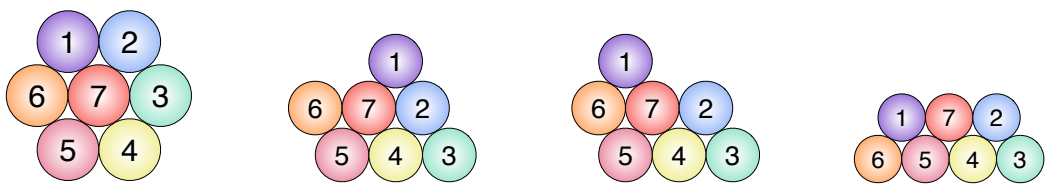
We consider this toy example since it has sufficient complexity to exhibit, in the parareal context, the same issues as those appearing with more complex choices while also being amenable to theoretical analysis (see Section 6 below).

2.1.2 Two-dimensional Lennard-Jones 7-atom cluster (LJ7-2d)

For a more realistic model, we consider a 7-atom ($m = 7$) cluster in dimension $d = 2$ with pairwise Lennard-Jones interaction. The energy landscape has four local minima which correspond to four stable configurations (see Figure 1). The fine potential $V_f : \mathbb{R}^{14} \to \mathbb{R}$ is

$$V_f(q) = \frac{1}{2} \sum_{\substack{i,j \in \{1,\dots,7\} \\ i \neq j}} \phi_f(|q^i - q^j|), \qquad \phi_f(r) = r^{-12} - 2\, r^{-6},$$

where $|\cdot|$ is the 2-dimensional Euclidean norm. The coarse potential V_c is the harmonic approximation of V_f around the initial well, i.e. the well in which the initial condition lies (see Appendix A for explicit formulae). In practice, we choose the initial position to be the global minimizer of the potential energy, that is the deepest well (see Figure 1a). The choice of initial momenta is discussed in the following Section 2.2.1.



(a) $V = -12.53$ (b) $V = -11.50$ (c) $V = -11.48$ (d) $V = -11.40$

Figure 1: Local minima of the potential energy for the Lennard-Jones cluster with seven atoms in 2D. Also indicated are the associated potential energy levels.

2.2 Implementation issues

We now briefly discuss some implementation issues.

2.2.1 Initial conditions

We first discuss how initial conditions are chosen. The invariant measure for the Langevin dynamics is the Boltzmann-Gibbs measure with density

$$\mu(q,p) = Z^{-1} \exp\left(-\beta \left[\frac{|p|^2}{2} + V_f(q)\right]\right),$$

where $|\cdot|$ is the Euclidean norm in $\mathbb{R}^{md}$ and where Z is the normalisation constant which ensures that μ is a probability density. In the Har-1d case, since $V_f(q) = q^2/2$, we simply draw the initial condition q_0 and p_0 independently according to $\mathcal{N}(0, \beta^{-1})$. Throughout this article, for Har-1d, we work with $\beta = 3$, and the corresponding initial condition is $(q_0, p_0) = (-0.386593, 0.48453)$.

For the LJ7-2d, we cannot use the same strategy as above to choose an initial condition, since it is in practice very challenging to draw q_0 according to the correct probability measure. Instead we follow a *burn-in* procedure which we now describe. We start the particles with random initial momenta (drawn from the invariant measure $\mathcal{N}(0, \beta^{-1}\,\mathrm{Id})$ for the momenta, where Id is the identity matrix of size $md \times md$) and initial positions at the bottom of the deepest well corresponding to Figure 1a (six atoms located $\pi/3$ radians apart on a unit circle and one in the centre), and we next run the Langevin dynamics. We integrate this dynamics until the time $N_{\mathrm{burn-in}}\,\Delta t$ with $N_{\mathrm{burn-in}}$ chosen such that the average kinetic energy (averaged over time, dimension and number of atoms)

$$\frac{1}{14 N_{\mathrm{burn-in}}} \sum_{n=1}^{N_{\mathrm{burn-in}}} \sum_{i \in \{1,\ldots,7\}} \frac{1}{2} \left| p^i(n\Delta t) \right|^2$$

is sufficiently close to its target value $1/(2\beta)$. This configuration, obtained at time $N_{\mathrm{burn-in}}\,\Delta t$, is then chosen as the initial condition for the parareal procedure.

2.2.2 Translation-invariance for LJ7-2d

We impose translation invariance, since it ensures that translating the entire system does not contribute to the errors. This is implemented by using zero-mean noise in both directions. More precisely, for each particle $i \in \{1, \ldots, 7\}$ and at each time-step, we draw a Gaussian random variable G_x^i for the x-coordinate, and G_y^i for the y-coordinate (recall (2.2)). The idea is then to replace G_x^i by $G_x^i - \frac{1}{7}\sum_{j=1}^{7} G_x^j$, and likewise G_y^i by $G_y^i - \frac{1}{7}\sum_{j=1}^{7} G_y^j$. Using that V_f satisfies $\sum_{j=1}^{7} \frac{\partial V_f}{\partial q_x^j} = \sum_{j=1}^{7} \frac{\partial V_f}{\partial q_y^j} = 0$, we then deduce from (2.2) that, if $\sum_{j=1}^{7} p_n^j = 0$, then $\sum_{j=1}^{7} p_{n+1}^j = 0$. The center of mass of the system hence does not move. We note that this scheme actually corresponds to integrating the Langevin dynamics projected on the manifold $\left\{(q,p) \in \mathbb{R}^{14},\ \ \sum_{j=1}^{7} p^j = 0\right\}$. We underline that the conclusions drawn on that case also hold in the absence of translation invariance, that is when we directly implement (2.2) without modification of the Gaussian increments. Since it is commonly employed in MD, we have decided here to implement this translation-invariance feature.

2.2.3 Parareal within a single well of the potential energy

In all the experiments on LJ7-2d, we work with parameters (time horizon T, friction coefficient γ, inverse temperature β, ...) such that the fine trajectory remains within the deepest well in which it initially lies. This requirement is imposed keeping in mind that using a harmonic approximation around the initial well is not a reasonable choice when the trajectory is close to a saddle point or leaves the well. The question of choosing a good coarse potential is a challenging one and often requires an understanding of the system being studied. In this article we are interested in the general features of the parareal algorithm, and consequently the study of better coarse potentials on the entire state space is left to future work (see Section 5.2 below for additional details).

2.2.4 Choice of the friction coefficient

One possible way to choose the friction coefficient γ in the Langevin equation (2.1) is by using the synchronisation rate. Considering two initial conditions which are close one to each other, we write the two dynamics (2.1) starting from these two initial conditions. Assuming that we can linearize the potential, we take the difference between both dynamics (the Brownian motions thus cancelling each other) and we look for γ such that the difference (both in position and momentum) converges to zero with the largest

possible rate. This is the case when γ is set at the so-called synchronisation rate, which is equal to twice the square-root of the smallest eigenvalue of the Hessian of the potential at the bottom of the well.

In the Har-1d case, where $V_f(q) = q^2/2$, the synchronisation rate is thus equal to 2. In the LJ7-2d case, we ignore the first three eigenvalues of the Hessian, which correspond to translation and rotation in dimension $d = 2$, and are very close to zero. The first non-zero eigenvalue for this setup is 69.1, which gives a synchronisation rate of 16.62.

We will not systematically use exactly these values, but we will consider values of γ of this order of magnitude for each system. We will observe below that the efficiency of the parareal algorithm significantly depends on γ.

3 Numerical results using the classical parareal algorithm

We now introduce some quantities useful to study the parareal algorithm. For $k \in \{1, \ldots, N\}$, the relative error (for the entire trajectory) between two consecutive parareal trajectories is defined as

$$E(k, N, \gamma, \beta) = \frac{\sum_{n=1}^{N} |q_n^k - q_n^{k-1}|}{\sum_{n=1}^{N} |q_n^{k-1}|}. \tag{3.1}$$

Since the parareal algorithm always converges, for a fixed parameter $\delta_{\text{conv}} > 0$ (convergence criterion), we stop the algorithm at the first parareal iteration $\overline{k}$ for which $E(\overline{k}, N, \gamma, \beta) < \delta_{\text{conv}}$. For a given δ_{conv}, the gain of the parareal algorithm is then defined as

$$\text{gain}(N, \gamma, \beta) = \frac{N}{\overline{k}}. \tag{3.2}$$

Note that this definition of gain assumes in particular that the cost of the coarse propagator is negligible in comparison with the cost of the fine propagator. This definition is also centered around the notion of wall-clock time, in contrast to the notion of total CPU cost. As soon as several parareal iterations are used, the total CPU cost of the parareal algorithm is larger than the cost of a single, sequential computation of the trajectory using the fine propagator. But the wall-clock time may be smaller, and this is precisely quantified by (3.2). For the sake of simplicity we choose

$$\delta_{\text{conv}} = 10^{-5} \tag{3.3}$$

for all the numerical experiments in this article.

The relative error (for the entire trajectory) between the fine trajectory (which, we recall, is denoted $(q(t), p(t))$ for $0 \leq t \leq T = N\,\Delta t$) and the parareal trajectory is

$$E_f(k, N, \gamma, \beta) = \frac{\sum_{n=1}^{N} |q(n\Delta t) - q_n^k|}{\sum_{n=1}^{N} |q(n\Delta t)|}. \tag{3.4}$$

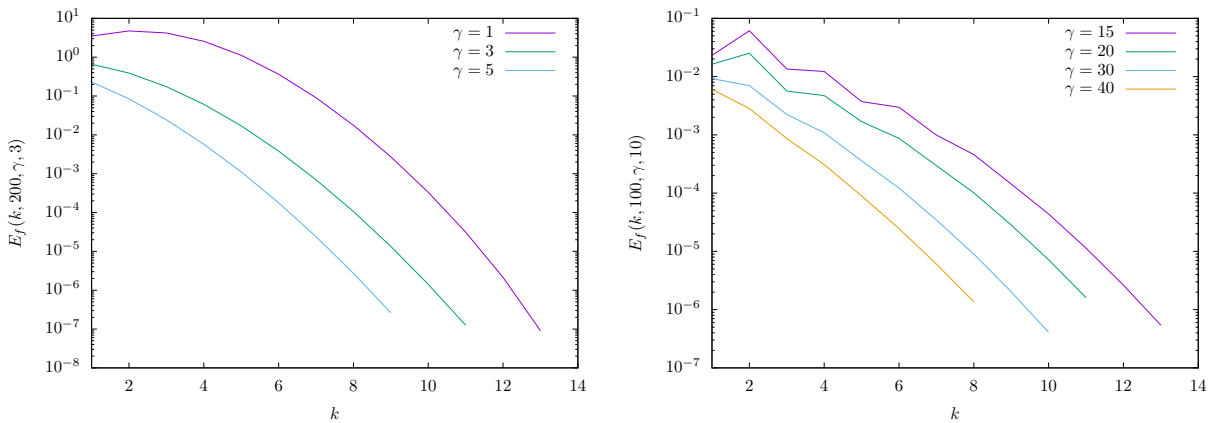
Since the parareal algorithm always converges, we have $E(N+1, N, \gamma, \beta) = 0 = E_f(N, N, \gamma, \beta)$.

Remark 3.1 (Error at terminal time)**.** Note that the relative error E defined by (3.1) is quite strong since it requires convergence over the entire trajectory. Another (weaker) measure of error is $\dfrac{|q_N^k - q_N^{k-1}|}{|q_N^{k-1}|}$, which keeps track of the relative error between two consecutive parareal trajectories only at the terminal time-step. On the examples we have considered, it turns out that this error scales the same way as E. Consequently, in this article, we use the relative errors for the entire trajectory (3.1) or (3.4) as a measure of error. We comment in Section 5.1 below on other measures of error, well adapted to the MD context. □

The standard parareal algorithm exhibits two crucial features when computing long trajectories (i.e., when $T = N\Delta t$ is large) which we now discuss.

3.1 Intermediate explosion of error for large times

For small N, the parareal trajectory converges to the reference trajectory fairly quickly when k increases. This is illustrated in Figure 2, where we plot the relative error E_f between the parareal and the fine trajectory as a function of k for various values of γ. For k of the order of 10, we have a converged trajectory over a time horizon $N\Delta t$ with N of the order of 100 (thus a gain of the order of 10). When γ increases, this convergence improves as well. This is explained by the fact that γ is the friction parameter in the Langevin dynamics (2.1), and therefore increasing γ leads to the system becoming increasingly dissipative and thus less movement of the particles. In addition, for very large values of γ, the physical forces term in the second line of (2.1) is dominated by the next two terms, and thus the coarse and the fine propagators integrate dynamics which are close one to each other.

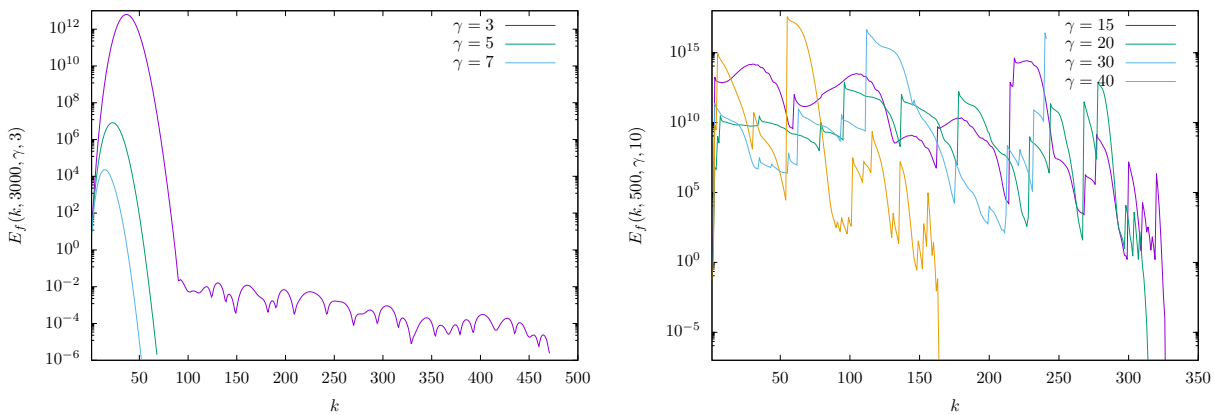


(a) Har-1d: $\omega = 0.1$, $N = 200$, $T = N\Delta t = 10$ (b) LJ7-2d: $N = 100$, $T = N\Delta t = 1$

Figure 2: Plot of $E_f(k, N, \gamma, \beta)$ as a function of k for various γ. Left: Har-1d with $\Delta t = 0.05$, $\omega = 0.1$, $\beta = 3$ and $N = 200$. Right: LJ7-2d with $\Delta t = 0.01$, $\beta = 10$ and $N = 100$.

We now comment upon the non-monotone profile of E_f for $\gamma = 15$ that we observe in Figure 2b. As we shall now see, this is a typical feature of the parareal algorithm. When $T = N\Delta t$ is large, the parareal trajectory has an intermediate non-monotone error profile with respect to the fine trajectory: the error on $[0, T]$ typically increases to very large values when k increases, before eventually decreasing and converging to zero. This is obvious in Figure 3, where we have run the same experiments as in Figure 2 (with essentially the same values of γ) but for a much larger value of N. We observe that the error E_f reaches extremely large values for intermediate values of k. In fact, in some cases, the parareal trajectory has undefined (NaN) values for certain parareal iterations, before converging when k becomes closer to N. For instance, the curve for $\gamma = 30$ in Figure 3b corresponds to such a situation: it is not complete and reaches NaN values at $k \approx 240$.

This phenomenon is explained by the fact that the parareal trajectory goes far away from the fine trajectory (and in fact takes extremely large values) before eventually converging. This is illustrated in Figure 4 for Har-1d (where we have set $N = 3000$ as in Figure 3a), where, for an intermediate value of k, the parareal trajectory has values that are orders of magnitude larger than the fine, reference trajectory. We work here with $\gamma = 3$, which corresponds to the purple curve in Figure 3a. We observe there that the error is maximal for $k \approx 40$. On the top right of Figure 4, we show the parareal trajectory for $k = 40$: for $n \geq 1400$, it takes values that are extremely large compared to the reference values, thus the very large error. When we perform additional parareal iterations and consider $k = 80$, the error is much smaller, as can be seen in Figure 3a, although still large. The parareal trajectory for $k = 80$ (see bottom left of Figure 4) now blows up only for $n \geq 2700$, and is a least of the correct order of magnitude (if not accurate) for $n \leq 2700$. We can expect from Figure 3a that the error is small for $k = 150$, a fact confirmed by the plot on the bottom right of Figure 4: the parareal trajectory for $k = 150$ can essentially be superimposed to the reference trajectory shown on the top left of Figure 4.



(a) Har-1d: $\omega = 0.1$, $N = 3000$, $T = N\Delta t = 150$ (b) LJ7-2d: $N = 500$, $T = N\Delta t = 5$

Figure 3: Plot of $E_f(k, N, \gamma, \beta)$ as a function of k for various γ. Left: Har-1d with $\Delta t = 0.05$, $\omega = 0.1$, $\beta = 3$ and $N = 3000$. Right: LJ7-2d with $\Delta t = 0.01$, $\beta = 10$ and $N = 500$. The gap in the right figure for $\gamma = 30$ corresponds to undefined (NaN) values for E_f.

Remark 3.2. In Figures 2, 3 and 4, we have chosen to work with $\omega = 0.1$ for the coarse potential in the Har-1d case (which is then a bad approximation of the fine potential) to mimick the fact that, for realistic examples, it may be difficult to design coarse potentials V_c which are at the same time less expensive to compute than V_f and sufficiently close to V_f. On purpose, we have chosen here a bad coarse potential. □

The fact that the parareal trajectory goes extremely far away for intermediate values of k is not the sole reason for slow convergence of the iterations. Actually, for large T, the parareal algorithm converges slowly even when stable. This is illustrated in Figure 5, where we use the following "trust-region based" fine and coarse potentials:

$$V_f^{\text{trust}}(q) = \begin{cases} \dfrac{q^2}{2} & \text{if } q \in [-3, 3] \\ 0 & \text{otherwise} \end{cases}, \qquad V_c^{\text{trust}}(q) = \begin{cases} \omega\, \dfrac{q^2}{2} & \text{if } q \in [-3, 3] \\ 0 & \text{otherwise} \end{cases}. \tag{3.5}$$

We consider the same numerical parameters as in Figure 4. We see on the top left plot of that figure that the fine trajectory always satisfies $|q(t)| \leq 3$. Changing the fine potential from V_f to V_f^{trust} hence does not affect the fine, reference trajectory. By using these potentials V_f^{trust} and V_c^{trust}, we make sure that the parareal trajectory experiences zero force when leaving the trust-region. As a result, the parareal trajectory (and consequently the error) stays bounded (see Figure 5). However, the parareal trajectory using these modified potentials converges to the reference trajectory as slowly as when using the original propagators. This substantiates the claim that the parareal algorithm, even when stable, converges slowly for large time horizon $T > 0$. This is consistent with the theoretical analysis for harmonic potentials presented in Section 6 (see in particular Figures 12a and 12b, in which the error remains bounded but may converge extremely slowly to zero).

The intermediate explosion discussed above leads to various practical issues. In MD, one is typically interested in certain features of the trajectory that need to be calculated online, such as the current basin of attraction. However, these quantities cannot be computed (or are meaningless) for such parareal trajectories due to the unphysically large values. Furthermore, the NaN values pose obvious issues since they are architecture dependent.

In Figure 6, we plot E (the relative error between two consecutive iterations) and E_f (the relative error with respect to the reference trajectory) as a function of k, on a simulation performed with again the same numerical parameters as in Figure 3. Recall that the parareal algorithm only uses E, and not E_f, as a

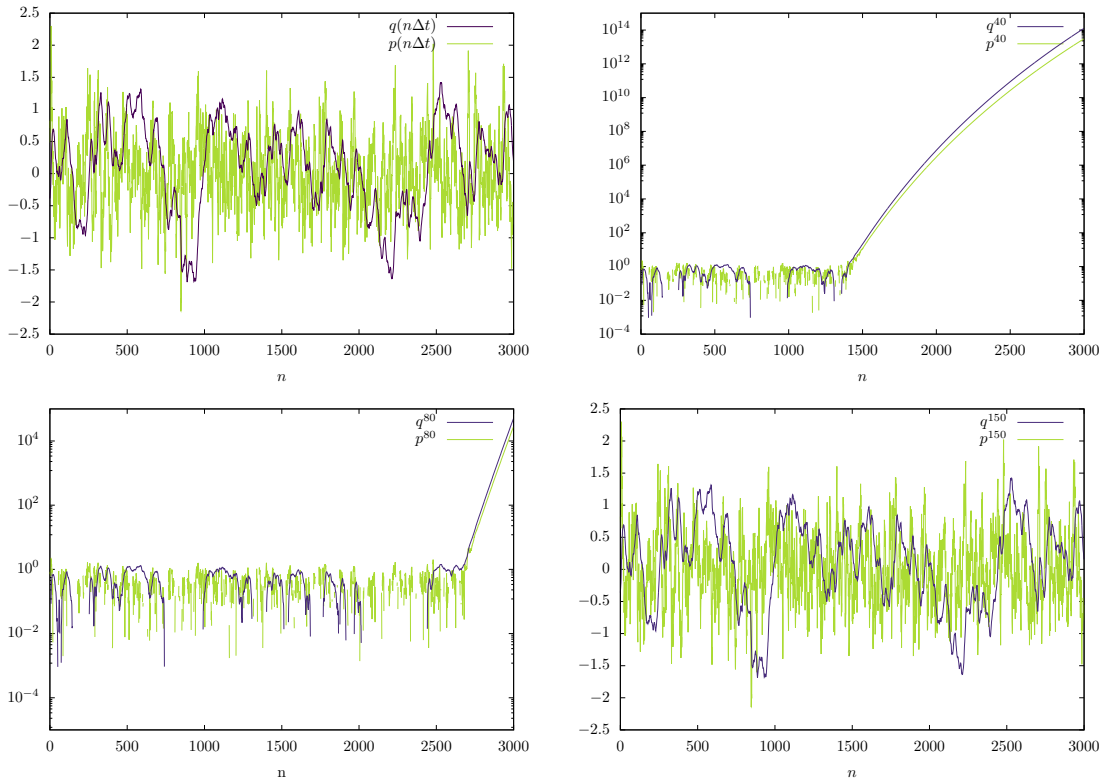


Figure 4: Plot of fine and parareal trajectories as a function of the time index for Har-1d with $\Delta t = 0.05$, $N = 3000$, $\beta = \gamma = 3$ and $\omega = 0.1$. Top left: plot of the fine trajectory. Top right: plot of (q_n^{40}, p_n^{40}) in log-scale for $k = 40$ (the gaps in the graph correspond to negative values). Bottom left: plot of (q_n^{80}, p_n^{80}) in log-scale for $k = 80$ (the gaps in the graph correspond to negative values). Bottom right: plot of (q_n^{150}, p_n^{150}) for $k = 150$. While the trajectory is stable at larger k, it blows up for intermediate k.

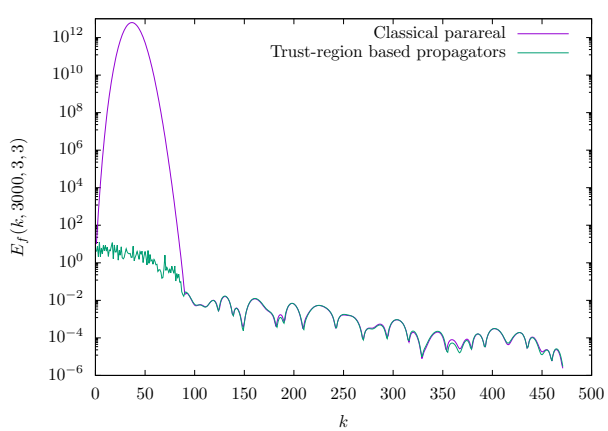
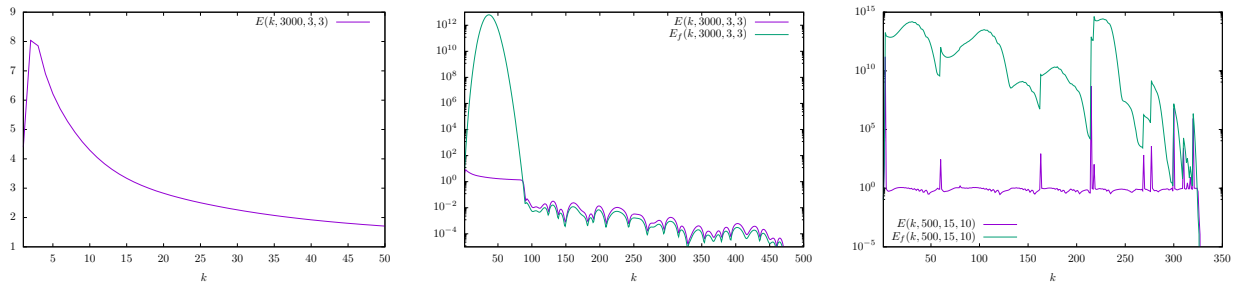


Figure 5: Plot of $E_f(k, 3000, 3, 3)$ as a function of k for Har-1d with $\omega = 0.1$, $\Delta t = 0.05$, $N = 3000$ and $\beta = \gamma = 3$ with different propagators. The classical parareal curve corresponds to the potentials defined by (2.4). The trust-region curve corresponds to the potentials defined by (3.5).

stopping criterion. However, as clearly seen on the figure, the explosion of the parareal trajectory and therefore E_f can be correlated to an increase in E beyond a certain threshold. This observation plays a key role in the adaptive parareal algorithm introduced in Section 4.



(a) Har-1d: $\omega = 0.1$, $\gamma = \beta = 3$, $\Delta t = 0.05$, $N = 3000$, $T = N\Delta t = 150$

(b) Har-1d: $\omega = 0.1$, $\gamma = \beta = 3$, $\Delta t = 0.05$, $N = 3000$, $T = N\Delta t = 150$

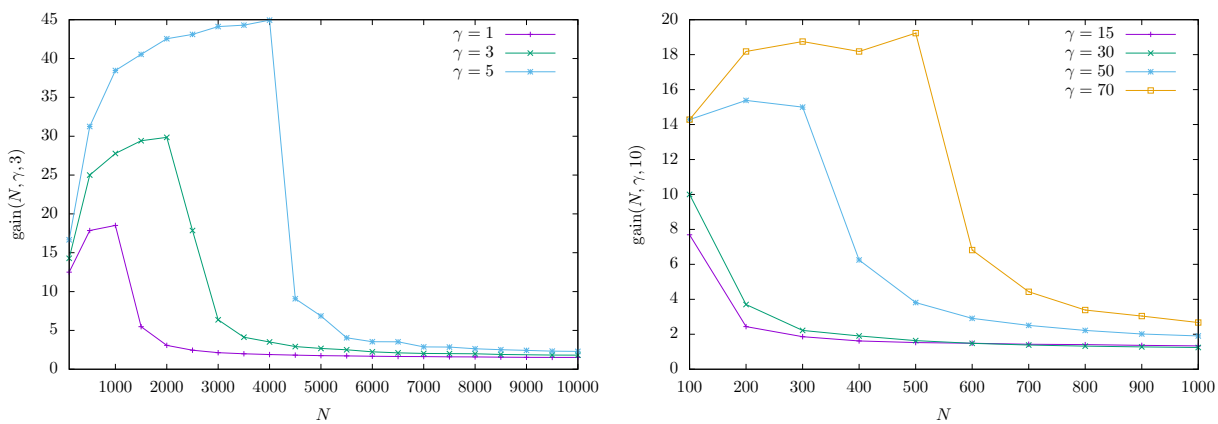
(c) LJ7-2d: $\gamma = 15$, $\beta = 10$, $\Delta t = 0.01$, $N = 500$, $T = N\Delta t = 5$

Figure 6: Left and Center: Plot of $E(k, N, \gamma, \beta)$ and $E_f(k, N, \gamma, \beta)$ for Har-1d for the first 50 (resp. 500) parareal iterations (the left plot is a zoom of the center plot for the first parareal iterations; it clearly shows the maximum value attained by the error E, a value which is to be compared with the threshold δ_{expl} introduced in Section 4 below). Right: Plot of $E(k, N, \gamma, \beta)$ and $E_f(k, N, \gamma, \beta)$ for LJ7-2d for the first 350 iterations. Note that an increase in $E(k, N, \gamma, \beta)$ beyond a threshold indicates explosion of parareal trajectory.

3.2 Gain converges to one for large N

We now focus our investigations on the gain provided by the algorithm, which is defined by (3.2).

We numerically observe that the gain converges to one when N (or equivalently T) becomes larger, and consequently the parareal algorithm does not offer any advantage, for long time simulations, over using the fine integrator in a sequential manner. This is illustrated in Figure 7, where we plot the gain for both examples as a function of N for different values of γ. Note that, while increasing γ improves the gain at fixed N (recall that we have already observed in Figure 2 that the error is smaller when considering larger values of γ), the gain nevertheless converges to one when $N \to \infty$, whatever the value of γ.



(a) Har-1d: $\omega = 0.1$, $T \in \{5, 10, \ldots, 500\}$

(b) LJ7-2d: $T \in \{1, 2, \ldots, 10\}$

Figure 7: Plot of $\text{gain}(N, \gamma, \beta)$ as a function of N for various γ. Left: Har-1d with $\Delta t = 0.05$ and $\beta = 3$. Right: LJ7-2d with $\Delta t = 0.01$ and $\beta = 10$.

An expected feature of the parareal algorithm is that the gain decreases as the coarse potential becomes a

worse approximation of the fine integrator. This is illustrated in Figure 8, where we plot the gain for Har-1d case, as a function of ω, for fixed N, γ and β. As expected, the gain decreases as ω takes values further away from the reference value $\omega_{\text{ref}} = 1$. A similar conclusion also follows from the theoretical analysis presented in Section 6. Comparing the left and the right plot in Figure 8, we also observe that the algorithm behaves differently whether $\omega > \omega_{\text{ref}}$ or $\omega < \omega_{\text{ref}}$, a fact again consistent with the analysis of Section 6, where the sign of y defined by (6.4) (or equivalently the sign of $\omega - \omega_{\text{ref}}$) matters.

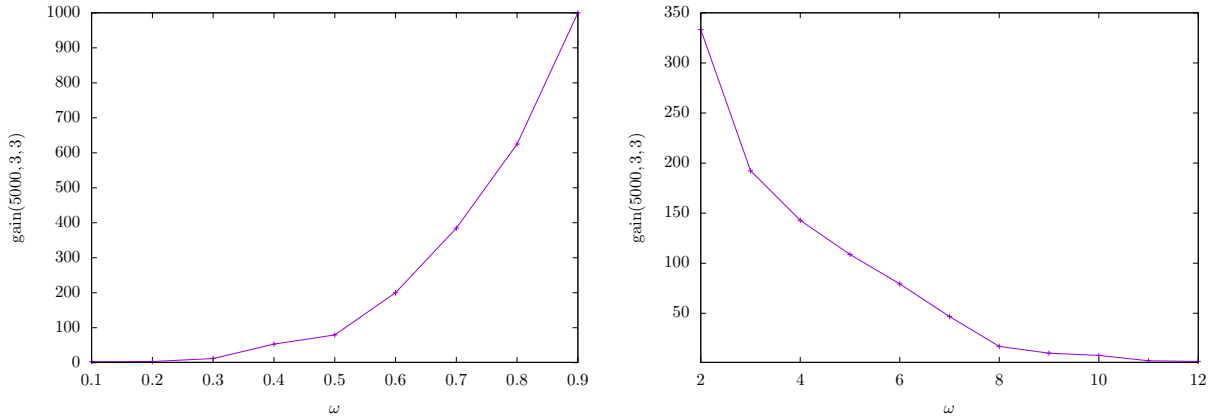


Figure 8: Plot of gain$(5000, 3, 3)$ for Har-1d (with $\beta = \gamma = 3$, $\Delta t = 0.05$ and $N = 5000$, hence $T = N\Delta t = 250$) as a function of ω. Left: $\omega < 1$. Right: $\omega > 1$. Note the different scale on the vertical axis of the two curves.

In Table 1 (given at the end of Section 4 below), we collect in blue the gains for LJ7-2d for various γ and β. Note that, for smaller β, the gain is rather insensitive to the value of β, whereas it considerably increases for $\beta \geq 18$. This is due to the fact that for larger β, the effect of the noise becomes less pronounced in the Langevin dynamics and the system tends to stay close to the bottom of the well, where the harmonic approximation of the potential is an excellent approximation. In contrast, for smaller β, the noise has a considerable effect on the trajectory, which explores a much larger part of the state space.

4 Adaptive algorithm

Let us summarise the two important observations from Section 3:

- The relative error between the fine and the parareal trajectory decreases when T decreases. We have observed this behaviour in all the above numerical experiments. This behaviour is also consistent with the theoretical analysis presented in Section 6.
- If the relative error between consecutive parareal trajectories remains larger than some threshold (say larger than 1), then the gain of the parareal algorithm is very limited (see in particular the discussion at the end of Section 3.1, where values of E larger than 1 are correlated with blow-up of the error).

Based on these observations, we now introduce an *adaptive parareal algorithm*.

Starting from the time-slab $[0, T]$, the idea is to run the parareal algorithm until the relative error between consecutive trajectories is either larger than an explosion threshold $\delta_{\text{expl}} > 0$ (we denote k_{cur} the parareal iteration at which this is attained), or smaller than the convergence threshold δ_{conv} (which, we recall, has been fixed at $\delta_{\text{conv}} = 10^{-5}$, see (3.3)). In the latter case, the method has converged on the time range $[0, T]$. In the former case, for the parareal iteration k_{cur}, we find the first time iteration $N_{\text{final}} \leq N$ for which the relative error exceeds δ_{expl}, and change the slab to $[0, (N_{\text{final}} - 1)\Delta t]$. We then proceed with the parareal iterations on the smaller time-slab $[0, (N_{\text{final}} - 1)\Delta t]$ (note that we do not need to perform again the first

k_{cur} iterations, we can just proceed from the knowledge of $\{x_n^{k_{\text{cur}}}\}_{0\leq n\leq N_{\text{final}}-1})$, until the relative error (now of course on $[0,(N_{\text{final}}-1)\Delta t])$ between consecutive trajectories is either larger than δ_{expl}, or smaller than δ_{conv}. In the latter case, the method has converged on the time range $[0,(N_{\text{final}}-1)\Delta t]$. In the former case, we again reduce the size of the time-slab. We hence continue to find smaller and smaller slabs until the parareal algorithm converges, and we denote $[0,\widetilde{T}_1]=[0,\widetilde{m}_1\Delta t]$ the slab which has eventually been found.

Once we have a converged parareal trajectory on $[0,\widetilde{m}_1\Delta t]$, we proceed with the next part of the time range and define the new time-slab as $[\widetilde{m}_1\Delta t,T]$. We proceed with the parareal iterations (starting from the converged configuration at time $\widetilde{m}_1\Delta t$), and possibly reduce the size of the time-slab to $[\widetilde{m}_1\Delta t,\widetilde{m}_2\Delta t]$ in order to reach convergence.

This procedure is repeated until we have reached the time horizon T.

The pseudocode for this algorithm is presented in Algorithm 1, where we use the following notation for the relative error over a given time interval: for $N_{\text{init}},N_{\text{final}}\in\mathbb{N}$ with $N_{\text{init}}\leq N_{\text{final}}$ and two trajectories $\{x_n^{\text{prev}}\}_{N_{\text{init}}\leq n\leq N_{\text{final}}}$ and $\{x_n^{\text{cur}}\}_{N_{\text{init}}\leq n\leq N_{\text{final}}}$, the relative error is

$$E(N_{\text{init}},N_{\text{final}};x^{\text{prev}},x^{\text{cur}})=\frac{\sum_{n=N_{\text{init}}}^{N_{\text{final}}}|x_n^{\text{cur}}-x_n^{\text{prev}}|}{\sum_{n=N_{\text{init}}}^{N_{\text{final}}}|x_n^{\text{prev}}|}.$$

The relative error between two consecutive parareal trajectories is computed in line 11 of the algorithm. We actually do not wait to reach the current end point of the time-slab to compute it, but rather compute it for each new time point in $[N_{\text{init}}\Delta t,N_{\text{final}}\Delta t]$, and exit the computation (to redefine a new endpoint) whenever this relative error is larger than δ_{expl}. Note that it is of course advantageous to compute independently the numerator and the denominator of this relative error (which are easy to increment for each new value of x_{n+1}^{cur}, with $N_{\text{init}}+1\leq n+1\leq N_{\text{final}}$). The `while` instruction on line 5 corresponds to the parareal iterations: as long as $\delta>\delta_{\text{conv}}$ (no convergence yet) and $\delta<\delta_{\text{expl}}$ (no explosion), we proceed with the iterations. Note that we always exit this `while` loop, since we know that, for sufficiently large k, the parareal algorithm converges (recall (1.3)): if the trajectory does not blow up (i.e. if δ remains smaller than δ_{expl}), then the algorithm reaches some iteration where $\delta<\delta_{\text{conv}}$. The `else` condition on line 20 corresponds to the case $\delta<\delta_{\text{conv}}$ (since we have exited the `while` loop, we can only have $\delta>\delta_{\text{expl}}$ or $\delta<\delta_{\text{conv}}$).

The key parameter in the adaptive algorithm is the threshold of explosion, which we have denoted by δ_{expl}. More precisely, the algorithm adaptively chooses the slab size for which the relative error is smaller than δ_{expl}. This means that for smaller δ_{expl}, these slabs are of smaller size and we need a larger number of slabs to reach the final time horizon T. In contrast, if δ_{expl} were chosen very large, the relative error would always be smaller than δ_{expl} and a unique slab, namely $[0,T]$, would be used. This is illustrated in Figure 9 for the Har-1d case for two large values of N. The influence of δ_{expl} on the gain is discussed below. We simply point out for now that, in this particular case and for $N=5000$ (see Figure 10a and also, for larger values of N, Figure 11a), the optimal choice for best gain is $\delta_{\text{expl}}=5$. Figure 9 shows that the algorithm builds 4 slabs (resp. 20 slabs) to converge on $[0,N\Delta t]$ with $N=5000$ (resp. $N=30000$), and the size of these slabs varies from one to the next. The number of CPUs required in practice to obtain a perfect parallelization of the fine propagators thus corresponds to the largest slab-size. For instance, the case $N=30000$ and $\delta_{\text{expl}}=1$ (resp. $\delta_{\text{expl}}=5$) requires close to 440 (resp. 2700) CPUs. While this may be considered as a large number (and while some of these CPUs are not used when the slab size is shorter), it is considerably less compared to the classical parareal algorithm which would require $N=30000$ CPUs. Furthermore the number of CPUs required is fairly robust when N increases (compare the left and the right plots of Figure 9, e.g. for $\delta_{\text{conv}}=1$ and for $\delta_{\text{conv}}=3$). We also observe that the fluctuation of the size of the slabs seems to depend on δ_{expl}, and seems to be larger when δ_{expl} is larger.

We define the cost (as above, in terms of wall clock time and neglecting the cost of the coarse propagator compared to the fine propagator) for the adaptive parareal algorithm as follows. Consider a tentative time-slab $[\widetilde{m}_j\Delta t,T]$, which has possibly been eventually shortened to $[\widetilde{m}_j\Delta t,\widetilde{m}_{j+1}\Delta t]$ in order for the parareal iterations to reach convergence on that interval. Since the fine propagations are performed in parallel, the cost of the algorithm over the time-slab $[\widetilde{m}_j\Delta t,\widetilde{m}_{j+1}\Delta t]$ is the total number C_j of parareal iterations that

Algorithm 1: Adaptive parareal algorithm

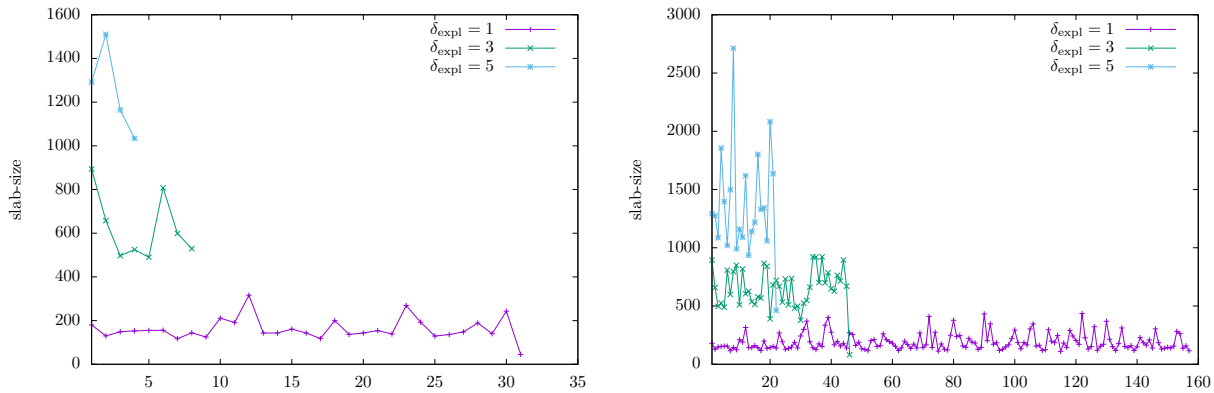
Numerical parameters: N, δ_{conv}, δ_{expl}

1 Set $N_{\text{init}} := 0$, $N_{\text{final}} := N$ and $\delta := (\delta_{\text{conv}} + \delta_{\text{expl}})/2$;
2 Compute $\{x_n^{\text{cur}}\}_{N_{\text{init}} \le n \le N_{\text{final}}}$: $x_0^{\text{cur}} := x_0$ and sequential update $x_{n+1}^{\text{cur}} := \mathcal{C}_{\Delta t}(x_n^{\text{cur}})$;
3 Set Cost := 0;
4 **while** $N_{\text{init}} < N$ **do**
5 **while** $\delta \in [\delta_{\text{conv}}, \delta_{\text{expl}}]$ **do**
6 Define $\{x_n^{\text{prev}}\}_{N_{\text{init}} \le n \le N_{\text{final}}}$ as $x_n^{\text{prev}} := x_n^{\text{cur}}$;
7 Compute $\{J_n\}_{N_{\text{init}} \le n \le N_{\text{final}}}$ (in parallel): $J_n := \mathcal{F}_{\Delta t}(x_n^{\text{prev}}) - \mathcal{C}_{\Delta t}(x_n^{\text{prev}})$;
8 Cost = Cost +1;
9 **for** $n \leftarrow N_{\text{init}}$ **to** $N_{\text{final}} - 1$ **by** 1 **do**
10 $x_{n+1}^{\text{cur}} := \mathcal{C}_{\Delta t}(x_n^{\text{cur}}) + J_n$;
11 Compute the relative error $\delta = E(N_{\text{init}}, n+1; x^{\text{prev}}, x^{\text{cur}})$;
12 **if** $\delta > \delta_{\text{expl}}$ **then**
13 $N_{\text{expl}} := n + 1$;
14 break; `// exit the for loop if condition satisfied; we also exit the while loop since` δ `is too large`
15 **end**
16 **end**
17 **end**
18 **if** $\delta > \delta_{\text{expl}}$ **then**
19 $N_{\text{final}} := N_{\text{expl}} - 1$;
20 **else**
21 $N_{\text{init}} := N_{\text{final}}$;
22 Compute $\{x_n^{\text{cur}}\}_{N_{\text{init}} \le n \le N}$: $x_{N_{\text{init}}}^{\text{cur}} := x_{N_{\text{final}}}^{\text{cur}}$ and sequential update $x_{n+1}^{\text{cur}} := \mathcal{C}_{\Delta t}(x_n^{\text{cur}})$;
23 $N_{\text{final}} := N$;
24 **end**
25 Set $\delta := (\delta_{\text{conv}} + \delta_{\text{expl}})/2$;
26 **end**

Output of the algorithm: $\{x_n^{\text{cur}}\}_{0 \le n \le N}$ and Cost

have been performed for the trajectory to converge on $[\widetilde{m}_j \Delta t, \widetilde{m}_{j+1} \Delta t]$ (or equivalently the number of times the fine integrator has been called to propagate the system from $\widetilde{m}_j \Delta t$ to $(1 + \widetilde{m}_j)\Delta t$; of course, on the time-slab $[\widetilde{m}_j \Delta t, \widetilde{m}_{j+1} \Delta t]$, the fine propagator has to be called to propagate the system further in time than $(1 + \widetilde{m}_j)\Delta t$, but this is performed in parallel and thus should not be accounted for in terms of wall-clock time). We then define the cost of the adaptive variant as the sum of these C_j over all the time-slabs that have been needed to reach the final time horizon T: if the time interval $[0, T]$ has been decomposed by the adaptive algorithm as $[0, T] = \cup_{0 \le j \le J-1} [\widetilde{m}_j \Delta t, \widetilde{m}_{j+1} \Delta t]$, then the cost is defined as $\sum_{0 \le j \le J-1} C_j$. The reason for this definition is that the computation from one time-slab to the next is performed in a sequential manner (the trajectory should be converged on $[\widetilde{m}_j \Delta t, \widetilde{m}_{j+1} \Delta t]$ before we move on to $[\widetilde{m}_{j+1} \Delta t, \widetilde{m}_{j+2} \Delta t]$), and thus the wall clock times should add up. The computation of this cost is performed in line 8 of the algorithm.

Following the definition (3.2) of the gain for the classical parareal algorithm, we define the gain of the adaptive parareal algorithm (compared to a sequential computation) as the ratio of N divided by the cost defined above. Note that, if the adaptive variant turns out to use a unique slab to reach the final time horizon, then both definitions of gain, (3.2) and the present one, coincide. In Figure 10 we compare the gain of the classical and the adaptive parareal algorithm as a function of N (the targeted time horizon), for various values of δ_{expl}. For moderate values of δ_{expl} (say $0.1 \le \delta_{\text{expl}} \le 5$ for the Har-1d case, and $1 \le \delta_{\text{expl}} \le 10^2$



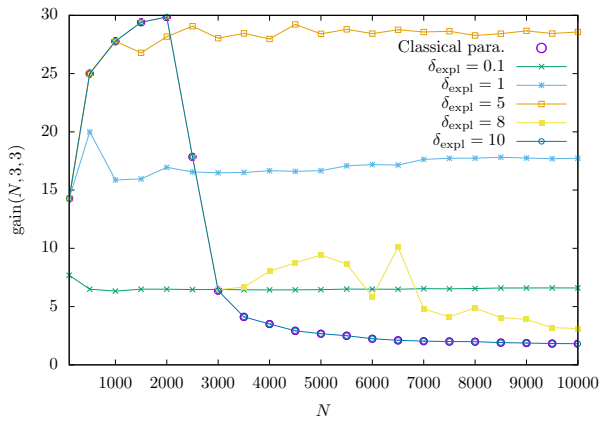
(a) $N = 5000$, $\Delta t = 0.05$ ($T = N\Delta t = 250$) (b) $N = 30000$, $\Delta t = 0.05$ ($T = N\Delta t = 1500$)

Figure 9: We use the Adaptive parareal algorithm for the Har-1d case with $\omega = 0.1$ and $\beta = \gamma = 3$. For two different values of the time horizon (left: $T = 250$; right: $T = 1500$), and different choices of δ_{expl}, we plot the list of the sizes of the time-slabs found by the algorithm. For instance, when $N = 5000$ and $\delta_{\text{expl}} = 1$, the total time interval $[0, T]$ is eventually divided into 31 slabs, each of which of size of the order of 160 time-steps. The 12-th slab is the longest and its length is of the order of 350 time-steps. The 31st slab, which is the last one needed to reach the final horizon, is the shortest with a length of the order of 50 time-steps. When $N = 5000$ and $\delta_{\text{expl}} = 5$, the total time interval $[0, T]$ is eventually divided into 4 slabs, of length close to 1300, 1500, 1200 and 1000 time-steps.

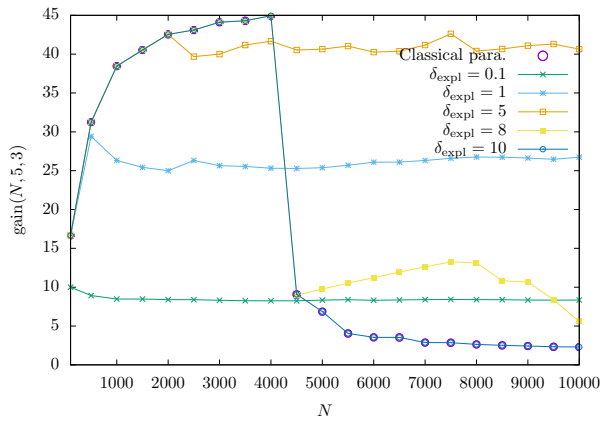
for the LJ7-2d case), the gain seems to be fairly robust and independent of N. Comparing the left and the right plots of Figure 10, we note that the gain increases when γ increases. As expected, for the largest values of δ_{expl}, the adaptive parareal algorithm is identical to the classical parareal algorithm. It thus leads to the same gain, which converges to 1 when N increases, as already discussed in Section 3.2. For large values of N (which is our regime of interest here), the adaptive algorithm always outperforms the classical version. For a good choice of δ_{expl}, we can reach a gain of the order of 30 for the Har-1d case, and of 7 for the LJ7-2d case. In the LJ7-2d case, and in contrast to the Har-1d case, we also observe that the adaptive variant outperforms the classical variant even for small values of N.

In Figure 11, we plot the gain as a function of δ_{expl}, for fixed N, γ and β. Note that the gain is fairly stable for long-times: we have already observed this in Figure 10, and we confirm this observation by comparing the two curves in the Har-1d case in Figure 11a. The large plateau in the LJ7-2d case is due to the fact that the relative error E has extremely high spikes (recall Figure 6c): whether we set $\delta_{\text{expl}} = 0.1$ or $\delta_{\text{expl}} = 100$, adaptivity of the time-slabs is triggered at the same moment, corresponding to the high spikes. The comparison between Figures 11b and 11c allows to investigate the influence of γ. We observe that, when γ increases, the plateau increases in size as well, which is in agreement with the fact that the algorithm is more stable when γ increases. In all the plots of Figure 11, the gain is maximal for a certain optimal choice of δ_{expl}, which we expect to be system-dependent. When δ_{expl} becomes larger than the optimal choice, the adaptive algorithm allows for larger slabs and thereby tends to behave like the classical parareal algorithm, which explains the diminishing gain. In contrast, when δ_{expl} is much smaller than the optimal choice, the time-slabs become very short, and there is almost no parallelisation over different subintervals anymore, thus again a diminishing gain.

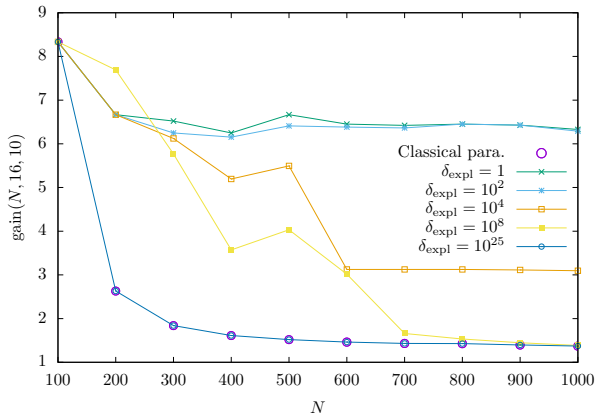
We collect in Table 1 the gains for LJ7-2d for various γ and β, for the classical parareal algorithm (in blue) and for the adaptive variant (in black). We observe that, when the classical variant does not perform satisfactorily (and provides a gain close to 1), then the adaptive variant significantly outperforms it. We have not observed gains smaller than 4 with the adaptive variant. When the classical variant already performs in an efficient manner, then the adaptive variant provides similar (if not larger) gains. For smaller values of



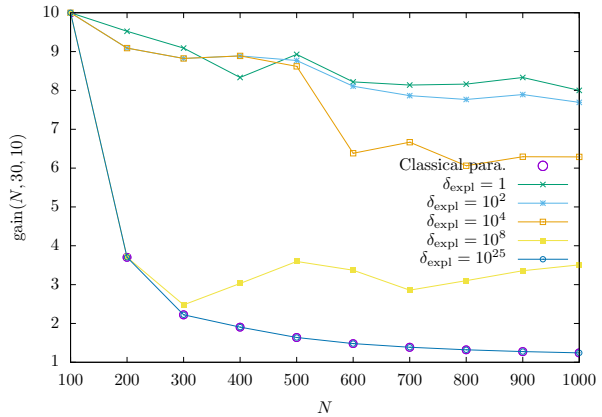
(a) Har-1d: $\omega = 0.1$, $\gamma = 3$, $T \in \{5, 10, \ldots, 500\}$



(b) Har-1d: $\omega = 0.1$, $\gamma = 5$, $T \in \{5, 10, \ldots, 500\}$

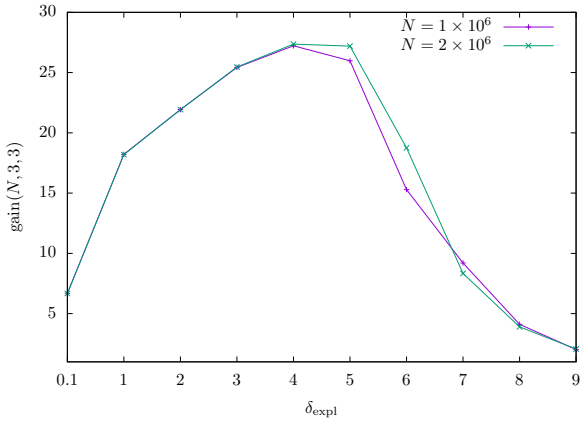


(c) LJ7-2d: $\gamma = 16$, $T \in \{1, 2, \ldots, 10\}$

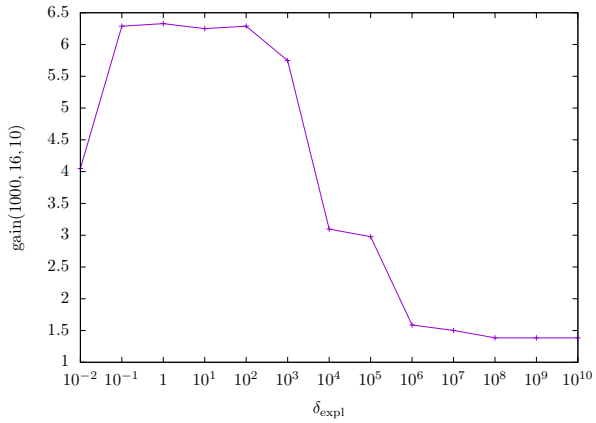


(d) LJ7-2d: $\gamma = 30$, $T \in \{1, 2, \ldots, 10\}$

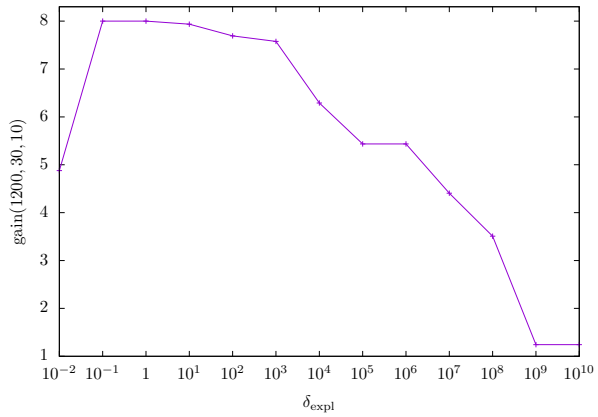
Figure 10: Plot of gain(N, γ, β) as a function of N for various γ for the classical and the adaptive parareal algorithm. Top: Har-1d with $\Delta t = 0.05$, $\beta = 3$ and $\gamma = 3$ (left) or $\gamma = 5$ (right). Bottom: LJ7-2d with $\Delta t = 0.01$, $\beta = 10$ and $\gamma = 16$ (left) or $\gamma = 30$ (right).



(a) Har-1d: $\omega = 0.1$, $T \in \{5000, 10000\}$



(b) LJ7-2d: $T = 10$



(c) LJ7-2d: $T = 12$

Figure 11: Plot of gain(N, γ, β) as a function of δ_{expl}. Top: Har-1d with $\Delta t = 0.05$, $\beta = \gamma = 3$, $N = 10^6$ and $N = 2 \times 10^6$. Bottom left: LJ7-2d with $\Delta t = 0.01$, $\gamma = 16$, $\beta = 10$, $N = 1000$. Bottom right: LJ7-2d with $\Delta t = 0.01$, $\gamma = 30$, $\beta = 10$, $N = 1200$.

β \ γ	1	5	10	15	20	25	30	35	40	45	50
5	1.16	1.26	1.29	1.36	1.31	1.58	1.46	1.65	1.78	1.97	2.07
	3.85	4.42	4.74	5.46	6.13	6.75	7.57	8.13	9	9.9	10
8	1.3	1.07	1.36	1.38	1.13	1.44	1.46	1.43	1.44	1.45	1.55
	4.38	4.69	4.97	5.68	6.13	6.41	7.3	7.63	8.2	8.85	9.71
10	1.22	1.29	1.21	1.34	1.41	1.4	1.24	1.45	1.72	1.83	1.9
	4.69	5.05	5.62	6.09	6.94	7.63	8	8.69	10.41	11.11	12.04
13	1.3	1.29	1.34	1.42	1.26	1.38	1.56	1.59	1.89	2.04	2.09
	5.05	5.71	6.58	7.04	7.81	8.69	10.1	11.24	12.19	12.5	12.99
15	1.29	1.3	1.42	1.44	1.24	1.69	1.69	1.93	2.22	2.07	2.96
	5.46	6.29	6.76	7.41	8.26	10.1	10.53	11.76	12.05	13.89	15.38
18	1.28	1.15	1.26	1.47	1.78	1.79	2.34	2.82	3.31	4.27	4.59
	5.88	6.71	7.46	8.93	9.71	10.75	11.76	13.33	14.92	15.38	16.39
20	1.22	1.42	7.75	12.34	14.08	15.87	17.86	19.61	21.74	23.81	25.64
	6.58	8.33	7.75	12.34	14.08	15.87	17.86	19.61	21.74	23.81	25.64

Table 1: This table collects the gain(N, γ, β) for LJ7-2d with $\Delta t = 0.01$ and $N = 1000$ for various γ and β. The values in blue correspond to the classical parareal algorithm and those in black to the adaptive parareal algorithm with $\delta_{\text{expl}} = 1$. Note that, for each different value of β, a different initial condition has been chosen, using the burn-in procedure described in Section 2.2.1.

β, the adaptive variant performances are slightly increasing when β increases. They dramatically improve for larger values of β. In contrast to the classical variant, the evolution of the performances of the adaptive variant as a function of β is smoother. In addition, when γ increases, the gain of the adaptive variant steadily increases, in contrast to the classical variant for which the gain increase may be very limited.

5 Discussions

In this article, we have considered a version of the parareal algorithm using integrators based on different dynamics and we have outlined its limitations, which include possible intermediate explosion and poor gain for longer trajectories. We have then introduced a modified version of the parareal algorithm which adaptively chooses the size of the time-slabs. The adaptive variant does not show the above mentioned shortcomings. We have observed that it systematically outperforms the classical version of the algorithm, and that it always provides significant gains on the problems we have studied.

5.1 Some comments on alternative ideas

We comment here on some related ideas from the literature about the parareal algorithm in the MD context and some associated numerical tests that we have performed.

Stabilising the parareal algorithm with projections. The authors of [DLBLM13] consider Hamiltonian systems (which have traditionally been challenging test cases for the parareal algorithm) and introduce symmetrised variants of the parareal algorithm (we also refer to [DM13] for a study of the wave equation). The additional idea in [DLBLM13] is to perform a projection within the parareal iteration, such that the energy (i.e. the Hamiltonian) of the parareal trajectory matches with that of the fine trajectory. Such a projection is of course motivated by the fact that, in the Hamiltonian setting, the energy is preserved along the exact dynamics. In principle, similar ideas can be applied to the Langevin dynamics, wherein we could ensure that the parareal trajectory matches the value of either the Hamiltonian (physical energy) or of some Lyapunov function (see e.g. [Tal02, Eq. (2.1)]) along the fine trajectory. Recall in particular that

such Lyapunov functions (which are built by considering the Langevin dynamics without noise) are very useful functions to explain convergence to equilibrium of the Langevin equation. Furthermore, in contrast to the physical energy, they depend on γ, and may thus appear to be a more adapted choice to a given Langevin dynamics than the Hamiltonian energy. It turns out that these matching algorithms lead to extremely slow convergence (if not lack of convergence at all). The reason is that the Langevin dynamics does not have a purely dissipative structure (which would be relevant for Lyapunov matching), neither of course a Hamiltonian structure.

γ-adaptive parareal algorithm. Recently, the authors of [MM20] consider a variant of the classical parareal algorithm, where the time-step of the fine integrator is adaptively decreased along the parareal iterations. The heuristics behing this strategy is that it is useless, for the first few parareal iterations, to accurately compute the fine propagator, since these first iterations only serve as an "initial guess" for the latter ones. This strategy is shown to lead to better gains. While decreasing the step-size is not a feasible strategy in the MD context, a similar idea could be implemented by choosing a fine integrator which starts at a larger value of γ (since parareal is more stable at larger γ) and then adaptively reduces γ to the original value along the parareal iterations. While this idea offers positive results when the original dynamics starts with large γ, it does not even converge for intermediate values of γ such as the ones discussed in this article. A typical result is that the first parareal iterations are somewhat stable. Then, when γ decreases below a certain threshold (irrespective of how slowly γ decreases along the parareal iterations), instabilities appear and prevent the algorithm from converging.

Trust-region based parareal algorithm. Another possible strategy, inspired by trust-region optimisation, is to introduce a modified coarse propagator which uses the coarse potential when inside the trust-region and the fine potential otherwise (it is thus a different – although related – idea than (3.5)). A motivation is that instabilities appear when the trajectory leaves the domain where it should be, and that reverting to the fine potential in that case may help fix the instabilities. It turns out that this strategy is costly in terms of wall-clock time, since the expensive fine propagator happens to be evaluated in a sequential manner, which defeats the purpose of the parareal algorithm in the first place. In fact, even when a zero potential (instead of V_f) is used for the coarse-propagator outside the trust-region, the gain behaves as in the classical parareal case (see Equation (3.5), Figure 5 and the corresponding discussion).

Trajectorial versus statistical properties. In the present context of MD, we recall that statistical properties of the system (as opposed to individual trajectories, which is the focus of this article in view of the definition (3.4) of the error) are of interest. They are typically less demanding in terms of accuracy for the numerical scheme: examples abound for which the numerical trajectory is plainly wrong but its statistical properties are very accurate. However, as we have observed, the individual trajectories provided by the classical parareal algorithm are far away from the fine trajectories at long times. They thus cannot be expected to yield correct averages. From the perspective of statistical quantities, the classical variant thus presents the same limitations as for trajectory computations.

5.2 Open questions

We now summarise some open problems and describe some perspectives of this work.

Why does the parareal trajectory explode before converging? We have shown that, in several MD cases, the parareal trajectory blows-up before eventually converging to the fine trajectory. It is yet unclear to us why exactly such an explosion occurs. Note that this feature is not observed in the (classical) setting when the fine and coarse propagators solve the same parabolic (i.e. dissipative) dynamics with different time-steps. We have observed it for parabolic problems when the two propagators are based on different dynamics (see Section 6 below, in particular Figure 12). This issue has also been observed when solving hyperbolic problems (see e.g. [DLBLM13, Figs. 4 and 5]).

Global coarse potential for LJ7-2d. In this article, we have used the harmonic approximation of the fine potential around the deepest well as the coarse potential, and we have considered initial conditions in that deepest well. However, such an approximation is a bad choice if one is interested in conformational changes which involve a trajectory visiting multiple wells. A natural next step of this work is thus to explore global coarse potentials and ascertain the gain achieved by the (classical and adaptive) parareal algorithms in this setting. We intend to explore this question in future works on more realistic physical systems, using various force fields.

Focus on dynamical quantities. Quantities of interest in MD include thermodynamic quantities and dynamical quantities. For the computation of the former, several methods have been proposed in the literature (see e.g. [LRS10]). Some of them rely on the long-time integration of the Langevin dynamics (and then a time-parallelisation tool such as the adaptive strategy introduced in this article may be very useful). But many other strategies, based on different ideas, have been proposed. In contrast, dynamical quantities (such as velocity autocorrelation plots, diffusion coefficients of defects – vacancies, interstitials, ... – in a material, exit times from a given well, to name but a few) are defined on the basis of the dynamics. In this context, it is of paramount interest to be able to speed-up the computation of long time trajectories, for instance using the adaptive strategy introduced here. Investigating the efficiency of parareal algorithms for the computation of such dynamical quantities is a very interesting question, and we intend to explore it on more realistic systems in forthcoming works.

6 Theoretical analysis on a toy problem

In the above sections, we have studied parareal algorithms for the Langevin dynamics. These are not straightforwardly amenable to numerical analysis, because the variable in a Langevin dynamics is necessarily multi-dimensional. In this section, we analyse the error between the fine (reference) and the parareal trajectories, for a simple ODE in dimension $d = 1$ driven by harmonic potentials (we are going to actually consider the same potentials V_f and V_c as in the Har-1d case, see (2.4)). As we shall see, this simpler setting offers explicit expressions of the propagators and of the solution x_n^k for any n and k, which allows us to theoretically discuss various aspects of the classical parareal algorithm that we have outlined in Section 3 on the basis of numerical simulations. Our main result is Proposition 6.5, which describes the behaviour of the relative error as a function of n and k, depending on the quality of the coarse propagator (namely, how close the coarse and the fine propagators are).

Let us briefly mention that several theoretical analysis of parareal algorithms have already been proposed in the literature. We refer e.g. to [MT02, MT05], to [Bal05, SR05] for stability investigation, to [GH08] for a numerical analysis in a nonlinear context and to [GH14] for an analysis in the specific Hamiltonian context. To the best of our knowledge, our result below is the first one to shed light on the interplay on the error between the quality of the coarse propagator (with respect to the fine one) and the length of the time interval on which the parareal algorithm is used.

Remark 6.1. Our analysis below applies to the setting of this article, namely when the coarse and fine propagators integrate different dynamics (governed by different potentials). But it also applies to the classical case when the coarse and fine propagators integrate the same dynamics with different time-steps and/or different numerical schemes. All what matters (see e.g. our main result, Proposition 6.5) is the value of y defined in (6.4) below. □

Consider the reference ODE

$$\frac{dx}{dt} = -V_f'(x) = -x, \qquad x(0) = x_0, \tag{6.1}$$

for $x : [0, T] \to \mathbb{R}$, where $T > 0$ is the finite-time horizon over which we study the dynamics and where V_f is the reference potential defined by (2.4). Without loss of generality, we assume that the initial condition x_0

is positive: $x_0 > 0$. As before, we divide the interval $[0, T]$ into subintervals of length Δt with $N = T/\Delta t$. The fine propagator $\mathcal{F}_{\Delta t}$ corresponding to (6.1) is linear and given by

$$\mathcal{F}_{\Delta t}(x) = Fx, \quad \text{where} \quad F = \exp(-\Delta t).$$

For $\omega > 0$, we define the coarse propagator as the propagator (over the time length Δt) of the coarse model $\dfrac{dx_t}{dt} = -V_c'(x_t) = -\omega\, x_t$ (where V_c is the coarse potential also defined by (2.4)). We therefore have

$$\mathcal{C}_{\Delta t}(x) = Cx, \quad \text{where} \quad C = \exp(-\omega \Delta t).$$

Denoting by x_n^k the solution at the k-th parareal iteration and n-th time-step, the parareal algorithm reads

$$x_{n+1}^{k+1} = \mathcal{C}_{\Delta t}(x_n^{k+1}) + \mathcal{F}_{\Delta t}(x_n^k) - \mathcal{C}_{\Delta t}(x_n^k) = Cx_n^{k+1} + (F - C)x_n^k, \qquad x_0^{k+1} = x_0 \tag{6.2}$$

and the 0-th parareal iteration is given by

$$x_{n+1}^0 = \mathcal{C}_{\Delta t}(x_n^0) = Cx_n^0, \qquad x_0^0 = x_0. \tag{6.3}$$

In what follows, we fix n, and study the relative error at the terminal point between the parareal and the fine trajectory

$$R_{n,k} = \frac{x(n\Delta t) - x_n^k}{x(n\Delta t)} = \frac{x_n^n - x_n^k}{x_n^n} \quad \text{where} \quad k \in \{0, \dots, n-1\},$$

where we have used the parareal algorithm property that $x_n^n = x(n\Delta t)$. Note that $R_{n,k}$ is mathematically more tractable than the relative error for the entire trajectory that we have used to present the numerical experiments in the above sections (recall Remark 3.1). A critical parameter in this study is

$$y = \frac{F}{C} - 1, \tag{6.4}$$

which quantifies how much the coarse and the fine propagators are far apart. Using the explicit expressions of F and C, we have $y = \exp\big((\omega - 1)\Delta t\big) - 1$, which satisfies

$$y > 0 \ \text{ if } \ \omega > 1 \qquad \text{and} \qquad y \in (-1, 0) \ \text{ if } \ \omega \in (0, 1).$$

Of course, if the coarse and fine propagators are identical, namely if $\omega = \omega_{\text{ref}} = 1$, then $y = 0$.

The following result provides an explicit expression for x_n^k in terms of y, and is useful to prove the results below.

Lemma 6.2. *For any integer $n \geq 1$, define $(0)_n = C^n$ and, for any $p \in \{1, \dots, n\}$, define $(p)_n = C^n\, y^p \binom{n}{p}$. Then, for any $n \geq 1$ and any $k \in \{1, \dots, n\}$, we have*

$$x_n^k = \sum_{p=0}^{k} (p)_n\, x_0. \tag{6.5}$$

Furthermore, we have $x_n^n = F^n\, x_0 = x(n\Delta t)$.

Proof. Using (6.2) and (6.3), the tuple $z_{n+1} := (x_{n+1}^0, x_{n+1}^1, \dots, x_{n+1}^k)^T \in \mathbb{R}^{k+1}$, which contains the $(n+1)^{\text{th}}$ time-iteration for the first k parareal iterations, evolves according to the matrix evolution

$$z_{n+1} = Lz_n, \quad \text{with} \quad L = \begin{bmatrix} C & 0 & \dots & 0 \\ F - C & C & & \\ & \ddots & \ddots & \\ 0 & & F - C & C \end{bmatrix},$$

where $L \in \mathbb{R}^{(k+1)\times(k+1)}$. Using this relation and denoting $z_0 = (x_0, \dots, x_0)^T$ the initial condition, the iterative formula for z_n is

$$z_n = L^n \, z_0.$$

Since L is a lower triangular matrix, we find

$$L^n = \begin{bmatrix} C^n & 0 & 0 & \dots & 0 \\ (1)_n & C^n & 0 & \dots & 0 \\ (2)_n & (1)_n & C^n & & \\ \vdots & \ddots & \ddots & \ddots & \\ (k)_n & \dots & (2)_n & (1)_n & C^n \end{bmatrix},$$

where, for any $p \in \{1, \dots, k\}$,

$$(p)_n = \sum_{\substack{i_1,\dots,i_p=0 \\ i_1+\dots+i_p \le n-p}} C^{i_1}(F-C)C^{i_2}(F-C)\dots C^{i_p}(F-C)\, C^{n-p-\sum_{\alpha=1}^{p} i_\alpha}.$$

Using that C and $F - C$ are scalar numbers (a fact that would not hold for a true Langevin equation) and therefore commute, we get

$$(p)_n = \sum_{\substack{i_1,\dots,i_p=0 \\ i_1+\dots+i_p \le n-p}} C^{n-p}\,(F-C)^p = C^{n-p}\,(F-C)^p \frac{n!}{p!(n-p)!} = C^n \left(\frac{F}{C} - 1\right)^p \binom{n}{p}.$$

Using the definition (6.4) of y, we obtain the expression of $(p)_n$ claimed in the lemma statement and thus (6.5).

In the particular case when $k = n$, we recover the fine propagator, since

$$x_0 \sum_{p=0}^{n} (p)_n = x_0\, C^n \sum_{p=0}^{n} \left(\frac{F}{C} - 1\right)^p 1^{n-p} \binom{n}{p} = C^n \left(\frac{F}{C} - 1 + 1\right)^n x_0 = F^n\, x_0.$$

This concludes the proof of Lemma 6.2. □

Using Lemma 6.2, the relative error $R_{n,k}$ can be written as

$$R_{n,k} = \frac{1}{F^n\, x_0}\Big(\sum_{p=0}^{n} (p)_n\, x_0 - \sum_{p=0}^{k} (p)_n\, x_0\Big) = \frac{1}{F^n} \sum_{p=k+1}^{n} C^n\, y^p \binom{n}{p}$$
$$= \frac{n!}{(1+y)^n} \sum_{p=k+1}^{n} \frac{y^p}{\Gamma(p+1)\Gamma(n-p+1)}, \quad (6.6)$$

where Γ is the gamma function satisfying $\Gamma(n+1) = n!$ for any integer n. Since we are interested in the analysis of $R_{n,k}$ as a function of k for fixed n, the term inside the summation in the right-hand side of (6.6) is the key term that dictates the behaviour of $R_{n,k}$. The following lemma (the proof of which is postponed until Appendix B) provides a complete characterisation of this term.

Lemma 6.3. *Consider some integer $n > 1$ and some real number $y \in (-1, 0) \cup (0, \infty)$. For $p \in \{1, \dots, n\}$, define*

$$\eta(p) = \frac{|y|^p}{\Gamma(p+1)\Gamma(n-p+1)}. \quad (6.7)$$

We introduce the digamma function defined by $\psi = \dfrac{\Gamma'}{\Gamma}$. Then y satisfies one (and only one) of the following three cases (hereafter, log *denotes the natural logarithm function):*

1. *If y satisfies*
$$\log|y| < \psi(2) - \psi(n), \tag{6.8}$$
then $\eta(p)$ is a strictly decreasing sequence in $p \in \{1, \dots, n\}$.

2. *If y satisfies*
$$\log|y| > \psi(n+1) - \psi(1), \tag{6.9}$$
then $\eta(p)$ is a strictly increasing sequence in $p \in \{1, \dots, n\}$.

3. *If there exists a real number $\overline{p} \in [1, n]$ such that*
$$\log|y| = \psi(\overline{p}+1) - \psi(n-\overline{p}+1), \tag{6.10}$$
then
$$\eta(p) \begin{cases} \text{is strictly increasing for } p \in \{1, \dots, \lfloor \overline{p} \rfloor\}, \\ \text{is strictly decreasing for } p \in \{\lceil \overline{p} \rceil, \dots, n\}, \end{cases} \tag{6.11}$$
where $\lfloor \cdot \rfloor$ (resp. $\lceil \cdot \rceil$) is the floor (resp. ceiling) function. In this case, $\overline{p}$ satisfies the following bound:
$$\min\left(\left\lceil \frac{n|y|-2}{1+|y|} \right\rceil, \left\lceil \frac{(n-1)|y|-1}{1+|y|} \right\rceil\right) \leq \overline{p} \leq \max\left(\left\lfloor \frac{(n+1)|y|+1}{1+|y|} \right\rfloor, \left\lfloor \frac{(n+2)|y|}{1+|y|} \right\rfloor\right). \tag{6.12}$$

As pointed out in the proof below, the function ψ is increasing on $(0, \infty)$. Assuming that $n > 2$, the condition (6.8) (resp. (6.9)) implies that $y \in (-1, 0) \cup (0, 1)$ (resp. $y > 1$). Remark 6.4 below discusses in more details the two conditions (6.8) and (6.9), and shows that the values of y such that either (6.8) or (6.9) is satisfied become somehow "less and less likely" when n increases. In the case (6.10) (which is hence the most "frequent" case when n is large), we see that (6.12) means that $\overline{p}$ essentially scales (when $n \gg 1$) as
$$\overline{p} \approx \frac{n|y|}{1+|y|}. \tag{6.13}$$

Remark 6.4. We discuss here the conditions (6.8) and (6.9), in terms of y (and hence of the parameter ω of the coarse model) and n. In view of (B.2) below, we note that $\psi(n)$ is equivalent to $\log n$ when $n \to \infty$. More precisely, there exists some $c > 0$ such that
$$\forall n \in \mathbb{N}^\star, \quad -c + \log n \leq \psi(n) \leq c + \log n.$$
Condition (6.8) hence yields not only that $|y| < 1$, but also that $|y| \leq c/n$ for some universal $c > 0$. The parameter y is thus very close to 0, and hence the coarse and the fine models are close one to each other. Using $y = \exp\big((\omega-1)\Delta t\big) - 1$, it follows that (6.8) is equivalent to
$$\omega \in \left(1 + \frac{1}{\Delta t}\log\left[1 - \exp\big(\psi(2) - \psi(n)\big)\right], 1\right) \cup \left(1, 1 + \frac{1}{\Delta t}\log\left[1 + \exp\big(\psi(2) - \psi(n)\big)\right]\right).$$
For $n = 1000$ and $\Delta t = 0.05$, which are typical values used in this article, this rewrites (after rounding) as $\omega \in (0.97, 1) \cup (1, 1.03)$, which thus corresponds to ω being very close to the reference value $\omega_{\text{ref}} = 1$. If n increases, this domain shrinks even further.

Similarly, Condition (6.9) not only yields that $y > 1$, but also that $y \geq c\,n$ for some universal $c > 0$. The parameter y is thus very large, and the coarse and the fine models are very far apart each other. Using again $y = \exp\big((\omega-1)\Delta t\big) - 1$, it follows that (6.9) for $y \in (1, \infty)$ holds if and only if
$$\omega > 1 + \frac{1}{\Delta t}\log\left[1 + \exp\big(\psi(n+1) - \psi(1)\big)\right].$$
As before, for $n = 1000$ and $\Delta t = 0.05$, this condition rewrites as $\omega > 150.72$, namely ω taking very large values compared to the reference value $\omega_{\text{ref}} = 1$. If n increases, this threshold value increases further.

For the sake of completeness, and to provide some rough orders of magnitude, we note that the value $y = 1$ corresponds to $F = 2C$, and thus $\omega = 1 + \dfrac{\log 2}{\Delta t}$. For $\Delta t = 0.05$, this corresponds to $\omega = 14.86$. □

Using Lemma 6.3, the following result, which is the main result of this section, describes the behaviour of the relative error $R_{n,k}$ as a function of k for fixed n. Its proof is postponed until Appendix C.

Proposition 6.5. *Fix $n \in \mathbb{N}$ with $n > 1$. If $y \in (0, \infty)$, then the sequence $\{0, \dots, n-1\} \ni k \mapsto R_{n,k}$ is strictly decreasing and we have the bound*

$$0 < R_{n,k} \leq 1 - \frac{1}{(1+y)^n}. \tag{6.14}$$

Furthermore, we have three (and only three) distinct possibilities:

1. *If y satisfies (6.8), then $k \mapsto R_{n,k}$ is convex for $k \in \{1, \dots, n-1\}$.*
2. *If y satisfies (6.9), then $k \mapsto R_{n,k}$ is concave for $k \in \{1, \dots, n-1\}$.*
3. *If there exists $\overline{p} \in [1, n]$ such that (6.10) is satisfied, then $k \mapsto R_{n,k}$ is concave for $k \in \{1, \dots, \lfloor \overline{p} \rfloor\}$ and convex for $k \in \{\lceil \overline{p} \rceil, \dots, n-1\}$, where we recall that $\overline{p}$ satisfies the bounds (6.12).*

If $y \in (-1, 0)$, then the series $R_{n,k} = \dfrac{n!}{(1+y)^n} \displaystyle\sum_{p=k+1}^{n} (-1)^p \, \eta(p)$ is an alternating series and we have two (and only two) distinct possibilities:

1. *If y satisfies (6.8), then $R_{n,2\ell} \leq 0$ and $R_{n,1+2\ell} \geq 0$ for any n and ℓ. The sequences $\ell \mapsto |R_{n,2\ell}|$ and $\ell \mapsto |R_{n,1+2\ell}|$ are both decreasing and we have the bound*

$$|R_{n,k}| \leq \max(|R_{n,0}|, |R_{n,1}|) = \max\left(\frac{1}{(1+y)^n} - 1, 1 - \frac{1}{(1+y)^n} - \frac{ny}{(1+y)^n}\right) \tag{6.15}$$

   for any $k \in \{0, \dots, n\}$. Furthermore, the oscillations of $R_{n,k}$, namely the sequence $k \mapsto |R_{n,k+1} - R_{n,k}|$, are decreasing when k increases.

2. *If there exists $\overline{p} \in [1, n]$ such that (6.10) is satisfied, then the largest oscillation of $R_{n,k}$ when k varies (namely $\max\limits_{k \in \{1, \dots, n\}} |R_{n,k} - R_{n,k-1}|$) is either $\left|R_{n,\lfloor \overline{p} \rfloor} - R_{n,\lfloor \overline{p} \rfloor - 1}\right|$ or $\left|R_{n,\lceil \overline{p} \rceil} - R_{n,\lceil \overline{p} \rceil - 1}\right|$, where we recall that $\overline{p}$ satisfies (6.12). In addition, we have the bound*

$$\forall k \in \{0, \dots, n\}, \qquad |R_{n,k}| \leq \max\left(\left|R_{n,\lfloor \overline{p} \rfloor - 1}\right|, \left|R_{n,\lfloor \overline{p} \rfloor}\right|, \left|R_{n,\lfloor \overline{p} \rfloor + 1}\right|\right), \tag{6.16}$$

   and the lower bound

$$\max_{k \in \{0, \dots, n\}} |R_{n,k}| \geq \frac{1}{2} \min_{q \in \{\lfloor \overline{p} \rfloor, \lceil \overline{p} \rceil\}} \frac{n!}{q!(n-q)!} \frac{|y|^q}{(1+y)^n}. \tag{6.17}$$

Some estimate of the right-hand side of (6.17), in terms of y and n, is given in (6.18)–(6.19) below.

When $y > 0$, the estimate (6.14) shows that, when n is fixed (i.e. the time-horizon is fixed) and y gets closer to 0 (i.e. the coarse and the fine models get closer one to each other), then the bound on the relative error becomes smaller (and converges to 0 when $y \to 0$). Conversely, for a given $y > 0$, the estimate (6.14) shows that, when n increases, the relative error remains bounded but may be not small. This is confirmed by numerical experiments, as illustrated below (see Figures 12a and 12b and the corresponding discussion).

When $y < 0$ and for a fixed value of n, Condition (6.8) is satisfied for sufficiently small $|y|$. The estimate (6.15) again shows that the bound on the relative error converges to 0 when $y \to 0$.

When we consider a fixed value of $y < 0$ and when n increases, then Condition (6.8) cannot be satisfied for sufficiently large n. Condition (6.10) is thus satisfied. The error (in absolute value) is maximal for k equal to $\lfloor \overline{p} \rfloor$ or $\lfloor \overline{p} \rfloor \pm 1$, as shown by (6.16). This is confirmed by numerical experiments (see Figure 12c

below). The estimate (6.17) provides a lower bound on this error. Recasting (6.13) in the form $\overline{p} \approx \alpha\, n$ for $\alpha = |y|/(1+|y|)$ and using Stirling formula, we see that the right-hand side of (6.17) is of the order of

$$\begin{aligned}
\frac{1}{2} \min_{q \in \{\lfloor \overline{p} \rfloor, \lceil \overline{p} \rceil\}} \frac{n!}{q!(n-q)!} \frac{|y|^q}{(1+y)^n} &\approx \frac{1}{2} \frac{\sqrt{2\pi n}\ (n/e)^n}{\sqrt{2\pi \overline{p}}\ (\overline{p}/e)^{\overline{p}}\ \sqrt{2\pi(n-\overline{p})}\ ((n-\overline{p})/e)^{n-\overline{p}}} \frac{|y|^{\overline{p}}}{(1+y)^n} \\
&\approx \frac{1}{2} \frac{n^n}{\sqrt{2\pi\alpha(1-\alpha)\, n}\ \ (\alpha\, n)^{\alpha\, n}\ \ ((1-\alpha)\, n)^{(1-\alpha)\, n}} \frac{|y|^{\alpha\, n}}{(1+y)^n} \\
&= \frac{1}{2} \frac{1}{\sqrt{2\pi\alpha(1-\alpha)\, n}\ \ \alpha^{\alpha\, n}\ \ (1-\alpha)^{(1-\alpha)\, n}} \frac{|y|^{\alpha\, n}}{(1+y)^n} \\
&= \frac{C_\alpha}{\sqrt{n}} (D_{\alpha,y})^n, \qquad (6.18)
\end{aligned}$$

for some constant C_α (only depending on α, which is itself a function of $|y|$) and with

$$D_{\alpha,y} = \frac{1}{\alpha^\alpha\ (1-\alpha)^{(1-\alpha)}} \frac{|y|^\alpha}{1+y} = \frac{1}{\alpha^\alpha\ (1-\alpha)^{(1-\alpha)}} \frac{\alpha^\alpha}{(1-\alpha)^\alpha} \frac{1}{1-\alpha/(1-\alpha)} = \frac{1}{1-2\alpha}, \qquad (6.19)$$

where we have used that $|y| = \alpha/(1-\alpha)$ to obtain the second equality. Recalling that $y \in (-1,0)$, we have $\alpha \in (0,1/2)$, and thus $D_{\alpha,y} > 1$. We thus infer from (6.18) that the maximum value of the error goes to ∞ when $n \to \infty$, exponentially fast with respect to n. The equation (6.19) shows that the rate is increasing when $|y| \in (0,1)$ increases. All these facts are confirmed by numerical experiments (see Figure 12d below).

Figure 12 illustrates the findings of Proposition 6.5. Except stated otherwise, we work with the parameters $n = 1000$ and $\Delta t = 0.05$. On the top left (Figure 12a), we plot $k \mapsto R_{n,k}$ as a function of k for various ω. All the chosen values of ω are larger than 1, hence $y > 0$ in all these cases. The value $\omega = 1.02$ falls in the case when (6.8) holds (recall the numerical values from Remark 6.4), and hence $R_{n,k}$ is convex with respect to k (and in practice very close to 0 for most values of k). The value $\omega = 160$ falls in the case when (6.9) holds (recall again Remark 6.4), and hence $k \mapsto R_{n,k}$ is concave (and in practice very close to 1 for most values of k). The remaining choices of ω correspond to intermediate values of y, falling in the case when (6.10) holds, and where $R_{n,k}$ is concave up to some $\overline{p}$ and then convex. We have shown with (6.13) that $\overline{p}$ essentially scales as $n|y|/(1+|y|)$. For instance, when $\omega = 14.86$, we have $y = 1$ and we thus expect $\overline{p}$ to be of the order of $n/2$. This is indeed what we observe, with an inflection point of the function $k \mapsto R_{n,k}$ close to $k = 500$. When ω becomes larger (resp. smaller) than 14.86, y increases (resp. decreases) and thus $n|y|/(1+|y|)$ increases (resp. decreases) as well. We thus expect the inflection point to move to larger (resp. smaller) values of k, which is indeed what we observe in Figure 12a.

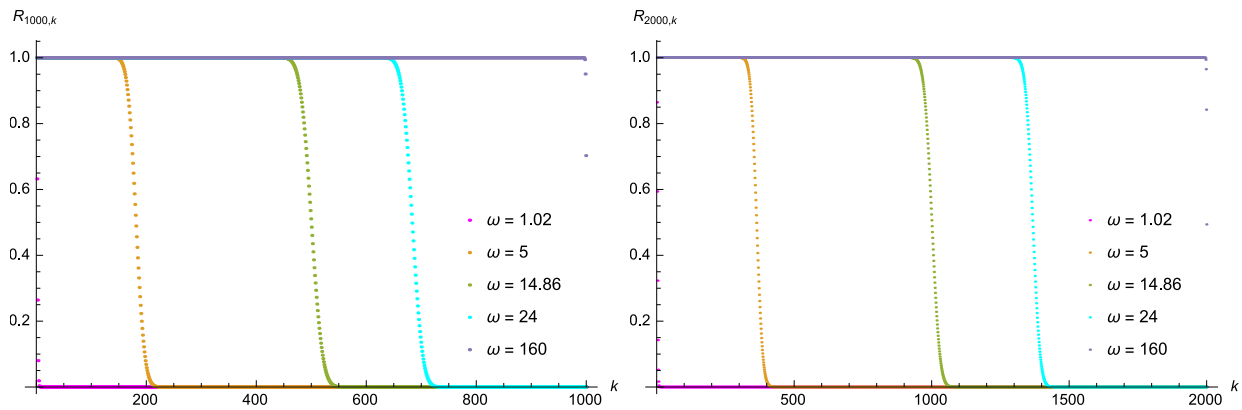
On the top right (Figure 12b), we again plot $k \mapsto R_{n,k}$ as a function of k, for the same ω as in Figure 12a, but for a value of $n = 2000$ twice larger. Comparing Figures 12a and 12b, we check that the inflection point lies at values twice larger when n is twice as large. We have also performed computations with $n = 500$ and $n = 1500$ (results not shown), which are also in perfect agreement with (6.13).

On the bottom left (Figure 12c), we plot $k \mapsto R_{n,k}$ for $\omega = 0.1$ and $n = 1000$. We are in the case when $y \in (-1,0)$ and Condition (6.10) holds for some $\overline{p}$. We compute that $y \approx -0.044$ and thus, according to (6.12), we have $41 \leq \overline{p} \leq 43$. We have stated in Proposition 6.5 that the maximum oscillation of $R_{n,k}$ occurs for $k \approx \overline{p}$, and that $|R_{n,k}|$ is maximal also for $k \approx \overline{p}$ (see (6.16)), which is indeed what we observe in Figure 12c.

On the bottom right (Figure 12d), we consider several values of ω all smaller than 1 (hence corresponding to $y \in (-1,0)$) and we consider values of n sufficiently large such that Condition (6.10) holds for all the values of ω that we consider (this occurs for any $n \geq 700$ for $\omega = 0.9$, and for any $n \geq 75$ for $\omega = 0.1$). We then plot $\left| R_{n,\lfloor \overline{p} \rfloor} \right|$ as a function of n, where $\overline{p}$ is approximated by (6.13). The figure shows that the maximum error dramatically increases as a function of n: as expected from (6.18), it grows exponentially with respect to n, with a rate which, as expected from (6.19), decreases when ω gets closer to $\omega_{\text{ref}} = 1$.

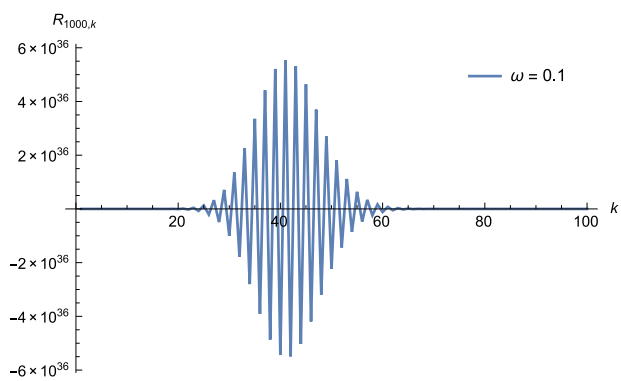
Overall, the best possible case is when the coarse and fine models and the time window are such that Condition (6.8) holds. If $y > 0$, this corresponds to a convex positive decreasing function $R_{n,k}$ (and thus

an error which converges very fast to 0). If $y < 0$, the function $R_{n,k}$ has decreasing oscillations and we essentially have that $|R_{n,k}|$ is decreasing. In both cases, the error decreases when the parareal iterations proceed. As pointed out in Remark 6.4, Condition (6.8) only holds for y close to 0, i.e. ω close to $\omega_{\text{ref}} = 1$ (and thus coarse and fine models close to each other), or for small enough values of n. In the intermediate case when (6.10) holds, and if $y > 0$, there is an inflection point in the behaviour of $R_{n,k}$ at $k \approx \overline{p}$. The error $R_{n,k}$ is always bounded, but it may take many parareal iterations to converge to zero (this is also true in the case (6.9)). If $y < 0$, then neither $R_{n,k}$ nor $|R_{n,k}|$ is decreasing, and the error can reach extremely large values.

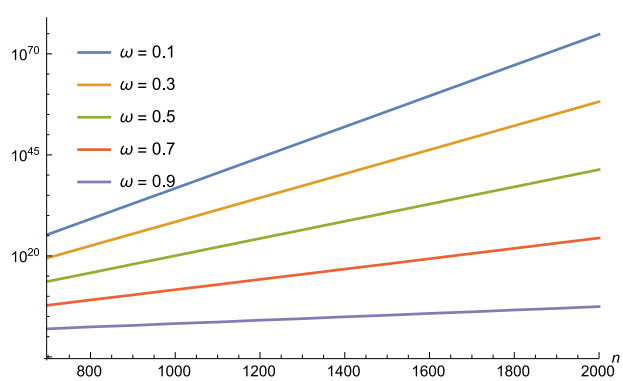


(a) Case $y > 0$: plot of $k \mapsto R_{n,k}$ for $n = 1000$

(b) Case $y > 0$: plot of $k \mapsto R_{n,k}$ for $n = 2000$



(c) Case $y < 0$ and (6.10): plot of $k \mapsto R_{n,k}$ for $n = 1000$



(d) Case $y < 0$: plot of $n \mapsto |R_{n,\lfloor\overline{p}\rfloor}|$

Figure 12: We work with $\Delta t = 0.05$. Top left: Plot of $k \mapsto R_{n,k}$ as a function of k for various ω, for $n = 1000$ (all these ω are larger than 1, hence $y > 0$ in all cases; the three possible cases (6.8), (6.9) and (6.10) are represented). Top right: Plot of $k \mapsto R_{n,k}$ as a function of k for various ω, for $n = 2000$. Bottom left: Plot of $k \mapsto R_{n,k}$ for $\omega = 0.1$ and $n = 1000$ ($y < 0$ and (6.10) holds). Bottom right: Plot of $|R_{n,\lfloor\overline{p}\rfloor}|$ as a function of n for various ω (in these cases, $y < 0$).

A Explicit formulae for LJ7-2d

The fine potential V_f for LJ7-2d is the sum of Lennard-Jones contributions:

$$V_f(q) = \frac{1}{2} \sum_{\substack{i,j \in \{1,\dots,7\} \\ i \neq j}} \phi_f(|q^i - q^j|), \qquad \phi_f(r) = r^{-12} - 2\, r^{-6},$$

where $|\cdot|$ is the d-dimensional Euclidean norm (here, $d = 2$). The corresponding force on the system is $F_f = -\nabla V_f$, and the force on the i-th particle (here onwards we do not explicitly state that $i, j \in \{1, \dots, 7\}$) is

$$F_f^i(q) = -\frac{\partial V_f}{\partial q^i}(q) = \sum_{j \neq i} \frac{12}{r_{ij}^2} \left(r_{ij}^{-12} - r_{ij}^{-6}\right) \begin{pmatrix} q_x^i - q_x^j \\ q_y^i - q_y^j \end{pmatrix}, \qquad r_{ij} = |q^i - q^j|.$$

The coarse potential is a harmonic approximation of V_f around the well where the system initially lies. To calculate this coarse force, we need the Hessian of the fine potential V_f, which is given (for any $\alpha, \beta \in \{x, y\}$ and any $i, j \in \{1, \dots, 7\}$) by

$$\frac{\partial^2 V_f}{\partial q_\beta^\ell \partial q_\alpha^i}(q) = \begin{cases} \theta'(r_{i\ell}) \, \dfrac{q_\beta^\ell - q_\beta^i}{r_{i\ell}} \, (q_\alpha^i - q_\alpha^\ell) - \theta(r_{i\ell}) \, \delta_{\alpha\beta} & \text{if } \ell \neq i, \\ \displaystyle\sum_{j \neq i} \theta'(r_{ij}) \, \frac{q_\beta^i - q_\beta^j}{r_{ij}} \, (q_\alpha^i - q_\alpha^j) + \sum_{j \neq i} \theta(r_{ij}) \, \delta_{\alpha\beta} & \text{if } \ell = i, \end{cases}$$

where $\delta_{\alpha\beta} = 1$ if $\alpha = \beta$ and zero otherwise, and

$$\theta(r) = \frac{\phi_f'(r)}{r} = -\frac{12}{r^{14}} + \frac{12}{r^8} \qquad \text{and} \qquad \frac{\theta'(r)}{r} = \frac{12 \times 14}{r^{16}} - \frac{12 \times 8}{r^{10}}.$$

Using $\overline{q}$ for the position of the well corresponding to the initial condition of the system, the (harmonic) coarse potential is given by

$$V_c(q) = \frac{1}{2} \sum_{i,j} \sum_{\alpha, \beta \in \{x,y\}} (q_\alpha^i - \overline{q}_\alpha^i) \, (q_\beta^j - \overline{q}_\beta^j) \, \frac{\partial^2 V_f}{\partial q_\beta^j \partial q_\alpha^i}(\overline{q}),$$

and the corresponding force on the j-th particle in the β-th coordinate is

$$F_c^{j,\beta}(q) = -\frac{\partial V_c}{\partial q_\beta^j}(q) = -\sum_i \sum_{\alpha \in \{x,y\}} (q_\alpha^i - \overline{q}_\alpha^i) \, \frac{\partial^2 V_f}{\partial q_\alpha^i \partial q_\beta^j}(\overline{q}).$$

B Proof of Lemma 6.3

Differentiating the function $[1, n] \ni p \mapsto \eta(p)$ and using the digamma function $\psi(p) = \Gamma'(p)/\Gamma(p)$, we find

$$\eta'(p) = \frac{|y|^p \big(\log|y| - \psi(p+1) + \psi(n-p+1) \big)}{\Gamma(p+1)\Gamma(n-p+1)}. \tag{B.1}$$

Using that $\psi' > 0$ (a fact which directly stems from the series expansion of the polygamma function of order one $\psi_1 = \psi'$), we observe that the mapping

$$[1, n] \ni p \mapsto f(p) = \psi(p+1) - \psi(n-p+1)$$

is strictly increasing, since $f'(p) = \psi'(p+1) + \psi'(n-p+1) > 0$. We next observe that $\eta'(p) = \dfrac{|y|^p \big(\log|y| - f(p) \big)}{\Gamma(p+1)\Gamma(n-p+1)}$. We can thus distinguish three (mutually exclusive) cases:

- either $\log|y| < \inf_{p \in [1,n]} f(p)$,
- or $\log|y| > \sup_{p \in [1,n]} f(p)$,
- or $\inf_{p \in [1,n]} f(p) \leq \log|y| \leq \sup_{p \in [1,n]} f(p)$.

We successively study these cases.

Since f is increasing, the first case corresponds to $\log|y| < f(1) = \psi(2) - \psi(n)$, and we recover (6.8). In that case, we have $\eta'(p) < 0$ for any $p \in [1, n]$, and thus η is strictly decreasing on $[1, n]$.

Since f is increasing, the second case corresponds to $\log|y| > f(n) = \psi(n+1) - \psi(1)$, and we recover (6.9). In that case, we have $\eta'(p) > 0$ for any $p \in [1, n]$, and thus η is strictly increasing on $[1, n]$.

Again using the fact that f is increasing, the third case corresponds to the fact that there exists some $\overline{p} \in [1, n]$ such that $\log|y| = f(\overline{p}) = \psi(\overline{p}+1) - \psi(n - \overline{p} + 1)$ and we recover (6.10). Using (B.1), we next obtain (6.11).

We now establish explicit bounds on $\overline{p}$. Using the fact that

$$\psi(p+1) = -\gamma_{\mathrm{EM}} + \sum_{i=1}^{p} \frac{1}{i} \quad \text{for any positive integer } p, \tag{B.2}$$

where γ_{EM} is the Euler-Mascheroni constant, we have

$$f(\lfloor \overline{p} \rfloor) = \begin{cases} -\displaystyle\sum_{i=\lfloor \overline{p} \rfloor + 1}^{n - \lfloor \overline{p} \rfloor} \frac{1}{i} & \text{if } 2\lfloor \overline{p} \rfloor < n \\ 0 & \text{if } 2\lfloor \overline{p} \rfloor = n \\ \displaystyle\sum_{i=n-\lfloor \overline{p} \rfloor + 1}^{\lfloor \overline{p} \rfloor} \frac{1}{i} & \text{if } 2\lfloor \overline{p} \rfloor > n \end{cases} \quad \text{and likewise} \quad f(\lceil \overline{p} \rceil) = \begin{cases} -\displaystyle\sum_{i=\lceil \overline{p} \rceil + 1}^{n - \lceil \overline{p} \rceil} \frac{1}{i} & \text{if } 2\lceil \overline{p} \rceil < n \\ 0 & \text{if } 2\lceil \overline{p} \rceil = n \\ \displaystyle\sum_{i=n-\lceil \overline{p} \rceil + 1}^{\lceil \overline{p} \rceil} \frac{1}{i} & \text{if } 2\lceil \overline{p} \rceil > n \end{cases}. \tag{B.3}$$

We now write some easy and useful estimates. For any integer k such that $k + 1 < n - k + 1$, we have the following upper and lower bounds:

$$\begin{aligned} \log\left(\frac{n-k+1}{k+1}\right) &= \int_{k+1}^{n-k+1} \frac{dx}{x} = \sum_{i=k+1}^{n-k} \int_{i}^{i+1} \frac{dx}{x} \le \sum_{i=k+1}^{n-k} \frac{1}{i}, \\ \sum_{i=k+1}^{n-k} \frac{1}{i} &= \sum_{j=k}^{n-k-1} \frac{1}{j+1} \le \sum_{j=k}^{n-k-1} \int_{j}^{j+1} \frac{dx}{x} = \int_{k}^{n-k} \frac{dx}{x} = \log\left(\frac{n-k}{k}\right). \end{aligned} \tag{B.4}$$

Proceeding similarly, we find, for any integer k such that $k + 1 > n - k + 1$, that

$$\begin{aligned} \log\left(\frac{k+1}{n-k+1}\right) &= \int_{n-k+1}^{k+1} \frac{dx}{x} = \sum_{i=n-k+1}^{k} \int_{i}^{i+1} \frac{dx}{x} \le \sum_{i=n-k+1}^{k} \frac{1}{i}, \\ \sum_{i=n-k+1}^{k} \frac{1}{i} &= \sum_{j=n-k}^{k-1} \frac{1}{j+1} \le \sum_{j=n-k}^{k-1} \int_{j}^{j+1} \frac{dx}{x} = \int_{n-k}^{k} \frac{dx}{x} = \log\left(\frac{k}{n-k}\right). \end{aligned} \tag{B.5}$$

Using (B.3) and (B.4), we see that, if $2\lfloor \overline{p} \rfloor < n$, then $f(\lfloor \overline{p} \rfloor) < 0$ and it satisfies

$$\log\left(\frac{\lfloor \overline{p} \rfloor}{n - \lfloor \overline{p} \rfloor}\right) \le f(\lfloor \overline{p} \rfloor) \le \log\left(\frac{\lfloor \overline{p} \rfloor + 1}{n - \lfloor \overline{p} \rfloor + 1}\right). \tag{B.6}$$

Similarly, if $2\lceil \overline{p} \rceil < n$, then $f(\lceil \overline{p} \rceil) < 0$ and it satisfies

$$\log\left(\frac{\lceil \overline{p} \rceil}{n - \lceil \overline{p} \rceil}\right) \le f(\lceil \overline{p} \rceil) \le \log\left(\frac{\lceil \overline{p} \rceil + 1}{n - \lceil \overline{p} \rceil + 1}\right). \tag{B.7}$$

Alternatively, if $2\lfloor \overline{p} \rfloor > n$, we infer from (B.3) and (B.5) that $f(\lfloor \overline{p} \rfloor) > 0$ and that it satisfies

$$\log\left(\frac{\lfloor \overline{p} \rfloor + 1}{n - \lfloor \overline{p} \rfloor + 1}\right) \le f(\lfloor \overline{p} \rfloor) \le \log\left(\frac{\lfloor \overline{p} \rfloor}{n - \lfloor \overline{p} \rfloor}\right), \tag{B.8}$$

and likewise, if $2\lceil\overline{p}\rceil > n$, then $f(\lceil\overline{p}\rceil) > 0$ and it satisfies

$$\log\left(\frac{\lceil\overline{p}\rceil+1}{n-\lceil\overline{p}\rceil+1}\right) \leq f(\lceil\overline{p}\rceil) \leq \log\left(\frac{\lceil\overline{p}\rceil}{n-\lceil\overline{p}\rceil}\right). \tag{B.9}$$

Since f is increasing, it follows from (6.10), that we write in the form $\log|y| = f(\overline{p})$, that

$$f(\lfloor\overline{p}\rfloor) \leq \log|y| \leq f(\lceil\overline{p}\rceil). \tag{B.10}$$

We now distinguish three (mutually exclusive) cases:

(I) either $2\lfloor\overline{p}\rfloor \leq 2\lceil\overline{p}\rceil < n$,

(II) or $n < 2\lfloor\overline{p}\rfloor \leq 2\lceil\overline{p}\rceil$,

(III) or $2\lfloor\overline{p}\rfloor \leq n \leq 2\lceil\overline{p}\rceil$,

that we successively study.

In the case (I), we have $2\lceil\overline{p}\rceil < n$, hence $\log|y| \leq f(\lceil\overline{p}\rceil) < 0$ and therefore $|y| < 1$. Inserting (B.6) and (B.7) into (B.10), we find

$$\log\left(\frac{\lfloor\overline{p}\rfloor}{n-\lfloor\overline{p}\rfloor}\right) \leq \log|y| \leq \log\left(\frac{\lceil\overline{p}\rceil+1}{n-\lceil\overline{p}\rceil+1}\right),$$

and thus

$$\lfloor\overline{p}\rfloor \leq \frac{n|y|}{1+|y|}, \qquad \frac{(n+1)|y|-1}{1+|y|} \leq \lceil\overline{p}\rceil.$$

Using $\lfloor\overline{p}\rfloor \geq \lceil\overline{p}\rceil - 1$, we infer that

$$\frac{n|y|-2}{1+|y|} \leq \lfloor\overline{p}\rfloor \leq \frac{n|y|}{1+|y|}, \qquad \frac{(n+1)|y|-1}{1+|y|} \leq \lceil\overline{p}\rceil \leq \frac{(n+1)|y|+1}{1+|y|},$$

and hence, since $\lfloor\overline{p}\rfloor$ and $\lceil\overline{p}\rceil$ are integer numbers, we get

$$\left\lceil\frac{n|y|-2}{1+|y|}\right\rceil \leq \lfloor\overline{p}\rfloor \leq \left\lfloor\frac{n|y|}{1+|y|}\right\rfloor, \qquad \left\lceil\frac{(n+1)|y|-1}{1+|y|}\right\rceil \leq \lceil\overline{p}\rceil \leq \left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor.$$

Using that $\lfloor\overline{p}\rfloor \leq \overline{p} \leq \lceil\overline{p}\rceil$, we eventually get the bound

$$\left\lceil\frac{n|y|-2}{1+|y|}\right\rceil \leq \overline{p} \leq \left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor,$$

and hence (6.12).

We next turn to the case (II), when $n < 2\lfloor\overline{p}\rfloor$. We then have $\log|y| \geq f(\lfloor\overline{p}\rfloor) > 0$, thus $|y| > 1$ and hence $y > 1$ (since $y \in (-1,0)\cup(0,\infty)$). Inserting (B.8) and (B.9) into (B.10) and proceeding as above, we obtain the bounds

$$\frac{(n-1)|y|-1}{1+|y|} \leq \lfloor\overline{p}\rfloor \leq \frac{(n+1)|y|-1}{1+|y|}, \qquad \frac{n|y|}{1+|y|} \leq \lceil\overline{p}\rceil \leq \frac{(n+2)|y|}{1+|y|},$$

and therefore

$$\left\lceil\frac{(n-1)|y|-1}{1+|y|}\right\rceil \leq \overline{p} \leq \left\lfloor\frac{(n+2)|y|}{1+|y|}\right\rfloor,$$

which again implies (6.12).

We eventually turn to the case (III), when $2\lfloor\overline{p}\rfloor \leq n \leq 2\lceil\overline{p}\rceil$. If $\overline{p}$ is an integer, then $2\overline{p} = n$. In addition, in view of (B.3), we have $f(\overline{p}) = 0$, and thus (B.10) implies that $|y| = 1$, and hence $y = 1$ (since $y \in (-1,0)\cup(0,\infty)$). The bound (6.12) again holds.

If $\overline{p}$ is not an integer, then either (i) $n = 2\lfloor\overline{p}\rfloor$ or (ii) $n = 1 + 2\lfloor\overline{p}\rfloor$ or (iii) $n = 2 + 2\lfloor\overline{p}\rfloor = 2\lceil\overline{p}\rceil$. In the case (i) (i.e. when $n = 2\lfloor\overline{p}\rfloor$), we have, in view of (B.3), that $f(\lfloor\overline{p}\rfloor) = 0$ and $f(\lceil\overline{p}\rceil) = 1/\lceil\overline{p}\rceil + 1/(\lceil\overline{p}\rceil - 1)$. In view of (B.10), this implies that

$$0 \leq \log|y| \leq \frac{1}{\lceil\overline{p}\rceil} + \frac{1}{\lceil\overline{p}\rceil - 1} \leq \log\left(\frac{\lceil\overline{p}\rceil}{\lceil\overline{p}\rceil - 2}\right) = \log\left(\frac{n+2}{n-2}\right)$$

and hence

$$1 \leq |y| \leq \frac{n+2}{n-2}.$$

The left and right-hand sides of (6.12) then satisfy

$$\min\left(\left\lceil\frac{n|y|-2}{1+|y|}\right\rceil, \left\lceil\frac{(n-1)|y|-1}{1+|y|}\right\rceil\right) = \left\lceil\frac{(n-1)|y|-1}{1+|y|}\right\rceil \leq \left\lceil\frac{n}{2}\right\rceil = \lfloor\overline{p}\rfloor \leq \overline{p}$$

and

$$\max\left(\left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor, \left\lfloor\frac{(n+2)|y|}{1+|y|}\right\rfloor\right) = \left\lfloor\frac{(n+2)|y|}{1+|y|}\right\rfloor \geq \left\lfloor\frac{n+2}{2}\right\rfloor = \lfloor\overline{p}\rfloor + 1 \geq \overline{p},$$

where we have used that $|y| \geq 1$ (for the first equalities) and the fact that $|y| \mapsto \dfrac{(n-1)|y|-1}{1+|y|} = \dfrac{n}{1+1/|y|} - 1$ and $|y| \mapsto \dfrac{(n+2)|y|}{1+|y|}$ are both increasing functions (for the first inequalities). This shows that (6.12) again holds.

In a similar fashion, in the case (iii) (i.e. when $n = 2\lceil\overline{p}\rceil$), we have, in view of (B.3), that $f(\lceil\overline{p}\rceil) = 0$ and $f(\lfloor\overline{p}\rfloor) = -1/(\lfloor\overline{p}\rfloor + 2) - 1/(\lfloor\overline{p}\rfloor + 1)$. In view of (B.10), this implies that

$$0 \geq \log|y| \geq -\frac{1}{\lfloor\overline{p}\rfloor + 2} - \frac{1}{\lfloor\overline{p}\rfloor + 1} \geq -\log\left(\frac{\lfloor\overline{p}\rfloor + 2}{\lfloor\overline{p}\rfloor}\right) = -\log\left(\frac{n+2}{n-2}\right)$$

and hence

$$1 \geq |y| \geq \frac{n-2}{n+2}.$$

The left and right-hand sides of (6.12) then satisfy

$$\min\left(\left\lceil\frac{n|y|-2}{1+|y|}\right\rceil, \left\lceil\frac{(n-1)|y|-1}{1+|y|}\right\rceil\right) = \left\lceil\frac{n|y|-2}{1+|y|}\right\rceil \leq \left\lceil\frac{n-2}{2}\right\rceil = \lceil\overline{p}\rceil - 1 \leq \overline{p}$$

and

$$\max\left(\left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor, \left\lfloor\frac{(n+2)|y|}{1+|y|}\right\rfloor\right) = \left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor \geq \left\lfloor\frac{n}{2}\right\rfloor = \lceil\overline{p}\rceil \geq \overline{p},$$

where we have used that $|y| \leq 1$ (for the first equalities) and the fact that $|y| \mapsto \dfrac{(n+1)|y|+1}{1+|y|} = \dfrac{n}{1+1/|y|} + 1$ and $|y| \mapsto \dfrac{n|y|-2}{1+|y|}$ are both increasing functions (for the first inequalities). This shows that (6.12) again holds.

We eventually consider the case (ii), when $n = 1 + 2\lfloor\overline{p}\rfloor = 2\lceil\overline{p}\rceil - 1$. In view of (B.3), we have that $f(\lfloor\overline{p}\rfloor) = -1/(\lfloor\overline{p}\rfloor + 1)$ and $f(\lceil\overline{p}\rceil) = 1/\lceil\overline{p}\rceil$. In view of (B.10), this implies that

$$-\log\left(\frac{n+1}{n-1}\right) = -\log\left(\frac{\lfloor\overline{p}\rfloor + 1}{\lfloor\overline{p}\rfloor}\right) \leq -\frac{1}{\lfloor\overline{p}\rfloor + 1} \leq \log|y| \leq \frac{1}{\lceil\overline{p}\rceil} \leq \log\left(\frac{\lceil\overline{p}\rceil}{\lceil\overline{p}\rceil - 1}\right) = \log\left(\frac{n+1}{n-1}\right)$$

and hence

$$\frac{n-1}{n+1} \leq |y| \leq \frac{n+1}{n-1}.$$

Since the functions that map $|y|$ to the left and right-hand sides of (6.12) are both increasing, we obtain

$$\min\left(\left\lceil\frac{n|y|-2}{1+|y|}\right\rceil,\left\lceil\frac{(n-1)|y|-1}{1+|y|}\right\rceil\right)\leq\min\left(\left\lceil\frac{n-1}{2}+\frac{1}{n}\right\rceil,\left\lceil\frac{n-1}{2}\right\rceil\right)=\left\lceil\frac{n-1}{2}\right\rceil=\lfloor\overline{p}\rfloor\leq\overline{p}$$

and

$$\max\left(\left\lfloor\frac{(n+1)|y|+1}{1+|y|}\right\rfloor,\left\lfloor\frac{(n+2)|y|}{1+|y|}\right\rfloor\right)\geq\max\left(\left\lfloor\frac{n+1}{2}\right\rfloor,\left\lfloor\frac{n+1}{2}-\frac{1}{n}\right\rfloor\right)=\left\lfloor\frac{n+1}{2}\right\rfloor=\lfloor\overline{p}\rfloor+1\geq\overline{p},$$

which again implies (6.12). This concludes the proof of Lemma 6.3.

C Proof of Proposition 6.5

The proof falls in two cases.

Case A. We first consider the case when $y\in(0,\infty)$. In view of (6.6), the sequence $k\mapsto R_{n,k}$ is a decreasing sequence and therefore, for any $k\in\{0,\ldots,n-1\}$, we have

$$0<R_{n,k}\leq R_{n,0}=\frac{1}{(1+y)^n}\left(\sum_{p=0}^{n}\frac{n!\;y^p}{p!(n-p)!}-1\right)=\frac{(1+y)^n-1}{(1+y)^n},$$

which yields (6.14). In view of (6.6) and (6.7), we have $R_{n,k}=\dfrac{n!}{(1+y)^n}\displaystyle\sum_{p=k+1}^{n}\eta(p)$. The convexity properties of $R_{n,k}$ directly follow from the monotonicity properties of $p\mapsto\eta(p)$ established in Lemma 6.3.

Case B. We next consider the case when $y\in(-1,0)$. We have pointed out below Lemma 6.3 that Condition (6.9) cannot be satisfied in that case, and we thus only have to consider (6.8) or (6.10). In view of (6.6) and (6.7), we have

$$R_{n,k}=\frac{n!}{(1+y)^n}\sum_{p=k+1}^{n}(-1)^p\,\eta(p),\tag{C.1}$$

which is an alternating series.

Case B1. If $y\in(-1,0)$ satisfies (6.8), we know from Lemma 6.3 that η is a decreasing function. In view of (C.1), we obtain that the oscillations of $R_{n,k}$ are decreasing when k increases.

We claim that $R_{n,0}<0$. Indeed, if n is even, then we write

$$R_{n,0}=\frac{n!}{(1+y)^n}\Big[\big(-\eta(1)+\eta(2)\big)+\ldots+\big(-\eta(n-1)+\eta(n)\big)\Big]$$

and each term $-\eta(j-1)+\eta(j)$ (for even values of j) is negative. If n is odd, then

$$R_{n,0}=\frac{n!}{(1+y)^n}\Big[\big(-\eta(1)+\eta(2)\big)+\ldots+\big(-\eta(n-2)+\eta(n-1)\big)-\eta(n)\Big]$$

and again, each term $-\eta(j-1)+\eta(j)$ (for even values of j) is negative, while $-\eta(n)$ is also negative. We hence have that $R_{n,0}<0$ for any $n\geq1$. With similar arguments, we can show that $R_{n,k}\leq0$ for any even k. Furthermore, the sequence $\ell\mapsto R_{n,2\ell}$ (namely, the relative error after an even number of parareal iterations) is increasing.

We next claim that $R_{n,1}>0$. Indeed, if n is odd, then we write

$$R_{n,1}=\frac{n!}{(1+y)^n}\Big[\big(\eta(2)-\eta(3)\big)+\ldots+\big(\eta(n-1)-\eta(n)\big)\Big]$$

and each term $\eta(j-1)-\eta(j)$ (for odd values of j) is positive. If n is even, then

$$R_{n,1} = \frac{n!}{(1+y)^n}\Big[\big(\eta(2)-\eta(3)\big)+\ldots+\big(\eta(n-2)-\eta(n-1)\big)+\eta(n)\Big]$$

and again, each term $\eta(j-1)-\eta(j)$ (for odd values of j) is positive, while $\eta(n)$ is also positive. We thus have that $R_{n,1}>0$ for any $n\geq 2$. With similar arguments, we can show that $R_{n,k}\geq 0$ for any odd k, and that the sequence $\ell\mapsto R_{n,1+2\ell}$ (namely, the relative error after an odd number of parareal iterations) is decreasing.

Since $\ell\mapsto |R_{n,2\ell}|$ and $\ell\mapsto |R_{n,1+2\ell}|$ are both decreasing, we have

$$\max_{k\in\{0,\ldots,n-1\}}|R_{n,k}| = \max(|R_{n,0}|,|R_{n,1}|),$$

with

$$0<-R_{n,0} = \frac{-1}{(1+y)^n}\sum_{p=1}^{n}\frac{n!\;y^p}{p!(n-p)!} = \frac{-1}{(1+y)^n}\Big((1+y)^n-1\Big) = \frac{1}{(1+y)^n}-1$$

and

$$0<R_{n,1} = \frac{1}{(1+y)^n}\sum_{p=2}^{n}\frac{n!\;y^p}{p!(n-p)!} = \frac{1}{(1+y)^n}\Big((1+y)^n-1-ny\Big) = 1-\frac{1}{(1+y)^n}-\frac{ny}{(1+y)^n},$$

thus the bound (6.15).

Case B2. We eventually consider the case when $y\in(-1,0)$ and (6.10) is satisfied. Using Lemma 6.3, we know that $s\mapsto\eta(s)$ is increasing (resp. decreasing) for $s\leq\overline{p}$ (resp. $s\geq\overline{p}$). In view of (C.1), we get that $|R_{n,k}-R_{n,k-1}| = \dfrac{n!}{(1+y)^n}\,\eta(k)$. The largest oscillation for $R_{n,k}$ is thus given by

$$\begin{aligned}\max_{k\in\{1,\ldots,n\}}|R_{n,k}-R_{n,k-1}| = \frac{n!}{(1+y)^n}\max_{k\in\{1,\ldots,n\}}\eta(k) &= \frac{n!}{(1+y)^n}\max\big(\eta(\lfloor\overline{p}\rfloor),\eta(\lceil\overline{p}\rceil)\big)\\ &= \max\Big(\Big|R_{n,\lfloor\overline{p}\rfloor}-R_{n,\lfloor\overline{p}\rfloor-1}\Big|,\Big|R_{n,\lceil\overline{p}\rceil}-R_{n,\lceil\overline{p}\rceil-1}\Big|\Big).\end{aligned}$$

For any $q\in\{1,\ldots,n\}$, we have

$$\max_{k\in\{0,\ldots,n\}}|R_{n,k}|\geq|R_{n,q}|\geq|R_{n,q}-R_{n,q-1}|-|R_{n,q-1}|\geq|R_{n,q}-R_{n,q-1}|-\max_{k\in\{0,\ldots,n\}}|R_{n,k}|,$$

and therefore $\max_{k\in\{0,\ldots,n\}}|R_{n,k}|\geq\dfrac{1}{2}|R_{n,q}-R_{n,q-1}|$. Since this holds for any $q\in\{1,\ldots,n\}$, we deduce, using our previous result on the largest oscillation, that

$$\max_{k\in\{0,\ldots,n\}}|R_{n,k}|\geq\frac{1}{2}\min_{q\in\{\lfloor\overline{p}\rfloor,\lceil\overline{p}\rceil\}}|R_{n,q}-R_{n,q-1}| = \frac{1}{2}\min_{q\in\{\lfloor\overline{p}\rfloor,\lceil\overline{p}\rceil\}}\frac{n!}{q!(n-q)!}\,\frac{|y|^q}{(1+y)^n},$$

which is (6.17). We are now left with showing (6.16).

Case B2.1. In order to study $|R_{n,k}|$, we first assume that n and $\lfloor\overline{p}\rfloor$ are even. We claim that the sequence $\ell\mapsto R_{n,2\ell}$ is decreasing when $0\leq\ell\leq\lfloor\overline{p}\rfloor/2$ and increasing when $\lfloor\overline{p}\rfloor/2\leq\ell\leq n/2$. Indeed, using (C.1), we have

$$R_{n,2\ell}-R_{n,2+2\ell} = \frac{n!}{(1+y)^n}\big(\eta(2+2\ell)-\eta(1+2\ell)\big),$$

which is positive when $0\leq\ell\leq\lfloor\overline{p}\rfloor/2-1$ and negative when $\ell\geq\lfloor\overline{p}\rfloor/2$ (since then $1+2\ell\geq 1+\lfloor\overline{p}\rfloor>\overline{p}$). In addition, for the final value $\ell=n/2$, we of course have $R_{n,2\ell}=0$.

Likewise, we claim that the sequence $\ell \mapsto R_{n,1+2\ell}$ is increasing when $0 \le \ell \le \lfloor \overline{p} \rfloor/2 - 1$ and decreasing when $\lfloor \overline{p} \rfloor/2 \le \ell \le n/2$. Indeed, using again (C.1), we have

$$R_{n,1+2\ell} - R_{n,3+2\ell} = \frac{n!}{(1+y)^n}\big(\eta(2+2\ell) - \eta(3+2\ell)\big),$$

which is negative when $0 \le \ell \le \lfloor \overline{p} \rfloor/2 - 2$ and positive when $\ell \ge \lfloor \overline{p} \rfloor/2$. We also note that we cannot compare the values of $R_{n,1+2\ell}$ when $\ell = \lfloor \overline{p} \rfloor/2 - 1$ and when $\ell = \lfloor \overline{p} \rfloor/2$, since their difference is

$$R_{n,\lfloor \overline{p} \rfloor - 1} - R_{n,\lfloor \overline{p} \rfloor + 1} = \frac{n!}{(1+y)^n}\big(\eta(\lfloor \overline{p} \rfloor) - \eta(1 + \lfloor \overline{p} \rfloor)\big),$$

the sign of which is unknown. In addition, for the final value $\ell = n/2-1$, we have $R_{n,1+2\ell} = \dfrac{n!\,(-1)^n\,\eta(n)}{(1+y)^n} > 0$ since we have assumed n to be even.

We now identify $\max\limits_{k\in\{0,\dots,n\}} |R_{n,k}|$, supposing first that $R_{n,0} \le 0$. Since $\ell \mapsto R_{n,2\ell}$ is first decreasing and next increasing and since it vanishes for $\ell = n/2$, we have $R_{n,2\ell} \le 0$ for all $0 \le \ell \le n/2$, and therefore $\max\limits_{\ell\in\{0,\dots,n/2\}} |R_{n,2\ell}| = \big|R_{n,\lfloor \overline{p} \rfloor}\big|$. If $R_{n,1} \ge 0$, then using that $\ell \mapsto R_{n,1+2\ell}$ is first increasing and next decreasing and that it is positive for $\ell = n/2 - 1$, we have $R_{n,1+2\ell} \ge 0$ for all $0 \le \ell \le n/2 - 1$. This implies that $\max\limits_{\ell\in\{0,\dots,n/2-1\}} |R_{n,1+2\ell}| = \max\big(\big|R_{n,\lfloor \overline{p} \rfloor - 1}\big|, \big|R_{n,\lfloor \overline{p} \rfloor + 1}\big|\big)$, and we thus obtain (6.16). If $R_{n,1} < 0$, then using again that $\ell \mapsto R_{n,1+2\ell}$ is first increasing and next decreasing and that it is positive for $\ell = n/2-1$, we have that $\max\limits_{\ell\in\{0,\dots,n/2-1\}} |R_{n,1+2\ell}| = \max\big(\big|R_{n,1}\big|, \big|R_{n,\lfloor \overline{p} \rfloor - 1}\big|, \big|R_{n,\lfloor \overline{p} \rfloor + 1}\big|\big)$. We also observe that

$$R_{n,0} - R_{n,1} = -\frac{n!}{(1+y)^n}\eta(1) < 0,$$

and thus $0 > R_{n,1} > R_{n,0}$. This implies that $\big|R_{n,1}\big| < \big|R_{n,0}\big| \le \max\limits_{\ell\in\{0,\dots,n/2\}} |R_{n,2\ell}|$, and we thus again obtain (6.16).

We next identify $\max\limits_{k\in\{0,\dots,n\}} |R_{n,k}|$ under the assumption that $R_{n,0} > 0$. Using the same arguments as above, we have $\max\limits_{\ell\in\{0,\dots,n/2\}} |R_{n,2\ell}| = \max\big(\big|R_{n,0}\big|, \big|R_{n,\lfloor \overline{p} \rfloor}\big|\big)$. Since $R_{n,1} > R_{n,0}$, we have $R_{n,1} > 0$, and thus $\max\limits_{\ell\in\{0,\dots,n/2-1\}} |R_{n,1+2\ell}| = \max\big(\big|R_{n,\lfloor \overline{p} \rfloor - 1}\big|, \big|R_{n,\lfloor \overline{p} \rfloor + 1}\big|\big)$. In addition, we have $0 < R_{n,0} < R_{n,1} \le \max\limits_{\ell\in\{0,\dots,n/2-1\}} |R_{n,1+2\ell}|$. We thus deduce (6.16).

Case B2.2. We now revisit the above arguments studying $|R_{n,k}|$, under the assumption that n is even and that $\lfloor \overline{p} \rfloor$ is odd. We then have that the sequence $\ell \mapsto R_{n,2\ell}$ is decreasing when $0 \le \ell \le (\lfloor \overline{p} \rfloor - 1)/2$ and increasing when $(\lfloor \overline{p} \rfloor + 1)/2 \le \ell \le n/2$. We also note that we cannot compare the values of $R_{n,2\ell}$ when $\ell = (\lfloor \overline{p} \rfloor - 1)/2$ and when $\ell = (\lfloor \overline{p} \rfloor + 1)/2$, and that, for the final value $\ell = n/2$, we have $R_{n,2\ell} = 0$.

Similarly, the sequence $\ell \mapsto R_{n,1+2\ell}$ is increasing when $0 \le \ell \le (\lfloor \overline{p} \rfloor - 1)/2$ and decreasing when $(\lfloor \overline{p} \rfloor - 1)/2 \le \ell \le n/2$. For the final value $\ell = n/2 - 1$, we again have $R_{n,1+2\ell} > 0$.

Using the same arguments as above for identifying $\max\limits_{k\in\{0,\dots,n\}} |R_{n,k}|$ again leads to (6.16).

Case B2.3. We eventually consider the case when n is odd:

- if $\lfloor \overline{p} \rfloor$ is even, then the sequence $\ell \mapsto R_{n,2\ell}$ is decreasing when $0 \le \ell \le \lfloor \overline{p} \rfloor/2$, increasing when $\lfloor \overline{p} \rfloor/2 \le \ell \le (n-1)/2$, and is negative for the final value $\ell = (n-1)/2$. The sequence $\ell \mapsto R_{n,1+2\ell}$ is increasing when $0 \le \ell \le \lfloor \overline{p} \rfloor/2 - 1$, decreasing when $\lfloor \overline{p} \rfloor/2 \le \ell \le (n-1)/2$, and vanishes for the final value $\ell = (n-1)/2$.

- if $\lfloor \overline{p} \rfloor$ is odd, the sequence $\ell \mapsto R_{n,2\ell}$ is decreasing when $0 \leq \ell \leq (\lfloor \overline{p} \rfloor - 1)/2$, increasing when $(\lfloor \overline{p} \rfloor + 1)/2 \leq \ell \leq (n-1)/2$, and negative for the final value $\ell = (n-1)/2$. The sequence $\ell \mapsto R_{n,1+2\ell}$ is increasing when $0 \leq \ell \leq (\lfloor \overline{p} \rfloor - 1)/2$, decreasing when $(\lfloor \overline{p} \rfloor - 1)/2 \leq \ell \leq (n-1)/2$, and vanishes for the final value $\ell = (n-1)/2$.

In both cases, we can show (6.16). This concludes the proof of Proposition 6.5.

Acknowledgements

The work of FL and US was supported in part by the ANR project CINE-PARA (ANR-15-CE23-0019-06). US also acknowledges support from the Alexander von Humboldt foundation and from DFG under Germany's Excellence Strategy-MATH+: The Berlin Mathematics Research Center (EXC-2046/1)-project ID:390685689 (subproject EF4-4). This work was also partially funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 810367; project EMC2) and from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701 (project Time-X). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland. The authors would like to thank Claude Le Bris, Yvon Maday, Danny Perez and Arthur Voter for stimulating interactions, and Olga Gorynina and Gabriel Stoltz for their remarks on a draft version of this article.

References

[Bal] G. Bal. Parallelization in time of (stochastic) ordinary differential equations. Preprint available at `https://www.stat.uchicago.edu/~guillaumebal/PAPERS/paralleltime.pdf`.

[Bal05] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 425–432. Springer Berlin Heidelberg, 2005.

[BBK10] A. Blouza, L. Boudin, and S.-M. Kaber. Parallel in time algorithms with reduction methods for solving chemical kinetics. *Communications in Applied Mathematics and Computational Science*, 5:241–263, 2010.

[BBM+02] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66(5):057701, 2002.

[BM02] G. Bal and Y. Maday. A parareal time discretization for nonlinear PDE's with application to the pricing of an American put. In L. Pavarino and A. Toselli, editors, *Recent developments in domain decomposition methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 189–202. Springer Verlag, 2002.

[DLBLM13] X. Dai, C. Le Bris, F. Legoll, and Y. Maday. Symmetric parareal algorithms for Hamiltonian systems. *Mathematical Modelling and Numerical Analysis*, 47(3):717–742, 2013.

[DM13] X. Dai and Y. Maday. Stable parareal in time method for first- and second-order hyperbolic systems. *SIAM J. Sci. Comput.*, 35:A52–A78, 2013.

[Eng09] S. Engblom. Parallel in time simulation of multiscale stochastic chemical kinetics. *SIAM Multiscale Modeling and Simulation*, 8:46–68, 2009.

[FC03] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *International Journal for Numerical Methods in Engineering*, 58:1397–1434, 2003.

[FHM05] P. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 433–440. Springer Berlin Heidelberg, 2005.

[GEF05] I. Garrido, M. Espedal, and G. Fladmark. A convergent algorithm for time parallelization applied to reservoir simulation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 469–476. Springer Berlin Heidelberg, 2005.

[GH08] M. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. In U. Langer, M. Discacciati, D. Keyes, O. Widlund, and W. Zulehner, editors, *Domain decomposition methods in science and engineering*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 45–56. Springer Berlin Heidelberg, 2008.

[GH14] M. Gander and E. Hairer. Analysis for parareal algorithms applied to Hamiltonian differential equations. *Journal of Computational and Applied Mathematics*, 259:2–13, 2014.

[GLFE06] I. Garrido, B. Lee, G. Fladmark, and M. Espedal. Convergent iterative schemes for time parallelization. *Mathematics of Computation*, 75:1403–1428, 2006.

[JS12] R. Joubaud and G. Stoltz. Nonequilibrium shear viscosity computations with Langevin dynamics. *SIAM Multiscale Modeling and Simulation*, 10(1):191–216, 2012.

[LLMS20] F. Legoll, T. Lelièvre, K. Myerscough, and G. Samaey. Parareal computation of stochastic differential equations with time-scale separation: a numerical convergence study. *Computing and Visualization in Science*, 23(9), 2020.

[LLS13] F. Legoll, T. Lelièvre, and G. Samaey. A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations. *SIAM J. Sci. Comput.*, 35(4):A1951–A1986, 2013.

[LMT01] J.-L. Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of PDEs. *C.R. Math. Acad. Sci. Paris*, 332:661–668, 2001.

[LRS10] T. Lelièvre, M. Rousset, and G. Stoltz. *Free Energy Computations*. Imperial College Press, 2010.

[Mad07] Y. Maday. Parareal in time algorithm for kinetic systems based on model reduction. In A. Bandrauk, M. Delfour, and C. Le Bris, editors, *High-dimensional partial differential equations in science and engineering*, volume 41 of *CRM Proceedings and Lecture Notes*, pages 183–194. American Mathematical Society, 2007.

[MM20] Y. Maday and O. Mula. An adaptive parareal algorithm. *Journal of Computational and Applied Mathematics*, 377:112915, 2020.

[MT02] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C.R. Math. Acad. Sci. Paris*, 335:387–392, 2002.

[MT05] Y. Maday and G. Turinici. The parareal in time iterative solver: a further direction to parallel implementation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 441–448. Springer Berlin Heidelberg, 2005.

[SR05] G. Staff and E. Rønquist. Stability of the parareal algorithm. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 449–456. Springer Berlin Heidelberg, 2005.

[Tal02] D. Talay. Stochastic Hamiltonian dissipative systems: exponential convergence to the invariant measure, and discretization by the implicit Euler scheme. *Markov Processes and Related Fields*, 8(2):163–198, 2002.

(F. Legoll) École des Ponts ParisTech and Inria, 6-8 Avenue Blaise Pascal, Cité Descartes, 77455 Marne-la-Vallée, France. E-mail address: frederic.legoll@enpc.fr
(T. Lelièvre) École des Ponts ParisTech and Inria, 6-8 Avenue Blaise Pascal, Cité Descartes, 77455 Marne-la-Vallée, France. E-mail address: tony.lelievre@enpc.fr
(U. Sharma) Fachbereich Mathematik und Informatik, Freie Universität Berlin, Arnimallee 9, 14195 Berlin, Germany. E-mail address: upanshu.sharma@fu-berlin.de (this work has been initiated when U. Sharma was at École des Ponts ParisTech, 6-8 Avenue Blaise Pascal, Cité Descartes, 77455 Marne-la-Vallée, France).