SIAM J. SCI. COMPUT.
Vol. 35, No. 1, pp. A52–A78

STABLE PARAREAL IN TIME METHOD FOR FIRST- AND SECOND-ORDER HYPERBOLIC SYSTEMS*

XIAOYING DAI† AND YVON MADAY‡

Abstract. The parareal in time algorithm allows one to perform parallel simulations of time-dependent problems. This algorithm has been implemented on many types of time-dependent problems with some success. Recent contributions have allowed one to extend the domain of application of the parareal in time algorithm so as to handle long-time simulations of Hamiltonian systems. This improvement has managed to avoid the fatally large lack of accuracy of the plain parareal in time algorithm, which does not conserve invariant quantities. A somewhat similar difficulty occurs for problems where the solution lacks regularity, either initially or during the evolution, as is the case for hyperbolic systems of conservation laws. In this paper we identify the reasons for instabilities of the parareal in time algorithm and propose a simple way to cure them. We use the new method to solve a linear wave equation and a nonlinear Burgers' equation. The results illustrate the stability of this variant of the parareal in time algorithm.

Key words. parareal in time algorithm, parallelization, time discretization, evolution equations, hyperbolic system, wave equation

AMS subject classifications. 65L05, 65P10, 65Y05

DOI. 10.1137/110861002

1. Introduction.

Parallel in time algorithms represent a new competitive way of using the ever-increasing number of cores in today's supercomputer platforms either in cases where classical domain decomposition methods are of no relevance (for systems of differential equations) or in cases where the scalability of the sole domain decomposition (for evolution PDEs) reaches saturation. This kind of algorithm in the fourth dimension has received too little attention in the literature in the past due to the inherent sequential nature of the time dimension: it seems difficult to simulate what will arrive in the far future, e.g., next week, without knowing in detail what will arrive in the near future, e.g., tomorrow and each day of the current week. Due to the paramount importance of finding new ways of parallelization, together with the ever-increasing number of processors present in the new generations of architectures, algorithms such as parareal in time, parallel deferred corrections, and parallel exponential integrators have become an active research topic showing promising applicability for large-scale computations and perhaps one way to approach exascale computations. We refer, e.g., to the book of K. Burrage [4] for a synthetic approach to the subject (see also [5]). In that book, the various techniques of parallel in time algorithms are classified into three categories: (i) parallelism *across the system*, (ii) parallelism *across the method,* and (iii) parallelism *across time*.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 3, 2010; accepted for publication (in revised form) August 22, 2012; published electronically January 8, 2013. This work was supported by ANR-06-CIS6-007 and National Science Foundation of China grant 11101416.

http://www.siam.org/journals/sisc/35-1/86100.html

†LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (daixy@lsec.cc.ac.cn).

‡UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France and CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France. Current address: Division of Applied Mathematics, Brown University, Providence, RI (maday@ann.jussieu.fr).

The parareal in time algorithm pioneered in [16], extended later in [1, 3, 19] under a form much better tuned to the treatment of nonlinear problems, belongs to the third category above (see [21] for the first algorithm in this category). It combines two propagators, one fine and one coarse, in a predictor-corrector manner that allows one to use the fine propagator in parallel on the various processors, while the coarse propagator is solved in a sequential manner. The algorithm provides good, provable convergence for diffusive PDEs such as parabolic-type problems; however, several studies have documented some instabilities for hyperbolic equations [9, 11, 20, 6].

A challenging field of application for parallel in time propagators deals with long-time simulation of Hamiltonian systems. Here also the plain parareal method does not maintain the geometric properties that are essential to guarantee the quality of the approximation. Recall that these geometric properties lead to the convergence of the algorithm toward the solution of the original systems in long-time simulations. These properties—either the symplecticity (exact preservation of the geometric frame) or the symmetry (time reversability) of the scheme—are not achieved for the parareal scheme even if the basic coarse and fine propagators are symplectic or symmetric; we refer to the book [14] for precise definitions of the geometric features of the Hamiltonian flow and the necessary conditions for the discretization schemes that allow one to maintain the numerical flow in this framework. A cure for this lack of geometric properties for the parareal scheme has been proposed in [8], based on two ingredients: The first one is symmetrization of the parareal in time algorithm, which leads to a new multistep scheme. This symmetrization alone is not sufficient, though, to provide a correct algorithm for Hamiltonian systems. Indeed, some resonances are artificially introduced by the parareal strategy itself, as is explained in [8]; they prevent the symmetric variant from behaving well. The second ingredient introduced in [8] consists simply, after each correction of the predictor-corrector scheme, of projecting the solution coming out of the parareal in time algorithm onto the manifold, expressing the preservation of some basic invariant quantities of the Hamiltonian system. This is quite simple to implement, at least in its basic formulation (there exists a symmetric version based on a more involved implicit resolution of projection on the manifold). We refer also to [15] for another parallel implementation applied to a celestial Hamiltonian system.

Following the second idea recalled above to stabilize the parareal in time algorithm for a Hamiltonian system, we propose in this paper an extension of it that resolves the possible instability that occurs when solving hyperbolic-type problems. A different strategy, based on providing a much more involved correction procedure that reutilizes previously computed information has been proposed in [10, 7] and has already improved the performance of the parareal in time algorithm for second-order hyperbolic systems. The method proposed here seems much simpler to implement and more general in scope and application.

The remainder of this paper is organized as follows. In section 2, we recall the basics of the plain parareal in time algorithm, including the current state of the art on the stability and convergence analysis of the algorithm (see [3, 2, 22]); we use the conclusion of this analysis to introduce the proposed stable variant of the parareal scheme. In section 3 we address the case of the wave equation, a simple second-order hyperbolic equation. The correction we propose is discussed in full detail, and the numerical simulation illustrates the good properties of the variant. In section 4 we address the case of Burgers' equation with small viscosity; here again the variant performs well. Finally, some conclusions are drawn in section 6.

2. Some elements of the parareal in time algorithm.

2.1. Basics of the parareal in time algorithm. Let us consider a possibly nonlinear time-dependent PDE of the form

$$\frac{\partial y}{\partial t} + \mathcal{A}(t; y) = 0, \qquad y(0) = y_0, \tag{2.1}$$

where $y(t)$ takes its values in a Banach space B and $\mathcal{A}$ is a partial differential operator defined over B. We assume here that this problem is well posed, at least for some time: the solution $y(t)$ exists for $t \in [0, T)$. More precisely, we assume that there exists a propagator $\mathcal{E}$ such that

$$y(t) = \mathcal{E}_0^t(y_0),$$

named the exact propagator. We note that, with obvious notation, we have

$$y(t) = \mathcal{E}_s^t(y(s)) \quad \forall s, t, \quad 0 \le s \le t \le T. \tag{2.2}$$

Let us assume now that we want to simulate this PDE on the interval $[0, T)$. We thus introduce a discrete propagator $\mathcal{F}$ that is assumed to be a fine and precise approximation of the exact solution based on a discretization in time only. The propagator $\mathcal{F}$ can be based on an approximation using a time discretization and, e.g., the use of an Euler-type or a more involved scheme based on a small enough time step δt.

For instance, let us denote as $T_0 = 0$, $T_N = T$, and $T_n = n\Delta T$ (with $\Delta T = \frac{T}{N}$) special times at which we are interested in considering snapshots of the solution $y(T_n)$. Then, with notation similar to (2.2) above, we have the approximation

$$Y_n = \mathcal{F}_{T_0}^{T_n}(y_0), \quad n = 1, \dots, N.$$

We also have

$$Y_{n+1} = \mathcal{F}_{T_n}^{T_{n+1}}(Y_n).$$

In what follows we present the parareal in time algorithm, which allows us to define a sequence $\{\mathbf{Y}^k\}_k$, $\mathbf{Y}^k \equiv (Y_1^k, Y_2^k, \dots, Y_N^k)$, such that, for any n, $\mathbf{Y}_n^k$ converges to $\mathbf{Y}_n$ in the space B as k goes to infinity. The definition of this sequence requires the availability of another approximated propagator, denoted as $\mathcal{G}$. This second propagator is assumed to be cheaper than $\mathcal{F}$ at the price of some inaccuracy. One can think immediately about having $\mathcal{G}$ based on a similar Euler scheme as $\mathcal{F}$ with the larger time step $dT, \delta t \ll dT \le \Delta T$. But other possibilities are also offered: in addition to the solution scheme in time, a discretization in space has to be performed; for instance, the spatial discretization can be based on a finite element method, and the coarse solver $\mathcal{G}$ can then be based both on a coarser time step and on a coarser spatial discretization; we refer to [11] and [18], where such a strategy is followed. Another possibility consists of providing $\mathcal{F}$ with all the physics of the phenomenon while $\mathcal{G}$ is based on simplified physics (see, e.g., [17] for an example).

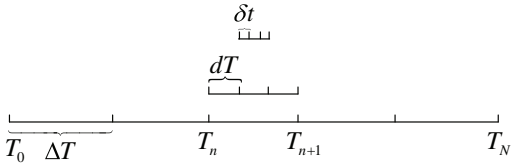
Assume $\mathcal{F}$ and $\mathcal{G}$ are given. Then we can proceed by giving the definition of the sequence. Let us initiate $\mathbf{Y}^0$ by

$$Y_n^0 = \mathcal{G}_{T_0}^{T_n}(y_0), \quad n = 1, \dots, N. \tag{2.3}$$

Assuming now that $\mathbf{Y}^k$ is known, then the iterative process proceeds by setting $Y_0^{k+1} = y(0)$ and

$$Y_{n+1}^{k+1} = \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1}) + \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k), \quad n = 1, \dots, N. \tag{2.4}$$

We first comment on the parallelization of the above algorithm. The initialization step (2.3) is sequential but involves a cheap, coarse propagator. Then, assuming that every $(Y_n^k)_n$ is known at step k, we first solve over each $]T_n, T_{n+1}[$ the totally independent propagation problems that provide $\mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k)$; we can either propagate the coarse solver again (and now in parallel) to get $\mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k)$ or just remember that these quantities were computed during the previous step and stored; in any case this is again parallel work. The corrector $\mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k)$ can then be built for any n, and the step can proceed. We finally solve the predictor based on the coarse solver $\mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1})$—sequentially—and immediately correct it by adding the previously computed corrections until convergence. Note that in this process, the fine solutions are only used on small propagation intervals and can be performed in parallel:



2.2. Numerical analysis of the parareal in time algorithm. We review in this subsection some elements of the stability and convergence of the parareal in time algorithm that were first presented in [2]. Let us assume that there exists a sequence of Banach spaces B_j such that $B_{j+1} \subset B_j$. It is also assumed that the problem (2.1) is stable in these spaces in the sense that there exists a constant $C > 0$ such that, for all $j, 1 \le j \le J$,

$$\|\mathcal{E}_0^t(y_0)\|_{B_j} \le C\|y_0\|_{B_j}. \tag{2.5}$$

The algorithm converges under the following natural hypothesis: denoting by $\delta\mathcal{F}$ (resp. $\delta\mathcal{G}$) the difference $\delta\mathcal{F}_s^t = \mathcal{E}_s^t - \mathcal{F}_s^t$ (resp. $\delta\mathcal{G}_s^t = \mathcal{E}_s^t - \mathcal{G}_s^t$), we assume that there exists a constant $C > 0$ such that, for all $j, 1 \le j \le J$,

$$\forall t \ge 0, \forall \tau \ge 0, \quad \|\mathcal{G}_t^{t+\tau}(x) - \mathcal{G}_t^{t+\tau}(y)\|_{B_j} \le (1 + C\tau)(\|x - y\|_{B_j}). \tag{2.6}$$

In addition let us assume that there exists a constant $C > 0$ such that, for all $j, 1 \le j \le J$,

$$\forall t \ge 0, \forall \tau \ge 0, \quad \|\delta\mathcal{F}_t^{t+\tau}(x) - \delta\mathcal{F}_t^{t+\tau}(y)\|_{B_j} \le C\tau\eta\|x - y\|_{B_{j+1}}, \tag{2.7}$$

$$\|\delta\mathcal{G}_t^{t+\tau}(x) - \delta\mathcal{G}_t^{t+\tau}(y)\|_{B_j} \le C\tau\varepsilon\|x - y\|_{B_{j+1}}, \tag{2.8}$$

where η and ε are small quantities (typically, for the Euler scheme these are $\eta = \delta t$ and $\varepsilon = dT$) and C is a generic constant, independent of the quantities on the right-hand side of the previous inequalities and on $j \le J$. We refer, e.g., to chapter 5 of [13] for details on this classical error analysis for the discretization of time-dependent functional equations.

Remark 1. For Euler schemes based on a time step δt for $\mathcal{F}$ and dT for $\mathcal{G}$, hypotheses (2.7) and (2.8) hold with $\eta = \delta t$ and $\varepsilon = dT$, respectively. For the heat equation, for instance, the spaces B_j would be the Sobolev spaces H^{2j}.

The convergence of the parareal approximation is stated in the following theorem (slight variation of theorem 1 and [2]).

THEOREM 2.1. *Assume that the discrete propagators $\mathcal{F}$ and $\mathcal{G}$ satisfy* (2.6), (2.7), *and* (2.8). *Then the error between the exact solution and the solution provided by the parareal scheme* (2.4) *satisfies, for any $k \geq 1$,*

$$\forall n \leq N, \quad \|Y_n^k - y(T_n)\|_{B_0} \leq C(\varepsilon^k + \eta)(1 + \|y_0\|_{B_k}). \tag{2.9}$$

Note that the constant C in (2.9) depends on k and T and this convergence result can be polluted on a long time range by a constant C that becomes too large. Note also that the convergence result implies, of course, stability. Hence, provided that the initial condition is regular enough and remains regular (when propagated through the PDE), the parareal in time algorithm is stable. It is well known that parabolic equations have regularizing effects in space; on the contrary, hyperbolic (nondissipative) equations generally lead to a decay of regularity, often with the creation of shocks.

In connection with Theorem 2.1, this actually explains the problem of the use of the parareal scheme when nondissipative PDEs or PDEs with small dissipation are simulated. This is the case for small-viscosity parabolic problems, as is explained in the next subsection; it is also the case for hyperbolic systems. We illustrate this statement in two situations: the periodic wave equation (a second-order hyperbolic equation) and the periodic Burgers' equation (a first-order nonlinear equation with an additional small viscosity), which, starting with regular enough initial condition, solves or corrects the instability of the parareal in time algorithm in this situation. This allows us to rectify somewhat previous statements as in, e.g., [7], where it is documented that the instability problem occurs for second-order hyperbolic problems and not for parabolic and first-order hyperbolic problems. The problem is not really the type of equations (and certainly not the order of the equations) to which the parareal algorithm is applied, but the space–time regularity of the solution of the problem, which of course depends strongly on the regularity of the initial condition.

2.3. Illustration of the stability and instabilities of the plain parareal in time algorithm.

2.3.1. A parabolic case. We have performed a series of simple tests on the linear parabolic equation provided with periodic boundary conditions,

$$\begin{cases} \dfrac{\partial u}{\partial t} - \mu \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial u}{\partial x} = 0, \\ u(x,0) = u_0(x), \\ u(x,t) = u(x + 2\pi, t), \end{cases} \tag{2.10}$$

in the case where the viscosity μ is positive yet very small: $\mu = 10^{-10}$. We have chosen a standard Fourier spectral approximation in space. That is, we choose

$$S_N = span\{e^{i\ell \frac{2\pi}{\mathbb{T}} x}, \ell = -N, \ldots, N\}, \quad N \in \mathbb{N}, \tag{2.11}$$

and approximate $u(x,t)$ by

$$u_N(x,t) = \sum_{\ell=-N}^{N} \hat{u}_\ell(t) e^{i\ell \frac{2\pi}{\mathbb{T}} x}. \tag{2.12}$$

For this example, $\mathbb{T} = 2\pi$, and we choose $N = 256$ in our tests. For the discretization in time, we choose a second-order implicit Euler scheme for propagation over

$[0, T = 2]$. The parareal in time algorithm is implemented with $\Delta T = 2.0 \times 10^{-2}$, $dT = 1.0 \times 10^{-2}$, and $\delta t = 1.0 \times 10^{-4}$.

These simulations are done in order to illustrate the behavior of the scheme as regards the regularity of the initial solution. We have thus tested different types of initial conditions with different regularities by choosing $u_0(x) = \sum \hat{u}_\ell e^{i\ell x}$ with

$$\hat{u}_\ell = \begin{cases} \dfrac{0.5i}{|\ell|^p} & \text{if } \ell > 0, \\ 0 & \text{if } \ell = 0, \\ -\dfrac{0.5i}{|\ell|^p} & \text{if } \ell < 0. \end{cases}$$

In the case of low regularity, $p = 0.5$ or $p = 1$, the relative error after 15 parareal iterations is about 10,000%! There is no blowup; the iterations can continue, with the error eventually going to zero. For $p = 4$ or even better $p = 10$, convergence is achieved after very few iterations, as is expected from Theorem 2.1.

2.3.2. A hyperbolic case. We have also done some tests on a second-order hyperbolic equation: the following one-dimensional wave equation with periodic boundary conditions:

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - c^2 \dfrac{\partial^2 u}{\partial x^2} = 0, \\ u(x,0) = f(x), \quad \dfrac{\partial u}{\partial t}(x,0) = g(x), \\ u(x,t) = u(x+\mathbb{T},t), \quad \dfrac{\partial u}{\partial x}(x,t) = \dfrac{\partial u}{\partial x}(x+\mathbb{T},t), \end{cases} \tag{2.13}$$

where f and g are periodic given initial conditions, c stands for the speed of the wave, and $\mathbb{T} \in \mathbb{R}^+$ represents the period. In our tests, we choose $\mathbb{T} = 5000, c = 2000$.

We choose again a standard Fourier spectral approximation as the discretization in space and a velocity Verlet scheme as the discretization scheme in time. The details of these discretization schemes for the wave equation above will be given in the next section.

We perform different simulations with the discretization parameters equal to $T = 100, \Delta T = 2.0 \times 10^{-1}, dT = 4.0 \times 10^{-3}, \delta t = 2.0 \times 10^{-6}$, and $N = 30$ for different initial conditions of different regularities: $f(x) = 0$, $g(x) = \sum_{l \in \mathbb{Z}} \hat{v}_l e^{il\frac{2\pi}{T}x}$ with $u_0(x) = \sum_{\ell \in Z} \hat{u}_\ell e^{i\ell\frac{2\pi}{\mathbb{T}}x}$ with

$$\hat{u}_\ell = \begin{cases} \dfrac{1}{|\ell|^p} & \text{if } \ell \neq 0, \\ 0 & \text{if } \ell = 0, \end{cases}$$

with $p = 1, 5$, and 10. The cases with $p = 1$ and $p = 5$ diverge, while the case with $p = 10$ converges.

These results provide a first illustration that the behavior of the parareal in time algorithm is deeply linked to the regularity of the initial solution.

3. The periodic wave equation.

3.1. The basic Fourier discretization. We consider further the one-dimensional wave equation with periodic boundary conditions (2.13). The weak form of this equation is as follows: Find u such that $\forall t > 0, u(\cdot, t) \in S = \{v \in H^1(0, \mathbb{T}), v(0) = v(\mathbb{T})\}$,

$$\left(\frac{\partial^2 u}{\partial t^2}, v\right) + c^2 \left(\frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}\right) = 0 \quad \forall v \in S; \tag{3.1}$$

here, $(u, v) = \int_0^{\mathbb{T}} u\bar{v}dx$.

We use now the Fourier spectral method to semidiscretize problem (3.1) in space. That is, we choose S_N as in (2.11) and approximate $u(x, t)$ as in (2.12).

The solution u_N is defined by a Galerkin approximation so as to satisfy

$$\left(\frac{\partial^2 u_N}{\partial t^2}, v_N\right) + c^2 \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0 \quad \forall v_N \in S_N. \tag{3.2}$$

As is classical, this Galerkin approximation reduces to a system of differential equations in the Fourier coefficients $\hat{u}_k(t)$:

$$\mathbb{T}\frac{\partial^2 \hat{u}_\ell}{\partial t^2}(t) + \frac{(2\pi)^2}{\mathbb{T}}\ell^2 c^2 \hat{u}_\ell(t) = 0 \quad \forall \ell \in [-N, N], \tag{3.3}$$

which can be exactly solved. This will be used in order to compute errors between exact and approximated solutions.

Next, using $v = \frac{\partial u}{\partial t}$ in (3.1) and $v_N = \frac{\partial u_N}{\partial t}$ in (3.2) allows us to derive, for any time t, the following conservations of energies:

$$\left\|\frac{\partial u}{\partial t}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})} + c^2 \left\|\frac{\partial u}{\partial x}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})} = \left\|\frac{\partial u}{\partial t}(\cdot, 0)\right\|^2_{L^2(0,\mathbb{T})} + c^2 \left\|\frac{\partial u}{\partial x}(\cdot, 0)\right\|^2_{L^2(0,\mathbb{T})} \tag{3.4}$$

and

$$\left\|\frac{\partial u_N}{\partial t}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})} + c^2 \left\|\frac{\partial u_N}{\partial x}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})} = \left\|\frac{\partial u_N}{\partial t}(\cdot, 0)\right\|^2_{L^2(0,\mathbb{T})} + c^2 \left\|\frac{\partial u_N}{\partial x}(\cdot, 0)\right\|^2_{L^2(0,\mathbb{T})}, \tag{3.5}$$

respectively. If we now introduce the Hamiltonian (energies)

$$H(u_N)(t) = \left\|\frac{\partial u_N}{\partial t}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})} + c^2 \left\|\frac{\partial u_N}{\partial x}(\cdot, t)\right\|^2_{L^2(0,\mathbb{T})}, \tag{3.6}$$

the conservation (3.5) reads, for any time t,

$$H(u_N)(t) = H(u_N)(0). \tag{3.7}$$

Of course further numerical discretization in time should preserve this invariant.

Remark 2. As is often the case for Hamiltonian systems, this basic invariant (which is here the total energy for the wave equation) is not the only one available; for instance it is immediately seen in (3.3) that by defining

$$h_\ell(v) = \left|\frac{\partial v}{\partial t}\right|^2 + \left(\frac{2\pi}{\mathbb{T}}\right)^2 \ell^2 c^2 |v|^2 \tag{3.8}$$

for any ℓ, $\ell = -N, \ldots, N$, we have,

$$\forall t, \quad h_\ell(\hat{u}_\ell)(t) = h_\ell(\hat{u}_\ell)(0), \tag{3.9}$$

which provides $2N + 1$ invariant quantities (note that $H = \sum_{\ell=-N}^{N} h_\ell$).

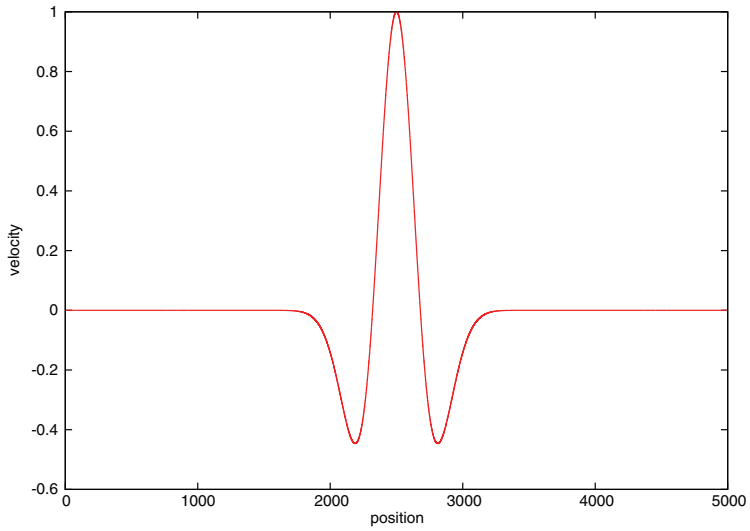


FIG. 3.1. *Initial velocity at different positions.*

In order to test the parareal in time algorithm on this simple second-order equation, we discretize the semidiscrete problem (3.2) by adding a time discretization that is based, for both the coarse $\mathcal{G}_{\Delta T}$ and the fine $\mathcal{F}_{\Delta T}$ propagators, on the velocity Verlet scheme. This is a second order in time, stable scheme that is consistent with the preservation of invariants (the symplectic nature of the velocity Verlet scheme makes it a method of choice for any Hamiltonian systems, such as this wave equation).

In all that follows, the discrete solutions will always, for any time t, belong to S_N and will result from a Galerkin process. We will thus denote only by u_n the approximation (in time) of the value of $u_N(T_n)$ (hence the full approximation of $u(T_n)$).

The plain parareal scheme reads

$$\begin{cases} u_{n+1}^0 = \mathcal{G}_{\Delta T}(u_n^0), \\ u_{n+1}^{k+1} = \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k). \end{cases} \tag{3.10}$$

3.2. Results for the plain parareal in time algorithm on a realistic test problem. In this example, the excitation is provided by a classical Ricker pulse, which is analytically defined as follows (see [20]):

1. $\mathbb{T} = 5000$ m, $c = 2000$ m/s.
2. The initial conditions are $f(x) = 0$, $g(x) = (1-2(f_s\pi\frac{x-x_s}{c})^2)\exp(-(f_s\pi\frac{x-x_s}{c})^2)$.

Here, $f_s = 2.5$ Hz is the frequency and $x_s = 2500$ m is the position of the vibration source. The initial velocity $g(x)$ is shown in Figure 3.1.

The numerical results are obtained with a coarse propagator $\mathcal{G}$ and a fine propagator $\mathcal{F}$ that differ only in their time steps dT and δt, respectively:

1. $T = 100, \Delta T = 2.0 \times 10^{-1}, dT = 4.0 \times 10^{-3}, \delta t = 2.0 \times 10^{-6}$.
2. $N = 30$.

The plot showing the evolution in time of the solution obtained after 15 iterations at different positions in the periodic space $[0, \mathbb{T}]$ in Figure 3.2 reveals an instability starting at time $t = 80$. The situation is even worse after 25 iterations. Actually this is only one simulation that illustrates the bad behavior of the parareal in time algorithm for this hyperbolic equation among many others for different choices of the discretization parameters δt or dT or even N but the very same initial condition, for

which the simulation is perfectly fine, stable, and convergent. The instability is thus not systematic, and changing the discretization parameters slightly is often enough to make the scheme work. The plain parareal in time algorithm does lack robustness, and this is what we want to correct in what follows.

Remark 3. Each plot representing the vibration (Figures 3.2, 3.3, and 3.8) corresponds to the vibration at different positions x indicated on the Y-axis. We thus display the displacement at some point x (magnified by a factor of 100) plus the value of the associated x position. The reason for the 100 times enlargement of the

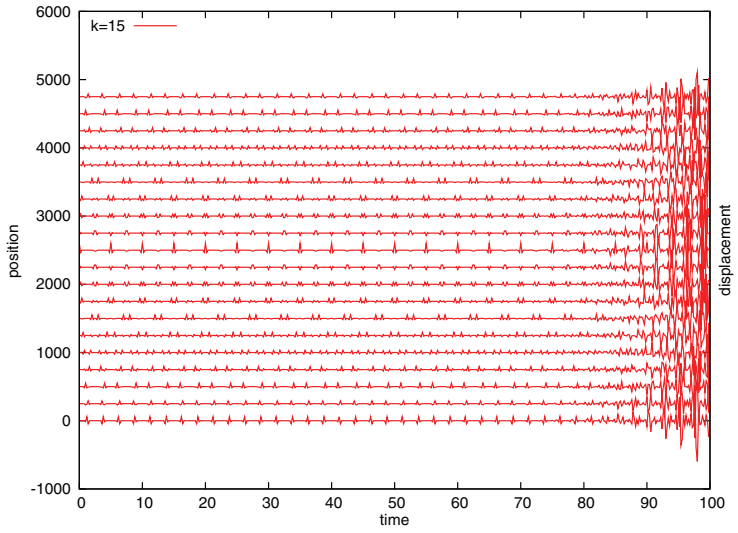


FIG. 3.2. *Each curve above represents the vibration at different positions (Y-axis) obtained by the parareal method* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *after* 15 *iterations as a function of time (X-axis). The instability is visible after time* = 80.

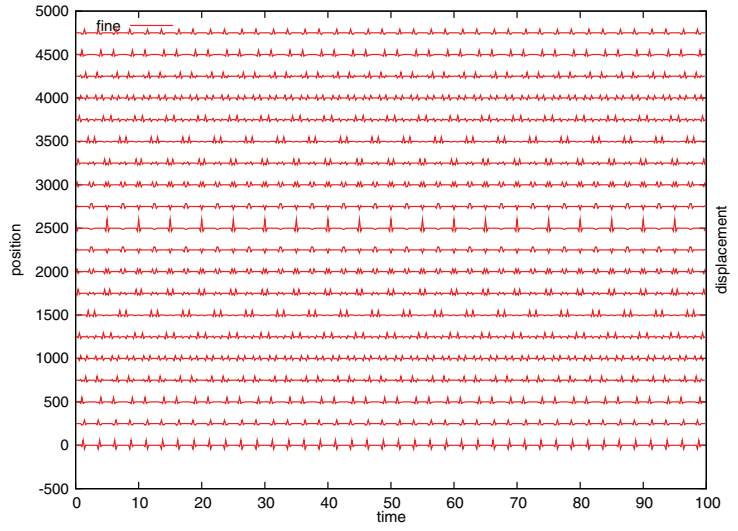


FIG. 3.3. *Each curve above represents the vibration at different positions (Y-axis) obtained by the fine sequential method* ($\delta t = 2.0 \times 10^{-6}$) *as a function of time (X-axis).*

displacement is that the absolute value of the displacement is too small compared to that of the position.

The exact solution shown on the next plot (Figure 3.3) allows us to see the differences.

The plots of the error between the exact solution and the solution obtained after k iterations, with $k = 1, \ldots, 15$ (see Figures 3.4 and 3.5), illustrate clearly that the solution obtained by the parareal in time algorithm has the following characteristics:

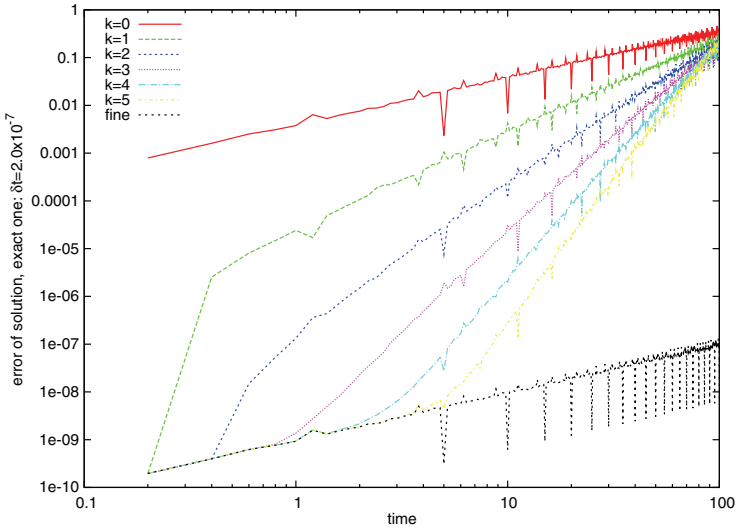


FIG. 3.4. *Error on the solution obtained by the parareal method* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

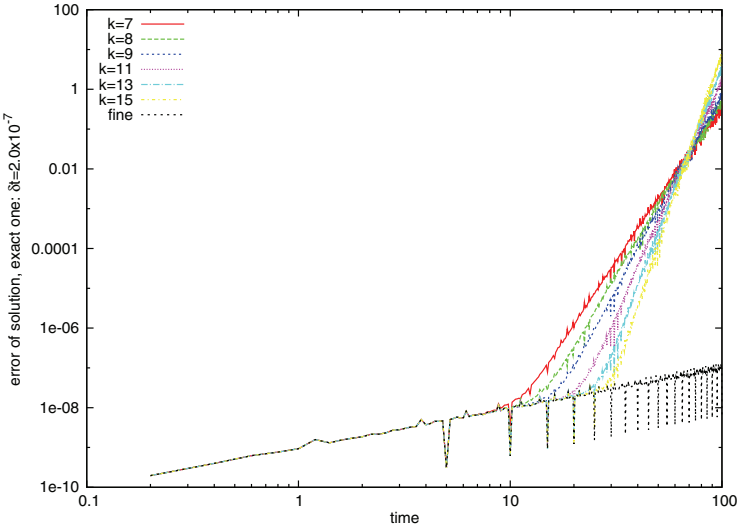


FIG. 3.5. *Error on the solution obtained by the parareal method* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 7, \ldots, 15$.

1. It converges for short time; e.g., only seven iterations are sufficient to recover the solution obtained with the sole sequential fine propagator at time equal to 10.
2. It does not converge for longer propagations, here for a time equal to 100.
3. During the first five or six iterations the error of the parareal in time algorithm is dominated by the error of the coarse scheme, but after $k = 6$, the error increases substantially to values that are order one and bigger.
4. After 15 iterations of the parareal in time algorithm, the parareal solution is computed with an acceptable error until time $t \simeq 30$ only.

Taking into account the variants of the parareal in time algorithm introduced in [8] to simulate Hamiltonian systems, we infer that the bad behavior of the parareal scheme comes from the nonpreservation of the invariant quantities. Let us for example observe the evolution of the value of the quantity H defined in (3.6).

Figures 3.6 and 3.7 illustrate the fact that the plain method does not conserve the proper quantities that are consistant with the divergence of the scheme in this case.

Finally, we want to report that we have performed the same simulation with a Ricker pulse of varying force (by modifying the value of f_s). The instability starts much earlier for larger f_s, confirming the relation between the instability and the singularity of the solution.

3.3. Parareal algorithm with projection. To start with, let us define the manifold $\mathcal{M}$ corresponding to the set of all functions v such that $H(v) = H_0 \equiv H(u(0))$. As in [8] we consider first the global invariant that deals with all the modes. Following [14, Chapter 4], we consider the following definition of the projection $\pi_{\mathcal{M}}$ onto the manifold $\mathcal{M}$. For any $\widetilde{u} = (\widetilde{q}, \widetilde{p}) \in \mathbb{R}^{2d}$, we define $u = (q, p) = \pi_{\mathcal{M}}(\widetilde{u}) \in \mathbb{R}^{2d}$ by

$$u = \pi_{\mathcal{M}}(\widetilde{u}) = \widetilde{u} + \lambda \nabla H(\widetilde{u}),$$

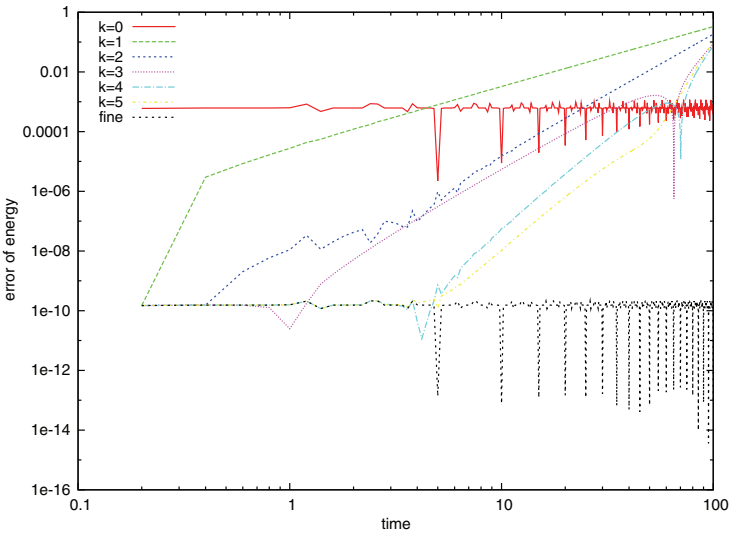


FIG. 3.6. *Error on the energy obtained by the parareal method* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

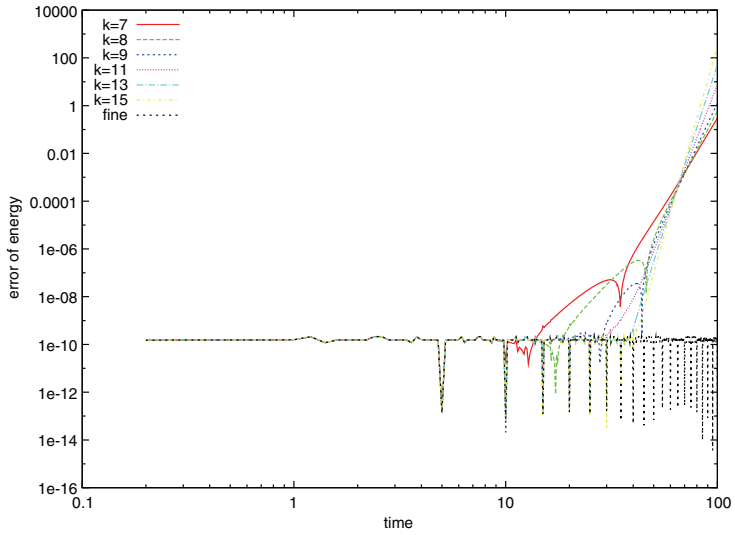


FIG. 3.7. *Error on the energy obtained by the parareal method* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 7, \ldots, 15$.

where $\lambda \in \mathbb{R}$ is such that $H(u) = H_0$, that is,

$$H(\widetilde{u} + \lambda \nabla H(\widetilde{u})) = H_0. \tag{3.11}$$

In the general case, (3.11) can only be solved by some iterative method, for example, Newton iterations, but in the current case, using the fact that H is quadratic, we take into account that (3.11) is a second-order polynomial in λ that can be solved exactly.

Then, combining the plain parareal scheme (3.10) with the above-defined projection method, we obtain the parareal method with projection to $\mathcal{M}$:

$$\begin{cases} u_{n+1}^0 = \mathcal{G}_{\Delta T}(u_n^0), \\ \tilde{u}_{n+1}^{k+1} = \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k), \\ u_{n+1}^{k+1} = \pi_{\mathcal{M}}(\tilde{u}_{n+1}^{k+1}). \end{cases} \tag{3.12}$$

We now show the results obtained by the parareal method with projection. We first note in Figure 3.8 that the global solution after $k = 15$ iterations does not seem to present the instabilities of the plain parareal in time algorithm. The same conclusion holds after 25 iterations.

This is confirmed by Figures 3.9 and 3.10, where we note an improvement in the error between the exact solution and the discrete one as the iterations evolve. The error is always at worst of the same order as the error of the coarse propagator, but it never uniformly reaches the accuracy of the fine solver.

Of course this is far from being what was expected, but the conservation of energy H does improve the behavior of the algorithm. Figures 3.11 and 3.12 illustrate the good preservation of H along the trajectory for the parareal method with projection. Although the solution does not diverge as in the plain parareal case, it does not converge as we would like.

In the results above, when doing the projection, we view all the Fourier modes as a whole system and treat them all together with a unique Lagrange multiplier.

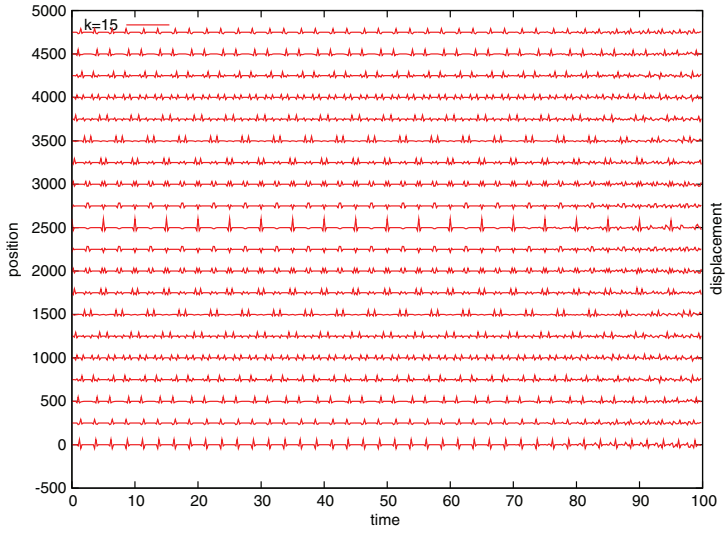


FIG. 3.8. *Each curve above represents the vibration at different positions (Y-axis) obtained by the parareal method with projection to the manifold* $\mathcal{M}$ *(*$\delta t = 2.0\times10^{-6}, dT = 4.0\times10^{-3}, \Delta T = 2.0\times 10^{-1}$*) after* 15 *iterations as a function of time (X-axis). The simulation is stable now; nevertheless, by comparison, we see that a suitable accuracy at time* $= 100$ *is not achieved.*

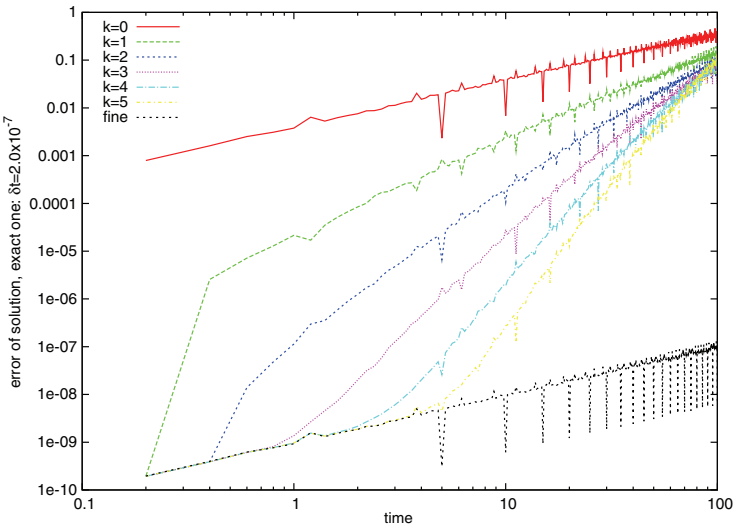


FIG. 3.9. *Error on the solution obtained by the parareal method with projection to the manifold* $\mathcal{M}$ *(*$\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$*) for different values of the iterations:* $k = 0, \ldots, 5$.

Remember that each Fourier coefficient individually has an energy that is preserved, as is explained in Remark 2. Instead of projecting on the manifold $\mathcal{M}$, where only one invariant is taken into account, we can have a manifold defined with more invariant properties. We illustrate this issue by choosing only two invariants based on two groups in the Fourier coefficients, those corresponding to high wave numbers and those corresponding to low wave numbers. That is, we partition all the Fourier modes into two groups, one containing the modes $|k| \geq K_0$ and the other containing all the

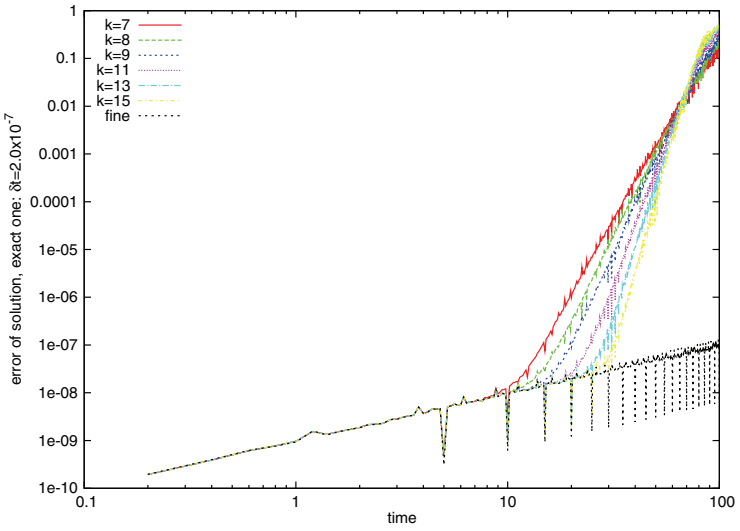


FIG. 3.10. *Error on the solution obtained by the parareal method with projection to the manifold* $\mathcal{M}$ *(*$\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$*) for different values of the iterations:* $k = 7, \ldots, 15$.

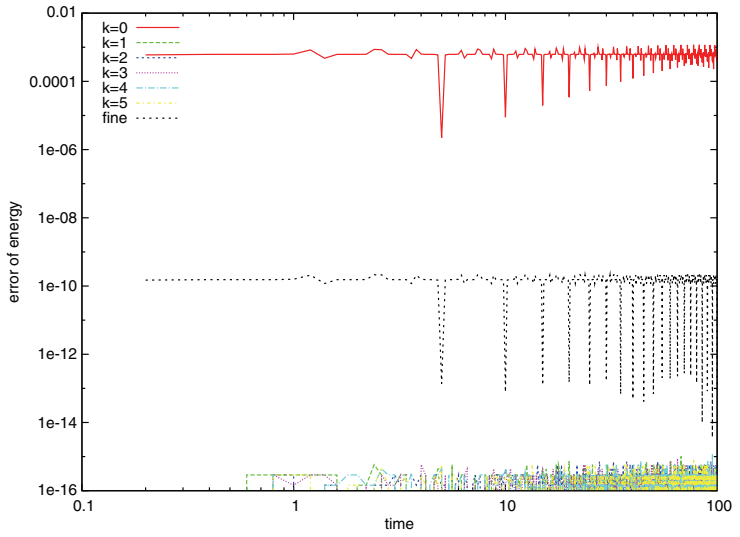


FIG. 3.11. *Error on the energy obtained by the parareal method with projection to the manifold* $\mathcal{M}$ *(*$\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$*) for different values of the iterations:* $k = 0, \ldots, 5$.

modes $|k| < K_0$. We then perform a projection for each group separately to conserve the energy of each group.

The motivation comes from the fact that, when we discretize the solution by Fourier series, the contribution of the high-frequency modes is much smaller than that of the low-frequency modes. This is what differs in the present Hamiltonian case (a PDE) from what was analyzed in the paper [8], which dealt only with ODEs. If we use the same Lagrange multiplier λ to correct all the modes, the contribution of the high-frequency modes may magnify them and pollute the solution.

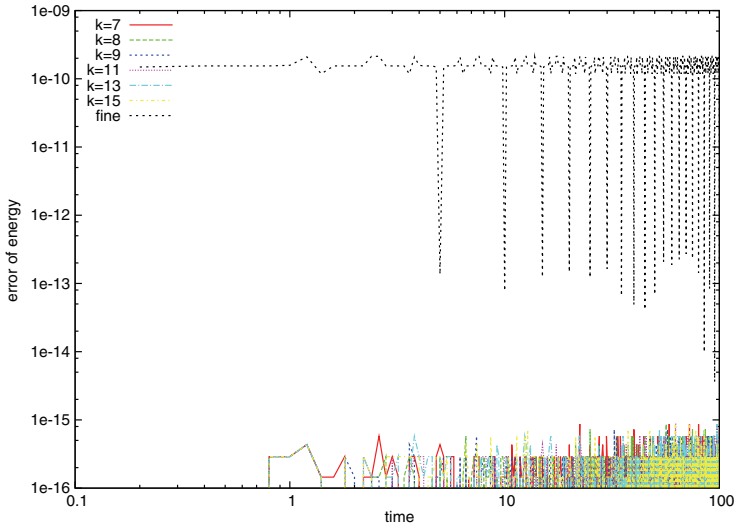


FIG. 3.12. *Error on energy obtained by the parareal method with projection to the manifold* $\mathcal{M}$ *(*$\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$*) for different values of the iterations:* $k = 7, \ldots, 15$.

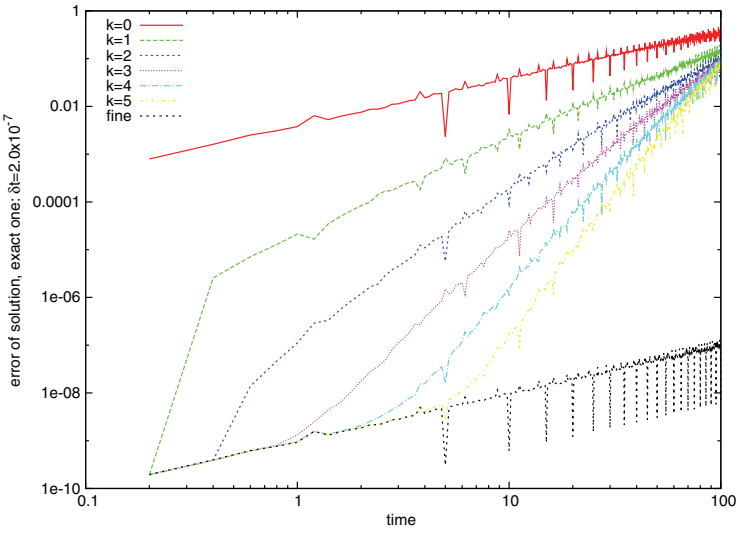


FIG. 3.13. *Error on the solution obtained by the parareal method with projection on two groups* $(\delta t = 2.0{\times}10^{-6}, dT = 4.0{\times}10^{-3}, \Delta T = 2.0{\times}10^{-1})$ *for different values of the iterations:* $k = 0, \ldots, 5$.

So, let us introduce two Lagrange multipliers, λ_1 and λ_2, one for the high-frequency modes and the other for the low-frequency modes.

The method is not more complex to implement. The numerical simulation, still with the same $N = 30$, is based on $K_0 = 20$; the results are much improved, as can be seen in Figures 3.13, 3.14, and 3.15.

The solution itself, after 15 iterations, is in excellent accordance with the very fine approximation, and the associated plot actually coincides with Figure 3.3.

For the linear second-order wave equation, the preservation of the invariant quantities based on the decomposition of the coefficients representing the solution into two

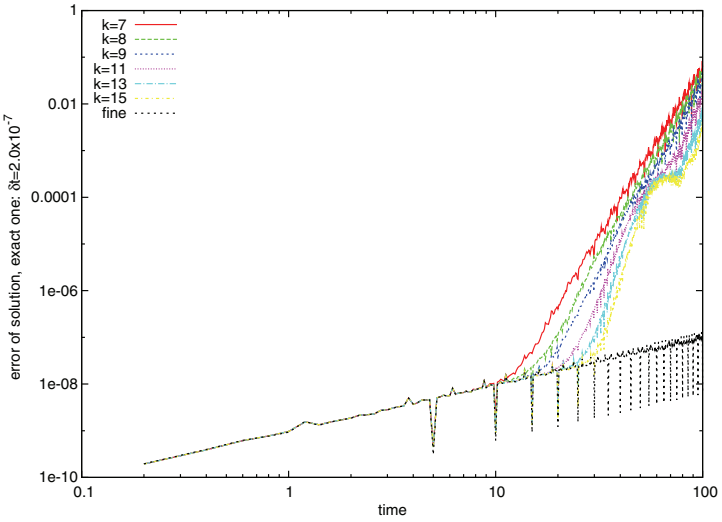


FIG. 3.14. *Error on the solution obtained by the parareal method with projection on two groups* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 7, \ldots, 15$.

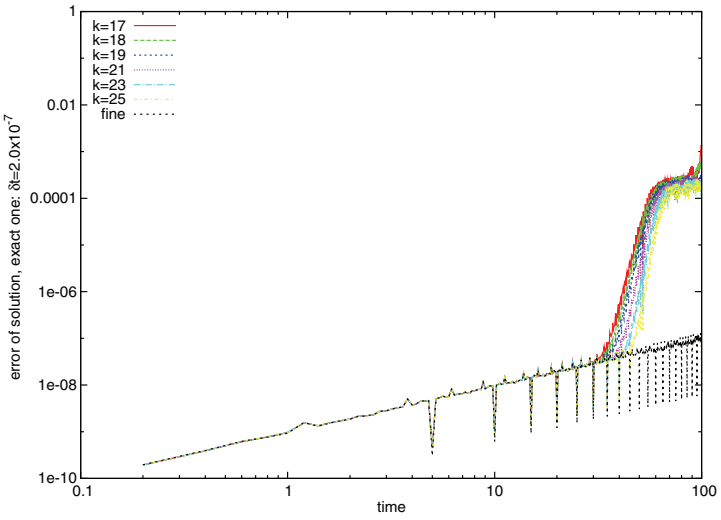


FIG. 3.15. *Error on the solution obtained by the parareal method with projection on two groups* ($\delta t = 2.0 \times 10^{-6}, dT = 4.0 \times 10^{-3}, \Delta T = 2.0 \times 10^{-1}$) *for different values of the iterations:* $k = 17, \ldots, 25$.

groups is thus a sufficiently mild ingredient to improve the behavior of the parareal in time algorithm. In the next section, we shall extend this variant of the parareal in time algorithm to a nonlinear PDE.

Note that, if the method converges, i.e., $\forall n = 1, \ldots, N$, $u_n^k \longrightarrow u_n^\infty$, the limit solution satisfies

$$u_{n+1}^\infty = \pi_{\mathcal{M}}(\mathcal{F}_{\Delta T}(u_n^\infty)),$$

thanks to (3.12), which, here, due to the fact that the propagator $\mathcal{F}_{\Delta T}$ is symplectic and thus preserves all Hamiltonian quantities, reads

$$u_{n+1}^{\infty} = \mathcal{F}_{\Delta T}(u_n^{\infty}),$$

which proves that the method converges toward the fine solution, as expected.

4. Burgers' equation.

4.1. The total Fourier/time-splitting discretization. Let us consider in this section the following periodic viscous Burgers' equation in one dimension:

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\frac{\partial (u^2)}{\partial x} &= 0, \\ u(x,0) &= u_0(x), \\ u(x+\mathbb{T},t) &= u(x,t). \end{aligned}$$

The viscosity $\nu > 0$ is planned to be small so that the solution will exhibit very sharp gradients after time $t = 1$.

Taking into account that the periodic frame of the problem allows us to use Fourier spectral approximation, we thus consider a semidiscretization based on the Galerkin Fourier approximation in S_N (see the definition in (2.11)):

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) + \frac{1}{2}\left(\frac{\partial (u_N)^2}{\partial x}, v_N\right) = 0 \quad \forall v_N \in S_N. \tag{4.1}$$

We then choose to use a second-order symmetrized splitting method based on the splitting between the diffusion effects (treated in the Fourier space exactly) and the nonlinear convection effects (treated in the physical space). The difference between the coarse propagator $\mathcal{G}$ and fine propagator $\mathcal{F}$ may be not only relative to the time step dT and δt, but also relative to the discrete step in space N_c and N_f. Denoting by δ either dT or dt, by N either N_c or N_f, the splitting schemes at each iteration proceeds as follows:

1. Solve the following problem in Fourier space in the interval $[\tau_n = n\delta, \tau_{n+\frac{1}{2}} = \tau_n + \frac{\delta}{2}]$:

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0 \quad \forall v_N \in S_N, \tag{4.2}$$

$$u_N(\tau_n) = u_{N,n}, \tag{4.3}$$

   and denote $\tilde{u}_{N,n+\frac{1}{2}} = u_N(\tau_{n+\frac{1}{2}})$. Note that, as in the discretization of the wave equation, by choosing iteratively $v_N = e^{ik\frac{2\pi}{\mathbb{T}}x}$, $k \in [-N, N]$, this results in a diagonal system of differential equations in the Fourier coefficients.

2. Solve the following problem in $[\tau_n, \tau_{n+1}]$:

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \frac{1}{2}\left(\frac{\partial (u_N)^2}{\partial x}, v_N\right) = 0 \quad \forall v_N \in S_N, \tag{4.4}$$

$$u_N(\tau_n) = \tilde{u}_{N,n+\frac{1}{2}}, \tag{4.5}$$

   and denote $u_N(\tau_{n+1})$ as $u_{N,n+\frac{1}{2}}$. Note that here, the solution procedure cannot be exact since the problem is nonlinear: the discretization in time of problem (4.4) is based on a third-order Runge–Kutta method and an exact evaluation of the Fourier coefficients of $\frac{\partial (u_N)^2}{\partial x}$ by a discrete Fourier method based on S_{2N}.

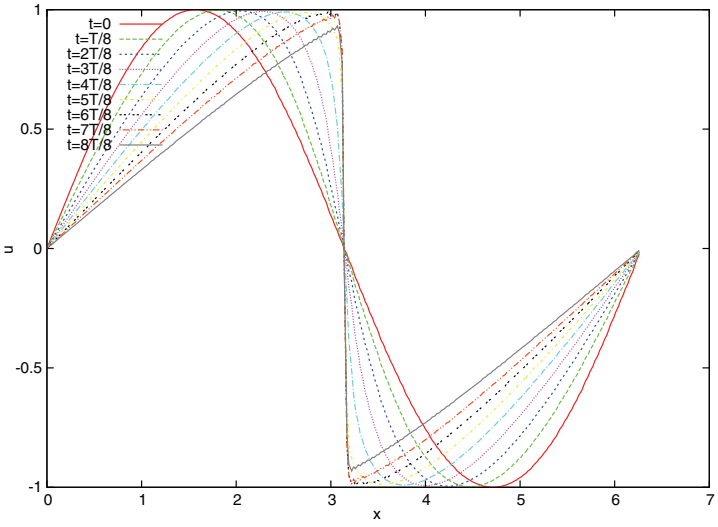


FIG. 4.1. *Case 1: solution obtained by the parareal method* ($\delta t = 1.0 \times 10^{-4}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$).

.

3. Solve the following problem in Fourier space in $[\tau_{n+\frac{1}{2}}, \tau_{n+1}]$:

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0 \quad \forall v_N \in S_N, \tag{4.6}$$

$$u_N(\tau_{n+\frac{1}{2}}) = u_{N,n+\frac{1}{2}}. \tag{4.7}$$

4.2. Results for the plain parareal in time algorithm. We have performed three tests for the plain algorithm. First we note that, provided that the viscosity is large enough, the parareal in time algorithm is working fine. Indeed, we list the choices for **Case 1** below:

Case 1:

1. $\nu = 1.0 \times 10^{-2}$.
2. $N_c = 2^8, N_f = 2^8$.
3. $T = 2, \Delta T = 2.0 \times 10^{-2}, dT = 1.0 \times 10^{-2}, \delta t = 1.0 \times 10^{-4}$.

Here and hereafter, N_c and N_f denote the truncated dimension for the Fourier spectral space used for the coarse propagator $\mathcal{G}$ and for the fine propagator $\mathcal{F}$ respectively.

The evolution corresponding to Case 1 is presented in Figure 4.1; the relative error in the solution with respect to a solution computed with a much smaller time step is presented in Figure 4.2, showing that after three parareal iterations, the same convergence as for the fine sequential simulation is achieved. Note that the target error of the fine propagator is quite nonuniform over the interval of time; indeed around $t = 1$ the error increases substantially, corresponding to the discontinuity that appears in the inviscid Burgers' equation.

For the second test, we have diminished the value of the viscosity. In this case, denoted as **Case 2**, we make the following choices:

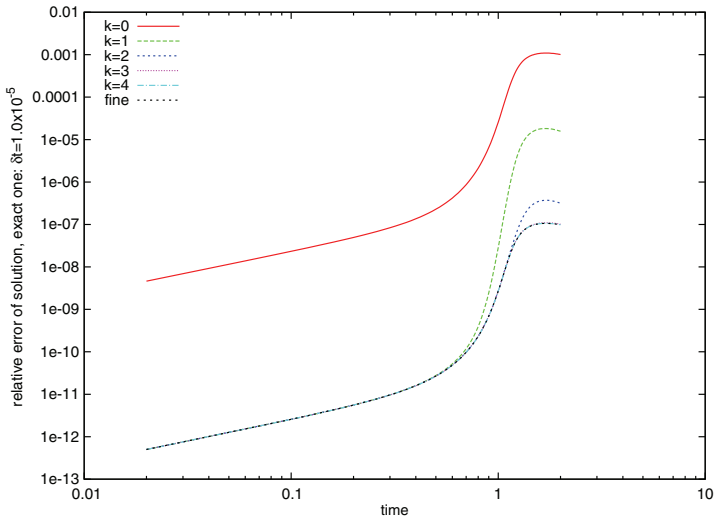


FIG. 4.2. *Case 1: relative error on the solution obtained by the parareal method* ($\delta t = 1.0 \times 10^{-4}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

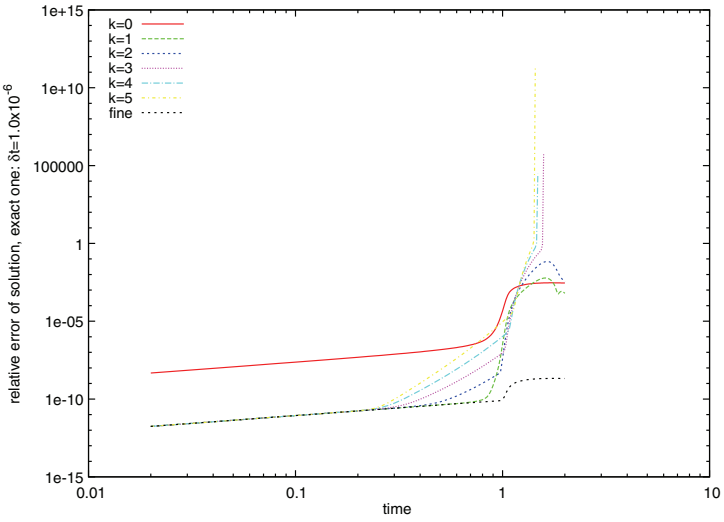


FIG. 4.3. *Case 2: relative error of the solution obtained by the parareal method* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

Case 2:

1. $\nu = 1.0 \times 10^{-3}$.
2. $N_c = 2^9, N_f = 2^9$.
3. $T = 2, \Delta T = 2.0 \times 10^{-2}, dT = 1.0 \times 10^{-2}, \delta t = 1.0 \times 10^{-5}$.

For this case, the plain parareal method is not stable, as can be seen in Figure 4.3; after three iterations instabilities occur soon after time 1, where the solution of the inviscid Burgers' equation becomes discontinuous.

It should be noted that, as explained in [18], using a coarser spatial representation helps in the stabilization of the process. Indeed, **Case 3** below provides a stable parareal approximation:

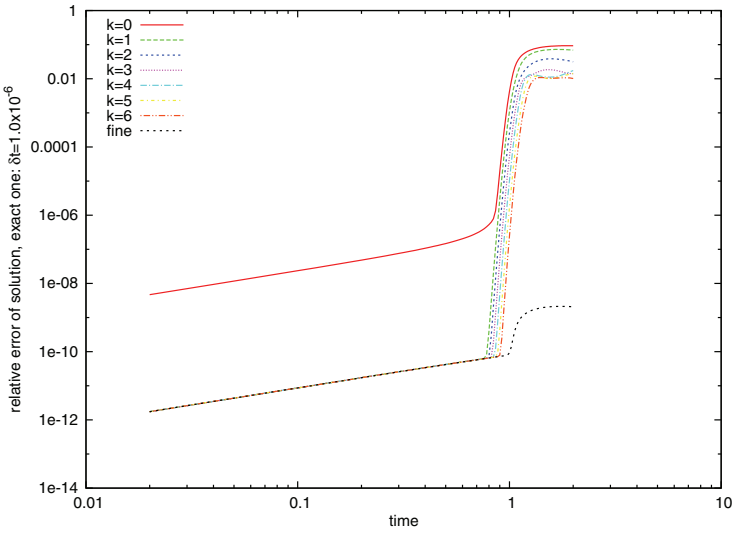


FIG. 4.4. *Case 3: relative error of the solution obtained by the parareal method* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

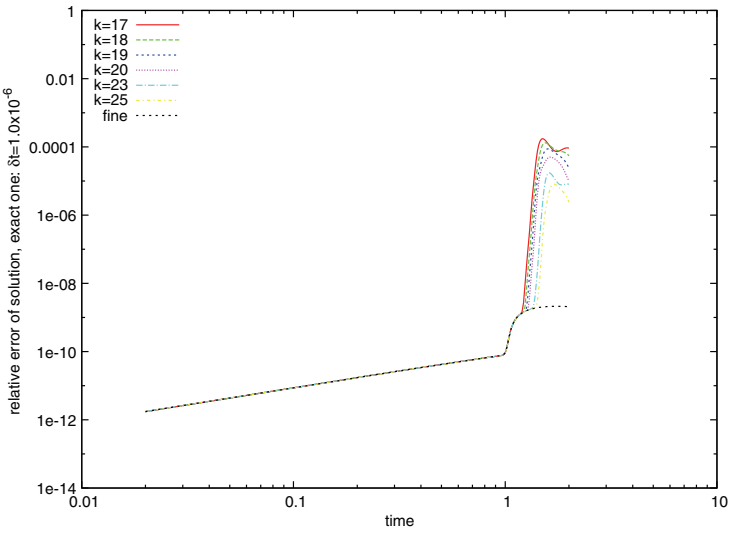


FIG. 4.5. *Case 3: relative error of the solution obtained by the parareal method* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 17, \ldots, 25$.

Case 3:

1. $\nu = 1.0 \times 10^{-3}$.
2. $N_c = 2^8, N_f = 2^9$.
3. $T = 2, \Delta T = 2.0 \times 10^{-2}, dT = 1.0 \times 10^{-2}, \delta t = 1.0 \times 10^{-5}$.

As can be seen in the series of Figure 4.4 (for $k = 0, \ldots, 6$) and Figure 4.5 (for $k = 17, \ldots, 25$), the method is stable but converges very slowly.

In the next section, we shall improve the last two simulations by extending the method presented in the previous section.

4.3. Results for the plain parareal in time algorithm with projection alone. To start with, let us note that, due to dissipation, no quantity is preserved along a trajectory; nevertheless, by using u_N as a test function in (4.1), we derive that

$$\|u_N(t)\|^2_{L^2(0,\mathbb{T})} + 2\nu \int_0^t \left[\frac{\partial u_N}{\partial x}\right]^2 = \|u_N(0)\|^2_{L^2(0,\mathbb{T})},$$

and actually for any two times $T_{n+1} > T_n \geq 0$,

$$\|u_N(T_{n+1})\|^2_{L^2(0,\mathbb{T})} + 2\nu \int_{T_n}^{T_{n+1}} \left[\frac{\partial u_N}{\partial x}\right]^2 = \|u_N(T_n)\|^2_{L^2(0,\mathbb{T})}. \tag{4.8}$$

In the parareal in time algorithm, starting from u_n^{k+1} (remember that this represents the Galerkin approximation that should be denoted as $u_{N,n}^{k+1}$), we expect that

$$\|u_{n+1}^{k+1}\|_{L^2(0,\mathbb{T})} = \|\mathcal{F}_{\Delta T}[u_n^{k+1}]\|_{L^2(0,\mathbb{T})}. \tag{4.9}$$

Since we do not want to compute $\mathcal{F}_{\Delta T}[u_n^{k+1}]$ at this stage, we approximate $\|\mathcal{F}_{\Delta T}[u_n^{k+1}]\|_{L^2(0,\mathbb{T})}$ by

$$\|\mathcal{F}_{\Delta T}[u_n^{k+1}]\|_{L^2(0,\mathbb{T})} \simeq \frac{\|\mathcal{F}_{\Delta T}[u_n^k]\|_{L^2(0,\mathbb{T})}}{\|u_n^k\|_{L^2(0,\mathbb{T})}} \|u_n^{k+1}\|_{L^2(0,\mathbb{T})} \tag{4.10}$$

and consider the manifold $\mathcal{M}_n^{k+1}$ corresponding to the set of all functions v such that $\|v\|_{L^2(0,\mathbb{T})} = \frac{\|\mathcal{F}_{\Delta T}[u_n^k]\|_{L^2(0,\mathbb{T})}}{\|u_n^k\|_{L^2(0,\mathbb{T})}} \|u_n^{k+1}\|_{L^2(0,\mathbb{T})}$.

Second, we have noted in the analysis of the wave equation that treating all the frequencies with a unique Lagrange multiplier is not effective enough. We thus partition the frequency space of functions in S_N into two (or more if necessary) groups G_i, $i = 1, 2$ (or $i = 1, \ldots, I$), defined by

$$G_i = span\{e^{i\ell\frac{2\pi}{\mathbb{T}}x}, L_i^- \leq |\ell| < L_i^+\}, \tag{4.11}$$

where $L_1^- = 0$, $0 < L_1^+ = L_2^-$, $L_2^+ = N + 1$ (or $L_1^- = 0$, $0 < L_1^+ = L_2^-$, $L_i^- < L_i^+ = L_{i+1}^-$, $L_I^+ = N + 1$ if I groups are considered). The L^2 projection on group G_i is denoted as P_i. In this case we consider the manifold $[\mathcal{M}^I]_n^{k+1}$ defined as being the intersection

$$[\mathcal{M}^I]_n^{k+1} = \cap_{i=1}^I [m^i]_n^{k+1}, \tag{4.12}$$

where the group manifolds $[m^i]_n^{k+1}$ are defined as follows:

$$[m^i]_n^{k+1} = \left\{ v \in S_N, \|P_i(v)\|_{L^2(0,\mathbb{T})} = \frac{\|P_i[\mathcal{F}_{\Delta T}[u_n^k]]\|_{L^2(0,\mathbb{T})}}{\|u_n^k\|_{L^2(0,\mathbb{T})}} \|u_n^{k+1}\|_{L^2(0,\mathbb{T})}. \right\} \tag{4.13}$$

To any given $\tilde{v} \in S_N$, we associate the element $\pi_{[\mathcal{M}^I]_n^{k+1}}(\tilde{v})$ defined by the standard projection method on $[\mathcal{M}^I]_n^{k+1}$, which involves the determination of I Lagrange multipliers.

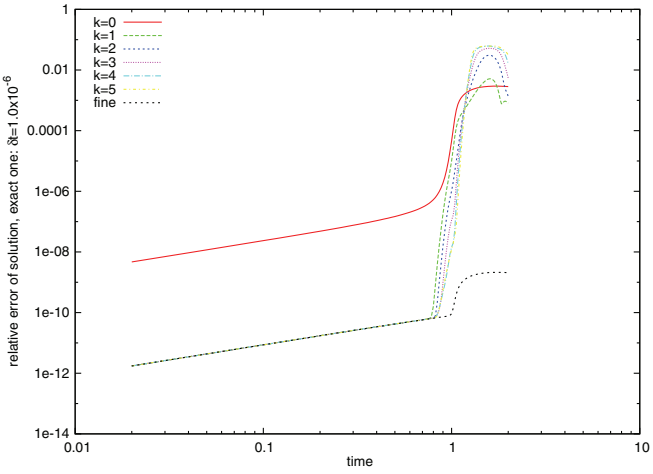


FIG. 4.6. *Case 2: relative error of the solution obtained by the parareal method with projection to two groups* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

With this manifold, the parareal scheme with projection reads

$$\begin{cases} u_{n+1}^0 = \mathcal{G}_{\Delta T}(u_n^0), \\ \tilde{u}_{n+1}^{k+1} = \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k), \\ u_{n+1}^{k+1} = \pi_{[\mathcal{M}^I]_n^{k+1}}(\tilde{u}_{n+1}^{k+1}). \end{cases} \tag{4.14}$$

The results are as follows (see Figure 4.6 (for $k = 0, \ldots, 5$) and Figure 4.7 (for $k = 17, \ldots, 25$)). We want to emphasize that the parareal in time algorithm is stabilized, but it does not converge much. Increasing the number of groups does not improve the behavior of this variant of the parareal in time algorithm.

Remark 4. If we knew the exact value of the energy of each group, the projection method would converge. The problem of course is that due to dissipation, and also to the nonlinear effect, which mixes up the low and high frequencies, the value of the exact energy in each group is not known. Since the method does not converge as it is, we could think of overrelaxing or underrelaxing the values of energy, but the preliminary tests we performed did not succeed either. This is why we get into these difficulties and why we propose in what follows a different strategy.

4.4. Results for the plain parareal in time algorithm with projection and damping. We note that, for this nonlinear problem, the projection allows us to stabilize the parareal in time algorithm, but a little more is required. We have chosen to damp the high frequencies at each coarse step before proceeding to the next parareal iteration. This means that we introduce a damping function Φ that to any element v_N in S_N provides an element $w_N = \Phi(v_N)$ in S_N such that

$$\forall \ell \in G_1, \quad \hat{w}_\ell = \hat{v}_\ell, \tag{4.15}$$

$$\forall \ell \in G_i, i = 2, \ldots, I, \quad \hat{w}_\ell = f(i)\hat{v}_\ell, \tag{4.16}$$

where $f(i)$ is chosen such that high-frequency parts are more and more filtered. In all the following results, we choose $f(i) = \frac{1}{i^2+1.0}$, except for two groups, where we have chosen $f(i) = \frac{1}{3i+1.0}$. With the same definition of the multigroup manifold, the parareal scheme with projection and damping reads

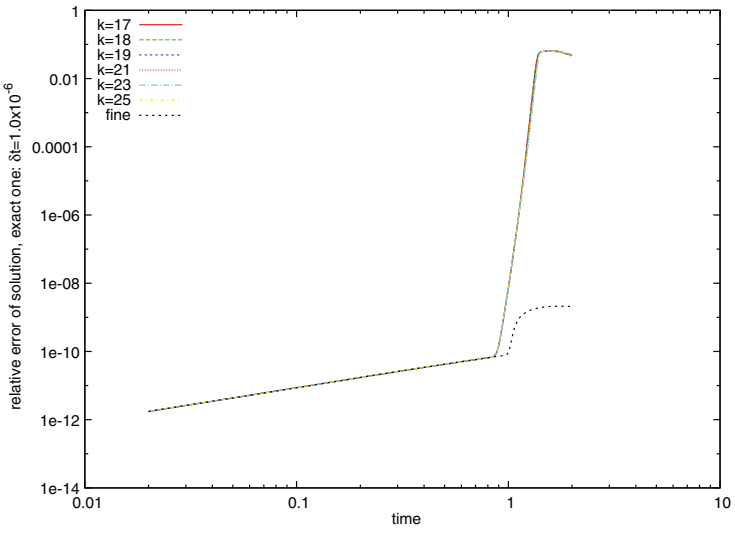


FIG. 4.7. *Case 2: relative error of the solution obtained by the parareal method with projection to two groups* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 7, \ldots, 25$.

$$
\begin{cases}
u_{n+1}^0 = \mathcal{G}_{\Delta T}(u_n^0), \\
\tilde{u}_{n+1}^{k+1} = \mathcal{G}_{\Delta T}(\bar{u}_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(\bar{u}_n^k), \\
u_{n+1}^{k+1} = \pi_{[\mathcal{M}^I]_n^{k+1}}(\tilde{u}_{n+1}^{k+1}), \\
\bar{u}_{n+1}^{k+1} = \Phi(u_{n+1}^{k+1}).
\end{cases} \tag{4.17}
$$

Figures 4.8–4.10 present the results of this improved parareal method. We want to emphasize that the parareal method is now converging.

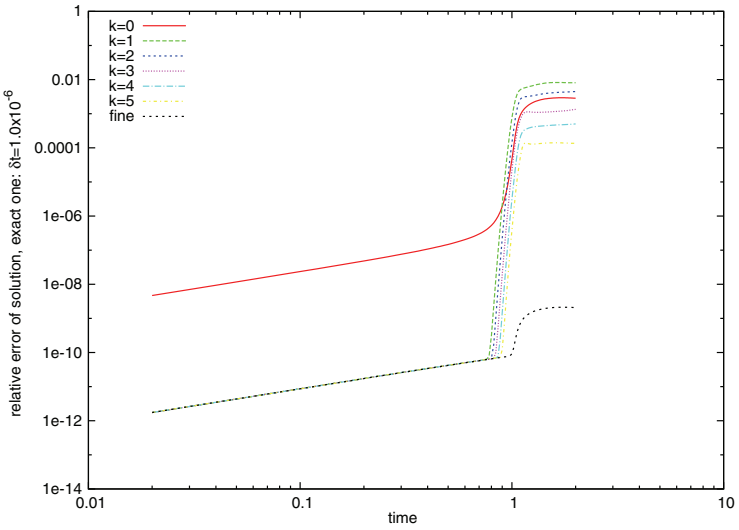


FIG. 4.8. *Case 2-2: relative error of the solution obtained by the parareal method with projection to two groups with damping* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 0, \ldots, 5$.

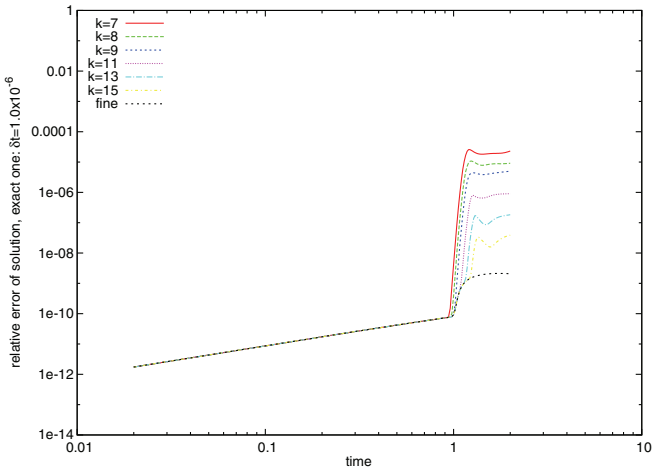


FIG. 4.9. *Case* 2-2*: relative error of the solution obtained by the parareal method with projection to two groups with damping* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 7, \ldots, 15$.

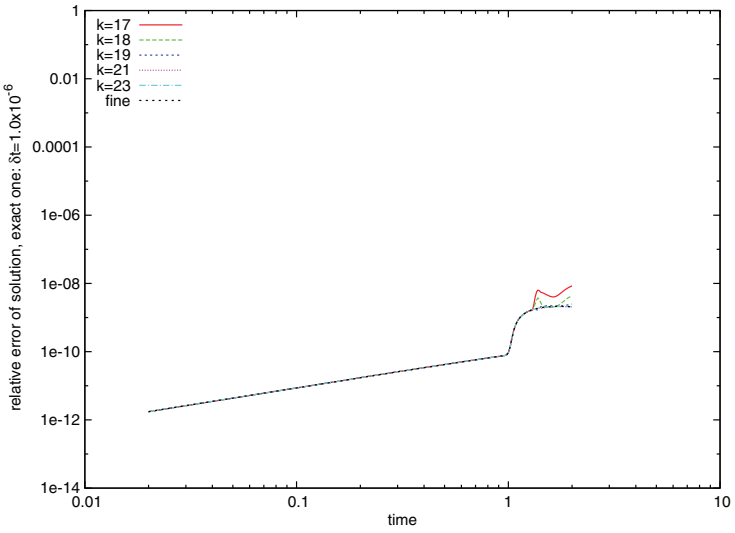


FIG. 4.10. *Case* 2-2*: relative error of the solution obtained by the parareal method with projection to two groups with damping* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 17, \ldots, 25$.

Remark 5. Note that the strategy used here consisting of adding a little bit of dissipation on the high modes to the parareal scheme goes in the direction suggested in [2] and is coherent with the spectral viscosity proposed in [23]. Note also that the solution should be measured after the projection and before the damping. Note finally that the fine propagator acts on the undamped solution so that the dissipativity that is added does not affect the quality of the solution (e.g., as in any parareal scheme, after N iterations, the solution is the same as the one for the plain fine propagator) but improves the quality of the scheme.

The projection with more than two groups performs slightly better; see, e.g., Figures 4.11 and 4.12, where only the results for iterations between 7 and 15 are

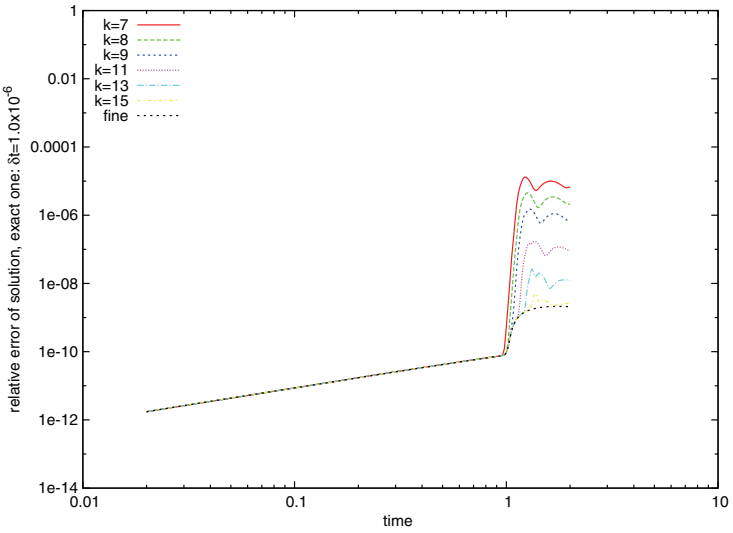


FIG. 4.11. *Case* 2-2*: relative error of the solution obtained by the parareal method with projection to* 10 *groups with damping* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 7, \ldots, 15$.

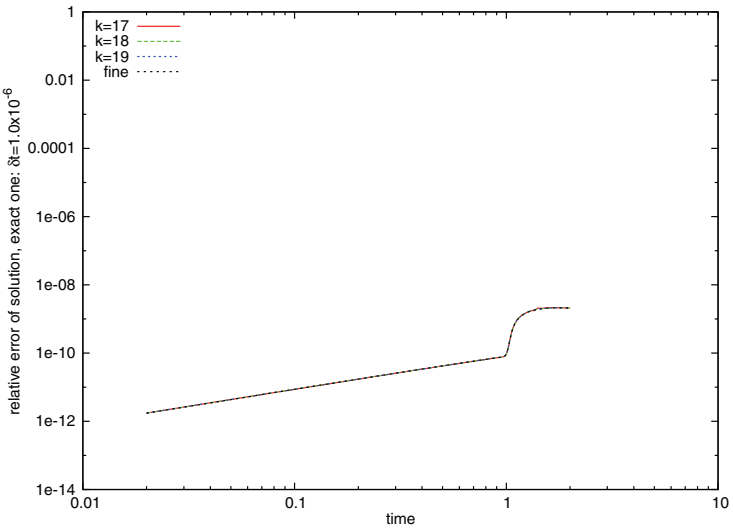


FIG. 4.12. *Case* 2-2*: relative error of the solution obtained by the parareal method with projection to* 10 *groups with damping* ($\delta t = 1.0 \times 10^{-5}, dT = 1.0 \times 10^{-2}, \Delta T = 2.0 \times 10^{-2}$) *for different values of the iterations:* $k = 17, \ldots, 25$.

presented. It may be interesting to keep this in mind even though the separation in more than two groups is not completely straightforward in the nonperiodic case where finite difference or finite volume techniques will be used.

5. Concluding remarks. In this paper we have analyzed the lack of robustness of the parareal in time algorithm applied to hyperbolic systems or convection-diffusion problems with small diffusion. This bad feature, leading sometimes to instabilities of the algorithm before convergence, is a consequence of the loss of regularity that the

solution develops when time runs. Based on this analysis, we have proposed a cure for this lack of robustness. The cure is inherited from the extension of the parareal in time algorithm for Hamiltonian systems and consists of projecting the solution over an energy manifold where the exact solution lies, which is defined on the fly and improved iteratively. This simple procedure allows us to stabilizes the algorithm, which now converges after some iterations. It is also quite general, and we do not see difficulties in the extension of the approach to other hyperbolic systems, other discretizations, or other boundary conditions; this will be the subject of a future paper.

Let us note that, provided that the fine propagator is L^2 stable in the sense that there exists a constant $\alpha > 0$ independent of the discretization parameters such that

$$\forall v \in L^2(0,\mathbb{T}), \quad \|\mathcal{F}_{\Delta T}(v)\|_{L^2(0,\mathbb{T})} \leq (1+\alpha\Delta T)\|v\|_{L^2(0,\mathbb{T})}, \tag{5.1}$$

then it is easy to obtain by induction that

$$\begin{aligned} \|u_{n+1}^{k+1}\|_{L^2(0,\mathbb{T})} &\leq (1+\alpha\Delta T)^{n+1}\|u_0^{k+1} = u_0\|_{L^2(0,\mathbb{T})} \\ &\leq e^{\alpha T}\|u_0\|_{L^2(0,\mathbb{T})}, \end{aligned} \tag{5.2}$$

hence the stability of the approach. Note that both the wave equation and Burgers' equation (5.1) hold with $\alpha = 0$.

Finally, it should be noted that, similar to other types of equations, the convergence of the parareal in time algorithm allows one to speed up the solution procedure but is not optimal in terms of parallel efficiency, since, roughly speaking, a convergence that is achieved after 10 or 15 iterations leads to a parallel efficiency of $\frac{1}{10}$ or $\frac{1}{15}$ at best, which is far from optimal. It is not the purpose of this paper to provide a full optimal implementation of the approach. Let us just emphasize that the combination of the parareal in time algorithm with standard domain decomposition methods is a way to improve the efficiency and use more of the processors that are available on current architectures over a plain domain decomposition technique. We refer to [12] for more on this matter.

REFERENCES

[1] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah, *Parallel in time molecular dynamics simulations*, Phys. Rev. E, 66 (2002), 055701.

[2] G. Bal, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in Domain Decomposition Methods in Science and Engineering, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, J. Xu, eds., Springer-Verlag, New York, 2005, pp. 425–432.

[3] G. Bal and Y. Maday, *A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put*, in Recent Developments in Domain Decomposition Methods, Lect. Notes Comput. Sci. Eng., 23, L. F. Pavarino, and A. Toselli, eds., Springer-Verlag, New York, 2002, pp. 189–202.

[4] K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations*, Numer. Math. Sci. Comput., Oxford Sci. Publ., Clarendon Press, Oxford University Press, New York, 1995.

[5] K. Burrage, *Parallel methods for ODEs*, Adv. Comput. Math., 7 (1997), pp. 1–3.

[6] F. Chouly and M. A. Fernández, *An enhanced parareal algorithm for partitioned parabolic-hyperbolic coupling*, AIP Conf. Proc. 1168, American Institute of physics, melville, NY, 2009, pp. 1517–1520.

[7] J. Cortial and C. Farhat, *A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems*, Internat. J. Numer. Methods Engrg., 77 (2009), pp. 451–470.

[8] X. Dai, C. Le Bris, F. Legoll, and Y. Maday, *Parareal algorithms for Hamiltonian systems*, M2AN Math. Model Numer. Anal., Cambridge University Press, 2012. Available online http://dx.doi.org/10.1051/m2an/2012046.

[9] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators part* I*: Theory and feasibility studies for fluid, structure, and fluid structure applications*, Internat. J. Numer. Methods Engrg., 58 (2003), pp. 1397–1434.
[10] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, Internat. J. Numer. Methods Engrg., 67 (2006), pp. 697–724.
[11] P. F. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, in Domain Decomposition Methods in Science and Engineering, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, J. Xu, eds., Springer-Verlag, New York, 2005, pp. 433–440.
[12] R. Guetat, *Methode de parallélisation en temps : Application aux méthodes de décomposition de domaine*, Ph.D. thesis, UPMC Paris, 2011.
[13] B. Gustafsson, H. O. Kreiss, and J. Oliger, *Time Dependent Problems and Difference Methods*, Wiley, New York, 1995.
[14] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations*, Springer Series in Computational Mathematics 31, Springer-Verlag, Berlin, Heidelberg, 2002.
[15] H. Jimnez-Prez and J. Laskar, *A Time-Parallel Algorithm for Almost Integrable Hamiltonian Systems*, arXiv:1106.3694.
[16] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel,"* C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.
[17] Y. Maday, *Parareal in time algorithm for kinetic systems based on model reduction*, in High Dimensional Partial Differential Equations in Science and Engineering, CRM Proc. Lecture Notes 41, AMS, Providence, RI, 2007, pp. 183–194.
[18] Y. Maday, *The "parareal in time" algorithm*, in Sub-Structuring Techniques and Domain Decomposition Methods, Computational Science, Engineering & Technology Series 24, Saxe-Coburg Publications, Stirling, UK, 2010, pp. 19–44.
[19] Y. Maday and G. Turinici, *A parareal in time procedure for the control of partial differential equations*, C. R. Acad Sci. Paris Sér. I Math., 335 (2002), pp. 387–392.
[20] D. Mercerat, L. Guillot, and J.-P. Vilotte, *Application of the parareal algorithm for acoustic wave propagation*, AIP Conf. Proc., 1168 (2009), pp. 1251–1254.
[21] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Commun. ACM, 7 (1964), pp. 731–733.
[22] G. Staff and E. Rønquist, *Stability of the parareal algorithm, in* Domain Decomposition Methods in Science and Engineering, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, and J. Xu, eds., Lect. Notes Comput. Sci. Eng. 40, Springer-Verlag, Berlin, 2005, pp. 449–456.
[23] E. Tadmor, *Convergence of spectral methods for nonlinear conservation laws*, SIAM J. Numer. Anal., 26 (1989), pp. 30–44.