



STABLE PARAREAL IN TIME METHOD FOR FIRST AND SECOND ORDER HYPERBOLIC SYSTEM

Xiaoying Dai, Yvon Maday

► To cite this version:

Xiaoying Dai, Yvon Maday. STABLE PARAREAL IN TIME METHOD FOR FIRST AND SECOND ORDER HYPERBOLIC SYSTEM. 2011. hal-00655643

HAL Id: hal-00655643

<https://hal.science/hal-00655643>

Submitted on 1 Jan 2012

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.

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

XIAOYING DAI[†] AND YVON MADAY^{‡§¶}

Abstract. The parareal in time algorithm allows 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 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 fatal large lack of accuracy of the plain parareal in time algorithm consequence of the fact that the plain parareal in time algorithm does not conserve invariants. A somehow similar difficulty occurs for problems where the solution lacks regularity, either initially or in the evolution, like for the solution to hyperbolic system of conservation laws. In this paper we identify the problem of lack of stability of the parareal in time algorithm and propose a simple way to cure it. The new method is used to solve a linear wave equation and a non linear Burger's equation, the results illustrate the stability of this variant of the parareal in time algorithm.

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

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

1. Introduction. Parallel in time algorithms represent a new competitive way of using the ever increasing number of cores in today's supercomputers platforms in cases where classical domain decomposition methods either are of no relevance (for systems of differential equations) or in cases where the scalability of the sole domain decomposition (for evolution PDE's) gets to saturation. This kind of algorithm in the 4th direction has received too little attention in the literature in the past due to the inherent sequential nature of the time direction: it seems indeed difficult to simulate what will arrive in the far future, e.g. next week, without knowing in details what will arrive in the close future, e.g. tomorrow and each day of the current week. Due to the paramount importance of finding new ways for parallelisation, going together with the ever increasing number of processors present in the new generations of architectures, algorithms such as parareal in time, parallel deferred corrections or parallel exponential integrators have become an active research topic showing promising applicability for large-scale computations and perhaps one way to go to exascale computations. We refer e.g. to the book of K. Burrage [4] for a synthetic approach on the subject (see also [5]). In this 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 the time*.

The parareal in time algorithm pioneered in [14] extended later in [1, 3, 18] under a form much better tuned to the treatment of nonlinear problems enters in the third category above (see [20] for a first algorithm in this category). It combines two propagators, one fine and one coarse, in a predictor-corrector manner that allows to use the fine propagator in parallel on the various processors, while the coarse

*This work was supported by ANR-06-CIS6-007.

[†]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

[§]CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

[¶]Division of Applied Mathematics, Brown University, Providence, RI, USA

propagator is solved in a sequential manner. The algorithm provides good, provable convergence, for diffusive PDEs such as parabolic type problem. However, several studies have shown some instabilities for hyperbolic equations [8, 11, 19, 6] and the numerical analysis performed on simple examples or partial differential equations in [3, 2, 21] allows us to understand this behavior.

A challenging field of application for parallel in time propagators is for long time simulation of Hamiltonian systems. Here also the plain method lacks the geometric properties that are essential to guarantee the quality of the approximation. Let us remind that these geometric properties lead to the convergence of the algorithm towards the solution of the original systems on long time simulations. These properties —either the symplecticity or the symmetry of the scheme — are not preserved for the parareal scheme even if the basics coarse and fine propagators are symplectic or symmetric. A cure to this lack of geometric properties has been proposed in [7], based on two ingredients: the first one is the symmetrization of the parareal in time algorithm that leads to a new multistep schemes. This ingredient 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 [7]; they prevent the symmetric variant of behaving well. The second ingredient that has been introduced in [7] consists simply, after each correction of the predictor corrector scheme, to project the solution coming out the parareal in time algorithm, on 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).

Following this second idea advocated to stabilize the parareal in time algorithm for Hamiltonian system, we propose in this paper an extension of it allowing to cure the lack of stability that occurs when solving hyperbolic type problem. Another strategy, based on providing a much more involved correction procedure re-utilizing previously computed information has been proposed in [9, 10] and already improved the performance of the parareal in time algorithm for second-order hyperbolic systems. The method proposed here seems much more simple to implement and more general in scope and applications.

The remainder of this paper is organized as follows. In Section 2, we remind the basics on the plain parareal in time algorithm including the current state of the art on the stability and convergence analysis of the algorithm; we use this 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 details and the numerical simulation illustrates the good properties of the variant. In Section 4 we address the case of the Burgers' equation, with small viscosity, here again the variant performs well. In Section 5 we present some elements of numerical analysis proving the stability of the proposed variant of the parareal scheme and finally some conclusions are drawn in Section 6.

2. Some elements on the parareal in time algorithm.

2.1. Basics on the parareal in time algorithm. Let us consider a possibly non-linear time dependent partial differential equation of the form

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

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 exact propagator. We note that, with obvious notations, we have

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

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 to consider snapshots of the solution $y(T_n)$, then, with notations similar as above (2.2), 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 that allows 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 $dT, \delta t \ll dT \leq \Delta T$. 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 spacial discretization can be based on a finite element method, the coarse solver \mathcal{G} can then be based both on a coarser time step and a coarser spacial discretization, we refer to [11] and [17] where such a strategy is followed. Another possibility consists in providing \mathcal{F} with all the physics of the phenomenon while \mathcal{G} will be based on a simplified physics (see eg [16] for an example).

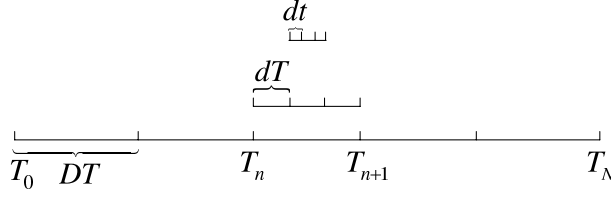
Assume \mathcal{F} and \mathcal{G} are given, we can proceed to the definition of the sequence. Let us initiate \mathbf{Y}^0 by

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

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

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

We first comment on the parallelization of the above algorithm. The initialization step (2.3) is sequential but involves the 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 again (and now in parallel) the coarse solver 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, any way 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 on the stability and convergence of the parareal in time algorithm that have been 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 $j, 1 \leq j \leq J$

$$\|\mathcal{E}_0^t(y_0)\|_{B_j} \leq C\|y_0\|_{B_j} \quad (2.5)$$

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 $j, 1 \leq j \leq J$

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

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

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

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

where η and ε are small quantities, typically, for the Euler scheme these are $\eta = \delta t$ and $\varepsilon = \Delta T$ and C is a generic constant, independent of the quantities on the right hand side of the previous inequalities and on $j \leq J$, but may depend on J .

REMARK 1. For Euler schemes based on a time step δt for \mathcal{F} and ΔT \mathcal{G} , hypothesis (2.7) and (2.8) hold with $\eta = \delta t$ and $\varepsilon = \Delta T$ respectively.

The convergence is stated in the following (see theorem 1 in [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 $k \geq 0$:

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

Proof: The proof is done by induction over $j \geq 0$ for the statement

$$\forall i \geq 0, \quad \|Y_n^j - y(T^n)\|_{B_i} \leq C(\varepsilon^{j+1} + \eta)(1 + \|y_0\|_{B_{i+j+1}}), \quad (2.10)$$

which is obvious for $j = 0$ since it corresponds to the analysis of the plain coarse operator. Let us assume that it is true for j and let us prove it for $j + 1$. From the definition of Y_{n+1}^{j+1} , we write

$$\begin{aligned} Y_{n+1}^{j+1} - y(T^{n+1}) &= \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{j+1}) + \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^j) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^j) - \mathcal{E}_{T_n}^{T_{n+1}}[y(T^n)] \\ &= \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{j+1}) - \mathcal{G}_{T_n}^{T_{n+1}}(y(T^n)) \\ &\quad + [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{G}_{T_n}^{T_{n+1}}](Y_n^j) - [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{G}_{T_n}^{T_{n+1}}](y(T^n)) \\ &\quad - [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{F}_{T_n}^{T_{n+1}}](Y_n^j) + [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{F}_{T_n}^{T_{n+1}}](y(T^n)) \\ &\quad - [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{F}_{T_n}^{T_{n+1}}](y(T^n)) \end{aligned}$$

or again with the notations introduced above

$$\begin{aligned} Y_{n+1}^{j+1} - y(T^{n+1}) &= \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{j+1}) - \mathcal{G}_{T_n}^{T_{n+1}}(y(T^n)) \\ &\quad + \delta \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^j) - \delta \mathcal{G}_{T_n}^{T_{n+1}}(y(T^n)) \\ &\quad - \delta \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^j) + \delta \mathcal{F}_{T_n}^{T_{n+1}}(y(T^n)) \\ &\quad - [\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{F}_{T_n}^{T_{n+1}}](y(T^n)) \end{aligned}$$

hence by triangular inequality

$$\begin{aligned} \|Y_{n+1}^{j+1} - y(T^{n+1})\|_{B_i} &\leq \|\mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{j+1}) - \mathcal{G}_{T_n}^{T_{n+1}}(y(T^n))\|_{B_i} \\ &\quad + \|\delta \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^j) - \delta \mathcal{G}_{T_n}^{T_{n+1}}(y(T^n))\|_{B_i} \\ &\quad + \|\delta \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^j) + \delta \mathcal{F}_{T_n}^{T_{n+1}}(y(T^n))\|_{B_i} \\ &\quad + \|[\mathcal{E}_{T_n}^{T_{n+1}} - \mathcal{F}_{T_n}^{T_{n+1}}](y(T^n))\|_{B_i} \end{aligned}$$

that from hypothesis (2.6), (2.7) and (2.8) can be upper bounded as follows

$$\begin{aligned} \|Y_{n+1}^{j+1} - y(T^{n+1})\|_{B_i} &\leq (1 + C\Delta T)\|Y_n^{j+1} - y(T^n)\|_{B_i} \\ &\quad + C\Delta T\varepsilon\|Y_n^j - y(T^n)\|_{B_{i+1}} \\ &\quad + C\Delta T\eta\|Y_n^j - y(T^n)\|_{B_{i+1}} + C\Delta T\eta\|y(T^n)\|_{B_{i+1}} \end{aligned}$$

the stability hypothesis (2.5) together with the induction hypothesis (2.10) allow to get

$$\begin{aligned} \|Y_{n+1}^{j+1} - y(T^{n+1})\|_{B_i} &\leq (1 + C\Delta T)\|Y_n^{j+1} - y(T^n)\|_{B_i} \\ &\quad + C\Delta T(\varepsilon + \eta)(\varepsilon^{j+1} + \eta)\|y_0\|_{B_{i+j+2}} + C\Delta T\eta\|y_0\|_{B_{i+1}} \\ &\leq (1 + C\Delta T)\|Y_n^{j+1} - y(T^n)\|_{B_i} \\ &\quad + C\Delta T(\varepsilon^{j+2} + \eta)\|y_0\|_{B_{i+j+2}} \end{aligned}$$

from which we get (2.10) for $j + 1$ from the discrete Gronwall lemma, and the proof is complete.

It is interesting to notice two things at this level. First 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. Second, the convergence results implies, of course, stability. Hence, provided that the initial condition is regular enough and remains regular the parareal in time algorithm is stable.

The previous remark explains actually the problem of the use of the parareal scheme when non dissipative PDE or PDE with small dissipation are simulated. This is the case for small viscosity parabolic problem as is explained in the next subsection.

This is also the case for hyperbolic systems. We can illustrate this statement on two equations: the periodic wave equation and the periodic Burger's equation for which, starting with regular enough initial condition solves or corrects the instability of the parareal in time algorithm in this situation. This allows to rectify somehow the statements that were done before in e.g. [10] where the stability problem occurs for second-order hyperbolic problems and not for parabolic and first-order hyperbolic problems. The problem is not really on the type of the equations (and certainly not on the order of the equations) but on the regularity of the solution, including the regularity of the initial condition.

2.3. Behavior on a parabolic case. We have performed a series of simple test on the linear parabolic equation, provided with periodic boundary conditions

$$\begin{cases} \frac{\partial u}{\partial t} - \mu \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} &= 0, \\ u(x, 0) &= u_0(x), \\ u(x, t) &= u(x + 2\pi, t), \end{cases} \quad (2.11)$$

in the case where the viscosity μ is positive yet very small $\mu = 10^{-10}$. We have chosen a standard spectral Fourier approximation in space (truncated series with $N = 256$ and chosen a second order in time implicit Euler scheme for a propagation over $[0, T = 2]$. The parareal in time algorithm is implemented with $\Delta T = 2 \cdot 10^{-2}$, $dT = 10^{-2}$ and $\delta t = 10^{-4}$.

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

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

In the case of low regularity: $p = 0.5$ or $p = 1$, the relative error after 15 parareal iterations is about 100! There is no blow up, the iterations can continue, the error eventually going to zero. For $p = 4$ or even better $p = 10$, the convergence is achieved after very few iterations, as is expected from Theorem 2.1.

These simple computations provide a preliminary illustration that the behavior of the parareal in time algorithm is deeply linked to the regularity of the solution.

3. The periodic wave equation.

3.1. The basic Fourier discretization. We consider the following one dimensional wave equation with periodic boundary conditions: Find u such that

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} &= 0, \\ u(x, 0) &= f(x), \quad \frac{\partial u}{\partial t}(x, 0) = g(x), \\ u(x, t) &= u(x + \mathbb{T}, t), \quad \frac{\partial u}{\partial x}(x, t) = \frac{\partial u}{\partial x}(x + \mathbb{T}, t) \end{cases} \quad (3.1)$$

where f and g are periodic given initial condition, c stands for the speed of the wave, and $\mathbb{T} \in \mathbb{R}^+$ represents the period.

The weak form of equation (3.1) is as follows: Find u such that $\forall t > 0, u(., t) \in S = \{v \in H^1(0, \mathbb{T}), v(0) = v(\mathbb{T})\}$

$$\left(\frac{\partial^2 u}{\partial t^2}, v\right) + c^2 \left(\frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}\right) = 0, \quad \forall v \in S, \quad (3.2)$$

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

We use now the Fourier spectral method to semi discretize in space problem (3.2). That is, we choose

$$S_N = \text{span}\{e^{i\ell \frac{2\pi}{\mathbb{T}} x}, \ell = -N, \dots, N\}, \quad (3.3)$$

and approximate $u(x, t)$ by

$$u_N(x, t) = \sum_{\ell=-N}^N \hat{u}_\ell(t) e^{i\ell \frac{2\pi}{\mathbb{T}} x}. \quad (3.4)$$

Note that $\hat{u}_\ell(t) = \frac{1}{\mathbb{T}}(u_N(x, t), e^{i\ell \frac{2\pi}{\mathbb{T}} x})$, as results from the equality $(e^{i\ell \frac{2\pi}{\mathbb{T}} x}, e^{ik \frac{2\pi}{\mathbb{T}} x}) = \mathbb{T} \delta_{\ell, k}$ where $\delta_{\ell, k}$ denotes the standard Kronecker symbol.

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

$$\left(\frac{\partial^2 u_N}{\partial t^2}, v_N\right) + c^2 \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0, \quad \forall v_N \in S_N. \quad (3.5)$$

As is classical, this Galerkin approximation reduces to a system of differential equations in the Fourier coefficients $\hat{u}_k(t)$. More precisely, let us choose iteratively $v_N = e^{ik \frac{2\pi}{\mathbb{T}} x}$, $k \in [-N, N]$ in (3.5). We then obtain the following system of independent equations

$$\mathbb{T} \frac{\partial^2 \hat{u}_\ell}{\partial t^2}(t) + \frac{(2\pi)^2}{\mathbb{T}} \ell^2 c^2 \hat{u}_\ell(t) = 0, \quad \forall \ell \in [-N, N], \quad (3.6)$$

that can be exactly solved. After resolution, we know all the coefficients $\hat{u}_\ell(t)$ in a closed form, then from equation (3.4) we get the Galerkin approximation of $u(x, t)$ in S_N .

Next, using respectively $v = \frac{\partial u}{\partial t}$ in (3.2) and $v_N = \frac{\partial u_N}{\partial t}$ in (3.5) allows to derive, for any time t

$$\left\| \frac{\partial u}{\partial t}(., t) \right\|_{L^2(0, \mathbb{T})}^2 + c^2 \left\| \frac{\partial u}{\partial x}(., t) \right\|_{L^2(0, \mathbb{T})}^2 = \left\| \frac{\partial u}{\partial t}(., 0) \right\|_{L^2(0, \mathbb{T})}^2 + c^2 \left\| \frac{\partial u}{\partial x}(., 0) \right\|_{L^2(0, \mathbb{T})}^2 \quad (3.7)$$

and

$$\left\| \frac{\partial u_N}{\partial t}(., t) \right\|_{L^2(0, \mathbb{T})}^2 + c^2 \left\| \frac{\partial u_N}{\partial x}(., t) \right\|_{L^2(0, \mathbb{T})}^2 = \left\| \frac{\partial u_N}{\partial t}(., 0) \right\|_{L^2(0, \mathbb{T})}^2 + c^2 \left\| \frac{\partial u_N}{\partial x}(., 0) \right\|_{L^2(0, \mathbb{T})}^2 \quad (3.8)$$

respectively. If we now introduce the following Hamiltonian

$$H(u_N)(t) = \left\| \frac{\partial u_N}{\partial t}(., t) \right\|_{L^2(0, \mathbb{T})}^2 + c^2 \left\| \frac{\partial u_N}{\partial x}(., t) \right\|_{L^2(0, \mathbb{T})}^2 \quad (3.9)$$

The conservation (3.8) reads for any time t

$$H(u_N)(t) = H(u_N)(0) \quad (3.10)$$

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

REMARK 2. *As is often the case for Hamiltonian system, 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 on (3.6) that by defining*

$$h_\ell(v) = \left| \frac{\partial v}{\partial t} \right|^2 + \left(\frac{2\pi}{\mathbb{T}} \right)^2 \ell^2 c^2 |v|^2, \quad (3.11)$$

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

$$\forall t, \quad h_\ell(\hat{u}_\ell)(t) = h_\ell(\hat{u}_\ell)(0) \quad (3.12)$$

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

3.2. Illustration of the stability and instabilities of the plain parareal in time algorithm. In all what 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 by u_n the approximation (in time) of the value of $u_N(T_n)$ (hence the full approximation of $u(T_n)$). In order to test the parareal in time algorithm on this simple second order equation, we further discretize the semi discrete problem (3.5) by adding a time discretization, that is based, both for the coarse $\mathcal{G}_{\Delta T}$ and the fine $\mathcal{F}_{\Delta T}$ propagators, on the velocity Verlet scheme which is a second order in time, stable and consistent with the preservation of invariants (the wave equation being an Hamiltonian system, the symplectic nature of the velocity Verlet scheme makes it a method of choice).

The plain parareal scheme reads:

$$\begin{cases} u_{n+1}^0 &= \mathcal{G}_{\Delta T}(u_n^0) \\ u_{n+1}^{k+1} &= \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k). \end{cases} \quad (3.13)$$

We perform simulations with the discretization parameters equal to

1. $T = 100, \Delta T = 2.10^{-1}, dT = 4.10^{-3}, \delta t = 2.10^{-6}$.
2. $N = 30$.

for different initial conditions of different regularities: $u_0(x) = \sum_{\ell \in \mathbb{Z}} \hat{u}_\ell e^{i\ell \frac{2\pi}{\mathbb{T}} x}$ with

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

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

3.3. Results with the plain parareal in time algorithm on a realistic test problem. In this example, the excitation is provided by a Ricker pulse, which is analytically defined as follows

1. $\mathbb{T} = 5000$ m, $c = 2000$ m/s
2. and 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 numerical results are obtained with a coarse propagator \mathcal{G} and fine propagator \mathcal{F} that differ only with time step dT and δt , respectively :

1. $T = 100, \Delta T = 2.10^{-1}, dT = 4.10^{-3}, \delta t = 2.10^{-6}$.
2. $N = 30$.

The plot showing the evolution in time of the solution obtained after 15 iterations at different positions in the periodic space $[0, T]$ on figure 3.1 reveals an instability visible starting at time $t = 80$. The situation is even worse after 25 iterations. Actually this is one case that illustrates the bad behavior of the parareal in time algorithm for this hyperbolic equation. Note however that for many other choices of the discretization parameters δt or dT or even N but the very same initial condition, the simulation is perfectly fine, stable and convergent. The instability is thus not systematic, and changing slightly the discretization parameters, is often enough to make the scheme work. The plain parareal in time algorithm does lack of robustness, and this is what we want to correct in what follows.

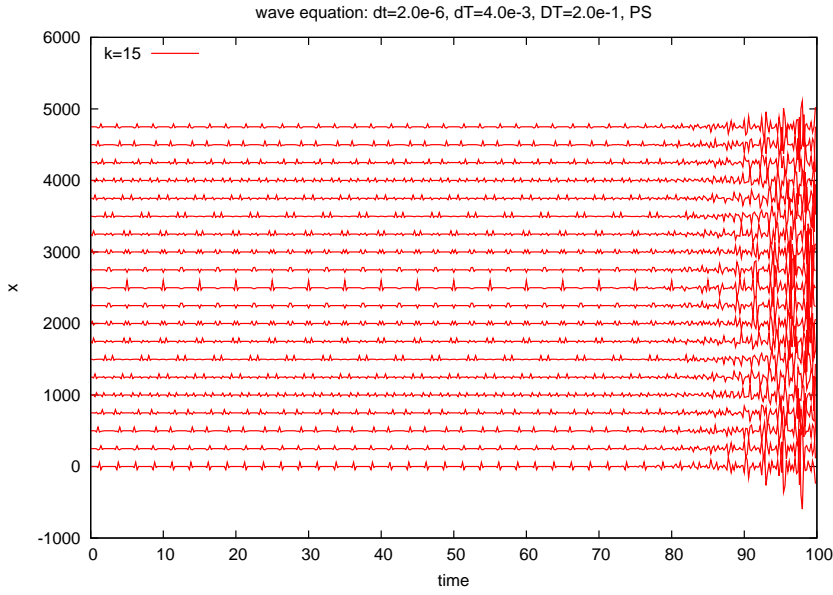


FIG. 3.1. *Vibration at different positions obtained by the parareal method ($\delta t = 2.10^{-6}, dT = 4.10^{-3}, \Delta T = 2.10^{-1}$) revealing the instabilities.*

The exact solution shown on the next plot (figure 3.2) illustrates the differences.

The plots of the error between the exact solution and the solution obtained after k iterations, with $k = 1, \dots, 15$ (see figures 3.3 and 3.4) illustrate clearly that the solution obtained by the parareal in time algorithm

1. converges for short time, e.g. only 7 iterations are sufficient to recover the solution obtained with the sole sequential fine propagator at time equal to 10.
2. does not converge for longer propagations, here for a time equal to 100
3. During the first 5 to 6 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 largely to values that are order 1 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.

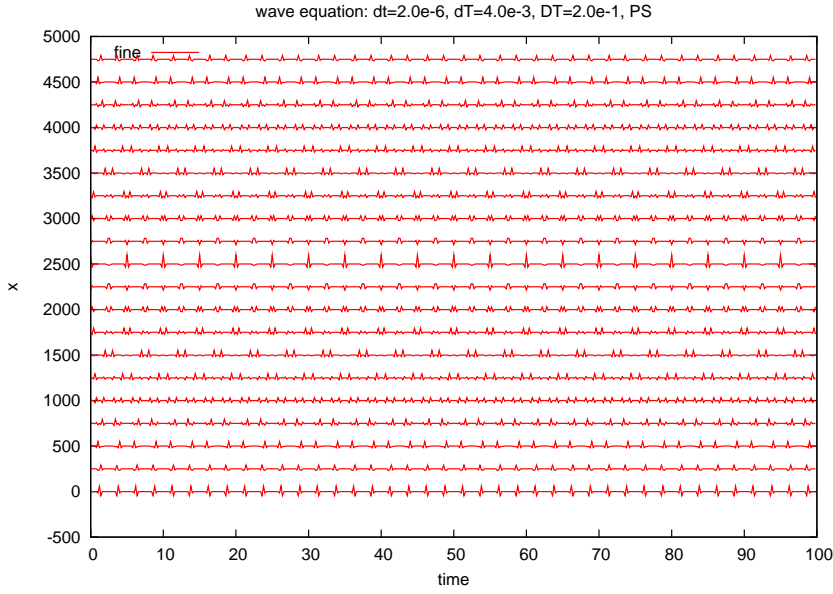


FIG. 3.2. *Vibration at different positions obtained by the fine sequential method ($\delta t = 2.10^{-6}$)*

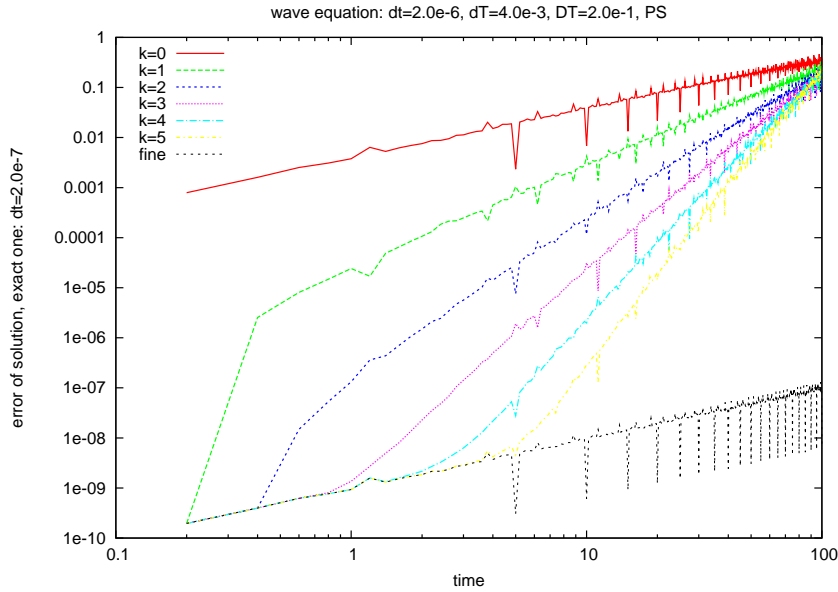


FIG. 3.3. *Error on the solution obtained by the parareal method ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)*

Taking into account the variants of the parareal in time algorithm introduced in [7] to simulate Hamiltonian systems we infer that the bad behavior of the parareal scheme comes from the non preservation of the invariant quantities. Let us for example observe the evolution of the value of the quantity H defined in (3.9)

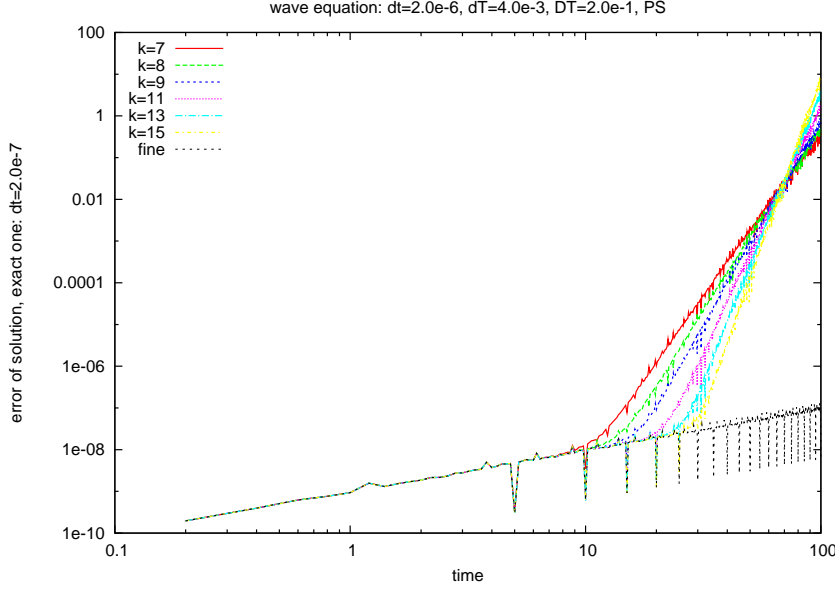


FIG. 3.4. Error on the solution obtained by the parareal method ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

The figures 3.5 and 3.6 illustrate the lack of conservation, that is coherent with the divergence of the scheme in this case.

Finally we want to notice 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.

3.4. Parareal algorithm with projection. Let us define the manifold \mathcal{M} corresponding to the set of all functions v such that $H(v) = H(u(0))$. The Standard Projection Method on \mathcal{M} (see chapter 4 of [13]) can be written as follows: For a given \tilde{u} , solve the system in (u, λ)

$$u = \tilde{u} + \lambda \nabla H(\tilde{u}),$$

where $\lambda \in \mathbb{R}$ is such that $H(u) = H(u(0))$, i.e.

$$H(\tilde{u} + \lambda \nabla H(\tilde{u})) = H(u(0)). \quad (3.14)$$

We then set

$$u = \pi_{\mathcal{M}}(\tilde{u}) \quad (3.15)$$

In the general case, equation (3.14) 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 notice and take into account that equation (3.14) is a second order polynomial in λ that can be solved exactly.

Then, combining the plain parareal scheme (3.13) with the projection method

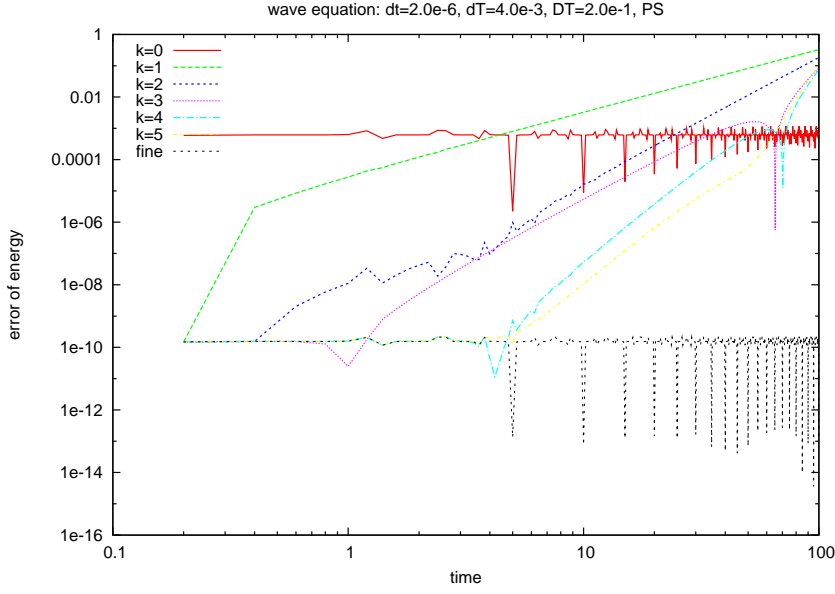


FIG. 3.5. Error on the energy obtained by the parareal method ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

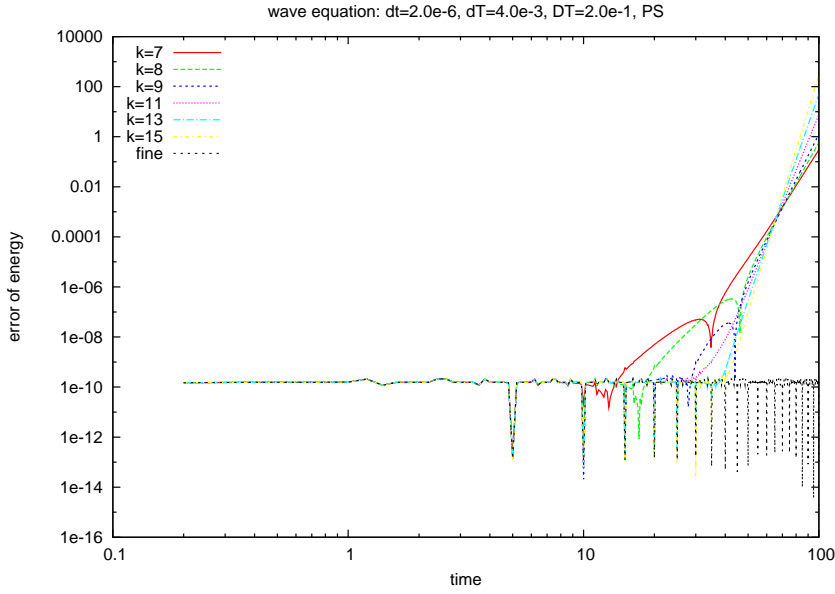


FIG. 3.6. Error on the energy obtained by the parareal method ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

above, we obtain the parareal method with projection to \mathcal{M} :

$$\begin{cases} u_{n+1}^0 &= \mathcal{G}_{\Delta T}(u_n^0) \\ \tilde{u}_{n+1}^{k+1} &= \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k), \\ u_{n+1}^{k+1} &= \pi_{\mathcal{M}}(\tilde{u}_{n+1}^{k+1}). \end{cases} \quad (3.16)$$

We now show the results obtained by parareal method with projection. We first note on figure 3.7 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.

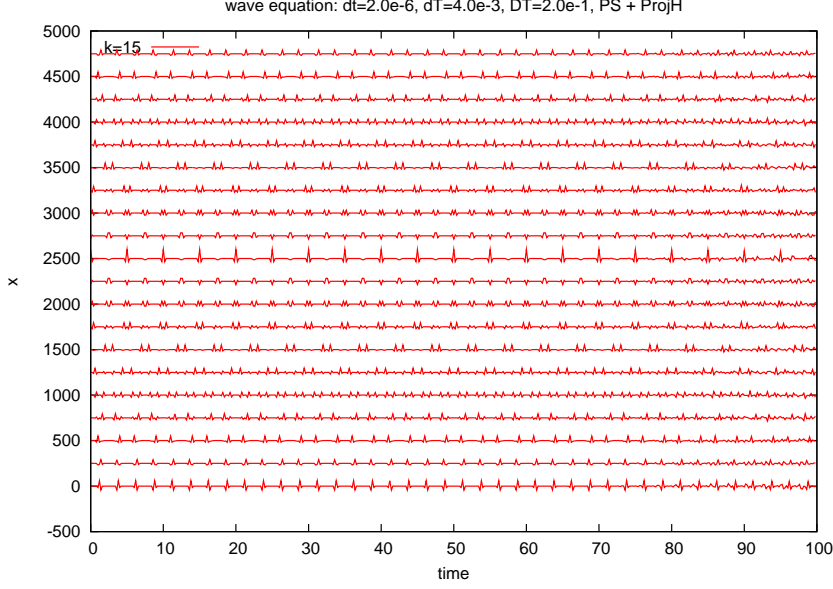


FIG. 3.7. *Vibration at different positions obtained by the parareal method with projection to the manifold \mathcal{M} ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)*

This is confirmed by the figures 3.8 and 3.9 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 worse of the same order as the error of the coarse propagator.

Of course this is far from being what is expected, but the conservation of H does improve the behavior of the algorithm. The following figures 3.10 and 3.11 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 altogether, with a unique Lagrange multiplier. 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 have defined 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 contains the modes $|k| \geq K_0$, the other one contains all the modes $|k| < K_0$, we then perform a projection for each group separately to make the energy of each group conserved.

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. If we use the same Lagrange multiplier λ to correct

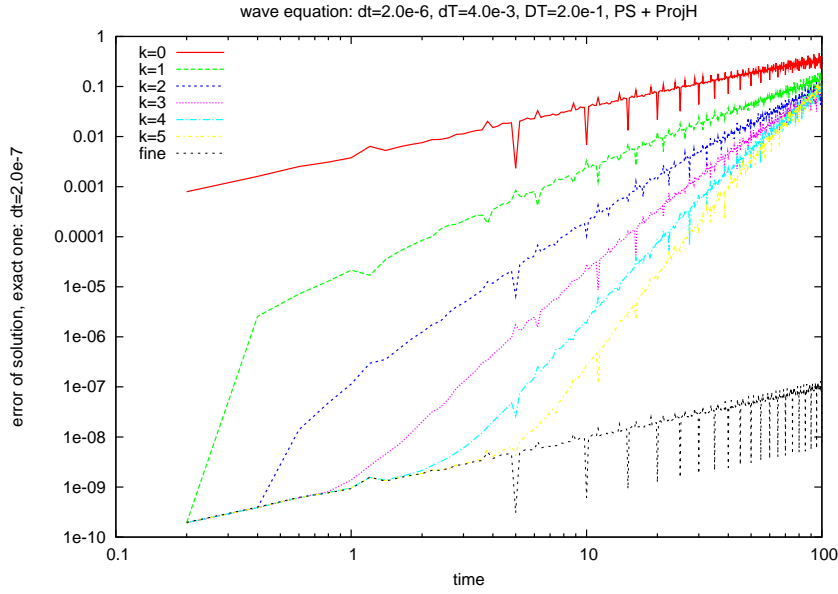


FIG. 3.8. Error on the solution obtained by the parareal method with projection to the manifold \mathcal{M} ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

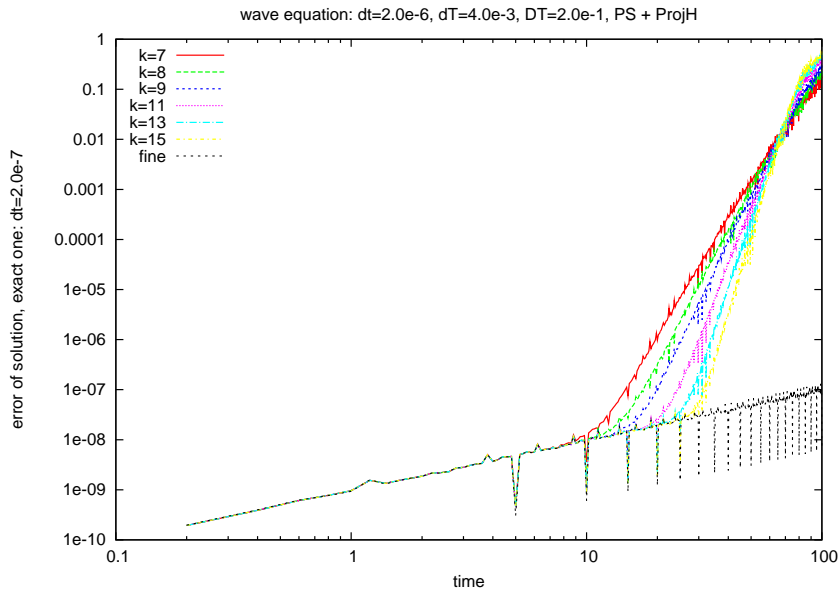


FIG. 3.9. Error on the solution obtained by parareal method with projection to the manifold \mathcal{M} ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

all the modes, the contribution of the high frequency modes may be to magnify them and pollute the solution.

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

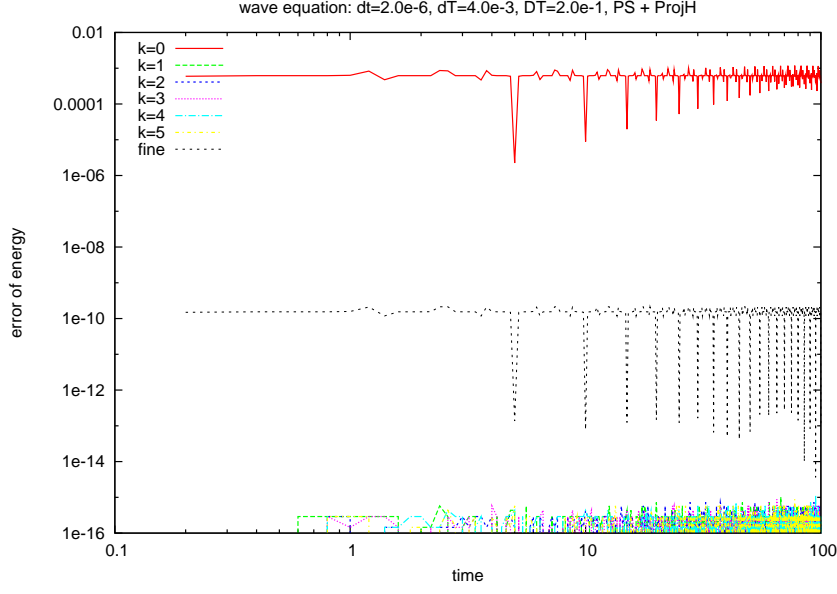


FIG. 3.10. *Error on the energy obtained by parareal method with projection to the manifold \mathcal{M} ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)*

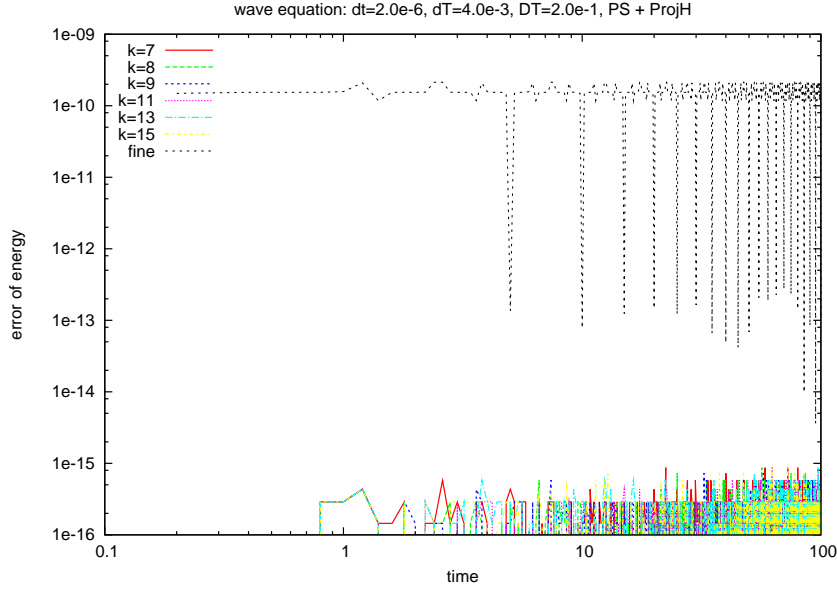


FIG. 3.11. *Error on energy obtained by parareal method with projection to the manifold \mathcal{M} ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)*

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 on figures 3.12, 3.13 and 3.14.

The solution itself, after 15 iterations, is in excellent accordance with the very

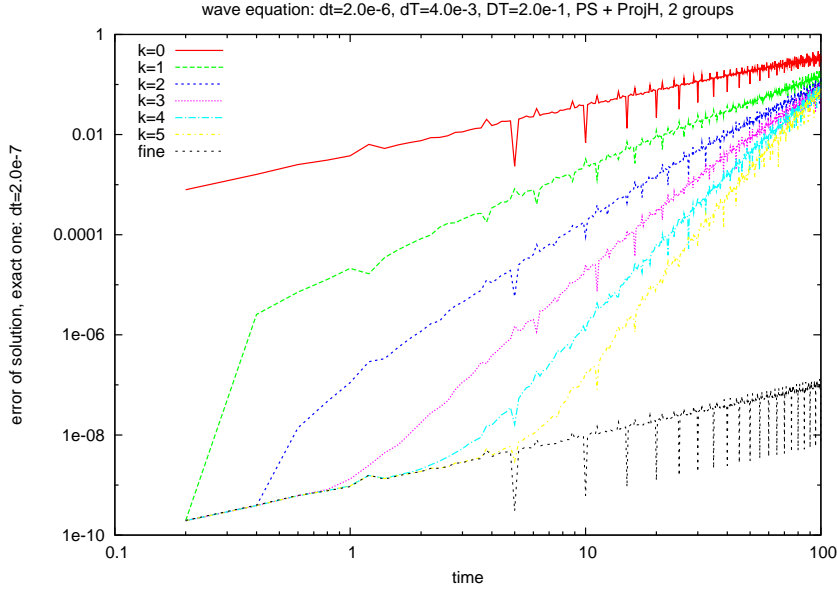


FIG. 3.12. Error on the solution obtained by the parareal with projection on two groups ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

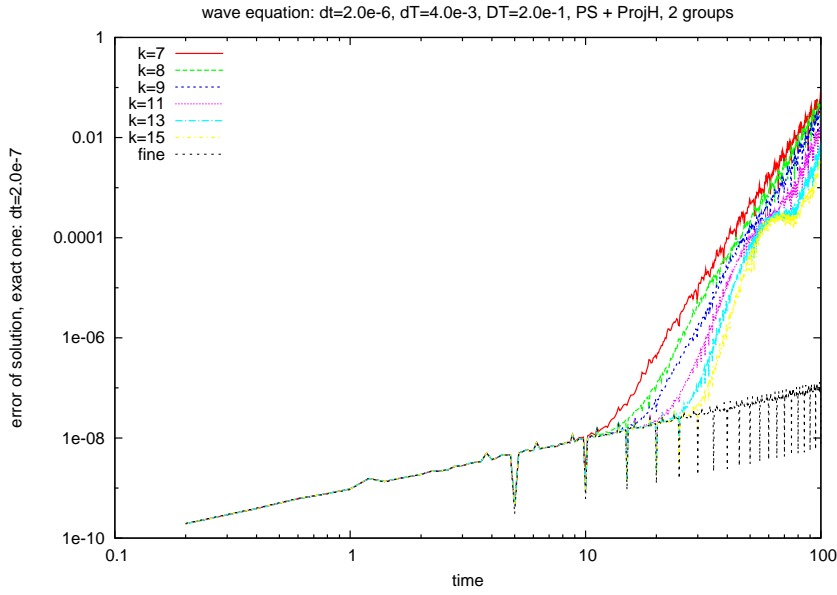


FIG. 3.13. Error on the solution obtained by the parareal with projection on two groups ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

fine approximation as can be seen on figure 3.15.

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 groups: the low modes and high frequency modes, is thus a sufficiently mild ingredi-

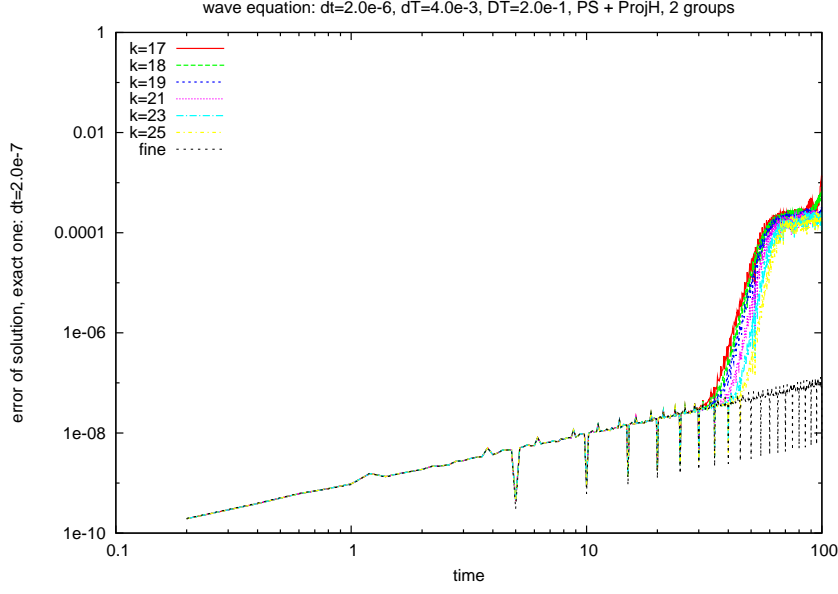


FIG. 3.14. Error on the solution obtained by the parareal with projection on two groups ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

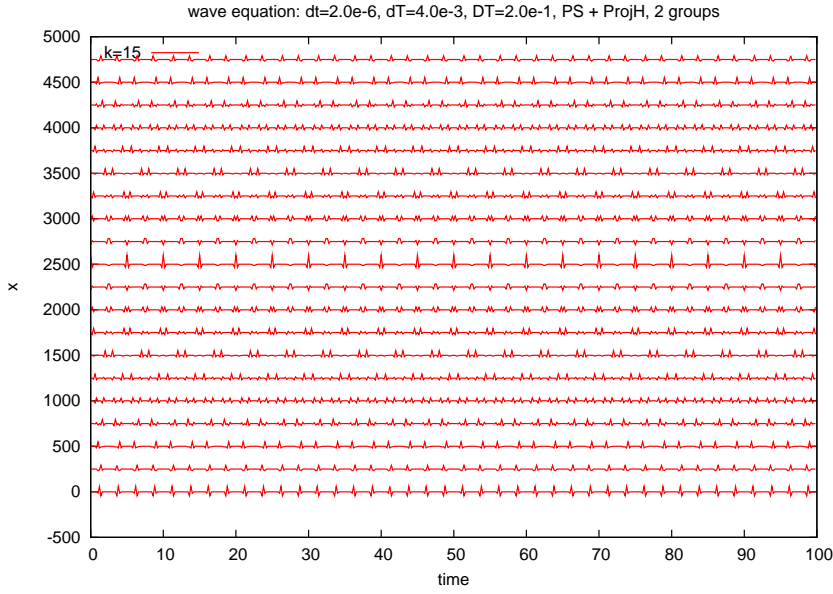


FIG. 3.15. Vibration at different positions obtained by parareal method with projection on two groups ($\delta t = 2.10^{-6}$, $dT = 4.10^{-3}$, $\Delta T = 2.10^{-1}$)

ent 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 when the method converges, i.e. $\forall n = 1, \dots, N$, $u_n^k \rightarrow u_n^\infty$, this limit

solution satisfies thanks to (3.16)

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

that, here, due to the fact that the propagator $\mathcal{F}_{\Delta T}$ is symplectic and thus preserves the Hamiltonian, reads

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

that proves that the method converges towards 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

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} + \frac{1}{2} \frac{\partial(u^2)}{\partial x} &= 0, \\ u(x, 0) &= u_0(x), \\ u(x + \mathbb{T}, t) &= u(x, t), \end{aligned} \quad (4.1)$$

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

Taking into account that the periodic frame of the problem allows to use Fourier spectral approximation, we thus consider a semi discretization based on the Galerkin Fourier approximation in S_N

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) + \frac{1}{2} \left(\frac{\partial(u_N)^2}{\partial x}, v_N\right) = 0, \forall v_N \in S_N. \quad (4.2)$$

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 only difference between the coarse propagator \mathcal{G} and fine propagator \mathcal{F} are relative to the time steps dT and fine time step δt , respectively. Denoting by δ either dT or δt , the splitting schemes at each iteration proceeds as follows:

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

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0, \forall v_N \in S_N \quad (4.3)$$

$$u_N(\tau_n) = u_{N,n}, \quad (4.4)$$

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 $v_N = e^{ik\frac{2\pi}{N}x}$, $k \in [-N, N]$, this results in a system of differential equations in the Fourier coefficients.

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

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \frac{1}{2} \left(\frac{\partial(u_N)^2}{\partial x}, v_N\right) = 0, \forall v_N \in S_N \quad (4.5)$$

$$u_N(\tau_n) = \tilde{u}_{N,n+\frac{1}{2}}, \quad (4.6)$$

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 non linear: the discretization in time is based on a 3rd 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} .

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

$$\left(\frac{\partial u_N}{\partial t}, v_N\right) + \nu \left(\frac{\partial u_N}{\partial x}, \frac{\partial v_N}{\partial x}\right) = 0, \forall v_N \in S_N \quad (4.7)$$

$$u_N(\tau_{n+\frac{1}{2}}) = u_{N,n+\frac{1}{2}}. \quad (4.8)$$

4.2. Results for the plain parareal in time algorithm. We have performed 3 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 in case referred later as **Case 1**, we have chosen

Case 1:

1. $\mu = 1e - 2$,
2. $N_c = 2^8, N_f = 2^8$,
3. $T = 2, \Delta T = 2.10^{-2}, dT = 1.10^{-2}, \delta t = 1.10^{-4}$

The evolution is presented on figure 4.1, the relative error in the solution with respect to a solution computed with a much smaller time step is presented on figure 4.2 showing that after 3 parareal iterations, the same convergence as for the fine sequential simulation is achieved.

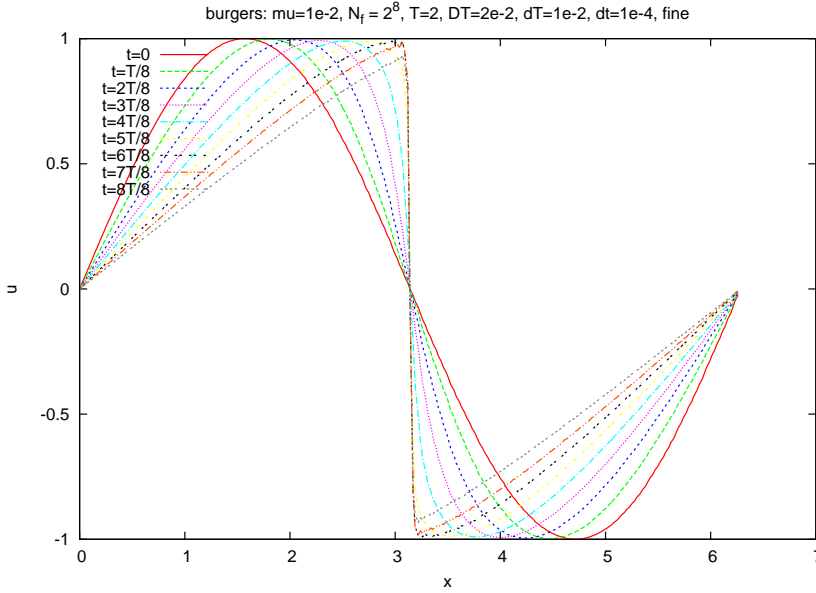


FIG. 4.1. Case 1: solution obtained by the parareal method ($\delta t = 1e - 4, dT = 1e - 2, DT = 2e - 2$)

For the second test, we have diminished the value of the viscosity. This case, referred to as **Case 2**, corresponds to the choices

Case 2:

1. $\mu = 1e - 3$;
2. $N_c = 2^9, N_f = 2^9$,
3. $T = 2, \Delta T = 2.10^{-2}, dT = 1.10^{-2}, \delta t = 1.10^{-5}$

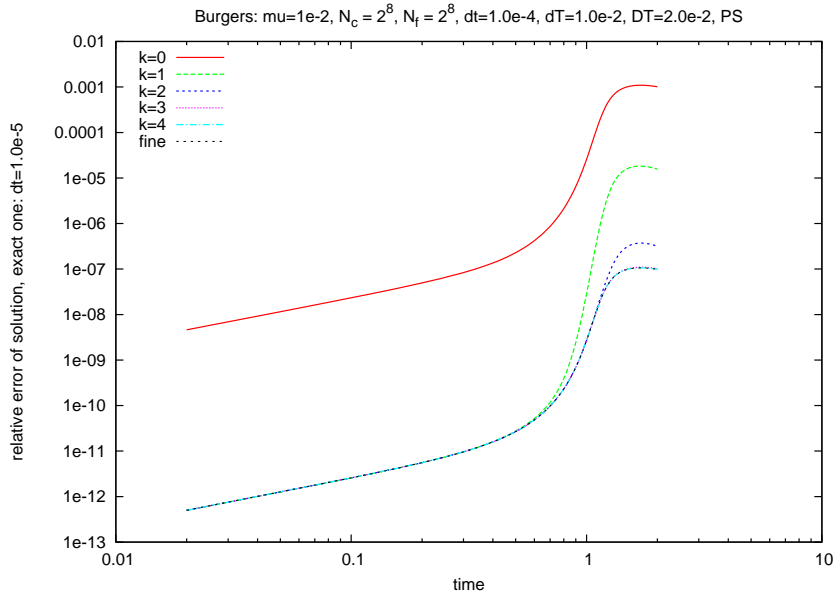


FIG. 4.2. *Case 1: relative error on the solution obtained by the parareal method ($\delta t = 1e-4$, $dT = 1e-2$, $DT = 2e-2$)*

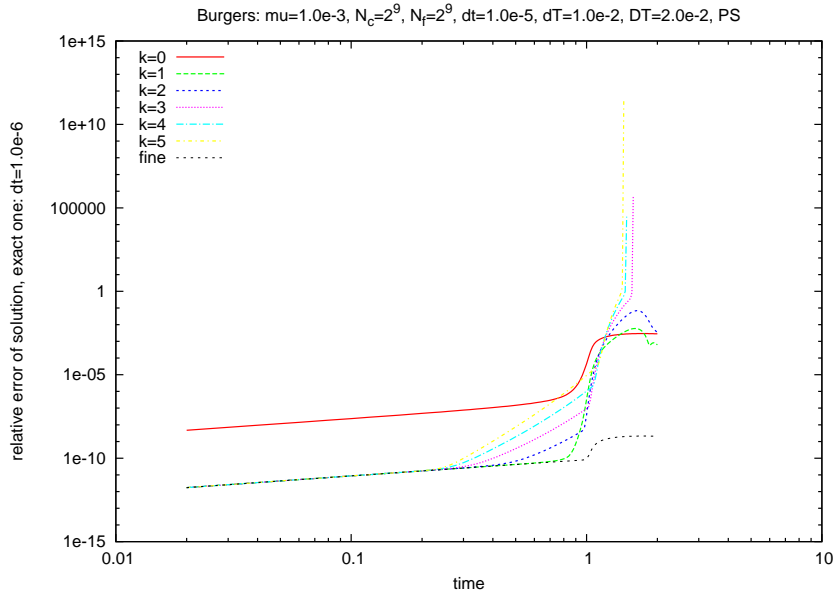


FIG. 4.3. *Case 2-2: relative error of the solution obtained by the parareal method ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)*

For this case, the plain parareal method is not stable, as can be seen on the figure 4.3, after 3 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 [17], that using a coarser spacial representation helps in the stabilization of the process. Indeed, **Case 3** below provides a stable parareal approximation

Case 3:

1. $\mu = 1e - 3$;
2. $N_c = 2^8, N_f = 2^9$,
3. $T = 2, \Delta T = 2.10^{-2}, dT = 1.10^{-2}, \delta t = 1.10^{-5}$.

As can be seen on the series of figures 4.4, 4.5 and 4.6, the method is stable, converges but very slowly.

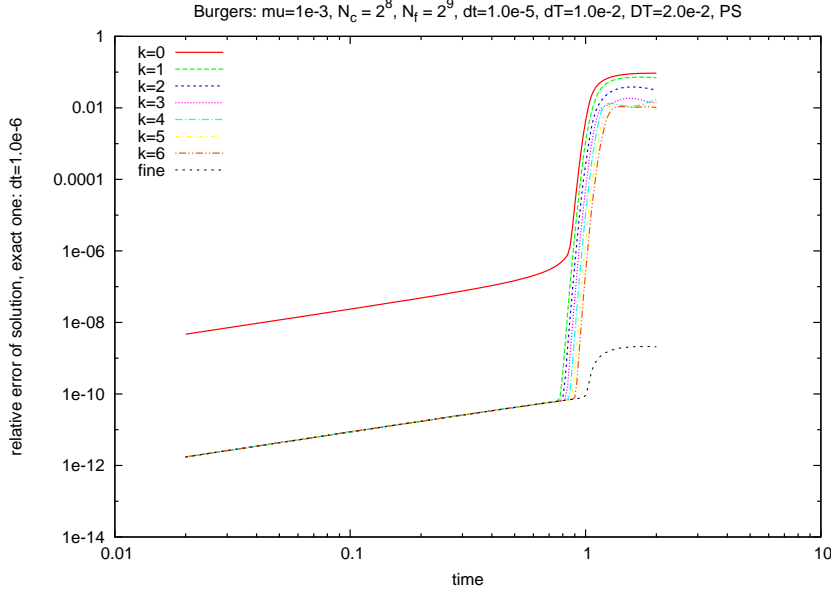


FIG. 4.4. *Case 2-1: relative error of the solution obtained by the parareal method ($\delta t = 1e - 5, dT = 1e - 2, DT = 2e - 2$)*

In the next sections, we shall improve the two last 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 notice that, due to dissipation, there is no quantity that is preserved along a trajectory, nevertheless, by using u_N as a test function in (4.2), we derive that

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

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

$$\|u_N(T_{n+1})\|_{L^2(0,T)}^2 + 2\nu \int_{T_n}^{T_{n+1}} \left[\frac{\partial u_N}{\partial x} \right]^2 = \|u_N(T_n)\|_{L^2(0,T)}^2. \quad (4.9)$$

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

$$\|u_{n+1}^{k+1}\|_{L^2(0,T)} = \|\mathcal{F}_{\Delta T}[u_n^{k+1}]\|_{L^2(0,T)} \quad (4.10)$$

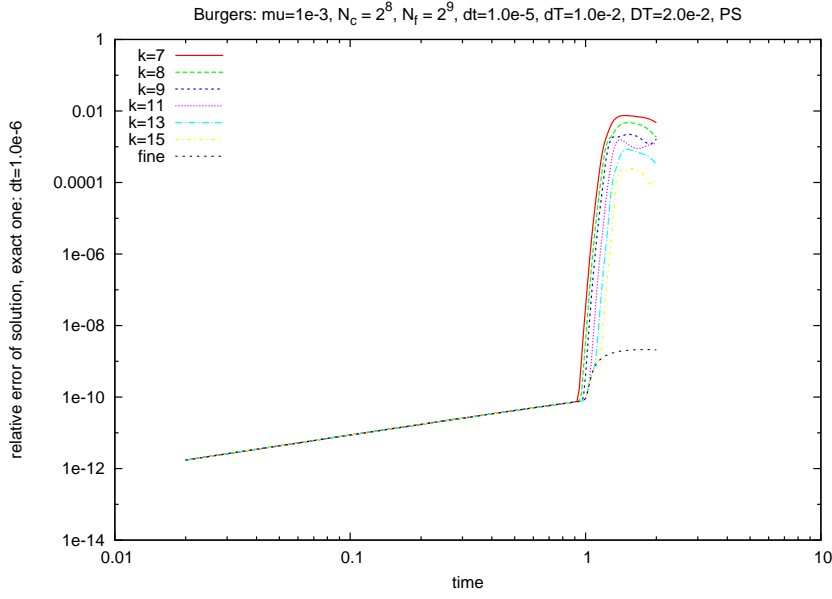


FIG. 4.5. Case 2-1: relative error of the solution obtained by the parareal method ($\delta t = 1e - 5$, $dT = 1e - 2$, $DT = 2e - 2$)

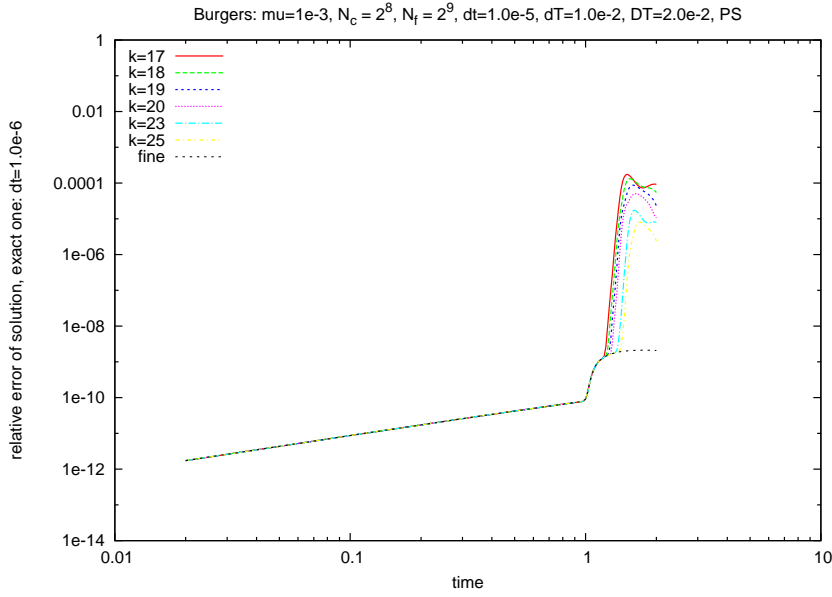


FIG. 4.6. Case 2-1: relative error of the solution obtained by the parareal method ($\delta t = 1e - 5$, $dT = 1e - 2$, $DT = 2e - 2$)

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

$$\|\mathcal{F}_{\Delta T}[u_n^{k+1}]\|_{L^2(0,\mathbb{T})} \simeq \frac{\|\mathcal{F}_{\Delta T}[u_n^k]\|_{L^2(0,\mathbb{T})}}{\|u_n^k\|_{L^2(0,\mathbb{T})}} \|u_n^{k+1}\|_{L^2(0,\mathbb{T})}. \quad (4.11)$$

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

Second, we have noticed 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 $i = 1, \dots, I$) defined by

$$G_i = \text{span}\{e^{i\ell \frac{2\pi}{T}x}, N_i^- \leq |\ell| < N_i^+\}, \quad (4.12)$$

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

$$[\mathcal{M}_n^I]^{k+1} = \cap_{i=1}^I [m_n^i]^{k+1} \quad (4.13)$$

where the group manifolds $[m_n^i]^{k+1}$ are defined as follows .

$$[m_n^i]^{k+1} = \{v \in S_N, \|P_i(v)\|_{L^2(0,T)} = \frac{\|P_i[\mathcal{F}_{\Delta T}[u_n^k]]\|_{L^2(0,T)}}{\|u_n^k\|_{L^2(0,T)}} \|u_n^{k+1}\|_{L^2(0,T)}\} \quad (4.14)$$

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

With this manifold the parareal scheme with projection reads

$$\begin{cases} u_{n+1}^0 &= \mathcal{G}_{\Delta T}(u_n^0) \\ \tilde{u}_{n+1}^{k+1} &= \mathcal{G}_{\Delta T}(u_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(u_n^k), \\ u_{n+1}^{k+1} &= \pi_{[\mathcal{M}_n^I]^{k+1}}(\tilde{u}_{n+1}^{k+1}). \end{cases} \quad (4.15)$$

The results are as follows (see figures 4.7, 4.8, 4.9). We notice 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 3. *If we would know the exact value of the energy of each groups, the projection method would converge. The problem of course is that due to dissipation, and also to nonlinear effect that mixes up the low and high frequencies, the value of the exact energy in each groups is not known. This is the reason why we get into these difficulties.*

4.4. Results for the plain parareal in time algorithm with projection and damping. We notice that, for this non linear problem, the projection allows 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

$$\forall \ell \in G_1, \quad \hat{w}_\ell = \hat{v}_\ell \quad (4.16)$$

$$\forall \ell \in G_i, i = 2, \dots, I, \quad \hat{w}_\ell = f(i)\hat{v}_\ell \quad (4.17)$$

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 2 groups, where we have chosen

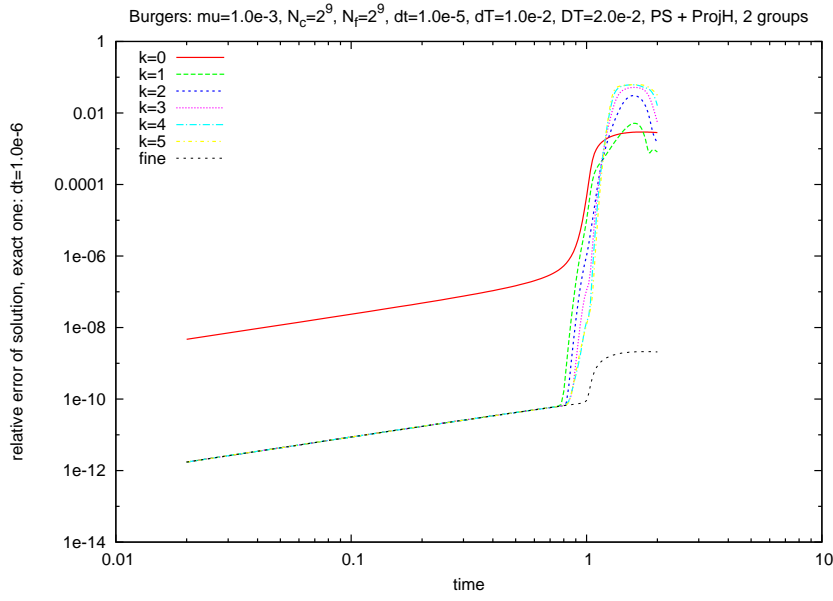


FIG. 4.7. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

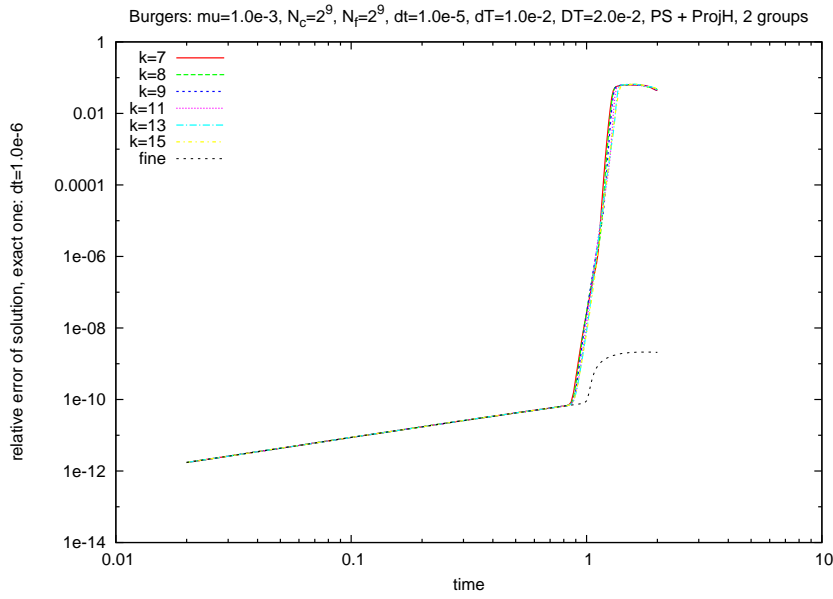


FIG. 4.8. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

$f(i) = \frac{1}{3i+1.0}$. With the same definition of the multigroup manifold the parareal

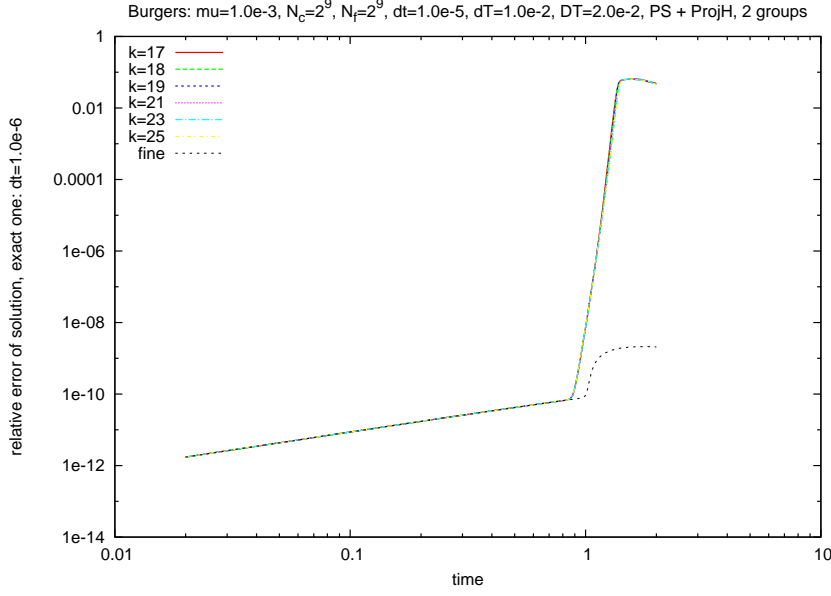


FIG. 4.9. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

scheme with projection and damping reads

$$\begin{cases} u_{n+1}^0 &= \mathcal{G}_{\Delta T}(u_n^0) \\ \tilde{u}_{n+1}^{k+1} &= \mathcal{G}_{\Delta T}(\bar{u}_n^{k+1}) + \mathcal{F}_{\Delta T}(u_n^k) - \mathcal{G}_{\Delta T}(\bar{u}_n^k), \\ u_{n+1}^{k+1} &= \pi_{[\mathcal{M}]_n^{k+1}}(\tilde{u}_{n+1}^{k+1}) \\ \bar{u}_{n+1}^{k+1} &= \Phi(u_{n+1}^{k+1}). \end{cases} \quad (4.18)$$

The following figures 4.10, 4.11 and 4.11 present the results of this improved parareal method. We notice that the parareal method is now converging.

REMARK 4. *Note that the strategy used here consisting in 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 [22]. 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 any parareal scheme, after N^{th} iterations, the solution is the same as the one of the plain fine propagator) but improves the quality of the scheme.*

The projection with more than two groups performs slightly better, see e.g. figure 4.13 where only the results for iterations between 7 and 15 are 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 non periodic case where finite difference, finite differences or finite volume techniques will be used.

5. Preliminary analysis. In this section, we provide some simple elements of analysis that prove the stability of the previous approximation with the parareal method with projection (and damping).

Let us first assume that the fine propagator is L^2 stable in the sense that there

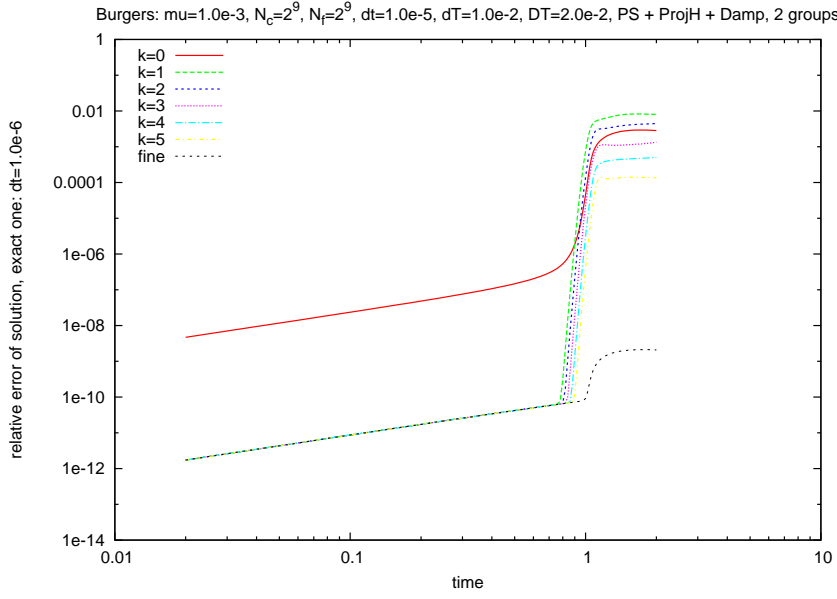


FIG. 4.10. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups with damping ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

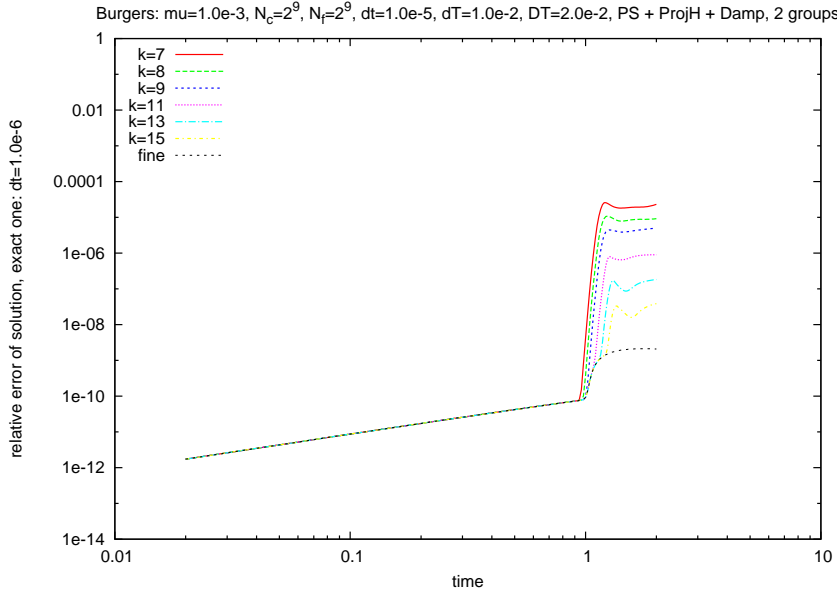


FIG. 4.11. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups with damping ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

exists a constant $\alpha > 0$ independent of the discretization parameters such that,

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

Note that both for the wave equation and the Burgers' equation, the previous

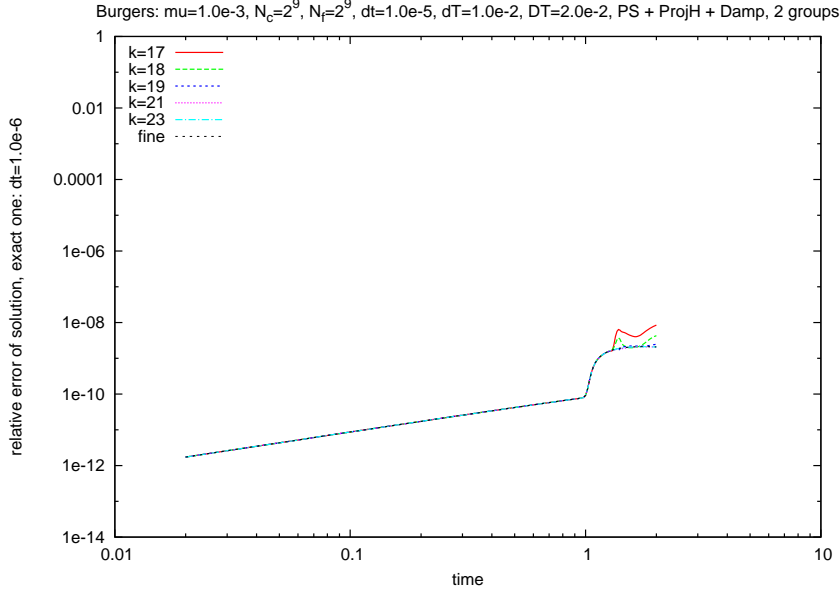


FIG. 4.12. Case 2-2: relative error of the solution obtained by the method parareal with projection to 2 groups with damping ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

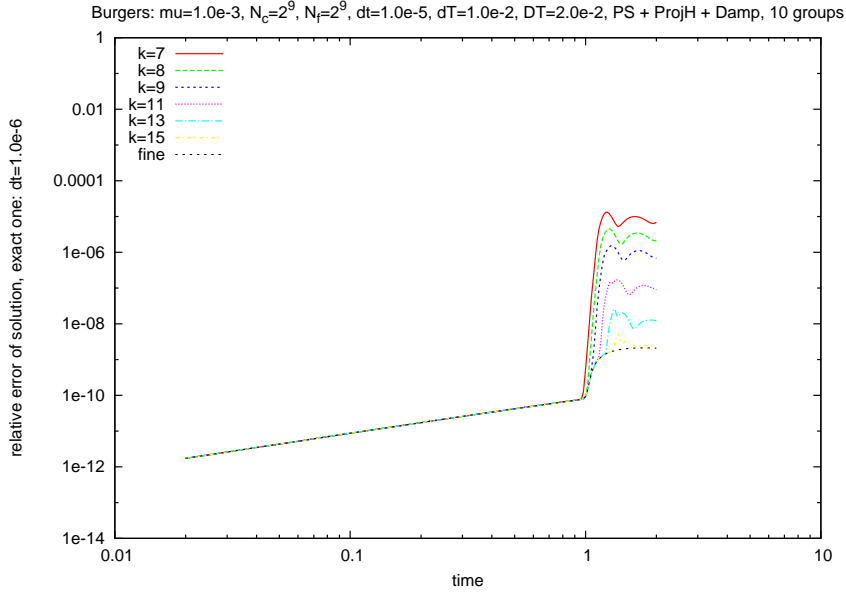


FIG. 4.13. Case 2-2: relative error of the solution obtained by the method parareal with projection to 10 groups with damping ($\delta t = 1e-5$, $dT = 1e-2$, $DT = 2e-2$)

inequality holds with $\alpha = 0$.

Let us start with the case where no dumping is implemented, then, from (4.15)

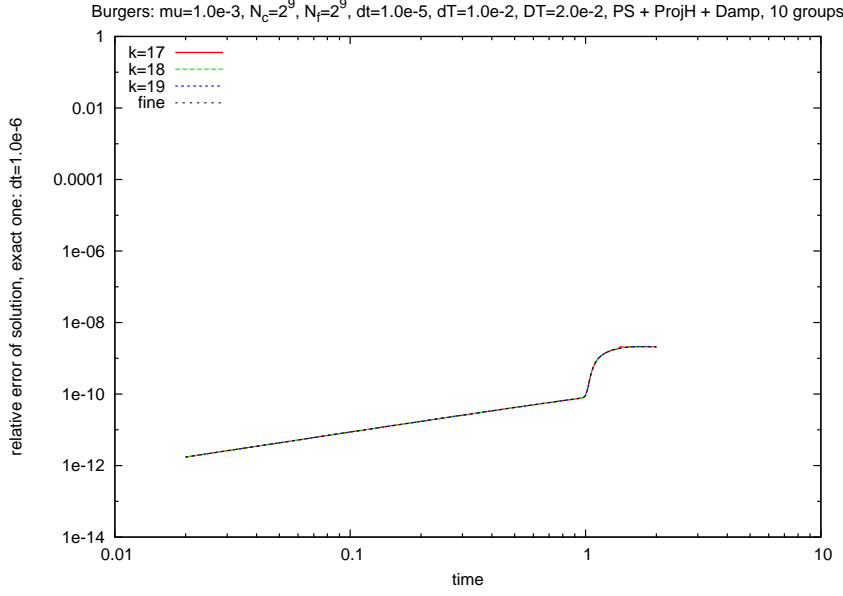


FIG. 4.14. Case 2-2: relative error of the solution obtained by the method parareal with projection to 10 groups with damping ($\delta t = 1e - 5, dT = 1e - 2, DT = 2e - 2$)

we derive

$$\begin{aligned}
 \|u_{n+1}^{k+1}\|_{L^2(0,T)} &= \pi_{[\mathcal{M}^I]_n^{k+1}}(\tilde{u}_{n+1}^{k+1})\|_{L^2(0,T)}^2 = \sum_{i=1}^I \|P_i[u_{n+1}^{k+1}]\|_{L^2(0,T)}^2 \\
 &= \sum_{i=1}^I \frac{\|P_i[\mathcal{F}_{\Delta T}[u_n^k]]\|_{L^2(0,T)}^2}{\|u_n^k\|_{L^2(0,T)}^2} \|u_n^{k+1}\|_{L^2(0,T)}^2 \\
 &= \frac{\|\mathcal{F}_{\Delta T}[u_n^k]\|_{L^2(0,T)}^2}{\|u_n^k\|_{L^2(0,T)}^2} \|u_n^{k+1}\|_{L^2(0,T)}^2 \\
 &\leq (1 + \alpha \Delta T)^2 \|u_n^{k+1}\|_{L^2(0,T)}^2
 \end{aligned} \tag{5.2}$$

This proves the stability by induction

$$\begin{aligned}
 \|u_{n+1}^{k+1}\|_{L^2(0,T)} &\leq (1 + \alpha \Delta T)^{n+1} \|u_0^{k+1} = u_0\|_{L^2(0,T)} \\
 &\leq e^{\alpha T} \|u_0\|_{L^2(0,T)}
 \end{aligned} \tag{5.3}$$

6. Concluding remarks. In this paper we have analyzed and proposed a cure to the lack of robustness of the parareal in time algorithm applied to hyperbolic systems or convection diffusion problems with small diffusion. The lack of robustness, leading sometimes to instabilities of the algorithm before convergence is a consequence of the lack of regularity that the solution develops when time runs. The cure is inherited from the extension of the parareal in time algorithm for Hamiltonian system and consists in projecting the solution over an energy manifold defined on the fly, and improved iteratively, where the exact solution lies. This simple procedure allows to stabilize the algorithm that now converges after some iterations. It should be noted that, similarly as for other types of equations, the convergence of the parareal in time algorithm allows 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 notice 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 than the one that a plain domain decomposition technique is able to optimally use. We refer to [12] for more on this matter.

Acknowledgments. This work has been supported in part by the Agence Nationale de la Recherche, under the grant ANR-06-CIS6-007 (PITAC).

REFERENCES

- [1] L. BAFFICO, S. BERNARD, Y. MADAY, G. TURINICI, G. ZÉRAH, *Parallel in time molecular dynamics simulations*, *Physical Review. E* **66**(2002)
- [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*, Kornhuber R, Hoppe RHW, Keyes DE, Périaux J, Pironneau O, Xu J (eds). *Lecture Notes in Computer Science and Engineering*, vol. 40. Springer: Berlin, 2004; 425432.
- [3] G. BAL, 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*, L. F. Pavarino, A. Toselli (eds), *Lecture Notes in Computational Science and Engineering*, **23**, Springer, 189—202 (2002)
- [4] K. BURRAGE, *Parallel and sequential methods for ordinary differential equations*, *Numerical Mathematics and Scientific Computation*, Oxford Science Publications, The Clarendon Press, Oxford University Press, New York, 1995.
- [5] K. BURRAGE, *Parallel methods for ODEs*, *Advances in Computational Mathematics* **7**(1-2), 1–3, 1997.
- [6] F. CHOULY AND M. A. FERNÁNDEZ, *An Enhanced parareal Algorithm for Partitioned Parabolic-Hyperbolic Coupling*, *AIP Conference Proceedings*, **1168** 1, pp 1517–1520, 2009.
- [7] X. DAI, C. LE BRIS, F. LEGOLL, Y. MADAY, *parareal algorithms for Hamiltonian systems*, submitted to *M²AN*
- [8] C. FARHAT, M. CHANDESIS, *Time-decomposed parallel time-integrators part I: theory and feasibility studies for fluid, structure, and fluidstructure applications*, *International Journal for Numerical Methods in Engineering* 2003; 58(9):13971434.
- [9] C. FARHAT, J. CORTIAL, C. DASTILLUNG, H. BAVESTRELLO, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, *International Journal for Numerical Methods in Engineering* 2006; 67(5):697724.
- [10] J. CORTIAL, C. FARHAT, *A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems* *International Journal for Numerical Methods in Engineering* 2009; 77 :451470.
- [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*, volume 40 of *Lect. Notes in Comput. Sci. Eng.*, pages 433440. Springer, Berlin, 2005.
- [12] R. GUÉTAT, *Méthode de parallélisation en temps : Application aux méthodes de décomposition de domaine* PhD Thesis, UPMC Paris 2011.
- [13] E. HAIRER, C. LUBICH, AND G. WANNER, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, ??
- [14] J.-L. LIONS, Y. MADAY, G. TURINICI, *Résolution d'EDP par un schéma en temps "pararéel"*, *Comptes Rendus de l'Académie des Sciences de Paris — Série I — Mathématique* 2001; 332(7):661—668.
- [15] Y. LIU, AND J. HU, *Modified propagators of parareal in time algorithm and application to Princeton Ocean model*, *International Journal for Numerical Methods in Fluids*, 57: 17931804. (2008).
- [16] Y. MADAY, *parareal in time algorithm for kinetic systems based on model reduction*, In *High dimensional partial differential equations in science and engineering*, **41** CRM Proc. Lecture Notes, pages 183 — 194. Amer. Math. Soc., Providence, RI, 2007.

- [17] Y. MADAY. , *The “parareal in time” algorithm*, In *Sub-Structuring Techniques and Domain Decomposition Methods*, **24** Computational Science, Engineering & Technology Series, pages 19—44. Saxe-Coburg Publications, 2010.
- [18] Y. MADAY, G. TURINICI , *A parareal in time procedure for the control of partial differential equations* *C.R.Acad Sci. Paris Sér. I Math* **335** 387—392 (2002)
- [19] D. MERCERAT AND L. GUILLOT AND J.-P. VILOTTE , *Application of the parareal Algorithm for Acoustic Wave Propagation*, *AIP Conference Proceedings*, **1168** 1, pp 1521–1524, 2009.
- [20] J. NIEVERGELT. , *Parallel methods for integrating ordinary differential equations*, *Commun. ACM*, 7(12):731—733, 1964.
- [21] G. STAFF AND E. RÖNQVIST , *Stability of the parareal Algorithm In Domain decomposition methods in science and engineering*, volume 40 of *Lect. Notes in Comput. Sci. Eng.*, pages 449—456. Springer, Berlin, 2005.
- [22] E. TADMOR, *Convergence of Spectral Methods for Nonlinear Conservation Laws*, *SIAM Journal on Numerical Analysis* Vol. 26, No. 1 (Feb., 1989), pp. 30-44