The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation

Yvon Maday[1], Gabriel Turinici[2,3]

[1] Université Pierre et Marie Curie, Laboratoire Jacques Louis Lions, Boite courrier 187, 75252 Paris cedex 05, France.

[2] INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt B.P. 105, 78153 Le Chesnay cedex, France

[3] CERMICS-ENPC, Champs sur Marne, 77455 Marne la Vallée Cedex 2, France

Summary. This paper is the basic one of the series resulting from the minisymposium entitled "Recent Advances for the Parareal in Time Algorithm" that was held at DD15. The parareal in time algorithm is presented in its current version (predictor-corrector) and the combination of this new algorithm with other more classical iterative solvers for parallelization which makes it possible to really consider the time direction as fertile ground to reduce the time integration costs.

1 Introduction

In the seminal paper Lions et al. [2001] the generalization of the concept of domain decomposition for time solution was first proposed. Even though the time direction seems intrinsically sequential, the combination of a coarse and a fine solution procedure have proven to converge and allow for more rapid solution if parallel architectures are available. This has led to the name "parareal in time" that has been proposed for this method. Since then, this scheme has received some attention and a presentation under the format of a predictor-corrector algorithm has been made in Bal and Maday [2002] and also in Baffico et al. [2002]. It is this last presentation that we shall use in what follows. Before this let us mention that a matricial form of the scheme was also presented in Maday and Turinici [2002] were the parareal methodology appears as a preconditioner.

Let us consider the partial differential equation (P.D.E.)

$$\frac{\partial u}{\partial t} + Au = 0, \quad \text{over the time interval } [T_0, T] \tag{1}$$

where A is some functional operator, linear or not, from a Hilbert space V into its dual space V'. This P.D.E. is complemented with initial conditions $u(t = T_0) = u_0$ and appropriate boundary conditions that are implicitly incorporated in the formulation and the space V.

It is well known that, when it exists, the solution of this P.D.E. can be written as

$$u(t) = \mathcal{E}_{t-T_0}(u_0; T_0) \tag{2}$$

and that we have the following semigroup property for any T^*, $T_0 \le T^* \le t$

$$u(t) = \mathcal{E}_{t-T^*}(\mathcal{E}_{T^*-T_0}(u_0; T_0); T^*). \tag{3}$$

that formalizes the sequential nature of the this Cauchy problem.

Associated with this formal operator, the numerical solution of this problem leads to an approximate operator $\mathcal{F}$ based on a discretization scheme with time steps δt and some order m. In addition to the time discretization, a discretization in space (with discretization parameter δx) can also be used that leads to an error of the order $\delta t^m + \delta x^\nu$ at any final time T.

Let us assume that a time range $\Delta T >> \delta t$ is being given and that we are interested in the collection of snapshots $\{u(T_n)\}_{0\le n\le N}$ where $T_n = T_0 + n\Delta T$ and $T_N = T$. The proper approximation of these values are given by $\{\lambda_n = \mathcal{F}_{n\Delta T}(u_0; T_0)\}_{0\le n\le N}$ (hence $\lambda_0 = u_0$) as in (3) we note that

$$\lambda_n = \mathcal{F}_{\Delta T}(\lambda_{n-1}; T_{n-1}). \tag{4}$$

The parareal algorithm makes it possible to define iteratively a sequence λ_n^k that converges toward λ_n as k goes to infinity. It involves a coarse solver $\mathcal{G}$, less accurate than $\mathcal{F}$, but much cheaper. It can be based for example on the time step ΔT (or any coarser time step than δt) together, as was proposed already in Lions et al. [2001], with a coarser discretization in space ΔX (see also Fischer et al. [2003]) or even, a simpler physical model, as was implemented in Maday and Turinici [2003]. The assumptions that are made are that

- $\|D(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})\| \le c\varepsilon\Delta T$ where ε depends on both ΔT, δt, ΔX and δx, and goes to zero when ΔT, and ΔX go to zero. The symbol D denotes the first derivative with respect to the first variable.
- $\|D\mathcal{G}_{\Delta T}\| \le c$

In the parareal algorithm, starting from $\lambda_n^0 = \mathcal{G}_{n\Delta T}(u_0; T_0)$ the sequence λ_n^k, $k \ge 1$ is determined by

$$\lambda_n^k = \mathcal{G}_{\Delta T}(\lambda_{n-1}^k; T_{n-1}) + \mathcal{F}_{\Delta T}(\lambda_{n-1}^{k-1}; T_{n-1}) - \mathcal{G}_{\Delta T}(\lambda_{n-1}^{k-1}; T_{n-1}). \tag{5}$$

and we can prove the following error

$$\|\lambda_n - \lambda_n^k\| \le C \sum_{m=k}^{n} \binom{n}{m} \|D(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})\|^m \|D\mathcal{G}_{\Delta T}\|^{n-m}. \tag{6}$$

from which the convergence in k follows since our hypothesis leads to

$$\max_{0\le n\le N} \|\lambda_n - \lambda_n^k\| \le C(T)\varepsilon^k. \tag{7}$$

In order to prove this result we remark that from Eqn (5) one obtains:

$$\begin{aligned}\lambda_n^k - \lambda_n &= \big(\mathcal{G}_{\Delta T}(\lambda_{n-1}^k;T_{n-1}) - \mathcal{G}_{\Delta T}(\lambda_{n-1};T_{n-1})\big)\\ + \quad &(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})(\lambda_{n-1}^{k-1};T_{n-1}) - (\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})(\lambda_{n-1};T_{n-1}) \qquad (8)\end{aligned}$$

Suppose now that $\mathcal{G}_{\Delta T}(\cdot;\cdot)$ and $(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})(\cdot;\cdot)$ are differentiable with respect to the (first) variable uniformly over all its values and over all values of the time parameter. Denoting by $e_n^k = \|\lambda_n^k - \lambda_n\|$ one can write

$$\begin{aligned}&\mathcal{G}_{\Delta T}(\lambda_{n-1}^k;T_{n-1}) - \mathcal{G}_{\Delta T}(\lambda_{n-1};T_{n-1})\\ &= D\mathcal{G}_{\Delta T}(\lambda_{n-1};T_{n-1})(\lambda_{n-1}^k - \lambda_{n-1}) + o(e_{n-1}^k)\end{aligned}$$

and obtain the estimate

$$\|\mathcal{G}_{\Delta T}(\lambda_{n-1}^k;T_{n-1}) - \mathcal{G}_{\Delta T}(\lambda_{n-1};T_{n-1})\| \le 3/2\|D\mathcal{G}_{\Delta T}(\cdot;\cdot)\|e_{n-1}^k$$

for any $e_{n-1}^k \le \mu_{3/2}^g$. Using the same technique for $(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})(\lambda_{n-1}^{k-1};T_{n-1})$ one obtains:

$$e_n^k \le 3/2\|D\mathcal{G}_{\Delta T}(\cdot;\cdot)\|e_{n-1}^k + 3/2\|D(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})(\cdot;\cdot)\|e_{n-1}^{k-1} \qquad (9)$$

provided that e_{n-1}^k and e_{n-1}^{k-1} are smaller than some constants $\mu_{3/2}^g$ and $\mu_{3/2}^{f-g}$ respectively. Provided that the initial guess λ_n^0, $n = 1, ..., N$ is chosen sufficiently close to the solution λ_n from Eqn (4), one can prove by induction the result in Eqn (6). Then, we notice that

$$\begin{aligned}&\sum_{m=k}^{n} \binom{n}{m} \|D(\mathcal{F}_{\Delta T} - \mathcal{G}_{\Delta T})\|^m \|D\mathcal{G}_{\Delta T}\|^{n-m} \le \epsilon^k \sum_{m=k}^{n} \binom{n}{m} (c\Delta T)^m \|D\mathcal{G}_{\Delta T}\|^{n-m}\\ &\le \epsilon^k (\Delta T)^k \sum_{m=0}^{n} \binom{n}{m} c^m \|D\mathcal{G}_{\Delta T}\|^{n-m} = \epsilon^k (\Delta T)^k (c + \|D\mathcal{G}_{\Delta T}\|)^n,\end{aligned}$$

and thus we obtain the result in Eqn (7).

We refer also to Farhat and Chandesris [2003], Staff and Rønquist [2003] and to Bal [2003a] for other issues about stability and approximation of the parareal in time scheme.

2 Combination with domain decomposition – the overlapping case

2.1 The iterative procedure

Let Ω be a domain decomposed into P subdomains that, in this section, we shall assume to be overlapping to make things easier. More precisely let $\overline{\Omega} = \cup_{p=1}^{P} \overline{\Omega}^p$ with $\Omega^p \cap \Omega^q = \emptyset$ whenever $p \neq q$, in addition, we assume that there exists $\omega^{p,q}$ – called here “bandages” – associated with any pair

(p, q) such that $\overline{\Omega}^p \cap \overline{\Omega}^q \neq \emptyset$ so that $\Omega = \{\cup \Omega^p\} \cup \{\cup \omega^{p,q}\}$. Note that those bandages may (and most often have to) overlap.

In what follows, we shall propose a space-time parallel iterative method for solving the following type of problem

$$\begin{aligned} \tfrac{\partial u}{\partial t} - \Delta u = f, &\quad \text{in } \Omega \times [0, T] \\ u(0, x) = u_0(x), &\quad \text{in } \Omega, \\ u(t, x) = g(t, x), &\quad \text{over } \partial\Omega \times [0, T], \end{aligned} \tag{10}$$

To make it easy, for the definition of the algorithm, we assume no discretization is used in space neither for the coarse nor for the fine propagator, similarly we assume that the fine propagator does not involve any discretization in time. We are going to define an iterative procedure that involves the fine and accurate solution (here actually exact) only over each block $\Omega^p \times [T_n, T_{n+1}]$ or $\omega^{p,q} \times [T_n, T_{n+1}]$. The solution at iteration k will be denoted as $u^k_{p,n}$ over $\Omega^p \times [T_n, T_{n+1}]$ and $v^k_{p,q,n}$ over $\omega^{p,q} \times [T_n, T_{n+1}]$. By construction, the function u^k_n built by "concatenation" of the various $\left(u^k_{p,n}\right)_p$ is an element of $H^1(\Omega)$ for almost each time (continuity enforced at the interfaces). We will also have the snapshots λ^k_n available at each time T_n.

The coarse propagator

Once the solution is known at iteration k, the definition of the solution at iteration $k + 1$ involves a coarse operator that can be defined as follows:

$$\frac{\mathcal{G}_{\Delta T}(\lambda^k_n) - \lambda^k_n}{\Delta T} - \Delta(\mathcal{G}_{\Delta T}(\lambda^k_n)) = f(T_{n+1}). \tag{11}$$

The fine propagator

The fine propagator actually involves not only the knowledge of λ^k_n but also of u^k_n. It proceeds as follows

Step one. We first propagate the solution over $\omega^{p,q} \times [T_n, T_{n+1}]$ by solving

$$\begin{aligned} \frac{\partial v^{k+1}_{p,q,n}}{\partial t} - \Delta v^{k+1}_{p,q,n} = f, &\quad \text{in } \omega^{p,q} \times [T_n, T_{n+1}] \\ v^{k+1}_{p,q,n}(T_n, x) = \lambda^k_n(x), &\quad \text{in } \omega^{p,q}, \\ v^{k+1}_{p,q,n}(t, x) = u^k_{p,n}(t, x) + \lambda^k_n(x) &- u^k_{p,n}(T_n, x), \\ &\quad \text{over } \partial\omega^{p,q} \cap \partial\Omega^p \times [T_n, T_{n+1}], \end{aligned} \tag{12}$$

Note that the correction: $\lambda^k_n - u^k_{p,n}(T_n, .)$, allows us to have the boundary conditions compatible with the initial condition for each local problem.

Step two. We now define from the various $v^{k+1}_{p,q,n}$ a current global boundary value, named v^{k+1}_n over $(\cup_p \partial\Omega^p) \setminus \partial\Omega$. In the case where the subdomains $\omega^{p,q}$ do not overlap, then v^{k+1}_n is, over each $(\cup_p \partial\Omega^p) \cap \omega^{p,q}$, equal to the unique possible value that is $v^{k+1}_{p,q,n}$. In case of overlapping $\omega^{p,q}$'s, there is a conflict between the $v^{k+1}_{p,q,n}$ that is solved by choosing a continuous convex combination of the different $v^{k+1}_{p,q,n}$'s.

Step three. We now propagate the solution over $\Omega^p \times [T_n, T_{n+1}]$ by solving

$$\begin{aligned}\frac{\partial u_{p,n}^{k+1}}{\partial t} - \Delta u_{p,n}^{k+1} &= f, \quad \text{in } \Omega^p \times [T_n, T_{n+1}] \\ u_{p,n}^{k+1}(T_n, x) &= \lambda_n^k(x), \quad \text{in } \Omega^p, \\ u_{p,n}^{k+1}(t, x) &= g(x), \quad \text{over } \partial\Omega^p \cap \partial\Omega \times [T_n, T_{n+1}], \\ u_{p,n}^{k+1}(t, x) &= v_n^{k+1}(t, x), \quad \text{over } \partial\Omega^p \cap \partial\omega^{p,q} \times [T_n, T_{n+1}],\end{aligned} \tag{13}$$

This allows us to define a new global solution u_n^{k+1} over each $\Omega \times [T_n, T_{n+1}]$ since, as we already said, the $u_{p,n}^{k+1}$ do match at the interfaces.

The $k+1$ iteration

The definition of each λ_n^{k+1}, $1 \leq n \leq N$ then proceeds similarly as for (5)

$$\lambda_n^{k+1} = \mathcal{G}_{\Delta T}(\lambda_n^k) + u_n^{k+1}(T_{n+1}) - \mathcal{G}_{\Delta T}(\lambda_n^{k-1}). \tag{14}$$

2.2 Numerical results

The first set of computations has been done on a rectangular domain $]0,4[\times]0,1[$, decomposed into 2 equal rectangles $\Omega^1 =]0,2[\times]0,1[$ and $\Omega^2 =]2,4[\times]0,1[$ plus a rectangular "bandage" $\omega^{1,2}$ of various width ($]1,3[\times]0,1[$ or $]1.5,2.5[\times]0,1[$). The P.D.E. that we have solved is

$$\frac{\partial u}{\partial t} - \nu\Delta u = f, \tag{15}$$

with $\nu = 1$ and $f = 50\sin(2\pi(x+t))\cos(2\pi(y+t))$ over a time range $T - T_0 = 1$. We have used a P_1-finite element discretization in space and an implicit Euler scheme of first order in time. The fine propagator is based on a time step δt that is 50 times smaller than the large time step. In the experiments reported below in Table 1, the size of the large time step $\Delta T = 1/N$ varies. A priori N is related to the number of parallel processors we have. Here this figure should be $2N$ as there are two subdomains that can be run at the same time. Table 1 summarizes the error between λ_n^k and the finite element solution with a very fine discretization in time. Note that in all the situations the error after 5 (resp. 4) iterations remains constant and is (resp. is of the order of) the error resulting from $\delta t = 1/(50N)$. Note that if we double N, achieving thus an error that is, at convergence, twice smaller, the number of iterations remains the same. This indicates the perfect scalability of our global (parareal + Schwarz) scheme.

Note that to be completely legal in the former statement, we assume that the cost of the coarse solvers should be considered as negligible with respect to the cost of the fine solver. To do so a coarse discretization in space should be added, we are currently working in that direction.

Table 1

	width of $\omega^{1,2}$= 2				width of $\omega^{1,2}$ = 1			
k=	N=15	N=30	N=60	N=120	N=15	N=30	N=60	N=120
1	0.95	0.49	0.28	0.17	0.50	0.38	0.31	0.27
2	0.076	0.040	0.031	0.020	0.10	0.068	0.065	0.042
3	0.045	0.022	0.016	0.009	0.056	0.022	0.016	0.009
4	0.045	0.024	0.014	0.005	0.041	0.022	0.016	0.005
5	0.045	0.022	0.011	0.006	0.041	0.020	0.010	0.005
6	0.045	0.022	0.011	0.006	0.041	0.020	0.010	0.005

Another indication on this scalability is that, if we maintain the accuracy, by having the product $n \times N$ constant, then the number of iterations required for convergence remains also constant. Hence provided that you have twice the number of processors, then N can be multiplied by a factor of 2 and the cost of each iteration is divided by 2. Since the number of iterations remains constant, this means that the global time to wait is divided by 2.

We have also performed the same Schwarz method over $]0,4[\times]0,4[$ divided into 4 squares of size 2 (the Ω^p's) and 2 rectangular "bandage" $\omega^{p,q}$: $]1.5,2.5[\times]0,4[$ and $]0,4[\times]1.5,2.5[$. The results are reported in Table 2.

Table 2

k=	N=15	N=30	N=60	N=120
1	0.28	0.11	0.077	0.046
2	0.082	0.032	0.020	0.010
3	0.034	0.014	0.007	0.004
4	0.021	0.009	0.007	0.004
5	0.017	0.009	0.007	0.004
6	0.017	0.009	0.007	0.004

The same conclusion holds for this set of experiments. It is even better since the saturated convergence is achieved for smaller values of k when N (thus here both the accuracy and the number of processors) increases.

We refer also to Farhat and Chandesris [2003] and especially to Bal [2003b] for other issues about scalability of this algorithm.

3 Combination with domain decomposition – the non-overlapping case

We have generalized this approach to a non overlapping situation in the case were we only assume $\overline{\Omega} = \cup_{p=1}^{P} \overline{\Omega}^p$ with $\Omega^p \cap \Omega^q = \emptyset$ whenever $p \neq q$. We have chosen here the Neumann-Neumann strategy as in Bourgat et al. [1989] The approach results also in the fine solution of problems set over P subdomains times a time span of ΔT. The approach here differs from the overlapping case in the sense that the fine propagator involves both a Dirichlet and a Neumann solver:

The fine propagator

Let us assume that $\{\lambda_n^k\}_n$ are given together with the values $\{\beta_n^k\}_n$ corresponding to a predictor for the Dirichlet value of the solution over $\cup\partial\Omega^p\setminus\partial\Omega$. We first propagate the solution over each $\Omega^p\times]T_n, T_{n+1}[$ from λ_n^k with the boundary conditions β_n^k. In order to correct these boundary conditions, as in the Neumann-Neumann algorithm, we transform the jump in the normal derivatives of the solutions that have been computed at interface by a harmonic lifting that provides a corrector for the boundary condition. A relaxation parameter is adjusted (in our case through an optimal gradient approach) in order to minimize the final jump in the solution.

The coarse propagator

The coarse propagator is similar to that of the previous section. We can remark at this level that both in this case or in the overlapping case, there is room for reducing the cost of this global propagation, either by coarsening the spacial mesh size or by using a (sole) domain decomposition approach.

The numerical results

What we report here are only preliminary results that have to be extended to more complex cases. We should also replace the gradient method by a faster (at least a conjugate gradient or a GMRES) methods. We have considered the same problem as for the overlapping strategy, on the rectangular domain $]0,4[\times]0,1[$. We assume it is only decomposed into Ω^1 and Ω^2 (without any bandage). We have run the procedure and shown that, by keeping the fine time step constant, thus decreasing the number of fine time step within ΔT as we increase N, the number of iteration for convergence again remains constant. This gives evidences of the scalability of the method. We have to remark that the convergence rate of the iterative procedure is appreciably lower for this non-overlapping strategy than for the overlapping one. We are convinced that by replacing the crude gradient method that we have implemented by a better approach, the method will perform as nicely as in the overlapping case. This is the subject of a forthcoming paper to improve this strategy and extend it to other classes of classical iterative-domain decomposition based- methods in space (as FETI, Dirichlet Neuman, substructuring..).

Acknowledgement. The numerical results have been performed on the Freefem++ platform and we have benefitted from the advices of Frédéric Hecht and Olivier Pironneau.

References

Leonardo Baffico, Stephane Bernard, Yvon Maday, Gabriel Turinici, and Gilles Zérah. Parallel in time molecular dynamics simulations. *Phys. Rev. E.*, 66, 2002.

Guillaume Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Fifteen International Conference on Domain Decomposition Methods*, Berlin, 2003a. Springer, Lecture Notes in Computational Science and Engineering (LNCSE).

Guillaume Bal. Parallelization in time of (stochastic) ordinary differential equations. Math. Meth. Anal. Num. (submitted), 2003b.

Guillaume Bal and Yvon Maday. A "parareal" time discretization for nonlinear PDE's with application to the pricing of an American put. In *Recent developments in domain decomposition methods (Zürich, 2001)*, volume 23 of *Lect. Notes Comput. Sci. Eng.*, pages 189–202. Springer, Berlin, 2002.

Jean-François Bourgat, Roland Glowinski, Patrick Le Tallec, and Marina Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM.

Charbel Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *Int. J. Numer. Meth. Engng.*, 58(9):1397–1434, 2003.

Paul Fischer, Frédéric Hecht, and Yvon Maday. A parareal in time semi implicit approximation of the Navier Stokes equations. In *Fifteen International Conference on Domain Decomposition Methods*, Berlin, 2003. Springer, Lecture Notes in Computational Science and Engineering (LNCSE).

Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. A parareal in time discretization of PDE's. *C.R. Acad. Sci. Paris, Serie I*, 332:661–668, 2001.

Yvon Maday and Gabriel Turinici. A parareal in time procedure for the control of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 335 (4):387–392, 2002. ISSN 1631-073X.

Yvon Maday and Gabriel Turinici. Parallel in time algorithms for quantum control: the parareal time discretization scheme. *Int. J. Quant. Chem.*, 93 (3):223–228, 2003.

Gunnar Andreas Staff and Einar M. Rønquist. Stability of the parareal algorithm. In *Fifteen International Conference on Domain Decomposition Methods*, Berlin, 2003. Springer, Lecture Notes in Computational Science and Engineering (LNCSE).