Parareal in time intermediate targets methods for optimal control problem

Yvon Maday, Mohamed-Kamel Riahi and Julien Salomon

Abstract. In this paper, we present a method that enables solving in parallel the Euler-Lagrange system associated with the optimal control of a parabolic equation. Our approach is based on an iterative update of a sequence of intermediate targets that gives rise to independent subproblems that can be solved in parallel. This method can be coupled with the parareal in time algorithm. Numerical experiments show the efficiency of our method.

Mathematics Subject Classification (2010). Primary 49J20; Secondary 68W10.

Keywords. Control, Optimization, PDEs, parareal in time algorithm, hight performance computing, parallel algorithm.

1. Introduction

In the last decade, parallelism across the time [3], based on the decomposition of the time domain has been exploited to accelerate the simulation of systems governed by time dependent partial differential equations [4]. Among others, the parareal algorithm [5] or multi-shooting schemes [2] have shown excellent results. In the framework of optimal control, this approach has been used to control parabolic systems [7, 8].

In this paper, we introduce a new approach to tackle such problems. The strategy we follow is based on the concept of target trajectory that has been introduced in the case of hyperbolic systems in [6]. Because of the irreversibility of parabolic equations, a new definition of this trajectory is considered. It enables us to define at each end point of the time sub-domains relevant initial conditions and intermediate targets, so that the initial problem is split up into independent optimization problems.

The paper is organized as follows: the optimal control problem is introduced in Section 2 and the parallelization setting is described in Section 3.

Part of this work was funded by by the ANR-06-CIS6-007-01 project PITAC.

The properties of the cost functionals involved in the control problem are studied in Section 4. The general structure of our algorithm is given in Section 5 and its convergences is proven in Section 6. In Section 7, we propose a fully parallelized version of our algorithm. Some numerical tests showing the efficiency of our approach are presented in Section 8.

In the sequel, we consider the optimal control problem associated with the heat equation on a compact set Ω and a time interval $[0, T]$, with $T > 0$. We denote by $\|.\|_\Omega$ the space norm associated with $L^2(\Omega)$, and by $\|.\|_{\Omega_c}$ the L^2-norm corresponding to a sub-domain $\Omega_c \subset \Omega$. Also, we use the notations $\|.\|_v$ (resp. $\|.\|_{v_n}$) and $\langle ., . \rangle_v$ (resp. $\langle ., . \rangle_{v_n}$) to represent the norm and the scalar product of the Hilbert space $L^2(0, T; \Omega_c)$ (resp. $L^2(I'; \Omega_c)$), with I' a sub-interval of $[0, T]$). Given a function y defined on the time interval $[0, T]$, we denote by $y_{|I'}$ the restriction of y to I'.

2. Optimal control problem

Given $\alpha > 0$, consider the optimal control problem defined by:

$$\min_{v \in L^2([0,T];L^2(\Omega_c))} J(v),$$

with

$$J(v) = \frac{1}{2}\|y(T) - y_{target}\|^2_\Omega + \frac{\alpha}{2}\int_0^T \|v(t)\|^2_{\Omega_c} dt,$$

where y_{target} is a given state in $L^2(\Omega)$. The state y evolves from y_0 on $[0, T]$ according to

$$\partial_t y - \nu \Delta y = \mathcal{B}v.$$

In this equation, Δ denotes the Laplace operator, v is the control term, applied on Ω_c and $\mathcal{B}$ is the natural injection from Ω_c into Ω. We assume Dirichlet conditions for y on the boundary of Ω.
The corresponding optimality system reads as

$$\begin{cases} \partial_t y - \nu \Delta y &= \mathcal{B}v \quad \text{on } [0, T] \times \Omega \\ y(0) &= y_0, \end{cases} \tag{2.1}$$

$$\begin{cases} \partial_t p + \nu \Delta p &= 0 \quad \text{on } [0, T] \times \Omega \\ p(T) &= y(T) - y_{target}, \end{cases} \tag{2.2}$$

$$\alpha v + \mathcal{B}^* p = 0, \tag{2.3}$$

where $\mathcal{B}^*$ is the adjoint operator of $\mathcal{B}$.

Note that for any $\alpha > 0$, the functional J is continuous, α-convex in $L^2(\Omega_c)$ and consequently the system (2.1–2.3) has a unique solution by $v^\star$. We denote by $y^\star$, $p^\star$ the associated state and adjoint state.

3. Time parallelization setting

In this section, we describe the relevant setting for a time parallelized resolution of the optimality system.
Consider $N \geq 1$ and a subdivision of $[0,T]$ of the form:

$$[0,T] = \cup_{n=0}^{N-1} I_n,$$

with $I_n = [t_n, t_{n+1}]$, $t_0 = 0 < t_1 < ... < t_{N-1} < t_N = T$. For the sake on simplicity, we assume here that the subdivision is uniform, i.e. for $n = 0, \ldots, N-1$ we assume that $t_{n+1} - t_n = T/N$; we denote $\Delta T = T/N$. Given a control v and its corresponding state y and adjoint state p, we define the *target trajectory* by:

$$\chi = y - p \qquad \text{on } [0,T] \times \Omega. \tag{3.1}$$

The trajectory χ is not governed by a partial differential equation, but reaches $\chi(T) = y_{target}$ at time T from (2.2_b), hence its denomination.

For $n = 0, \ldots, N-1$, consider the sub-problems

$$\min_{v_n \in L^2(I_n; L^2(\Omega_c))} J_n(v_n), \tag{3.2}$$

with

$$J_n(v_n) = \frac{1}{2}\|y_n(t_{n+1}) - \chi(t_{n+1})\|_\Omega^2 + \frac{\alpha}{2}\int_{I_n} \|v_n(t)\|_{\Omega_c}^2 dt, \tag{3.3}$$

where the function y_n is defined by

$$\left\{ \begin{array}{rcll} \partial_t y_n - \nu \Delta y_n & = & \mathcal{B} v_n & \text{on } I_n \times \Omega \\ y_n(t_n) & = & y(t_n). & \end{array} \right. \tag{3.4}$$

Recall that this optimal control problem is parameterized by v (and y and p) through the local target $\chi(t_{n+1})$, we note that this sub-problem has the same structure as the original one, and is also strictly convex. The optimality system associated with this optimization problem is given by (3.4) and the equations

$$\left\{ \begin{array}{rcll} \partial_t p_n + \nu \Delta p_n & = & 0 & \text{on } I_n \times \Omega \\ p_n(t_{n+1}) & = & y(t_{n+1}) - \chi(t_{n+1}), & \end{array} \right. \tag{3.5}$$

$$\alpha v_n + \mathcal{B}^* p_n = 0, \tag{3.6}$$

we denote by $v_n^\star$ its solution.

4. Some properties of J and J_n

The introduction of the target trajectory in the last section is motivated by the following result.

Lemma 1. *Denote by $\chi^\star$ the target trajectory defined by (3.1) with $y = y^\star$ and $p = p^\star$ and by $y_n^\star, p_n^\star, v_n^\star$ the solutions of (3.4–3.6) with $y = y^\star$ and $\chi = \chi^\star$. One has:*

$$v_n^\star = v_{|I_n}^\star.$$

Proof. Thanks to the uniqueness of the solution of the sub-problem, it is enough to show that $v_{|I_n}^\star$ satisfies the optimality system (3.4–3.6).
First, note that $y_{|I_n}^\star$ obviously satisfies (3.4) with $v_n = v_{|I_n}^\star$. It directly follows from the definition of $\chi^\star$ (see (3.1)), that:

$$p^\star(t_{n+1}) = y^\star(t_{n+1}) - \chi^\star(t_{n+1}),$$

so that $p_{|I_n}^\star$ satisfies (3.5). Finally, Equation (3.6) is a consequence of (2.3). The result follows. □

Let HJ denote the hessian operator associated with J; there exists a strong connection between the hessian operators HJ and HJ_n of J and J_n, as indicated in the next lemma.

Lemma 2. *The hessian operator HJ_n coincides with the restriction of HJ to controls whose time supports are included in $[t_{N-1}, T]$.*

Proof. First note that J is quadratic so that HJ is a constant operator. Given an increase $\delta v \in L^2([0,T]; L^2(\Omega_c))$, we have:

$$\langle HJ(\delta v), \delta v\rangle_v = \|\delta y(T)\|_\Omega^2 + \alpha \int_0^T \|\delta v(t)\|_{\Omega_c}^2 dt,$$

where δy is the solution of

$$\left\{ \begin{array}{rcll} \partial_t \delta y - \nu \Delta \delta y & = & \mathcal{B}\delta v & \text{on } [0,T]\times\Omega \\ \delta y(0) & = & 0. & \end{array} \right. \tag{4.1}$$

Given $1 \le n \le N$, consider now an increase $\delta v_n \in L^2(I_n; L^2(\Omega_c))$. One finds in the same way that:

$$\langle HJ_n(\delta v_n), \delta v_n\rangle_{v_n} = \|\delta y_n(t_{n+1})\|_\Omega^2 + \alpha \int_{t_n}^{t_{n+1}} \|\delta v_n(t)\|_{\Omega_c}^2 dt,$$

where δy_n is the solution of

$$\left\{ \begin{array}{rcll} \partial_t \delta y_n - \nu \Delta \delta y_n & = & \mathcal{B}\delta v_n & \text{on } [t_n, t_{n+1}]\times\Omega \\ \delta y_n(t_n) & = & 0. & \end{array} \right. \tag{4.2}$$

Suppose now that $\delta v = 0$ on $[0, t_{N-1}]$, it is a simple matter to check that $\delta y \equiv 0$ over $[0, t_{N-1}]$. The restriction of δy on the interval $[t_{N-1}, T]$ thus satisfies $\delta y(t_{N-1}) = 0$ and is consequently (up to a time translation) the solution of (4.2). □

We end this section with an estimate on these hessian operators.

Lemma 3. *Given $\delta v \in L^2([0,T]; L^2(\Omega_c))$, one has:*

$$\alpha \int_0^T \|\delta v(t)\|_{\Omega_c}^2 dt \le \langle HJ(\delta v), \delta v\rangle_v \le \beta \int_0^T \|\delta v(t)\|_{\Omega_c}^2 dt, \tag{4.3}$$

where $\beta = \alpha + C/\sqrt{2}$, with C the Poincaré's constant associated with $L^2(\Omega)$.

The proof of this result is standard and given in Appendix for the sake of completeness. Because of Lemma 2, the hessian operator HJ_n also satisfies (4.3).

5. Algorithm

We are now in a position to propose a time parallelized procedure to solve (2.1–2.3). In what follows we describe the principal steps of a parallel algorithm named "SITPOC" (Serial Intermediate Targets for Parallel Optimal Control).

Algorithm 4 (SITPOC). Consider an initial control v^0 and suppose that, at step k one knows v^k. The computation of v^{k+1} is achieved as follows:

I. Compute y^k, p^k and the associated target trajectory χ^k according to (2.1), (2.2) and (3.1) respectively.
II. Solve approximately the N sub-problems (3.2) in parallel. For $n = 0, \ldots, N-1$, denote by $\tilde{v}_n^{k+1}$ the corresponding solutions and by $\tilde{v}^{k+1}$ the concatenation of $(\tilde{v}_n^{k+1})_{n=0,\ldots,N-1}$.
III. Define v^{k+1} by $v^{k+1} = (1-\theta^k)v^k + \theta^k\tilde{v}^{k+1}$, where θ^k is defined to minimize $J((1-\theta^k)v^k + \theta^k\tilde{v}^{k+1})$.

Note that we do not explain in detail here the optimization step (Step II) and rather present a general structure of our algorithm. Because of the strictly convex setting, some steps of, e.g., a gradient method or a small number of conjugate gradient method step can be used.

6. Convergence

The convergence of Algorithm 4 can be guaranteed under some assumptions. In what follows, we denote by ∇J the gradient of J.

Theorem 6.1. *Suppose that the sequence $(v^k)_{k\in\mathbb{N}}$ defined in Algorithm 4 satisfies, for all $k \geq 0$:*

$$J(v^k) \neq J(v^\infty), \tag{6.1}$$

$$\langle \nabla J(v^k), v^{k+1} - v^k \rangle_v \leq 0, \tag{6.2}$$

and

$$\|\nabla J(v^k)\|_v \leq \eta \|v^{k+1} - v^k\|_v, \tag{6.3}$$

for a given $\eta > 0$. Then $(v^k)_{k\in\mathbb{N}}$ converges linearly with a rate $(1 - \frac{2\alpha^2}{\eta^2}) \in [0,1)$ to the solution of (2.1–2.3)

Note that in the case (6.1) is not satisfied, there exists $k_0 \in \mathbb{N}$ such that $v^{k_0} = v^\infty$ and the optimum is reached in a finite number of steps.

Proof. Define the shifted functional

$$\widetilde{J}(v) = J(v) - J(v^\star),$$

and note that because of the definition of $v^\star$, one has

$$\widetilde{J}(v) = \frac{1}{2}\langle HJ(v - v^\star), v - v^\star\rangle_v \leq \frac{\beta}{2}\|v - v^\star\|_v^2. \tag{6.4}$$

Since J is quadratic, for any $v \in L^2(\Omega_c)$

$$\nabla J(v) = HJ(v - v^\star),$$

and consequently

$$\langle \nabla J(v), v - v^\star\rangle_v = \langle HJ(v - v^\star), v - v^\star\rangle_v \geq \alpha\|v - v^\star\|_v^2,$$

so that

$$\|v - v^\star\|_v \leq \frac{1}{\alpha}\|\nabla J(v)\|_v. \tag{6.5}$$

Combining (6.4) and (6.5), one gets

$$\forall v \in L^2(\Omega_c), \qquad \sqrt{\widetilde{J}(v)} \leq \gamma\|\nabla J(v)\|_v, \tag{6.6}$$

with $\gamma = \frac{1}{2\sqrt{\alpha}}$.
On the other hand, the variations in the functional between two iterations of our algorithm reads as

$$\begin{aligned} J(v^k) - J(v^{k+1}) &= \langle \nabla J(v^k), v^k - v^{k+1}\rangle_v + \frac{1}{2}\langle HJ(v^k - v^{k+1}), v^k - v^{k+1}\rangle_v \\ &\geq \langle \nabla J(v^k), v^k - v^{k+1}\rangle_v + \frac{\alpha}{2}\|v^k - v^{k+1}\|_v^2. \end{aligned}$$

Combining this last inequality with (6.2), one finds that :

$$J(v^k) - J(v^{k+1}) \geq \frac{\alpha}{2}\|v^k - v^{k+1}\|_v^2. \tag{6.7}$$

Since $\widetilde{J}(v^k) - \widetilde{J}(v^{k+1}) = J(v^k) - J(v^{k+1}) \geq 0$, we have:

$$\begin{aligned} \sqrt{\widetilde{J}(v^k)} - \sqrt{\widetilde{J}(v^{k+1})} &\geq \frac{1}{2\sqrt{\widetilde{J}(v^k)}}\left(J(v^k) - J(v^{k+1})\right) \\ &\geq \frac{\alpha}{4\sqrt{\widetilde{J}(v^k)}}\|v^k - v^{k+1}\|_v^2 \qquad (6.8) \\ &\geq \frac{\alpha}{4\gamma\|\nabla J(v)\|_v}\|v^k - v^{k+1}\|_v^2 \qquad (6.9) \\ &\geq \frac{\alpha}{4\gamma\eta\|v^k - v^{k+1}\|_v}\|v^k - v^{k+1}\|_v^2 \qquad (6.10) \\ &\geq c\|v^k - v^{k+1}\|_v, \qquad (6.11) \end{aligned}$$

where $c = \frac{\alpha}{2\gamma\eta} = \frac{\alpha^{\frac{3}{2}}}{\eta}$. Indeed (6.8) follows from (6.7), (6.9) from (6.6) and (6.10) from (6.3). It follows from the monotonic convergence of $\sqrt{\widetilde{J}}(v^k)$ that the sequence v^k is Cauchy, thus its convergence.

Let us now study the convergence rate. Define $r^k = \sum_{\ell=k}^{+\infty} \|v^{\ell+1} - v^\ell\|_v$. Summing (6.11) between k and $+\infty$, we obtain:

$$\sqrt{\widetilde{J}(v^k)} \geq cr^k.$$

Using again (6.6) and (6.3), one finds that:

$$\eta\gamma(r^k - r^{k+1}) \geq cr^k. \tag{6.12}$$

Note that this inequality implies that $1 - \frac{c}{\eta\gamma} \geq 0$. Define $C := \frac{2\alpha^2}{\eta^2} = \frac{c}{\eta\gamma}$, we have $0 < C \leq 1$. Because of (6.12):

$$(1-C)^{-k} r^k \geq (1-C)^{-(k+1)} r^{k+1},$$

and the result follows. □

We now give an example where hypothesis (6.2–6.3) are satisfied.

Corollary 6.2. *Assume that Step* II *of Algorithm* 4 *is achieved using only one step of a locally optimal step gradient method and that at step k, the algorithm is initialized with* $v_n^k := v^k_{|I_n}$*, then (*6.2–6.3*) are satisfied hence the algorithm converges to the solution of (*2.1–2.3*).*

Proof. Because of the assumptions, the optimization step (Step. II) reads:

$$\tilde{v}_n^{k+1} = v_n^k - \rho_n^k \nabla J_n(v_n^k).$$

Since the functionals J_n are quadratic, one has:

$$\rho_n^k = \frac{\|\nabla J_n(v_n^k)\|^2_{v_n}}{\langle HJ_n(\nabla J_n(v_n^k)), \nabla J_n(v_n^k)\rangle_{v_n}},$$

A first consequence of these equalities is that:

$$\langle \nabla J_n(v_n^k), \tilde{v}_n^{k+1} - v_n^k\rangle_{v_n} = -\rho_n^k \|\nabla J_n(v_n^k)\|^2_{v_n} \leq 0. \tag{6.13}$$

Moreover Lemmas 2 and 3 imply:

$$\frac{1}{\beta} \leq \rho_n^k \leq \frac{1}{\alpha}. \tag{6.14}$$

One can also obtain similar estimates of θ^k. In this view, note first that since the only iteration which is considered uses as directions of descent $\nabla J_n(v_n^k) = \nabla J(v^k)_{|I_n}$. Then:

$$\begin{aligned}
\theta^k &= -\frac{\langle \nabla J(v^k), \tilde{v}^{k+1} - v^k\rangle_v}{\langle HJ(\tilde{v}^{k+1} - v^k), \tilde{v}^{k+1} - v^k\rangle_v}, \\
&= -\frac{1}{\langle HJ(\tilde{v}^{k+1} - v^k), \tilde{v}^{k+1} - v^k\rangle_v} \sum_{n=1}^{N} \frac{1}{\rho_n^k} \|\tilde{v}_n^{k+1} - v_n^k\|^2_{v_n}.
\end{aligned}$$

Using (6.14), one deduces:

$$\frac{\alpha}{\beta} \leq \theta^k \leq \frac{\beta}{\alpha}. \tag{6.15}$$

This preliminary results will now be used to prove the theorem. The proof of (6.2), follows from (6.13):

$$\begin{aligned}\langle \nabla J(v^k), v^{k+1} - v^k \rangle_v &= \theta^k \langle \nabla J(v^k), \tilde{v}^{k+1} - v^k \rangle_v, \\ &= \theta^k \sum_{n=1}^{N} \langle \nabla J_n(v_n^k), \tilde{v}_n^{k+1} - v_n^k \rangle_{v_n} \leq 0.\end{aligned}$$

This last estimate is a consequence of (6.13). It remains to prove (6.3). We have:

$$\begin{aligned}\|v^{k+1} - v^k\|_v &= \theta^k \|\tilde{v}^{k+1} - v^k\|_v \\ &= \theta^k \sqrt{\sum_{n=1}^{N} \|\tilde{v}_n^{k+1} - v_n^k\|_{v_n}^2} \\ &= \theta^k \sqrt{\sum_{n=1}^{N} \left(\rho_n^k\right)^2 \|\nabla J_n(v_n^k)\|_{v_n}^2} \\ &\leq \frac{\alpha}{\beta} \sqrt{\sum_{n=1}^{N} \frac{1}{\beta^2} \|\nabla J_n(v_n^k)\|_{v_n}^2} \\ &= \frac{\alpha}{\beta^2} \|\nabla J(v^k)\|_v,\end{aligned}$$

and the result follows.

□

7. Parareal acceleration

The method we have presented with algorithm 4 requires in Step I two sequential resolutions of the evolution Equation (2.1) on the whole interval $[0, T]$, which does not fit with the parallel setting. In this section, we make use of the parareal algorithm to parallelize the corresponding computations.

7.1. Setting

Let us first recall the main features of the parareal algorithm. We consider the example of Equation (2.1). In order to solve in parallel an evolution equation, for the parareal scheme [4] we introduce intermediate initial conditions at times $(t_n)_{n=0,\ldots,N-1}$ that are updated iteratively. Suppose that these values (λ_n^k) are known at step k. Denote by $\mathcal{G}_n(\lambda_n)$ and $\mathcal{F}_n(\lambda_n)$ coarse and fine solutions of (3.4) at time t_{n+1} with λ_n as initial value. The update is done according to the following iteration:

$$\lambda_{n+1}^{k+1} = \mathcal{G}_n(\lambda_n^{k+1}) + \mathcal{F}_n(\lambda_n^k) - \mathcal{G}_n(\lambda_n^k).$$

We use this procedure in Step I of Algorithm 4. The idea we follow consists in merging the two procedures, i.e. doing one parareal iteration at each iteration of our algorithm.

7.2. Algorithm

We now give details on the resulting procedure. Since the evolution equations depend on the control, we replace the notations $\mathcal{G}_n(\lambda_n)$ and $\mathcal{F}_n(\lambda_n)$ by $\mathcal{G}_n(\lambda_n, v_n)$ and $\mathcal{F}_n(\lambda_n, v_n)$ respectively. As we need backward solvers to compute p, see (2.2), we also introduce $\widetilde{\mathcal{G}}_n(\mu_{n+1})$ and $\widetilde{\mathcal{F}}_n(\mu_{n+1})$ to denote coarse and fine solutions of (3.5) at time t_n with μ_{n+1} as "initial" value (given at time t_{n+1}). Note that these bakward solvers $\widetilde{\mathcal{F}_n}$ (resp: $\widetilde{\mathcal{G}_n}$) do not depend on the control.

We describe in the following the principal steps of an enhanced version of the SITPOC algorithm which we give the name "PITPOC" as Parareal Intermediate Targets for Optimal Control.

Algorithm 5 (PITPOC**).** Denote by $v_n^k = v^k_{|I_n}$. Consider a control $(v_n^0)_{n=0,\dots,N-1}$, initial values $(\lambda_n^0)_{n=0,\dots,N}$ (through forward scheme $\lambda_{n+1}^0 = \mathcal{G}_n(\lambda_n^0, v_n^0)$), final values $(\mu_n^0)_{n=1,\dots,N}$ (through backward scheme $\mu_n^0 = \widetilde{\mathcal{G}}_n(\mu_{n+1}^0)$.
Suppose that, at step k one knows v^k, $(\lambda_n^k)_{n=0,\dots,N}$ and $(\mu_n^k)_{n=1,\dots,N}$. The computation of v^{k+1}, $(\lambda_n^{k+1})_{n=0,\dots,N}$ and $(\mu_n^{k+1})_{n=1,\dots,N}$ is achieved as follows:

I. Build the target trajectory $(\chi_n^k)_{n=1,\dots,N}$ according to a definition similar to (3.1):
$$\chi_n^k = \lambda_n^k - \mu_n^k.$$
II. Solve approximately the N sub-problems (3.2) in parallel. For $n = 0, \dots, N-1$, denote by $\tilde{v}_n^{k+1}$ the corresponding solutions.
III. Define $\tilde{v}^{k+1}$ as the concatenation of the sequence $(\tilde{v}_n^{k+1})_{n=0,\dots,N-1}$.
IV. Compute $(\tilde{\lambda}_n^{k+1})_{n=0,\dots,N}$, $(\mu_n^{k+1})_{n=1,\dots,N}$ by:
$$\begin{aligned}\tilde{\lambda}_{n+1}^{k+1} &= \mathcal{G}_n(\tilde{\lambda}_n^{k+1}, \tilde{v}_n^{k+1}) + \mathcal{F}_n(\lambda_n^k, \tilde{v}_n^{k+1}) - \mathcal{G}_n(\lambda_n^k, v_n^k),\\ \mu_n^{k+1} &= \widetilde{\mathcal{G}}_n(\mu_{n+1}^{k+1}) + \widetilde{\mathcal{F}}_n(\mu_{n+1}^k) - \widetilde{\mathcal{G}}_n(\mu_{n+1}^k),\end{aligned}$$
V. Define v^{k+1} and $(\lambda_n^{k+1})_{n=0,\dots,N}$
$$\begin{aligned}v^{k+1} &= (1-\theta^k)v^k + \theta^k \tilde{v}^{k+1},\\ \lambda_n^{k+1} &= (1-\theta^k)\lambda_n^k + \theta^k \tilde{\lambda}_n^{k+1}\end{aligned}$$
where θ^k is defined to minimize
$$\frac{1}{2}\|(1-\theta^k)\lambda_N^k + \theta^k \tilde{\lambda}_N^{k+1} - y_{target}\|_\Omega^2 + \frac{\alpha}{2}\int_0^T \|(1-\theta^k)v^k(t) + \theta^k \tilde{v}^{k+1}(t)\|_{\Omega_c}^2 dt.$$
VI. $k = k+1$ and return to I.

8. Numerical Results

In this section, we test the efficiency of our method and show how robust the approach is. We consider two independent parts describing numerical results of the selected algorithm.

8.1. Setting

We consider a 2D example, where $\Omega = [0,1] \times [0,1]$ and $\Omega_c = [\frac{1}{3}, \frac{2}{3}] \times [\frac{1}{3}, \frac{2}{3}]$. The parameters related to our control problem are $T = 6.4$, $\alpha = 10^{-2}$ and $\nu = 10^{-2}$. The time interval is discretized using a uniform step $\delta t = 10^{-2}$, and an Implicit-Euler solver is used to approximate the solution of Equations (2.1–2.2). For the space discretization, we use $\mathbb{P}_1$ finite elements. Our implementation makes use of the freeware FreeFem [9] and the parallelization is achieved thanks to the Message Passing Interface library. The independent optimization procedures required in Step II are simply carried out using one step of an optimal gradient method.

8.2. Influence of the number of sub-intervals

In this section, Step II of Algorithm 4 and Algorithm 5 are achieved by using one step of an optimal step gradient method. We first test our algorithm by varying the number of sub-intervals. The evolution of the cost functional values are plotted with respect to the number of iteration (Figure 1), the number of matrix multiplication (Figure 2) and the number of wall-clock time of computation (Figure 3). We first note that Algorithm 4 actually acts as

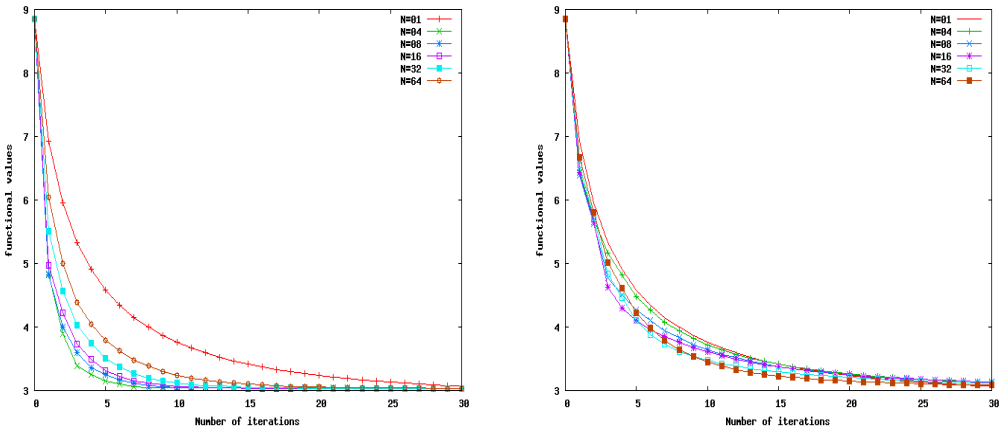

FIGURE 1. Decaying cost functional values according to the iterations count with respect to SITPOC algorithm (left) and PITPOC algorithm (right).

a preconditioner, since it improves the convergence rate of the optimization process. The introduceion of the intermediates targets allows to accelerate the decrease of the functional values, as shown in Figure 2 (left). Note that this property holds mostly for small numbers of sub-intervals, and disapears when dealing with large subdivisions. This feature is lost when considering Algorithm 5, whose convergence does not significantly depend on the number of sub-intervals that is considered, see Figure 2 (right).

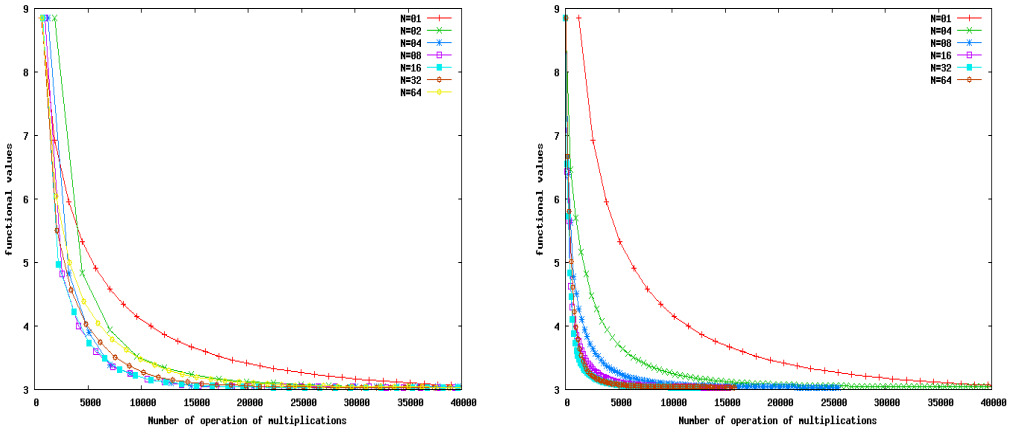

FIGURE 2. Decaying cost functional values according to the multiplication operations count with respect to SITPOC algorithm (left) and PITPOC algorithm (right).

On the contrary, Algorithm 5 achieves a good acceleration when considering the number of mutliplications involved in the computations. The corresponding results are shown in Figure 2, where the parallel operations have been counted only once. We see that Algorithm is close to the full efficiency, since the number of multiplications required to obtain a given value for the cost functional is roughly proportional to $\frac{1}{N}$.

We finally consider the wall-clock time required to carry out our algorithms. As the main part of the operations involved in the computation consists in matrix multiplications, the results we present in Figure 3 are close to the ones of Figure 2.

8.3. Influence of the number of steps in the optimization method

We now vary the number of steps of the gradient method used in Step II of our algorithm. The results are presented in Figure 4. Subdivisions of $N = 4$ and $N = 16$ intervals are considered. In both cases, we see that an increase in the number of gradient steps improves the preconditionning feature of our algorithm. However, we also observe that this strategy saturates for large numbers of gradient steps which probably reveals that the sub-problems considered in Step II are practically solved after 5 sub-iterations.

More results can be found in [10].

Appendix

For the sake of completeness, we recall here the proof of Lemma 3.
Because of (4.1) and thanks to Young's inequality, one has for all $t \in [0, T]$

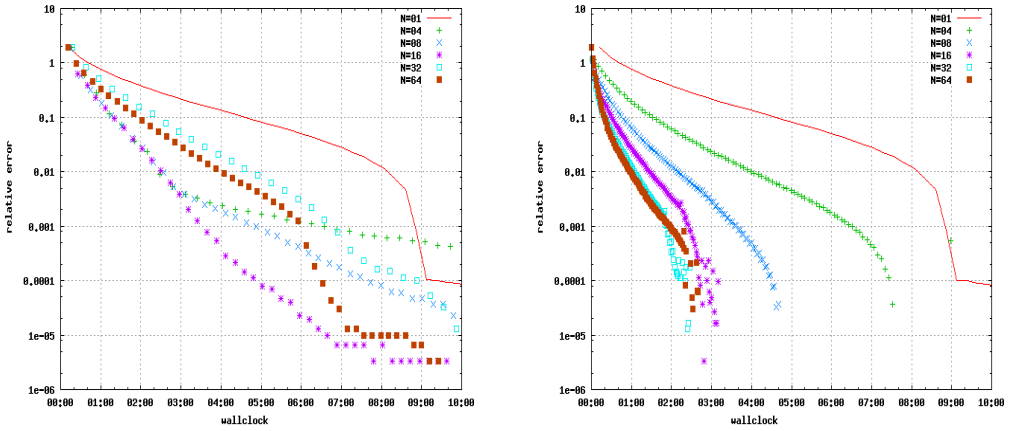

FIGURE 3. Decaying cost functional values according to elapsed real time with respect to SITPOC algorithm (left) and PITPOC algorithm (right).

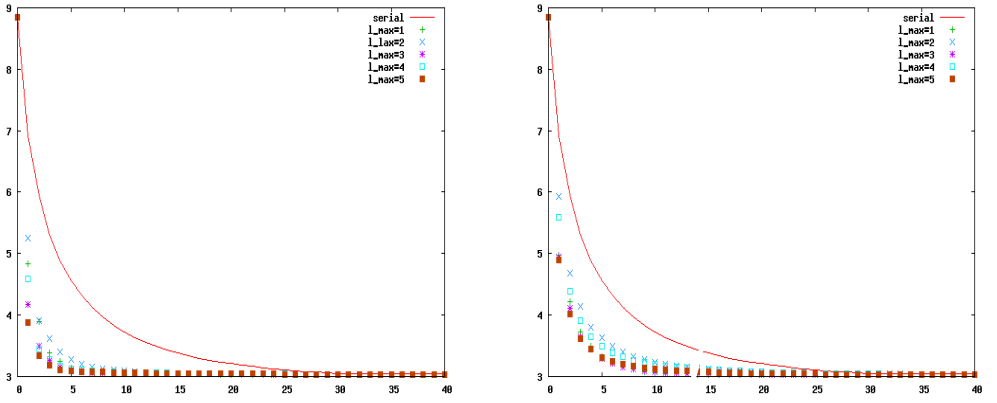

FIGURE 4. SITPOC algorithm with 4 subdivisions (left) and 16 subdivision (right): variation of the number of (lower/local)inner-iterations $\ell_{\max}$.

and all $\varepsilon > 0$:

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}\|\delta y(t)\|_\Omega^2 + \nu\|\nabla_x \delta y(t)\|_\Omega^2 &= \int_\Omega \delta y(t)\mathcal{B}\delta v(t)dt \\ &\leq \frac{1}{2}\left(\varepsilon\|\delta y(t)\|_\Omega^2 + \frac{1}{\varepsilon}\|\mathcal{B}\delta v(t)\|_v^2\right), \quad (8.1) \end{aligned}$$

where ∇_x denotes the gradient with respect to the space variable. As δy is supposed to satisfies Dirichlet conditions, one can apply Poincaré's inequality to obtain:

$$\|\delta y(t)\|_\Omega \leq C\|\nabla_x \delta y(t)\|_\Omega,$$

for a given $C > 0$. Combining this last estimate with (8.1), one gets:

$$\frac{1}{2}\frac{d}{dt}\|\delta y(t)\|_\Omega^2 \leq \left(\frac{\varepsilon}{2} - \frac{\nu}{C^2}\right)\|\delta y(t)\|_\Omega^2 + \frac{1}{2\varepsilon}\|\mathcal{B}\delta v(t)\|_v^2.$$

Now, setting $\varepsilon = \frac{2\nu}{C^2}$ gives:

$$\frac{d}{dt}\|\delta y(t)\|_\Omega^2 \leq \frac{1}{\varepsilon}\|\mathcal{B}\delta v(t)\|_v^2.$$

Since $\|\delta y(0)\|_\Omega^2 = 0$, the result follows with the fact that $\|\mathcal{B}\|_2 \leq 1$.

References

[1] G. Bal and Y. Maday A parareal time discretization for non-linear PDEs with application to the pricing of an american put, *Springer,Lect Notes Comput. Sci. Eng.* , 189-202, 2002.

[2] A. Bellen and M. Zennaro Parallel algorithms for initial value problems for nonlinear vector difference and differential equations, *J. Comput. Appl. Math.*, 25, 341-350, 1989.

[3] 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.

[4] J.-L. Lions, Virtual and effective control for distributed systems and decomposition of everything, *J. Anal. Math.* 80, 257-297 , 2000.

[5] J.-L. Lions, Y. Maday and G Turinici, Résolution d'EDP par un shéma pararréel, *C. R. Acad. Sci. Paris*, I 332 , 661-668, 2001.

[6] Y. Maday, J. Salomon and G. Turinici, Parareal in time control for quantum systems, *SIAM J. Num. Anal.*, 45 (6), 2468-2482, 2007.

[7] Y. Maday and G. Turinici, A parareal in time procedure for the control of partial differential equations, *C. R. Math. Acad. Sci. Paris 335*, 4, 387-392, 2002.

[8] T. P. Mathew, M. Sarkis and C. E. Schaerer, Analysis of block parareal preconditioners for parabolic optimal control problems, *SIAM J. Sci. Comp.*, 32 (3), 1180-1200, 2010.

[9] O. Pironneau, F. Hecht and K. Ohtsuka, Free soft : FreeFem++-mpi , http://www.freefem.org.

[10] M.-K. Riahi, Thèse de doctorat de l'université Pierre et Marie Curie, Paris6, *Submitted in december 06, 2011.*

Yvon Maday
UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, and Division of Applied Mathematics, Brown University, Providence, RI, USA.
e-mail: `maday@ann.jussieu.fr`

Mohamed-Kamel Riahi
UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.
e-mail: riahi@ann.jussieu.fr

Julien Salomon
CEREMADE, Université Paris-Dauphine, Pl. du Mal. de Lattre de Tassigny, F-75016, Paris, France.
e-mail: salomon@ceremade.dauphine.fr