C. R. Acad. Sci. Paris, Ser. I 335 (2002) 387–392

Analyse numérique/*Numerical Analysis*

A parareal in time procedure for the control of partial differential equations

Yvon Maday [a], Gabriel Turinici [b,c]

[a] **Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, Boîte courrier 187, 75252 Paris cedex 05, France**

[b] **INRIA Rocquencourt, BP 105, 78153 Le Chesnay cedex, France**

[c] **CERMICS, ENPC, Marne la Vallée, France**

Received 5 March 2002; accepted 7 June 2002

Note presented by Olivier Pironneau.

Abstract We have proposed in a previous note a time discretization for partial differential evolution equation that allows for parallel implementations. This scheme is here reinterpreted as a preconditioning procedure on an algebraic setting of the time discretization. This allows for extending the parallel methodology to the problem of optimal control for partial differential equations. We report a first numerical implementation that reveals a large interest. ***To cite this article: Y. Maday, G. Turinici, C. R. Acad. Sci. Paris, Ser. I 335 (2002) 387–392.***

Une technique de contrôle d'équations aux dérivées partielles en temps pararéel

Résumé On a proposé dans une précédente note, un schéma permettant de profiter d'une architecture parallèle pour la discrétisation en temps d'équation d'évolution aux dérivées partielles. Ce schéma est ici interprété sous un angle différent et matriciel, permettant d'étendre le concept pour le contrôle d'EDP. Les premières expériences numériques sont extrèmement encourageantes. ***Pour citer cet article : Y. Maday, G. Turinici, C. R. Acad. Sci. Paris, Ser. I 335 (2002) 387–392.***

Version française abrégée

Pour un opérateur A linéaire ou pas, agissant d'un espace de Hilbert V dans V' on considère le problème de contrôle (1) où v est un contrôle d'un espace $\mathcal{U}$ et B est un opérateur de $\mathcal{L}(\mathcal{U}; V')$. Ce problème est complété par la donnée de la fonctionnelle de coût (2) ou y^T est une cible. On souhaite étendre le schéma pararéel à ce cadre et on introduit donc une suite d'instants T_n vérifiant (3) puis les fonctions $\{y_0, y_1, \ldots, y_n, \ldots, y_{N-1}\}$ telles que y_n est la solution de (4) pour tout $n = 0, \ldots, N-1$. Il est clair que l'ensemble $\{y_0, y_1, \ldots, y_n, \ldots, y_{N-1}\}$ de (4) est lié à la solution y du problème initial (1) si pour tout $n = 0, \ldots, N-1$ on a (6). On peut ainsi interpréter dans (4) λ_n comme un contrôle virtuel (à la Lions *cf.* [2]) ce qui amène à introduire une nouvelle fonctionnelle de coût $\mathcal{J}_\varepsilon(v, \Lambda)$ comme dans (7) où

E-mail address: maday@ann.jussieu.fr (Y. Maday).

S1631-073X(02)02467-6/FLA

$\Lambda = \{\lambda_0 = y^0, \ldots, \lambda_n, \ldots, \lambda_{N-1}\}$ et $\varepsilon > 0$ est petit. La méthode du gradient pour minimiser $\mathcal{J}_\varepsilon$ conduit à introduire les états adjoints p_n, $n = N-1, \ldots, 0$, solutions de (9) et (10). On a alors (13) et donc la procédure de gradient (14). Il est clair que la convergence de cette procédure dépend de N (souvent assez grand) — et ce, indépendamment du véritable contrôle v — et donc qu'il convient de l'accélérer si l'on souhaite profiter de la décomposition en temps.

C'est là qu'intervient la procédure en temps pararéelle introduite dans [3] que l'on peut expliquer déjà sur le problème sans contrôle (15) pour lequel il ne reste que le contrôle virtuel dans la fonctionnelle de coût qui s'écrit simplement selon (16). Par exemple, si l'on considère que A est indépendant du temps, on peut introduire le propagateur $\mathcal{F}_{\Delta T}$ tel que pour tout μ donné, $\mathcal{F}_{\Delta T}(\mu)$ est la solution au temps ΔT du problème (17). On note alors que la succession de la résolution des problèmes (4), avec $B = 0$ est équivalente à la résolution du problème initial (15) si et seulement si $\lambda_n = \mathcal{F}_{\Delta T}(\lambda_{n-1})$ ou encore, sous forme matricielle comme (18) ou encore (19) avec des notations évidentes. Ce système, sous forme d'une matrice triangulaire inférieure, peut être inversé en $\mathcal{O}(N)$ opérations. Le schéma pararéel permet d'accélérer cette résolution en construisant une suite Λ^k qui converge vers la solution de (19). On introduit pour celà un propagateur grossier $\mathcal{G}_{\Delta T}$, approximation de $\mathcal{F}_{\Delta T}$, par exemple par le schéma de type Euler implicite (20) et on pose (21) qui s'écrit encore, sous forme matricielle (23) avec la matrice (22) et où le résidu $\mathrm{Res}^k = F - M\Lambda^k$.

On a montré dans [3] et [1] que cette procédure itérative converge en peu d'itérations, indépendamment de N et que les calculs les plus coûteux (reposant sur $\mathcal{F}_{\Delta T}$) peuvent être faits en parallèle. On en déduit que $\widetilde{M}^{-1}$ peut etre considérée comme proche de M^{-1}, au sens où la matrice d'amplification $\widetilde{M}^{-1}M$ est proche de l'identité.

De retour au problème de contrôle virtuel, on remarque que la résolution de (19) s'écrit $\delta\widetilde{\mathcal{J}}(\Lambda) = 0$ qui peut aussi prendre la forme (24). On remarque également que le saut $p_n^k(T_n^+) - p_{n-1}^k(T_n^-)$ dans le problème adjoint est égal au vecteur $M^*\mathrm{Res}^k$ c'est-à-dire au résidu de (24) au pas k. On prétend ainsi maintenant que $\widetilde{M}^{-1}(\widetilde{M}^{-1})^*$ est un bon préconditionneur de M^*M et on propose la méthode de gradient préconditionnée (25).

Afin de vérifier la pertinence de cette proposition, on l'a testée sur une équation d'évolution très simple posée sur l'intervalle $]0,1[$: trouver y solution de $\partial y/\partial t - y'' = v\chi$ où v est le contrôle et χ est la fonction indicatrice de $]1/2, 2/3[$. On a contrôlé cette équation sur l'intervalle de temps $]0, 100[$ partant de la condition initiale $y^0 = 10x(1-x)$ vers la cible $y^T = \sin(2\pi x)$. Les simulations fines (correspondant à $\mathcal{F}_{\Delta T}$) sont réalisées avec le pas de temps 2×10^{-2} soit sans décomposition en temps, soit avec la procédure de contrôle parallélisée avec $\Delta T = 1$. Il est interessant de noter qu'après seulement 25 itérations du schéma préconditionné, la valeur de la fonctionnelle de coût est du même ordre de grandeur qu'après 100 itérations de l'algorithme de contrôle seul (sans décomposition en temps) ; ceci conduit à une accélération d'un facteur environ 400 ! Nous ne nous attendons pas à ce que ce soit le cas pour tous les problèmes néanmoins, même avec un doublement du nombre d'itérations du gradient avec pas optimal, la procédure pararéelle donne une accélération d'un facteur proportionel à $T/\Delta T$.

Remerciements. Ce travail a été initialisé en collaboration avec J.-L. Lions et c'est avec une très grande émotion que nous souhaitons l'associer à ce travail qui s'inspire en particulier de [2].

1. Introduction

Let A be a linear or nonlinear operator from a Hilbert space V into V' and let us consider the following state equation:

$$\begin{cases} \frac{\partial y}{\partial t} + Ay = Bv, \\ y(0) = y^0, \end{cases} \tag{1}$$

where the boundary conditions are imposed implicitly, where the control $v(= v(t))$ (boundary or distributed) belongs to some space $\mathcal{U}$ and where B is an operator in $\mathcal{L}(\mathcal{U}; V')$. We assume that for any given $v \in \mathcal{U}$, this problem is well posed.

We complement this problem with a cost functional to be minimized

$$\mathcal{J}(v) = \frac{1}{2}\int_0^T \big\|v(t)\big\|_{\mathcal{U}}^2 \,\mathrm{d}t + \frac{\alpha}{2}\big\|y(T) - y^T\big\|^2, \tag{2}$$

where $\alpha > 0$, y^T is a given target and the norm is the H norm if, for instance $V \subset H \subset V'$ (where H is a pivot Hilbert space).

Let us increase slightly the complexity of this problem by decomposing time as follows

$$0 = T_0 < T_1 < \cdots < T_n = n\Delta T < T_{n+1} < \cdots < T_N = T. \tag{3}$$

Then we define the functions $\{y_0, y_1, \ldots, y_n, \ldots, y_{N-1}\}$ such that y_n is the solution of

$$\begin{cases} \frac{\partial y_n}{\partial t} + Ay_n = Bv_n \quad \text{over } (T_n, T_{n+1}), \\ y_n\big(T_n^+\big) = \lambda_n, \end{cases} \tag{4}$$

for any $n = 0, \ldots, N-1$. It is a triviality to note that the collection of solutions $\{y_0, y_1, \ldots, y_n, \ldots, y_{N-1}\}$ of (4) is connected with the solution y of the original problem (1) (in the sense that $y_n = y_{|(T_n, T_{n+1})}$) if and only if, for any $n = 0, \ldots, N-1$

$$v_n = v_{|(T_n, T_{n+1})} \quad \text{and} \quad \lambda_n = y(T_n), \tag{5}$$

or again, and more intrinsincally, if and only if, for any $n = 0, \ldots, N-1$

$$v_n = v_{|(T_n, T_{n+1})} \quad \text{and} \quad \lambda_n = y_{n-1}\big(T_n^-\big) \quad \text{with } y_{-1}\big(T_0\big) = y^0. \tag{6}$$

This way, in (4), λ_n can be interpreted as a "virtual" control (à la Lions *cf.* [2]) that leads to the following development. In order to satisfy (6) approximatively, we propose to modify slightly the cost functional $\mathcal{J}$ as follows

$$\mathcal{J}_\varepsilon(v, \Lambda) = \frac{1}{2}\int_0^T \big\|v(t)\big\|_{\mathcal{U}}^2 \,\mathrm{d}t + \frac{\alpha}{2}\big\|y_{N-1}(T) - y^T\big\|^2 + \frac{1}{2\varepsilon\Delta T}\sum_{n=1}^{N-1}\big\|y_{n-1}\big(T_n^-\big) - \lambda_n\big\|^2, \tag{7}$$

where $\Lambda = \{\lambda_0 = y^0, \ldots, \lambda_n, \ldots, \lambda_{N-1}\}$ and $\varepsilon > 0$ is small.

In order to solve this minimization problem, we compute the derivative of $\mathcal{J}_\varepsilon(v, \Lambda)$

$$\begin{aligned} \delta\mathcal{J}_\varepsilon(v, \Lambda)(\delta v, \delta\Lambda) = {} & \sum_{n=0}^{N-1}\int_{T_n}^{T_{n+1}} (v_n, \delta v_n)_{\mathcal{U}} + \alpha\big(y_{N-1}(T) - y^T, \delta y_{N-1}(T)\big) \\ & + \frac{1}{\varepsilon\Delta T}\sum_{n=1}^{N-1}\big(y_{n-1}\big(T_n^-\big) - \lambda_n, \delta y_{n-1}\big(T_n^-\big) - \delta\lambda_n\big). \end{aligned} \tag{8}$$

The introduction of an adjoint state is the standard argument for getting an expression of this derivative. Let p_{N-1} be the solution over (T_{N-1}, T_N) of

$$\begin{cases} -\frac{\partial p_{N-1}}{\partial t} + A^* p_{N-1} = 0 \quad \text{over } (T_{N-1}, T_N), \\ p_{N-1}(T) = \alpha(y_{N-1}(T) - y^T), \end{cases} \tag{9}$$

and the collection p_n, $n = N-2, N-1, \ldots, 0$, of solutions of

$$\begin{cases} -\frac{\partial p_n}{\partial t} + A^* p_n = 0 \quad \text{over } (T_n, T_{n+1}), \\ p_n\big(T_{n+1}^-\big) = \frac{1}{\varepsilon\Delta T}\big(y_n\big(T_{n+1}^-\big) - \lambda_{n+1}\big). \end{cases} \tag{10}$$

We multiply now (9) by δy_{N-1} and we integrate in time between T_{N-1} and T_N, then integrate by parts and use (4) so as to obtain

$$-\big(p_{N-1}(T^-),\delta y_{N-1}(T^-)\big)+\big(p_{N-1}(T_{N-1}^+),\delta y_{N-1}(T_{N-1}^+)\big)+\int_{T_{N-1}}^{T_N}(p_{N-1},B\delta v_{N-1})=0$$

another use of (9) gives

$$-\alpha\big(y_{N-1}(T)-y^T,\delta y_{N-1}(T^-)\big)+\big(p_{N-1}(T_{N-1}^+),\delta y_{N-1}(T_{N-1}^+)\big)+\int_{T_{N-1}}^{T_N}\big(B^*p_{N-1},\delta v_{N-1}\big)=0 \quad (11)$$

similarly, starting from (10) over (T_n,T_{n+1}) we obtain for any $n=0,\dots,N-2$

$$-\frac{1}{\varepsilon\Delta T}\big(y_n(T_{n+1}^-)-\lambda_{n+1},\delta y_n(T_{n+1}^-)\big)+\big(p_n(T_n^+),\delta y_n(T_n^+)\big)+\int_{T_n}^{T_{n+1}}(B^*p_n,\delta v_n)=0. \quad (12)$$

Using this in (8) then yields

$$\begin{aligned}\delta\mathcal{J}_\varepsilon(v,\Lambda)(\delta v,\delta\Lambda)&=\sum_{n=0}^{N-1}\int_{T_n}^{T_{n+1}}(v_n+B^*p_n,\delta v_n)_{\mathcal{U}}\\&\quad+\sum_{n=0}^{N-1}\big(p_n(T_n^+),\delta y_n(T_n^+)\big)-\frac{1}{\varepsilon\Delta T}\sum_{n=1}^{N-1}\big(y_{n-1}(T_n^-)-\lambda_n,\delta\lambda_n\big)\end{aligned}$$

recalling first that $p_{n-1}(T_n^-)=\frac{1}{\varepsilon\Delta T}(y_{n-1}(T_n^-)-\lambda_n)$, second that $\delta y_n(T_n^+)=\delta\lambda_n$, we get

$$\begin{aligned}\delta\mathcal{J}_\varepsilon(v,\Lambda)(\delta v,\delta\Lambda)&=\sum_{n=0}^{N-1}\int_{T_n}^{T_{n+1}}(v_n+B^*p_n,\delta v_n)_{\mathcal{U}}+\sum_{n=0}^{N-1}\big(p_n(T_n^+),\delta y_n(T_n^+)\big)-\sum_{n=1}^{N-1}\big(p_{n-1}(T_n^-),\delta\lambda_n\big)\\&=\sum_{n=0}^{N-1}\int_{T_n}^{T_{n+1}}(v_n+B^*p_n,\delta v_n)_{\mathcal{U}}+\sum_{n=1}^{N-1}\big(p_n(T_n^+)-p_{n-1}(T_n^-),\delta\lambda_n\big),\end{aligned} \quad (13)$$

since $\delta\lambda_0=0$. From these computations, we can easily implement a gradient method (or better a conjugate gradient method) an iteration of which reads: assume $y_n^k,\ p_n^k,\ v_n^k,\ \lambda_n^k$ are known, then

$$\begin{aligned}v_n^{k+1}&=v_n^k-\rho\big(v_n^k+B^*p_n^k\big)\quad\text{in }(T_n,T_{n+1}),\ n=0,\dots,N-1,\\\lambda_n^{k+1}&=\lambda_n^k-\rho\big(p_n^k(T_n^+)-p_{n-1}^k(T_n^-)\big),\quad n=1,\dots,N-1.\end{aligned} \quad (14)$$

It is quite easy to realize that the speed of convergence of the latter algorithm depends on the number N, of time steps T_n. Indeed, the transfer of information between time 0 and time T for y and between time T and time 0 for p can only be done by successive iterations through the subintervals (T_n,T_{n+1}), and requires at least N steps. Actually, the (real) control is not involved in this N-dependancy since for a standard control problem this gradient procedure converges quite rapidly. The dependancy here clearly comes from the fact that we have decomposed time into pieces. In order to understand what kind of preconditioner can be added to the previous iterative algorithm, we shall investigate in the next section the (only) virtual control. This is done by letting $B=0$ and $\alpha=0$.

2. The totally virtual problem

What we want to solve here is thus simply

$$\begin{cases}\frac{\partial y}{\partial t}+Ay=0,\\ y(0)=y^0\end{cases} \quad (15)$$

over the time interval $(0,T)$. Let us assume that it admits a unique solution y. The method of resolution through the virtual control involves a decomposition of the time interval. It is interesting to note that the cost functional becomes a function of Λ only: i.e. $\mathcal{J}_\varepsilon(v,\Lambda)$ can be identified with $\widetilde{\mathcal{J}(\Lambda)}$ since the control v is multiplied by zero and $\varepsilon\Delta T$ is just a multiplicative factor

$$\widetilde{\mathcal{J}(\Lambda)} = \sum_{n=1}^{N-1} \left\| y_{n-1}\left(T_n^-\right) - \lambda_n \right\|^2. \tag{16}$$

The minimum of $\widetilde{\mathcal{J}}$ is zero and is obtained by the choice $\lambda_n = y(T_n)$. We shall not use this information however but try instead to develop a link between this frame and the parareal algorithm developped in [3]. Actually, let us assume that A is time independent so that we can introduce the (time invariant) propagator $\mathcal{F}_{\Delta T}$ such that, for any given μ, $\mathcal{F}_{\Delta T}(\mu)$ is the solution, at time ΔT of the problem

$$\begin{cases} \frac{\partial y}{\partial t} + Ay = 0, \\ y(0) = \mu. \end{cases} \tag{17}$$

With this notation, it is important to notice that the succession of resolution of problems (4), with $B=0$ is equivalent to the resolution of the initial problem (15) if and only if $\lambda_n = \mathcal{F}_{\Delta T}(\lambda_{n-1})$ as noted in (6), or again, in a matricial form

$$\begin{pmatrix} \mathrm{Id} & 0 & \dots & 0 \\ -\mathcal{F}_{\Delta T} & \mathrm{Id} & 0 & \dots \\ 0 & -\mathcal{F}_{\Delta T} & \mathrm{Id} & \dots \\ 0 & \dots & -\mathcal{F}_{\Delta T} & \mathrm{Id} \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \dots \\ \lambda_{N-1} \end{pmatrix} = \begin{pmatrix} y^0 \\ 0 \\ \dots \\ 0 \end{pmatrix} \tag{18}$$

that can also be written, with obvious notations

$$M\Lambda = F. \tag{19}$$

The standard inversion of this triangular system involves $\mathcal{O}(N)$ resolutions of system (17). In order to accelerate, we have introduced an iterative procedure — the parareal scheme — that allows to construct a sequence Λ^k that converges toward the exact solution of (19). This procedure involves a coarse propagator $\mathcal{G}_{\Delta T}$, that is an approximation of $\mathcal{F}_{\Delta T}$, for instance through an implicit Euler scheme as follows

$$\frac{\mathcal{G}_{\Delta T}(\mu) - \mu}{\Delta T} + A\mathcal{G}_{\Delta T}(\mu) = 0. \tag{20}$$

It is defined as follows (*see* [1]) for this interpretation)

$$\lambda_{n+1}^{k+1} = \mathcal{G}_{\Delta T}\left(\lambda_n^{k+1}\right) + \mathcal{F}_{\Delta T}\left(\lambda_n^k\right) - \mathcal{G}_{\Delta T}\left(\lambda_n^k\right), \tag{21}$$

where it has to be noticed that, assuming Λ^k is known, the computation of Λ^{k+1} involves a parallel procedure (corresponding to the expensive computation of the $\mathcal{F}_{\Delta T}(\lambda_n^k)$) and a sequential one (corresponding to the fast computation of $\mathcal{G}_{\Delta T}(\lambda_n^{k+1})$).

By introducing the matrix

$$\widetilde{M} = \begin{pmatrix} \mathrm{Id} & 0 & \dots & 0 \\ -\mathcal{G}_{\Delta T} & \mathrm{Id} & 0 & \dots \\ 0 & -\mathcal{G}_{\Delta T} & \mathrm{Id} & \dots \\ 0 & \dots & -\mathcal{G}_{\Delta T} & \mathrm{Id} \end{pmatrix} \tag{22}$$

this iterative procedure takes the matricial form

$$\Lambda^{k+1} = \Lambda^k + \widetilde{M}^{-1}\,\mathrm{Res}^k, \tag{23}$$

where the residual Res^k is defined by $\mathrm{Res}^k = F - M\Lambda^k$.

It is proven and numerically checked that this method converges rapidly, independently of N, whenever the governing part in A is linear positive definite. It results that $\widetilde{M}^{-1}$ can be considered as close to M^{-1}, in the sense that the amplification matrix $\widetilde{M}^{-1}M$ is close to Identity.

3. Back to the control problem

Let us go back now to our virtual control problem. It can be easily guessed, due to the symmetric form of the Lagrange equations arising from (9), (10) that the resolution of $\delta\widetilde{\mathcal{J}}(\Lambda) = 0$ can also be writen as

$$M^*M\Lambda = M^*F. \tag{24}$$

Indeed, the vector of jumps $p_n^k(T_n^+) - p_{n-1}^k(T_n^-)$ in the dual state is exactely equal to the vector $M^*\,\mathrm{Res}^k$ i.e. to the residual of (24) at step k. The reason why the original gradient scheme is slow comes from the fact that the conditionning of M is $\mathcal{O}(N)$. The fact that we have produced a good preconditioner for M allows to forsee that $\widetilde{M}^{-1}(\widetilde{M}^{-1})^*$ may be a good preconditionner for M^*M so that, going back to the original control problem, we propose the following preconditionned gradient method

$$\begin{aligned} v_n^{k+1} &= v_n^k - \rho\big(v_n^k + B^*p_n\big) \quad \text{in } (T_n, T_{n+1}),\ n = 0, \dots, N-1, \\ \Lambda^{k+1} &= \Lambda^k - \rho\big[\widetilde{M}^{-1}\big(\widetilde{M}^{-1}\big)^*\big]\big(p_n^k\big(T_n^+\big) - p_{n-1}^k\big(T_n^-\big)\big)_{n-1}^{N-1}. \end{aligned} \tag{25}$$

In order to check the pertinence of this approach, we have tested on a very simple example of a one-dimensional parabolic equation on the interval $]0,1[$: find y such that $\frac{\partial y}{\partial t} - y'' = v\chi$ where v is the control and χ is the indicator of $]1/2, 2/3[$. We have simulated this equation over the time interval $]0, 100[$ from the initial condition $y^0 = 10x(1-x)$ so as to drive it to the target $y^T = \sin(2\pi x)$. The fine simulations (corresponding to $\mathcal{F}_{\Delta T}$ are performed with the time step 2×10^{-2} either without decomposition in time or by using the preconditionned controled problem with $\Delta T = 1$. It is impressive (and not totally understood) to obtain that after 25 iterations of the preconditionned scheme, the cost function is about the same as after 100 iterations of the plain control procedure. We have used a gradient method with optimal step. The parareal scheme thus achieves a speedup of more that 400! We do not expect that the parareal scheme will provide an improvement of the iteration count (that may be due here to the long time evolution with a quite viscous term) on more complex problems, nevertheless we expect that the number will be about the same so that it will allow for a speedup proportional to $T/\Delta T$.

References

[1] L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zerah, Parallel in time molecular dynamics simulations, CEMRACS'01 Proceedings, submitted.

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

[3] J.-L. Lions, Y. Maday, G. Turinici, Résolution d'EDP par un schéma en temps « pararéel », C. R. Acad. Sci. Paris, Série I 332 (7) (2001) 661–668.