

Sur le contrôle parallèle des systèmes distribués

J.L. Lions, O. Pironneau

Résumé — On présente des algorithmes de contrôle de systèmes distribués qui peuvent être calculés en parallèle, avec le "minimum" de transfert de données entre "actionneurs". Ces algorithmes sont basés, entre autres, sur les méthodes systématiques de décomposition des problèmes aux limites introduits dans [1]. Tout cela est esquissé ici dans deux situations simples, au n°1 pour un problème stationnaire, au n°2 dans un cas d'évolution (parabolique). Les méthodes sont extrêmement générales et s'adaptent aux problèmes de contrôle non linéaire et aux équations d'évolution de type quelconque, comme on indiquera dans plusieurs publications séparées [2], [4].

On the Control of Distributed Systems in Parallel

Abstract — We introduce, in a rather systematic manner, control algorithms for distributed systems which are "parallel" and "local". We use the general decomposition methods introduced in [1] and we show on two examples how these methods can be used for control problems. We choose firstly an elliptic problem for a linear second order operator, with Boundary Control. This example has been chosen as simple as possible to present the method, which is otherwise completely general for *linear* problems. Other methods for non linear problems can also be based on [1], as we shall explain elsewhere. The second example presented here concern a linear parabolic equation. A decomposition parallel algorithm is given. One can apply a further decomposition of the Riccati type, as indicated in the text and as it will be presented in detail elsewhere [2], [4].

1. Le cas stationnaire Soit Ω un ouvert de \mathbb{R}^d , de frontière $\Gamma = \partial\Omega$. Dans Ω , on considère l'opérateur elliptique A défini par la forme bilinéaire continue sur $V = H^1(\Omega)$ par

$$a(y, \hat{y}) = \sum_{k,l} \int_{\Omega} a_{kl} \frac{\partial y}{\partial x_l} \frac{\partial \hat{y}}{\partial x_k} dx + \int_{\Omega} a_o y \hat{y} dx \quad (1.1)$$

où

$$\begin{aligned} a_{kl} &\in C^o(\bar{\Omega}), a_o \in L^\infty(\Omega), \\ \sum_{k,l} a_{kl} \xi_k \xi_l &\geq c \sum_k \xi_k^2, \quad c > 0, \\ a_o(x) &\geq c. \end{aligned}$$

Remarque 1.1 La forme $a(y, \hat{y})$ est symétrique *ou non*. Elle peut aussi contenir des termes du 1er ordre. \square

On suppose que l'on peut agir sur l'état du système par un *contrôle frontière* v . Plus précisément, soit

$$\mathcal{S} \subset \Gamma \text{ une portion de } \Gamma. \quad (1.2)$$

Le contrôle v est concentré sur \mathcal{S} :

$$v \in L^2(\mathcal{S}). \quad (1.3)$$

On pose

$$s(v, \hat{y}) = \int_S v \hat{y} d\Gamma \quad (1.4)$$

ce qui définit une forme $\hat{y} \rightarrow s(v, \hat{y})$ linéaire continue sur V .

Remarque 1.2 Il suffirait que v soit dans $H^{-\frac{1}{2}}(\Gamma)$ (avec les notations usuelles des espaces de Sobolev d'ordre fractionnaire). \square

L'état $y = y(v)$ du système est alors donné par

$$\begin{cases} y \in V, \\ a(y, \hat{y}) = s(v, \hat{y}) \quad \forall \hat{y} \in V. \end{cases} \quad (1.5)$$

On introduit maintenant la fonction coût

$$J(v) = \frac{1}{2}s(v) + \frac{1}{2}c(y - y_o), \quad y = y(v) \quad (1.6)$$

où

$$\begin{aligned} s(v) &= s(v, v), \\ c(f) &= \alpha \int_{\mathcal{G}} f^2 dx, \quad \alpha > 0, \quad \mathcal{G} \text{ ouvert contenu dans } \Omega \end{aligned} \quad (1.7)$$

et où y_o est donné dans $L^2(\mathcal{G})$. \square

Remarque 1.3 Tout ce qui suit vaut pour d'autres fonctions coûts, d'autres conditions aux limites etc. \square

On cherche

$$\inf_{v \in L^2(S)} J(v), \quad . \quad (1.8)$$

Le problème admet une solution unique. Cf. [3]. C'est cette solution que nous nous proposons d'approcher par un algorithme parallèle. \square

Remarque 1.4 Ce qui suit peut s'adapter au cas où il y a des *contraintes* sur v . Nous reviendrons là-dessus. \square

On considère maintenant (avec les notations de [1]) une famille d'ouverts Ω_i , $i = 1, \dots, N$ tels que

$$\Omega = \cup \Omega_i. \quad (1.9)$$

Il y a *recouvrement* des ouverts (overlapping) : Pour tout i , il existe au moins un j tel que

$$\Omega_j \cap \Omega_i \neq \emptyset.$$

Remarque 1.5 Ce qui suit s'adapte au cas où la famille des Ω_i est *sans recouvrement*. Alors $\bar{\Omega} = \cup \bar{\Omega}_i$. \square

On définit :

$$V_i = H^1(\Omega_i)$$

$$a_i(y_i, \hat{y}_i) = \int_{\Omega_i} \sum_{k,l} a_{kl} \frac{\partial y_i}{\partial x_l} \frac{\partial \hat{y}_i}{\partial x_k} dx + \int_{\Omega_i} a_o y_i \hat{y}_i dx, \quad (1.10)$$

$$s_i(v_i, \hat{y}_i) = \int_{\mathcal{S}} \sigma_i v_i \hat{y}_i d\Gamma, \quad (1.11)$$

où

$$\sigma_i = \text{famille de fonctions sur } \mathcal{S}, \sigma_i \text{ à support dans } \mathcal{S} \cap \Omega_i,$$

$$\sum_{i=1}^N \sigma_i(x) = 1 \quad \text{sur } \mathcal{S}, \quad (1.12)$$

et où $v_i \in L^2(\mathcal{S})$. \square

Remarque 1.6 "En général" $\bar{\Omega}_i$ ne rencontre pas \mathcal{S} , de sorte que $\sigma_i \neq 0$ pour seulement quelques incidences "i". \square

On désigne par $\delta\Omega_i$ la frontière de $\partial\Omega_i$ dont on enlève éventuellement la partie sur Γ . On considère les équations

$$a_i(y_i, \hat{y}_i) = s_i(v_i, \hat{y}_i) + (\lambda_i, \hat{y}_i) \quad (1.13)$$

où

$$(\lambda_i, \hat{y}_i) = \int_{\delta\Omega_i} \lambda_i \hat{y}_i d\Gamma, \quad \lambda_i \in L^2(\delta\Omega_i). \quad (1.14)$$

Désignons maintenant par r_{ji} l'opérateur de trace de $H^1(\Omega_i)$ sur $\delta(\Omega_i \cap \Omega_j) =$ frontière de $\Omega_i \cap \Omega_j$ dont on enlève éventuellement les parties sur Γ . Alors

$$\left\{ \begin{array}{l} \text{si les } \lambda_i \text{ dans (1.13) sont tels que les solutions } y_i \text{ vérifient} \\ r_{ji} y_i = r_{ij} y_j \quad \forall i, j \\ \text{alors} \\ y_i = \text{restriction à } \Omega_i \text{ de } y, \\ y = y(v) = \text{solution de (1.5) pour } v = \sum \sigma_i v_i. \end{array} \right. \quad (1.15)$$

On introduit une partition de l'unité sur \mathcal{G} par

$$\tau_i \text{ à support dans } \mathcal{G} \cap \Omega_i, \sum \tau_i = 1 \text{ sur } \mathcal{G} \quad (1.16)$$

(naturellement $\tau_i = 0$ si $\mathcal{G} \cap \Omega_i = \emptyset$), et l'on pose

$$c_i(f_i) = \int_{\mathcal{G}} \tau_i f_i^2 dx. \quad (1.17)$$

On peut maintenant introduire

$$K(v, \lambda) = \frac{1}{2} \sum_{i=1}^N s_i(v_i) + \frac{1}{2} \sum_{i=1}^N c_i(y_i - y_{oi}) + \frac{1}{2} \sum_{i=1}^N \|\lambda_i\|_{L^2(\delta\Omega_i)}^2, \quad (1.18)$$

où

$$\begin{aligned} v &= \{v_i\} \in \Pi L^2(\mathcal{S} \cap \bar{\Omega}_i) \text{ (seul un petit nombre d'indice "i" interviennent ici),} \\ \lambda &= \{\lambda_i\} \in \Pi L^2(\delta\Omega_i), \\ y_{oi} &= \text{restriction de } y_o \text{ à } \Omega_i \cap \mathcal{G}. \end{aligned}$$

Le problème (1.8) peut être remplacé par la recherche de

$$\inf_{v, \lambda} K(v, \lambda), \quad (1.19)$$

sous les contraintes sur l'état $r_{ji}y_i = r_{ij}y_j$ (cf. (1.15)). \square

Remarque 1.7 Après ces transformations, les problèmes d'analyse *sans* contrôle et *avec* contrôle entrent tous dans le même moule. "Sans contrôle", on a un problème de contrôle avec les *contrôles virtuels* λ . "Avec contrôle", on a deux familles de contrôles, les contrôles virtuels λ et les *contrôles effectifs* v . Les contrôles virtuels λ n'ont pas de contraintes, ou des contraintes linéaires simples (cf. [1], Exemple 3). Les contrôles effectifs v *peuvent* avoir des contraintes, imposées par la physique ou par l'économie. *A ces différences près, toutes les méthodes introduites dans [1] s'adaptent à la situation générale présentée. Nous en donnons un exemple.* \square

Pour tenir compte des contraintes sur l'état, on remplace (1.19) par

$$\inf_{v, \lambda} \sup_q K(v, \lambda) + \sum_{i,j} (q_{ji}, r_{ji}y_i - r_{ij}y_j)_{H_{ij}}, \quad (1.20)$$

où $H_{ij} = L^2(\delta(\Omega_i \cap \Omega_j))$, $q = \{q_{ji}\} \in \Pi H_{ij}$.

Utilisant un résultat de min-max, on introduit

$$\mathcal{J}(q) = -\inf_{v, \lambda} [K(v, \lambda) + \sum_{i,j} (q_{ji}, r_{ji}y_i - r_{ij}y_j)_{H_{ij}}] \quad (1.21)$$

et on arrive à définir un algorithme de descente vers $\inf \mathcal{J}(q)$. \square

Si l'on introduit les états adjoints p_i par

$$a_i^*(p_i, \hat{p}_i) = c_i(y_i - y_{oi}, \hat{p}_i) + \sum_j (q_{ji} - q_{ij}, r_{ji}\hat{p}_i)_{H_{ij}} \quad (1.22)$$

alors le inf dans (1.21) est atteint par $v_i = -p_i$, $\lambda_i = -p_i$ (il s'agit des traces de p_i sur $\mathcal{S} \cap \partial\Omega_i$ ou sur $\delta\Omega_i$). On obtient ainsi

$$\mathcal{J}(q) = \frac{1}{2} \sum_i s_i(p_i) + \frac{1}{2} \sum \|p_i\|_{L^2(\delta\Omega_i)}^2 + \frac{1}{2} \sum_i e_i(y_i) - c \quad (1.23)$$

où

$$c = \text{constante} = \frac{1}{2} \sum c_i(y_{oi}) = \frac{1}{2} \int_{\mathcal{G}} y_o^2 dx.$$

On calcule alors $\delta\mathcal{J}(q)$ et on en déduit *l'algorithme de descente*

$$q_{ji}^{m+1} = q_{ji}^m + \rho(r_{ji}y_i^m - r_{ij}y_j^m) \quad (1.24)$$

qui est implémenté en parallèle. \square

Exemple. Nous avons testé la méthode sur le problème

$$\min_{v \in L^2(\Gamma)} J(v) = \frac{1}{2} \int_{\mathcal{G}} |y - y_o|^2 + \frac{\epsilon}{2} \int_S v^2$$

avec $y - y_o \in U = \{\hat{y} \in H^1(\Omega) : \hat{y}|_{\Gamma \setminus S} = 0\}$ et

$$a(y, \hat{y}) \equiv \int_{\Omega} (\mu y \hat{y} + \nu \nabla y \cdot \nabla \hat{y}) = \int_{\Omega} f \hat{y} + \int_S v \cdot \hat{y}, \quad \forall \hat{y} \in U$$

$\mu = 3.3$, $\nu = 2.2$, $f = (\mu + 2\nu) \sin(x + y)$, $y_o = \sin(x + y)$.

Le domaine Ω est divisé en 3 rectangles se recouvrant partiellement,

$$\Omega_1 = (0, 3) \times (0, 1), \quad \Omega_2 = (-1, 0.3) \times (0, 1), \quad \Omega_3 = (-0.3, 0.7) \times (0.7, 2)$$

La frontière \mathcal{S} est composée de la partie verticale à gauche dans $\partial\Omega_1$ plus la partie verticale à droite dans $\partial\Omega_2$ plus la partie horizontale en haut dans $\partial\Omega_3$:

$$\mathcal{S} = \partial\Omega \cap (\{x = -1\} \cup \{x = 3\} \cup \{y = 2\})$$

Nous avons choisi pour \mathcal{G} une partie de la réunion des intersections des Ω_i

$$\mathcal{G} = (\cup_{i \neq j} (\Omega_i \cap \Omega_j)) \cap \{x, y : y > 0.5\}$$

Les données ont été choisies pour que la solution exacte soit connue pour $\epsilon = 0$: $y = y_o = \sin(x + y)$, $v = \cos(x + y)(n_1 + n_2)$, (n_1, n_2) étant la normale à $\partial\Omega$.

Le découplage est fait par minimisation des sauts des dérivées normales aux interfaces (exemple 2 dans [1]) :

$$\sup_q \inf_{v, \lambda} J(q, \lambda, v) = \frac{1}{2} \left[\int_S \epsilon v^2 + \sum_i \int_{G_i} |y_i - y_o|^2 + \sum_{ij} \int_{\Gamma_{ij}} (\lambda_{ij}^2 + 2q_{ij}(y_i - y_j)) \right]$$

avec

$$a_i(y_i, \hat{y}) + \sum_j \int_{\Gamma_{ij}} \lambda_{ij} \hat{y} = \int_{\Omega_i} f \hat{y} + \int_{S_i} v \hat{y} \quad \forall \hat{y} \in U_i, \quad y_i - y_o|_{\Omega_i} \in U_i = U \cap H^1(\Omega_i)$$

Les sommations en i sont avec $i = 1, 2, 3$ et en ij avec $ij = 12, 21, 23, 32, 13, 31$. Par ailleurs $\Gamma_{ij} = (\partial\Omega_i) \cap (\partial(\Omega_i \cap \Omega_j))$ et $\mathcal{G}_i = \mathcal{G} \cap \Omega_i$

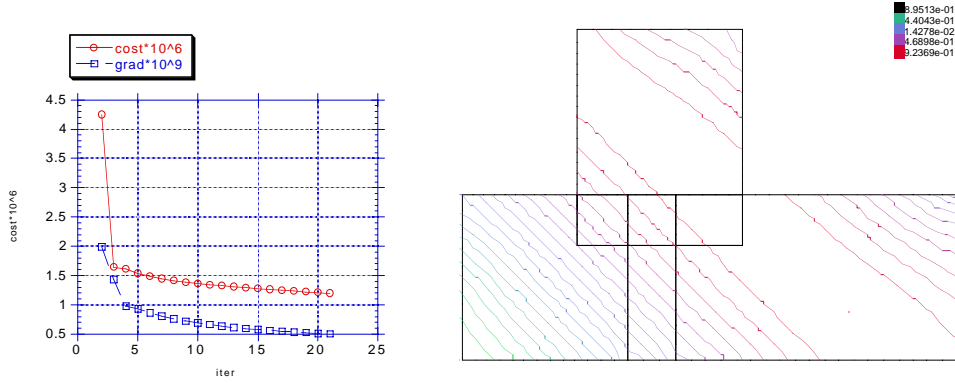


Figure 1 Résultats pour $\epsilon = 0.01$. Courbe de convergence et lignes de niveau de la solution calculée. A convergence les lignes de niveau sont des droites.

2. Un problème d'évolution On conserve les notations du n°1 (avec cette fois a_o de signe quelconque). L'état du système est donné par la solution de l'équation parabolique

$$\left(\frac{\partial y}{\partial t}, \hat{y}\right) + a(y, \hat{y}) = s(v, \hat{y}) \quad \forall \hat{y} \in V, \quad (2.1)$$

$$y \in L^2(0, T; V) \quad (2.2)$$

$$y(0) = 0, \quad (2.3)$$

où $(f, g) = \int_{\Omega} f g dx$, $T > 0$ donné, et

$$v \in L^2(0, T; L^2(S)). \quad (2.4)$$

Soit $y = y(v) = y(t; v)$ la solution unique de (2.1), (2.2), (2.3).

On définit alors la fonction coût

$$J(v) = \frac{1}{2} \int_0^T s(v) dt + \frac{1}{2} c(y(T) - y^T) \quad (2.5)$$

où y^T est donné dans $L^2(\Omega)$. On considère le problème

$$\inf J(v), \quad (2.6)$$

qui admet une solution unique [3]. Nous cherchons des algorithmes décentralisés, (décomposés), parallèles pour approcher cette solution. \square

On introduit les équations (comparer à (1.13))

$$\begin{cases} \left(\frac{\partial y_i}{\partial t}, \hat{y}_i\right)_{\Omega_i} + a_i(y_i, \hat{y}_i) = s_i(v_i, \hat{y}_i) + (\lambda_i, \hat{y}_i)_{L^2(\delta\Omega_i)} & \forall \hat{y}_i \in V_i, \\ y_i \in L^2(0, T; V_i), \\ y_i(0) = 0 \end{cases} \quad (2.7)$$

où

$$\lambda_i \in L^2(0, T; L^2(\delta\Omega_i)), \quad v_i \in L^2(0, T; L^2(\mathcal{S} \cap \partial\Omega_i)), \quad (f, g)_{\Omega_i} = \int_{\Omega_i} fg dx.$$

Chaque équation (2.7) admet une solution unique

$$\begin{aligned} y_i &= y_i(v, \lambda) = y_i(t; v, \lambda), \\ v &= \{v_i\} \in \Pi L^2(0, T; L^2(\mathcal{S} \cap \partial\Omega_i)), \\ \lambda &= \{\lambda_i\} \in \Pi L^2(0, T; L^2(\delta\Omega_i)), \end{aligned}$$

On introduit alors

$$\begin{aligned} K(v, \lambda) &= \frac{1}{2} \sum \int_0^T s_i(v_i) dt + \frac{1}{2} \sum \int_0^T \|\lambda_i\|_{L^2(\delta\Omega_i)}^2 dt + \\ &+ \frac{1}{2} \sum_i c_i (y_i(T) - y_i^T) \end{aligned} \quad (2.8)$$

où $y_i^T =$ restriction de y^T à Ω_i .

Le problème (2.6) est ainsi transformé en la recherche de

$$\left| \begin{array}{l} \inf_{v, \lambda} K(v, \lambda), \\ \text{avec les contraintes sur l'état} \\ r_{ji} y_i = r_{ij} y_j \quad \forall i, j. \end{array} \right. \quad (2.9)$$

Remarque 2.1 Toutes les méthodes présentées dans [1] s'adaptent à la situation présente, comme on l'a déjà indiqué dans la Remarque 1.7. \square

Remarque 2.2 Jusqu'ici ce qui a été présenté s'adapte aux *problèmes d'évolution non linéaires*. Nous présenterons dans un article séparé le cas des équations de Navier-Stokes. \square

Remarque 2.3 Tout ce qui est présenté ici s'adapte aux équations d'évolution de type Petrowsky (ou hyperbolique) ou de Schroedinger. \square

Dans le cas linéaire présent, on peut remplacer (2.9) par

$$\inf_{v, \lambda} \sup_q [K(v, \lambda) + \sum_{i, j} \int_0^T (q_{ji}, r_{ji} y_i - r_{ij} y_j)_{H_{ij}} dt] \quad (2.10)$$

où

$$q = \{q_{ji}\} \in \Pi L^2(0, T; H_{ij}).$$

On introduit p_i par

$$-\left(\frac{\partial p_i}{\partial t}, \hat{p}_i\right)_{\Omega_i} + a_i^*(p_i, \hat{p}_i) = \sum_j (q_{ji} - q_{ij}, \hat{p}_i)_{H_{ij}}, \quad \forall \hat{p}_i \in V_i, \quad (2.11)$$

$$p_i(T) = \alpha \tau_i(y_i(T) - y_i^T) \quad (2.12)$$

$$p_i \in L^2(0, T; V_i). \quad (2.13)$$

Alors le inf dans l'expression entre crochets de (2.10) est atteint pour $v_i + p_i = 0$, $\lambda_i + p_i = 0$ et tout revient à trouver un *algorithme de descente* pour

$$\left| \begin{aligned} \mathcal{J}(q) &= \frac{1}{2} \sum_i \int_0^T s_i(p_i) dt + \frac{1}{2} \sum_i \int_0^T \|\lambda_i\|_{L^2(\Omega_i)}^2 + \frac{1}{2} \sum c_i(y_i(T)) - \\ &- \text{constante} (= \frac{1}{2} \|y^T\|_{L^2(\Omega)}^2). \end{aligned} \right. \quad (2.14)$$

On obtient

$$q_{ji}^{m+1} = q_{ji}^m + \rho(r_{ji}y_i^m - r_{ij}y_j^m) \quad (2.15)$$

analogues à (1.24) mais où maintenant les fonctions dépendent de t . \square

Remarque 2.4 Tenant compte de $v_i + p_i = 0$, $\lambda_i + p_i = 0$, le système (2.7) s'écrit

$$\left| \begin{aligned} \left(\frac{\partial y_i}{\partial t}, \hat{y}_i \right)_{\Omega_i} + a_i(y_i, \hat{y}_i) + s_i(p_i, \hat{y}_i) + (p_i, \hat{y}_i)_{L^2(\delta\Omega_i)} &= 0, \\ y_i(0) &= 0 \end{aligned} \right. \quad (2.16)$$

les autres conditions étant inchangées.

On peut alors décomposer le système (2.11), (2.16), (2.12), cette fois à "i" fixé, comme dans [3], en

$$p_i = P_i y_i + R_i, \quad (2.17)$$

où les $P_i(x, \hat{x}, t)$ sont des noyaux donnés par la résolution "off-line" de N équations intégral-différentielles de Riccati dans Ω_i et où les R_i sont les solutions "on-line" d'équations paraboliques, où l'on itère par rapport à l'algorithme (2.15). Nous reviendrons là-dessus [4].

Remarque 2.5 On peut aussi introduire la décomposition par rapport aux domaines Ω_i des *Systèmes d'Optimalité* (cf. [3]) pour les problèmes elliptiques, paraboliques (cf. [5]) ou hyperboliques [6], les ouverts Ω_i étant sans recouvrement. Pour obtenir des schémas parallèles, il faut alors introduire une itération sur le Système d'Optimalité. \square

Références bibliographiques

- [1] J.L. LIONS and O. PIRONNEAU, Algorithmes parallèles pour la solution de problèmes aux limites. C.R.A.S., 1998.
- [2] J.L. LIONS and O. PIRONNEAU, Parallel Decomposition of Control Problems for Navier-Stokes Equations, dedicated to A. CHORIN, (Berkeley, June 1998).
- [3] J.L. LIONS, *Contrôle Optimal de systèmes gouvernés par des équations aux dérivées partielles*. Dunod-Gauthier Villars, 1968.
- [4] J.L. LIONS and O. PIRONNEAU, Domain Decomposition and Real Time Algorithms. A paraître.

[5] J.D. BENAMOU, Décomposition de domaines pour le contrôle de systèmes gouvernés sur des équations d'évolution. C.R.A.S., Paris I, 234, (1997), p. 1065-1070.

[6] J.D. BENAMOU and B. DESPRES, A domain decomposition method for the Helmholtz equations and related optimal control problems. J. Comp. Physics, 136 (1997), p. 68-82.

[7] J.L. LAGNESE and G. LEUGERING, Dynamic domain decomposition in approximate and exact boundary control in problems of transmission for wave equations. A paraître.

Jacques-Louis Lions, Collège de France, 3, rue d'Ulm, 75231 Paris Cedex 05

Olivier Pironneau, Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, 4, place Jussieu, 75231 Paris Cedex 05.