A Genetic Algorithm Compared with a Gradient-Based Method for the Solution of an Active-Control Model Problem

Nathalie Marco, Cyril Godart, Jean-Antoine Desideri, Bertrand Mantel, Jacques Périaux

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A genetic algorithm compared with a gradient-based method for the solution of an active-control model problem

Nathalie Marco, Cyril Godart, Jean-Antoine Désidéri,

Bertrand Mantel and Jacques Périaux

N° 2948

Juillet 1996

THÈME 4

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

A genetic algorithm compared with a gradient-based method for the solution of an active-control model problem

Nathalie Marco[*] , Cyril Godart[*] , Jean-Antoine Désidéri[*] ,

Bertrand Mantel[**] and Jacques Périaux[**]

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Sinus

Rapport de recherche n ° 2948 — Juillet 1996 — 33 pages

Abstract: In this study, a linear active-control model problem is employed to conduct a preliminary performance comparison between a genetic algorithm and a classical optimization method based on the evaluation of a gradient of functional. For this purpose, we consider a system modeled by the heat equation in one space dimension controled by a source term. The cost functional is a quadratic form of the distance between the final state and a prescribed function augmented of a penalty involving the control quadratically. The problem is first solved by a direct identification of the "optimal-control law" from the solution of a Riccati system. A simple genetic algorithm is implemented afterwards and compared.

Key-words: Optimal control - Heat equation - Objective functional - Riccati equation - Optimal control law - Spectral approximation - Genetic algorithm

(Résumé : tsvp)

[*] INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia Antipolis Cédex-France
[**] Dassault-Aviation, AMD-BA, 78, Quai Marcel Dassault, B.P. 300, 92214 Saint-Cloud

Un algorithme génétique comparé à une méthode de gradient pour la résolution d'un problème modèle de contrôle actif

Résumé : Le but de cette étude est d'utiliser un problème modèle linéaire de contrôle actif pour effectuer une première comparaison de performance d'un algorithme génétique avec une méthode classique d'optimisation basée sur une évaluation du gradient de fonctionnelle. A cette fin, on considère un système modélisé par l'équation de la chaleur monodimensionnelle contrôlée par un terme source. Le critère est une forme quadratique de l'écart de l'état final à une cible, augmentée d'une perturbation quadratique du contrôle. Le problème de contrôle est d'abord résolu par identification d'une loi de pilotage optimal issu d'un système d'équations de Riccati. Un algorithme génétique simple est ensuite construit et comparé à la méthode précédente.

Mots-clé : Contrôle optimal - Equation de la chaleur - Fonctionnelle coût - Equation de Riccati - Loi de pilotage optimal - Approximation spectrale - Algorithme génétique

Contents

1 The optimal control problem **4**
1.1 State Equation 4
1.2 Cost functional 4

2 Evaluation of the gradient **5**

3 Optimal Solution : Riccati equation **7**

4 Numerical solution of the Riccati system by a spectral method **9**
4.1 Spectral approximation of the S-equation in (12) 9
4.2 Spectral approximation of the μ-equation in (12) 10
4.3 Final conditions 11
4.4 Numerical results 12

5 Optimal control solutions **16**
5.1 Spectral approximation of the state 16
5.2 Numerical results 18

6 A Genetic approach **21**
6.1 Essential points about Genetic Algorithms (GAs) 21
6.2 Dynamics of GAs 22
6.3 Numerical experiments 25

7 Concluding remarks **28**

References **33**

1 The optimal control problem

For the purpose of comparing the performance of different optimization approaches, we consider here the following model control problem.

1.1 State Equation

Let t, the time variable, be in $[0,T]$, T being fixed. Let x, the space variable, be in $[0,1]$.
$y(x,t) \in L^2([0,1]\times[0,T])$ is the state variable, $v(t) \in L^2(0,T)$ is the control. Both are real valued.

We consider the following 1D heat equation with control in the R.H.S. :

$$\begin{cases} \dfrac{\partial y}{\partial t}(x,t) - \nu \dfrac{\partial^2 y}{\partial x^2}(x,t) = v(t)\ \delta(x-b(t)) & \text{on} \quad \Omega = [0,1]\times[0,T] \\ y(0,t) = y(1,t) = 0 \quad \forall t & \text{boundary condition} \\ y(x,0) = 0 \quad \forall x & \text{initial condition} \end{cases} \tag{1}$$

δ represents the **Dirac function**, whose definition is re-called :

$$\forall f \in L^2(a,b), \quad \int_a^b f(x)\ \ \delta(x-\widehat{x})dx = f(\widehat{x})$$

In (1), b is a smooth function from $[0,T]$ into $[0,1]$ and is defined by $b(t) = b_0 + b_1 \sin\omega t$ with $b_0 = 1$, $b_1 = 10^{-3}$, $\omega = 100\pi$.

In future applications, $b(t)$ could also be optimized and would then represent the "active control" aspect of the problem.

1.2 Cost functional

We consider the problem of approaching a specified final-time target function $y_T(x)$ with minimal energy. Mathematically, the following functional is to be minimized :

$$J(v) = \frac{1}{2}\ \int_0^1 (y(x,T) - y_T(x))^2\ dx\ +\ \frac{\varepsilon}{2}\int_0^T v^2(t)\ dt \tag{2}$$

For example, $y_T(x) = \sin\pi x \in L^2(0,1)$.

In particular, we intend to assess the robustness of optimization algorithms as ε vanishes, since this would result in increased stiffness.

The first term in the expression of J denotes a penalization of the distance between the final state actually achieved and the target function. The second term denotes a penalization of the energy spent to achieve the final state.

2 Evaluation of the gradient

Let δv and δy be small perturbations in v and y and δJ the resulting perturbation in the cost functional. We then obtain from (2) :

$$\delta J = \int_0^1 (y(x,T) - y_T(x)).\delta y(x,T)\ dx\ +\ \varepsilon\ \int_0^T v(t).\delta v(t)\ dt \tag{3}$$

The second term in the expression of δJ depends directly on δv. Unfortunately, the first one contains δy which is an implicit function of δv. It is thus necessary to evaluate this part in terms of δv. This is achieved by the classical "adjoint-equation" approach.

δy is the solution of the linearized equation :

$$\begin{cases} \dfrac{\partial \delta y}{\partial t}(x,t) - \nu \dfrac{\partial^2 \delta y}{\partial x^2}(x,t) = \delta v(t)\ \ \delta(x - b(t)) & \text{on} \quad \Omega = [0,1] \times [0,T] \\ \delta y(0,t) = \delta y(1,t) = 0 \quad \forall t & \text{boundary condition} \\ \delta y(x,0) = 0 \quad \forall x & \text{initial condition} \end{cases} \tag{4}$$

Let us introduce the real-valued adjoint-state function $\lambda(x,t)$. Multiplying (4) by $\lambda(x,t)$ and integrating over $[0,T] \times [0,1]$, yields (regardless the choice of λ) :

$$\int_0^T \int_0^1 \lambda(\ \delta y_t - \nu\ \delta y_{xx} - \delta v(t)\ \delta(x - b(t))\ )\ dx\ dt = 0 \tag{5}$$

$\left(\ (.)_t = \dfrac{\partial (.)}{\partial t}, \quad (.)_{xx} = \dfrac{\partial^2 (.)}{\partial x^2}\ \right)$

Let us evaluate the terms $\int_0^T \int_0^1 \lambda\ \delta y_t\ dx\ dt$ and $\int_0^T \int_0^1 \lambda\ \delta y_{xx}\ dx\ dt$ by integration by parts :

- $$\begin{aligned} \int_0^T \int_0^1 \lambda\ \delta y_t\ dxdt &= \int_0^1 \left(\ [\lambda.\delta y]_0^T - \int_0^T \delta y.\lambda_t\ dt\ \right) dx \\ &= \int_0^1 \lambda(x,T)\delta y(x,T)\ dx - \int_0^T \int_0^1 \lambda_t\ \delta y\ dx\ dt \end{aligned}$$

- $$\begin{aligned} \int_0^T \int_0^1 \lambda\ \delta y_{xx}\ dx\ dt &= \int_0^T \left(\ [\lambda.\delta y_x]_0^1 - \int_0^1 \lambda_x \delta y_x\ dx\ \right) dt \\ &= \int_0^T \left(\lambda(1,t)\delta y_x(1,t) - \lambda(0,t)\delta y_x(0,t)\right) dt \\ &\quad + \int_0^T \int_0^1 \lambda_{xx}\delta y\ dx\ dt \end{aligned}$$

Then, $\forall \lambda$, (5) becomes :

$$\int_0^1 \lambda(x,T)\delta y(x,T)\ dx - \int_0^T \int_0^1 \lambda_t\ \delta y\ dx\ dt - \nu\ \int_0^T \left(\lambda(1,t)\delta y_x(1,t) - \lambda(0,t)\delta y_x(0,t)\right) dt \tag{6}$$

$$-\nu \int_0^T \int_0^1 \lambda_{xx} \delta y \; dx \; dt - \int_0^T \int_0^1 \lambda \; \delta v(t) \; \delta(x - b(t)) \; dx \; dt \; = \; 0$$

We then select λ to be the solution of the following "adjoint system" :

$$\begin{cases} \lambda_t + \nu \; \lambda_{xx} = 0 & \text{on } \; \Omega = [0,1] \times [0,T] \\ \lambda(0,t) = \lambda(1,t) = 0 \;\; \forall t & \text{boundary conditions} \\ \lambda(x,T) = y(x,T) - y_T(x) \quad \forall x & \textbf{final} \text{ conditions} \end{cases} \tag{7}$$

which is, as it is classical, a backward-time linear Cauchy problem. Then, (6) becomes :

$$\begin{aligned} \int_0^1 \Big(y(x,T) - y_T(x) \Big) \delta y(x,T) \; dx \; &= \int_0^T \int_0^1 \lambda(x,t) \; \delta v(t) \; \delta(x - b(t)) \; dx \; dt \\ &= \int_0^T \lambda(b(t),t) . \delta v(t) \; dt \end{aligned}$$

which provides the expression of the implicit term in δJ as a function of δv as desired. By substitution in (3), we obtain :

$$\delta J = \int_0^T \Big(\; \varepsilon v(t) + \lambda(b(t),t) \; \Big) . \delta v(t) \; dt \tag{8}$$

In summary, we have

- $\star$ $y(x,t)$ with $0 \le x \le 1, \;\; 0 \le t \le T$ the state variable, solution of heat equation with a right-hand-side :

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu \dfrac{\partial^2 y}{\partial x^2} = v(t) \; \delta(x - b(t)) \\ y(0,t) = y(1,t) = 0 \quad \forall t \\ y(x,0) = 0 \quad \forall x \end{cases}$$

- $\star$ $\lambda(x,t)$ the co-state variable, solution of the adjoint-system :

$$\begin{cases} \dfrac{\partial \lambda}{\partial t} + \nu \; \dfrac{\partial^2 \lambda}{\partial x^2} = 0 \\ \lambda(0,t) = \lambda(1,t) = 0 \;\; \forall t \\ \lambda(x,T) = y(x,T) - y_T(x) \quad \forall x \end{cases}$$

- $\star$ the cost functional, defined by :

$$J(v) = \frac{\varepsilon}{2} \int_0^T v^2(t) \; dt \; + \; \frac{1}{2} \; \int_0^1 (y(x,T) - y_T(x))^2 \; dx$$

(obtained after integration of the state system forward in time) and its perturbation :

$$\delta J = \int_0^T \Big(\; \varepsilon v(t) + \lambda(b(t),t) \; \Big) . \delta v(t) \; dt$$

(obtained after integration of the adjoint-system backward in time).

The optimality condition is thus evident :

$$\varepsilon \; v(t) + \lambda(b(t),t) = 0 \tag{9}$$

3 Optimal Solution : Riccati equation

The optimal trajectory is defined by the solution of the coupled system made of the state equation, adjoint-equation and optimality condition. Here, these are respectively (1), (7) and (9).

In the particular case where the cost functional is quadratic (in y and v) and the dynamic system is linear, it is well known (see for example Bryson & Ho [1], Lions [4]) that a linear relationship exists giving the co-state variable as the image of the state variable by an operator S solution of an independent Riccati equation. Then, the optimization is considerably simplified, since after the Riccati equation is solved, an "optimal control law" is identified permitting to express co-state variable and control explicitly in terms of the observation at time t (the state vector).

Here, the problem is linear but not homogeneous. Then, we have an additional term to identify. For this, the solution is sought in the form :

$$\lambda(x,t) = \int_0^1 S(x,\xi,t)\ y(\xi,t)\ d\xi - \mu(x,t) \tag{10}$$

where S is an unknown operator and μ is an unknown function in L^2. Then :

- $$\frac{\partial\lambda}{\partial t}(x,t) = \int_0^1 \Big(\frac{\partial S}{\partial t}(x,\xi,t)\ y(\xi,t) + S(x,\xi,t)\ (\nu\frac{\partial^2 y}{\partial\xi^2}(\xi,t) + v(t)\ \delta(\xi - b(t))) \Big)\ d\xi - \frac{\partial\mu}{\partial t}(x,t)$$

  $\star$ $$\int_0^1 S(x,\xi,t)\ \frac{\partial^2 y}{\partial\xi^2}(\xi,t)\ d\xi \ = \ \int_0^1 \frac{\partial^2 S}{\partial\xi^2}(x,\xi,t)\ y(\xi,t)\ d\xi,$$

  provided the following boundary conditions are imposed :

$$\boxed{S(x,0,t) = S(x,1,t) = 0 \quad \forall x,t} \tag{11}$$

  $\star$ $$\int_0^1 S(x,\xi,t)\ v(t)\ \delta(\xi - b(t))\ d\xi = v(t)\ S(x,b(t),t)$$

  From (9), $\quad v(t) = -\dfrac{1}{\varepsilon}\ \lambda(b(t),t), \quad$ thus :

$$\begin{aligned}\int_0^1 S(x,\xi,t)\ v(t)\ \delta(\xi - b(t))\ d\xi \ &= -\frac{1}{\varepsilon}\ \lambda(b(t),t) S(x,b(t),t) \\ &= -\frac{1}{\varepsilon}\ S(x,b(t),t) \int_0^1 S(b(t),\xi,t)\ y(\xi,t)\ d\xi + \\ &\quad \frac{1}{\varepsilon}\ S(x,b(t),t)\ \mu(b(t),t) \qquad \text{(from (10))}\end{aligned}$$

Finally,

$$\begin{aligned}\frac{\partial\lambda}{\partial t}(x,t) \ &= \int_0^1 \Big(\frac{\partial S}{\partial t}(x,\xi,t) + \nu\ \frac{\partial^2 S}{\partial\xi^2}(x,\xi,t) - \frac{1}{\varepsilon}\ S(x,b(t),t)\ S(b(t),\xi,t) \Big)\ y(\xi,t)\ d\xi \ + \\ &\quad \frac{1}{\varepsilon}\ S(x,b(t),t)\ \mu(b(t),t) - \frac{\partial\mu}{\partial t}(x,t)\end{aligned}$$

- $\dfrac{\partial^2 \lambda}{\partial x^2}(x,t) = \displaystyle\int_0^1 \frac{\partial^2 S}{\partial x^2}(x,\xi,t)\ y(\xi,t)\ d\xi - \frac{\partial^2 \mu}{\partial x^2}(x,t)$

One way to satisfy the PDE $\lambda_t + \nu\ \lambda_{xx} = 0$, consists in fixing the pair (S,μ) solution of the following system :

$$\boxed{\begin{cases} S_t + \nu\ (S_{xx} + S_{\xi\xi}) = \dfrac{1}{\varepsilon}\ S(x,b(t),t)\ S(b(t),\xi,t) \\ \mu_t + \nu\ \mu_{xx} = \dfrac{1}{\varepsilon}\ S(x,b(t),t)\ \mu(b(t),t) \end{cases}} \tag{12}$$

The first equation of system (12) is called the **Riccati Equation**. As usual, it contains linear terms (here of Laplacian-type) and a quadratic term (here a source term related to the point of application $b(t)$ of the control). We recall that this equation is submitted to boundary condition (11). In order to make the determination of (S,μ) unique, we have to complete the PDE system with boundary and final conditions.

Boundary conditions :

(i) One imposes

$$\boxed{S(x,0,t) = S(x,1,t) = 0, \quad \forall x,t} \tag{13}$$

and :

$$\boxed{S(0,\xi,t) = S(1,\xi,t) = 0, \quad \forall \xi,t} \tag{14}$$

in order to enforce the boundary conditions $\lambda(0,t) = \lambda(1,t) = 0$.

(ii)

$$\boxed{\mu(0,t) = \mu(1,t) = 0, \quad \forall t} \tag{15}$$

Final conditions :

(iii)

$$\boxed{S(x,\xi,T) = \delta(\xi - x), \quad \forall x,\xi} \tag{16}$$

(iv)

$$\boxed{\mu(x,T) = y_T(x), \quad \forall x} \tag{17}$$

Remark : S is then symmetric : $\forall x,\xi,t :\ S(x,\xi,t) = S(\xi,x,t)$.

We have obtained an **"Optimal-control Law"** : the pair (S,μ) is computed once for all, without prior knowledge of the state $y(x,t)$, which is then obtained independently by the integration of a Cauchy problem, since the adjoint-state $\lambda(x,t)$ has been eliminated after a quadrature. The control $v(t)$ is deduced algebraically. It is enough to "observe" y to know how to pilot optimally.

4 Numerical solution of the Riccati system by a spectral method

The Fourier functions are chosen to form a discrete functional basis. Let M be the number of modes.

4.1 Spectral approximation of the S-equation in (12)

We put :

$$S(x,\xi,t) = \sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}_{ml}(t)\ \sin(m\pi x)\ \sin(l\pi\xi)$$

So that,

- $\dfrac{\partial S}{\partial t}(x,\xi,t) = \displaystyle\sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}'_{ml}(t)\ \sin(m\pi x)\ \sin(l\pi\xi)$

- $\dfrac{\partial^2 S}{\partial x^2}(x,\xi,t) + \dfrac{\partial^2 S}{\partial \xi^2}(x,\xi,t) = \displaystyle\sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}_{ml}(t)\ (-m^2-l^2)\ \pi^2\ \sin(m\pi x)\ \sin(l\pi\xi)$

- $$\begin{aligned} S(x,b(t),t)\ S(b(t),\xi,t) &= \Big(\sum_{m=1}^{M}\sum_{l_1=1}^{M} \widehat{s}_{ml_1}(t)\ \sin(m\pi x)\ \sin(l_1\pi b(t))\Big) \\ &\quad \Big(\sum_{m_1=1}^{M}\sum_{l=1}^{M} \widehat{s}_{m_1 l}(t)\ \sin(m_1\pi b(t))\ \sin(l\pi\xi)\Big) \\ &= \Big(\sum_{m=1}^{M}\Big(\sum_{l_1=1}^{M} \widehat{s}_{ml_1}(t)\ \sin(l_1\pi b(t))\Big)\ \sin(m\pi x)\Big) \\ &\quad \Big(\sum_{l=1}^{M}\Big(\sum_{m_1=1}^{M} \widehat{s}_{m_1 l}(t)\ \sin(m_1\pi b(t))\Big)\ \sin(l\pi\xi)\Big) \\ &= \Big(\sum_{m=1}^{M} \widehat{A}_m(t)\ \sin(m\pi x)\Big)\Big(\sum_{l=1}^{M} \widehat{B}_l(t)\ \sin(l\pi\xi)\Big) \end{aligned}$$

with :

$$\widehat{A}_m(t) = \sum_{l_1=1}^{M} \widehat{s}_{ml_1}(t)\ \sin(l_1\pi b(t)) \quad \text{and} \quad \widehat{B}_l(t) = \sum_{m_1=1}^{M} \widehat{s}_{m_1 l}(t)\ \sin(m_1\pi b(t))$$

After substitution of these expressions, the S-equation in (12) becomes :

$$\sum_{m=1}^{M}\sum_{l=1}^{M}\Big(\ \widehat{s}'_{ml}(t) - \nu\ (m^2+l^2)\pi^2\ \widehat{s}_{ml}(t)\ \Big)\ \sin(m\pi x)\sin(l\pi\xi)$$

$$-\ \frac{1}{\varepsilon}\Big(\ \sum_{m=1}^{M} \widehat{A}_m(t)\ \sin(m\pi x)\ \sum_{l=1}^{M} \widehat{B}_l(t)\ \sin(l\pi\xi)\ \Big)\ =\ 0$$

As the Fourier basis is orthogonal, we have :

$$< \; S_t + \nu \; \Delta S - \frac{1}{\varepsilon} \; S(b(t),\xi,t) \; S(x,b(t),t) \; , \; \sin(m\pi x) \; \sin(l\pi\xi) \; > \; = \; 0 \qquad \forall m,l$$

So that :

$$\boxed{\widehat{s}'_{ml}(t) - \nu \; (m^2+l^2)\pi^2 \; \widehat{s}_{ml}(t) - \; \frac{1}{\varepsilon} \; \widehat{A}_m(t) \; \widehat{B}_l(t) = 0 \qquad \forall m,l} \tag{18}$$

Then, the determination of the (approximate) function $S(x,\xi,t)$ is completed by the solution of a coupled set of $M \times M$ non linear ordinary differential equations.

In order to make the unknown functions smoother, the following change of variables is done additionally :

$$\widehat{s}_{ml}(t) = \exp\Big[\; \nu(m^2+l^2)\pi^2(t-T)\Big] \; \widehat{\sigma}_{ml}(t) \qquad \forall m,l$$

(18) becomes :

$$\boxed{\begin{aligned} \widehat{\sigma}'_{ml}(t) \; = \frac{1}{\varepsilon} \; \sum_{m_1=1}^{M} & \exp(\nu\pi^2 m_1^2(t-T)) \; \widehat{\sigma}_{m_1 l}(t) \; \sin(m_1\pi b(t)) \\ & \sum_{l_1=1}^{M} \exp(\nu\pi^2 l_1^2(t-T)) \; \widehat{\sigma}_{m l_1}(t) \; \sin(l_1\pi b(t)) \end{aligned}} \tag{19}$$

4.2 Spectral approximation of the μ-equation in (12)

We put :

$$\mu(x,t) = \sum_{m=1}^{M} \widehat{\mu}_m(t) \; \sin(m\pi x)$$

So that :

- $\dfrac{\partial \mu}{\partial t}(x,t) = \sum_{m=1}^{M} \widehat{\mu}'_m(t) \; \sin(m\pi x)$
- $\dfrac{\partial^2 \mu}{\partial x^2}(x,t) = -\sum_{m=1}^{M} m^2\pi^2 \; \widehat{\mu}_m(t) \; \sin(m\pi x)$
- $S(x,b(t),t) \; \mu(b(t),t) = \sum_{m=1}^{M} \Big(\sum_{l=1}^{M} \widehat{s}_{ml}(t) \; \sin(l\pi b(t)) \sum_{m_1=1}^{M} \widehat{\mu}_{m_1}(t) \; \sin(m_1\pi b(t)) \Big) \sin(m\pi x)$

As previously, we have :

$$< \; \mu_t + \; \nu\mu_{xx} - \frac{1}{\varepsilon} \; S(x,b(t),t) \; \mu(b(t),t) \; , \; \sin(m\pi x) \; > \; = \; 0 \qquad \forall m$$

which yields :

$$\widehat{\mu}'_m(t) - \nu\ m^2\ \pi^2\ \widehat{\mu}_m(t) - \frac{1}{\varepsilon}\ \sum_{l=1}^{M} \widehat{s}_{ml}(t)\ \sin(l\pi b(t))\ \sum_{m_1=1}^{M} \widehat{\mu}_{m_1}(t)\ \sin(m_1\pi b(t))\ =\ 0 \quad \forall m \tag{20}$$

Again, we put :

$$\widehat{\mu}_m(t) = \exp(\nu m^2\pi^2(t-T))\ \widehat{\gamma}_m(t)$$

so that, (20) becomes :

$$\boxed{\begin{aligned} \widehat{\gamma}'_m(t)\ &= \frac{1}{\varepsilon}\ \sum_{l=1}^{M} \exp(\nu\ l^2\pi^2(t-T))\ \widehat{\sigma}_{ml}(t)\ \sin(l\pi b(t)) \\ &\qquad \sum_{m_1=1}^{M} \exp(\nu\ m_1^2\pi^2(t-T))\ \widehat{\gamma}_{m_1}(t)\ \sin(m_1\pi b(t)) \end{aligned}} \tag{21}$$

4.3 Final conditions

We recall that the following final conditions have to be imposed at $t = T$,

$$S(x,\xi,T) = \delta(\xi - x) \qquad \text{and} \qquad \mu(x,T) = y_T(x) = \sin(\pi x)$$

Thus, we have to determine what are the corresponding final conditions on $\widehat{\sigma}_{ml}(T)$ and $\widehat{\gamma}_m(T)$.

- Final condition on $S(x,\xi,T)$.

  $$S(x,\xi,T) = \sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}_{ml}(T)\ \sin(m\pi x)\ \sin(l\pi\xi)$$

  The aim is to define S at $t = T$ such that :

  $$\forall\varphi,\ \int_0^1 S(x,\xi,T)\ \varphi(\xi)\ d\xi = \varphi(x).\ \text{ We let }\ \varphi(\xi) = \sum_{m=1}^{M} \widehat{\varphi}_m\ \sin(m\pi\xi).$$

  Then, for all sequence $(\widehat{\varphi}_l)$,

  $$\begin{aligned} \int_0^1 S(x,\xi,T)\ \varphi(\xi)\ d\xi\ &= \int_0^1 \Big(\ \sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}_{ml}(T)\ \sin(m\pi x)\sin(l\pi\xi)\ \Big)\ \Big(\ \sum_{k=1}^{M} \widehat{\varphi}_k\ \sin(k\pi\xi)\ \Big)\ d\xi \\ &= \sum_{k=1}^{M} \widehat{\varphi}_k\ \sin(k\pi x) \end{aligned}$$

  $$\Leftrightarrow\ \sum_{k=1}^{M} \widehat{\varphi}_k \Big(\ \sum_{m=1}^{M}\sum_{l=1}^{M} \widehat{s}_{ml}(T)\ \sin(m\pi x)\ \int_0^1 \sin(l\pi\xi)\ \sin(k\pi\xi)\, d\xi\ \Big) = \sum_{k=1}^{M} \widehat{\varphi}_k\ \sin(k\pi x)$$

  But :

  $$\int_0^1 \sin(l\pi\xi)\ \sin(k\pi\xi)\ d\xi = \begin{cases} 0 & \text{if } l \neq k \\ \dfrac{1}{2} & \text{if } l = k \end{cases}$$

So that,

$$\sum_{k=1}^{M} \frac{1}{2}\ \widehat{\varphi}_k \ \sum_{m=1}^{M} \widehat{s}_{mk}(T)\sin(m\pi x) = \sum_{k=1}^{M} \widehat{\varphi}_k \ \sin(k\pi x), \qquad \forall\ \varphi$$

$$\Leftrightarrow \quad \sum_{m=1}^{M} \widehat{s}_{mk}(T)\sin(m\pi x) = 2\sin(k\pi x) \qquad \forall k$$

This requires that :

$$\boxed{\widehat{s}_{ml}(T) = \begin{cases} 2 & \text{for } m = l \\ 0 & \text{for } m \neq l \end{cases} \qquad \Leftrightarrow \qquad \widehat{\sigma}_{ml}(T) = \begin{cases} 2 & \text{for } m = l \\ 0 & \text{for } m \neq l \end{cases}} \tag{22}$$

which indeed implies that :

$$S(x,\xi,T) = \sum_{m=1}^{M} 2\sin(m\pi x)\ \sin(m\pi\xi)\ =\ \delta(\xi - x), \quad \text{as desired.}$$

- Final condition on $\mu(x,T)$.

  We recall that $\mu(x,t) = \sum_{m=1}^{M} \widehat{\mu}_m(t)\sin(m\pi x)$.

  Then, $\mu(x,T) = \sum_{m=1}^{M} \widehat{\mu}_m(T)\sin(m\pi x) = \sin(\pi x) \quad \forall x.$

  So,

$$\boxed{\widehat{\mu}_m(T) = \begin{cases} 1 & \text{for } m = 1 \\ 0 & \text{for } m \geq 2 \end{cases} \qquad \Leftrightarrow \qquad \widehat{\gamma}_m(T) = \begin{cases} 1 & \text{for } m = 1 \\ 0 & \text{for } m \geq 2 \end{cases}} \tag{23}$$

4.4 Numerical results

We have set :

$T = 3,$
$\nu = 0.1,$
$b(t) = b_0 + b_1\sin(\omega t) \quad$ with $\quad b_0 = \frac{1}{2}, \quad b_1 = 10^{-3}, \quad \omega = 100\pi.$

ε varies from 1 to 10^{-6}.

To solve the system of Riccati equations, we have used a NAG Fortran library routine (namely D02EJF). It consists in integrating a stiff system of first-order ordinary differential equations over a range with suitable initial conditions, using a variable-order, variable-step method implementing the Backward Differentiation Formulae, until a user-specified function, if supplied, of the solution is zero, and returns the solution at points specified by the user, if desired. As the routine requires initial conditions on

entry (we have final conditions), a change of variables was necessary.

We have set $\tau = \dfrac{T-t}{\varepsilon}$. Then, τ varies from 0 to $\dfrac{T}{\varepsilon}$ (as t varies from T to 0). However, the smaller ε becomes , the larger the interval work ($[0, \frac{T}{\varepsilon}]$) becomes. A second change of variables is then necessary.

We have set $\theta = \log(\dfrac{T-t}{\varepsilon} + 1)$, then θ varies from 0 to $\log(\dfrac{T}{\varepsilon} + 1)$. Then, after introducing these two changes of variables, equations (19) and (21) have become :

$$\left\{ \begin{array}{l} \widehat{\sigma}'_{ml}(\theta) = \; -10^{\theta}\ \ln(10) \Big(\displaystyle\sum_{m_1=1}^{M} \exp(-\varepsilon(10^{\theta}-1)\nu\pi^2 m_1^2)\ \widehat{\sigma}_{m_1 l}(\theta)\ \sin(m_1\ \pi\ b(T-\varepsilon(10^{\theta}-1)))\ \Big) \\ \qquad\qquad \Big(\displaystyle\sum_{l_1=1}^{M} \exp(-\varepsilon(10^{\theta}-1)\nu\pi^2 l_1^2)\ \widehat{\sigma}_{m l_1}(\theta)\ \sin(l_1\ \pi\ b(T-\varepsilon(10^{\theta}-1)))\ \Big) \\ \widehat{\sigma}_{ml}(0) = 2 \quad \text{if} \quad m = l, \quad \widehat{\sigma}_{ml}(0) = 0 \quad \text{if} \quad m \neq l \end{array} \right. \tag{24}$$

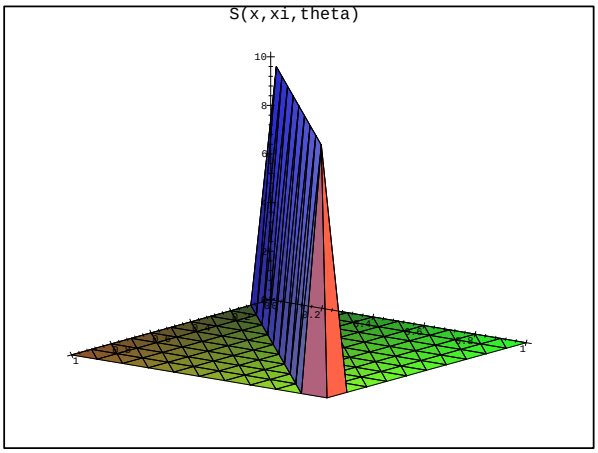
$$\left\{ \begin{array}{l} \widehat{\gamma}'_{m}(\theta) = \; -10^{\theta}\ \ln(10) \Big(\displaystyle\sum_{l=1}^{M} \exp(-\varepsilon(10^{\theta}-1)\nu\ l^2\pi^2)\ \widehat{s}_{ml}(\theta)\ \sin(l\ \pi\ b(T-\varepsilon(10^{\theta}-1)))\ \Big) \\ \qquad\qquad \Big(\displaystyle\sum_{m_1=1}^{M} \exp(-\varepsilon(10^{\theta}-1)\nu\ m_1^2\pi^2)\ \widehat{\gamma}_{m_1}(\theta)\ \sin(m_1\ \pi\ b(T-\varepsilon(10^{\theta}-1)))\ \Big) \\ \widehat{\gamma}_{m}(0) = 1 \quad \text{if} \quad m = 1, \quad \widehat{\gamma}_{m}(0) = 0 \quad \text{if} \quad m \geq 2 \end{array} \right. \tag{25}$$

In Figures 1 and 2, ε is fixed to 0.1.
Figure 1-a depicts $S(x, \xi, \theta = 0)$, (i.e. for $t = T$). It is a Dirac function located on $x = \xi$. Figure 1-b depicts $S(x, \xi, \theta = 1.2)$, (i.e. for $t = 1.5$), and finally, in Figure 1-c, $S(x, \xi, \theta = \log(\frac{T}{\varepsilon} + 1))$ (when $t = 0$), is presented.

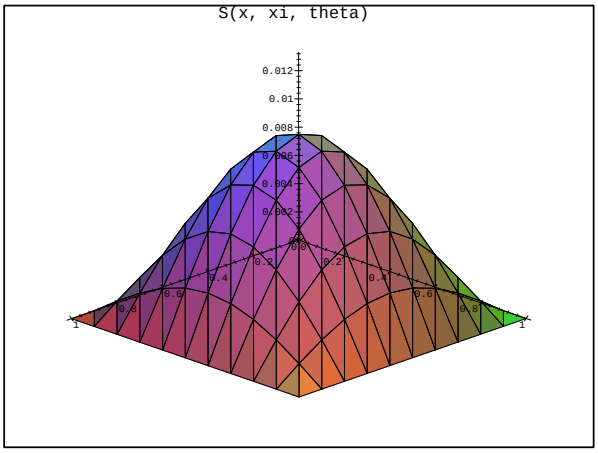
In Figure 2, we also present $\mu(x, \theta)$, for $\theta = 0,\ 1.2$ and $\log(\frac{T}{\varepsilon} + 1)$. We note that for $\theta = 0$ $(t = T)$, $\mu(x, t) = \sin(\pi x)$.

In Figures 3-a and 3-b, are presented the continuous L^2 norms of S and μ for θ varying from 0 to $\log(\frac{T}{\varepsilon} + 1)$, for different ε.

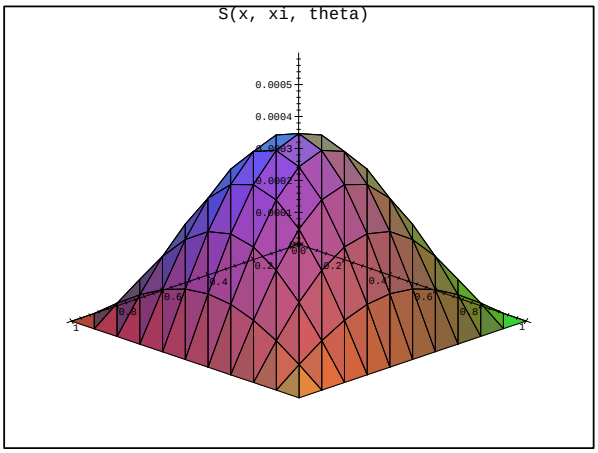
In Figures 4-a and 4-b, the same norms are depicted for t varying from 0 to T.



a- $S(x, \xi, \theta = 0)$, where $\theta = 0$ corresponds to $t = T$.



b- $S(x, \xi, \theta = 1.2)$, where $\theta = 1.2$ corresponds to $t = 1.5$.



c- $S(x, \xi, \theta = \log(\frac{T}{\varepsilon} + 1))$, where this value of θ corresponds to $t = 0$.

Figure 1: $S(x, \xi, \theta)$, where $\theta = 0,\ 1.2$ and $\log(\frac{T}{\varepsilon} + 1)$ (corresponding respectively to $t = T,\ 1.5$ and 0). $\varepsilon = 0.1$

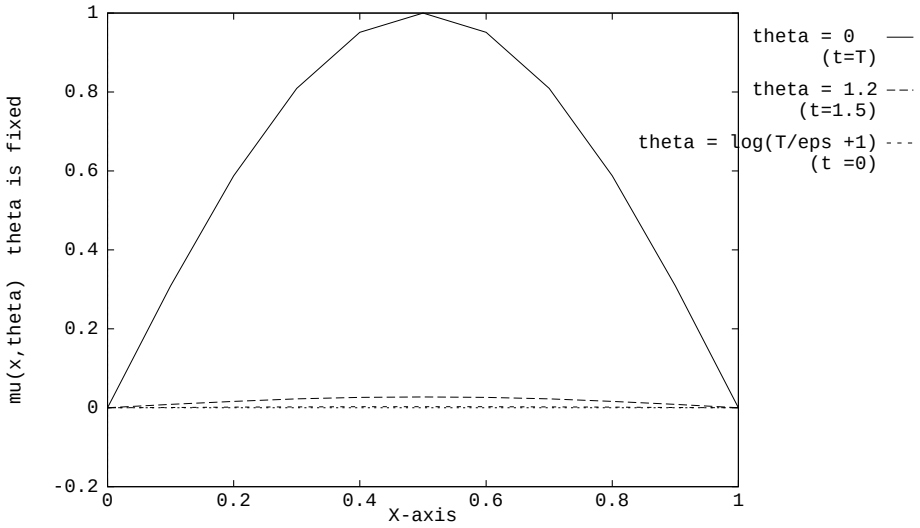
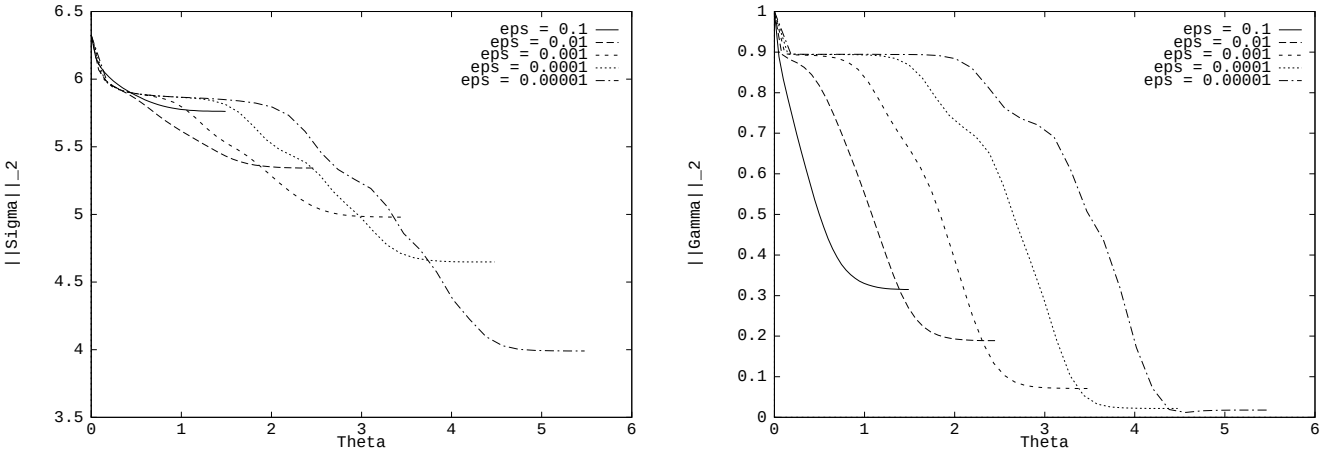
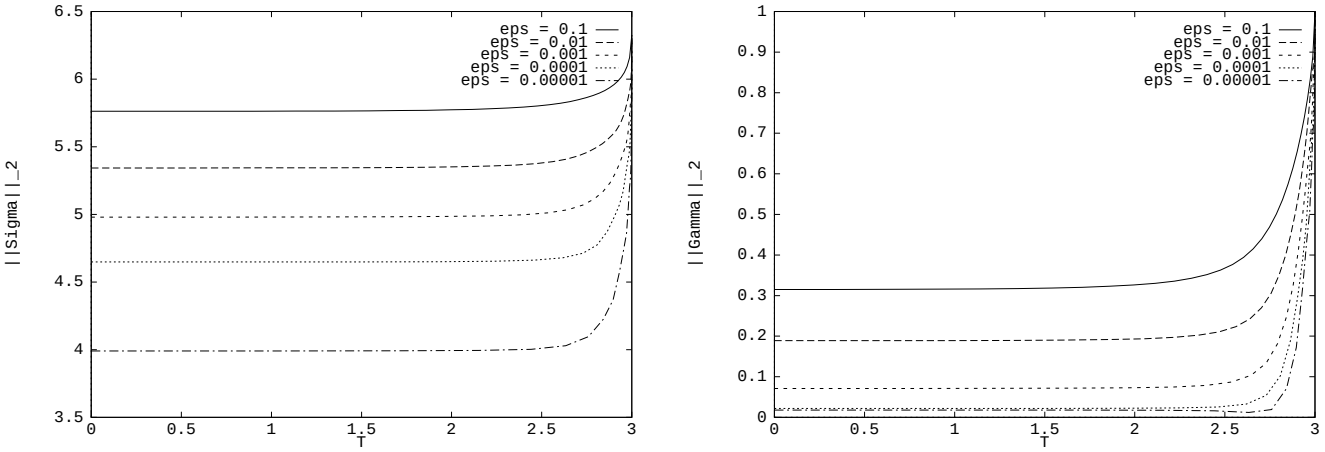


Figure 2: $\mu(x, \theta = 0,\ 1.2 \text{ and } \log(\frac{T}{\varepsilon}+1))$ (corresponding respectively to $t = T = 3, 1.5$ and 0). $\varepsilon = 0.1$



a- $||S||_{L^2}$. b- $||\mu||_{L^2}$.

Figure 3: $||S||_{L^2}$ and $||\mu||_{L^2}$, as functions of θ, for ε varying from 1 to 10^{-5}.



a- $||S||_{L^2}$. b- $||\mu||_{L^2}$.

Figure 4: $||S||_{L^2}$ and $||\mu||_{L^2}$, functions of t, for ε varying from 1 to 10^{-5}.

5 Optimal control solutions

5.1 Spectral approximation of the state

Now that system (12) has been solved, the state $y(x,t)$ can be calculated independently. Again for the purpose of a numerical integration, the state function $y(x,t)$ is approximated over a basis of Fourier functions :

$$y(x,t) = \sum_{m=1}^{M} \widehat{y}_m(t)\ \sin(m\pi x)$$

Let us apply a Fourier Transformation to the source term :

$$f(x,t) = v(t)\ \delta(x - b(t)) = \sum_{m=1}^{M} \widehat{f}_m(t)\ \sin(m\pi x)$$

In subsection 4.3, we found that $\delta(\xi - x) = \sum_{m=1}^{M} 2\sin(m\pi x)\ \sin(m\pi\xi)$, and we can deduce that :

$$\delta(x - b(t)) = \sum_{m=1}^{M} 2\sin(m\pi x)\ \sin(m\pi b(t))$$

So that,

$$f(x,t) = \sum_{m=1}^{M} 2\ v(t)\ \sin(m\pi x)\ \sin(m\pi b(t))$$

$$\Leftrightarrow \qquad \widehat{f}_m(t) = 2\ v(t)\ \sin(m\pi b(t)) \qquad \forall m \tag{26}$$

Hence, the Fourier approximation of (1) is :

$$\sum_{m=1}^{M} \left(\widehat{y}'_m(t) + \nu\ m^2\pi^2\ \widehat{y}_m(t) - \widehat{f}_m(t) \right)\ \sin(m\pi x) = 0$$

As the Fourier basis is orthogonal, we have :

$$< \frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} - v(t)\ \delta(x - b(t))\ ,\ \sin(m\pi x)\ > \ = \ 0 \qquad \forall m$$

Then, the heat equation is approximated by the following set of O.D.E's :

$$\widehat{y}'_m(t) + \nu\ m^2\pi^2\ \widehat{y}_m(t) - \widehat{f}_m(t)\ =\ 0 \qquad \forall m \tag{27}$$

But $\widehat{f}_m(t)$ depends on the control $v(t)$, which itself depends on the adjoint-state $\lambda(x,t)$, which itself depends on the state $y(x,t)$, from equation (10). We then have to evaluate $\widehat{f}_m(t)$ in terms of $(\widehat{y}_l(t))_{l=1,M}$.

We recall equation (10), where the adjoint-state depends on $S(x,\xi,t), y(x,t)$ and $\mu(x,t)$.

$$\lambda(x,t) = \int_0^1 S(x,\xi,t)\ y(\xi,t)\ d\xi - \mu(x,t)$$

$$\Leftrightarrow \qquad \lambda(x,t) = \sum_{m=1}^{M} \Big(\ \widehat{y}_m(t)\ \int_0^1 S(x,\xi,t)\ \sin(m\pi\xi)\ d\xi\ \Big)\ -\ \mu(x,t)$$

We recall that the optimal control $v(t)$ is given by $v(t) = -\dfrac{\lambda(b(t),t)}{\varepsilon}$, then,

$$v(t) = -\frac{1}{\varepsilon}\ \Big(\ \sum_{m=1}^{M} \widehat{y}_m(t)\ \int_0^1 S(b(t),\xi,t)\ \sin(m\pi\xi)\ d\xi\ -\ \mu(b(t),t)\ \Big) \tag{28}$$

from (26), it implies that :

$$\widehat{f}_m(t) = -\frac{2}{\varepsilon}\ \sin(m\pi b(t))\ \Big(\ \sum_{l=1}^{M} \widehat{y}_l(t)\ \int_0^1 S(b(t),\xi,t)\ \sin(l\pi\xi)\ d\xi\ -\ \mu(b(t),t)\ \Big)$$

By replacing the new expression of $\widehat{f}_m(t)$ in equation (27), we obtain a new ordinary differential equation :

$$\widehat{y}'_m(t) + \nu\ m^2\pi^2\ \widehat{y}_m(t) = -\frac{2}{\varepsilon}\ \sin(m\pi b(t))\ \Big(\ \sum_{l=1}^{M} \widehat{y}_l(t)\ \int_0^1 S(b(t),\xi,t)\ \sin(l\pi\xi)\ d\xi - \mu(b(t),t)\ \Big) \tag{29}$$

By setting

$$\widehat{y}_m(t) = \exp(-\nu m^2\pi^2(t-T))\ \widehat{z}_m(t),$$

we obtain :

$$\boxed{\begin{aligned} \widehat{z}'_m(t) = \ -\frac{2}{\varepsilon}\ \sin(m\pi b(t))\Big(\ \sum_{l=1}^{M} \exp(\nu\ (m^2 - l^2)\pi^2(t-T))\ \widehat{z}_l(t) \\ \int_0^1 S(b(t),\xi,t)\ \sin(l\pi\xi)\ d\xi\ -\ \mu(b(t),t)\ \Big) \end{aligned}} \tag{30}$$

Remark : $\widehat{z}_m(0) = \widehat{y}_m(0) = 0.$

By solving (30), we deduce the state $y(x,t) = \sum_{m=1}^{M} \exp(-\nu m^2\pi^2(t-T))\ \widehat{z}_m(t)\ \sin(m\pi x)$, then the optimal control $v(t)$ given by (28) and afterwards the optimal value of functional J given by (2).

5.2 Numerical results

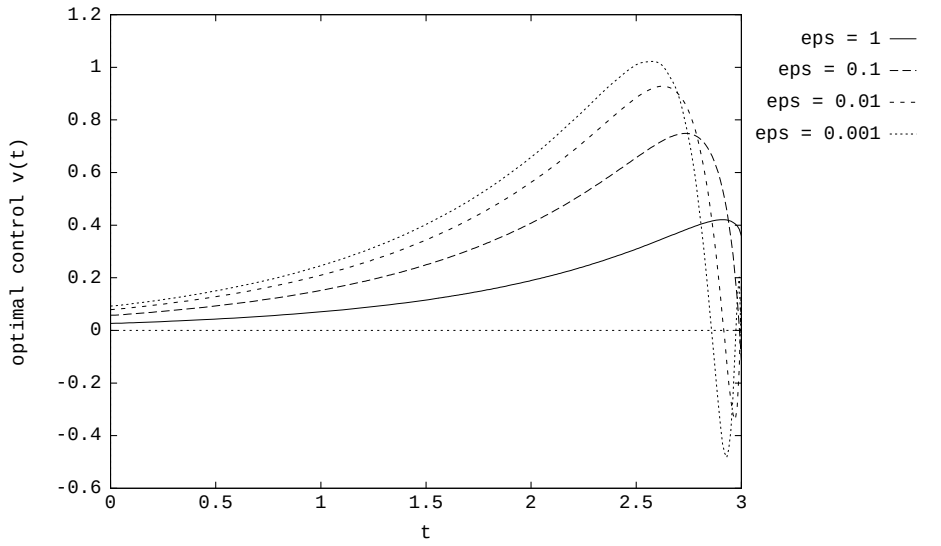
The state equation (29) is now written with $S(x,\xi,t)$ and $\mu(x,t)$ in their Fourier transformations. It is of the following form :

$$\left\{\begin{array}{l} \widehat{z}'_m(t) = -\dfrac{2}{\varepsilon}\ \exp(\nu m^2\pi^2(t-T))\ \sin(m\ \pi\ b(t))\ \Big(\ \dfrac{1}{2}\ \displaystyle\sum_{l=1}^{M} \exp(-\nu\pi^2 l^2(t-T))\ \widehat{z}_l(t) \\ \qquad \displaystyle\sum_{m_1}^{M} \exp(\nu\ (l^2+m_1^2)\pi^2(t-T))\ \widehat{\sigma}_{m_1 l}(\theta(t))\ \sin(m_1\ \pi\ b(t)) - \\ \qquad \displaystyle\sum_{m_1}^{M} \widehat{\gamma}_{m_1}(\theta(t))\ \exp(\nu m_1^2\pi^2(t-T))\ \sin(m_1\ \pi b(t))\ \Big) \\ \widehat{z}_m(0) = 0 \end{array}\right. \tag{31}$$

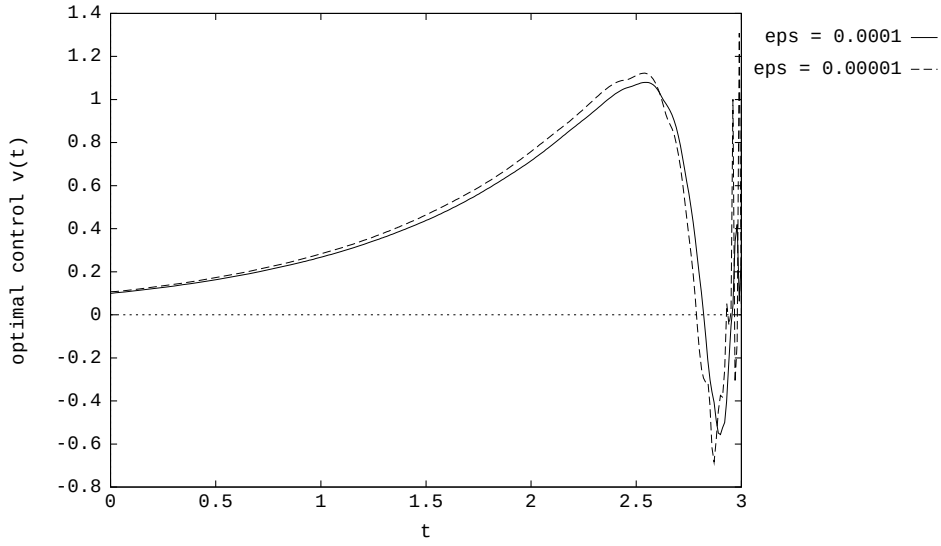
Equation (29) has been solved by the same NAG Fortran library routine as previously (D02EJF).

In Figure 5, the control $v(t)$ is depicted. The smaller ε is, the more oscillatory the control is when t is near T. In Figure 6, we also note that the smaller ε is, the better $y(x,T)$ (the state at $t = T$) approximates the target solution $y_T(x) = \sin(\pi x)$. In Figure 7, is depicted the relative error in the final state $\|y(x,T) - y_T(x)\|/\|y_T(x)\|$, (where the norm $\|.\|$ represents the continuous L^2 norm) in terms of ε. The norm tends linearly to 0, as ε vanishes.
Finally, in Figure 8, the cost functional is plotted as a function of ε. The table below indicates the values of J as a function of ε.

ε	$J(v)$
10	20.5
1	7.11
10^{-1}	6.9610^{-1}
10^{-2}	5.8510^{-2}
10^{-3}	4.9810^{-3}
10^{-4}	4.5110^{-4}
10^{-5}	4.8510^{-5}



Control $v(t)$ for ε varying from 1 to 10^{-3}.



Control $v(t)$ for $\varepsilon = 10^{-4}$ and 10^{-5}.

Figure 5: Control $v(t)$ for t varying from 0 to T, for different ε (varying from 1 to 10^{-5}). We note many oscillations when t is near the final time T.

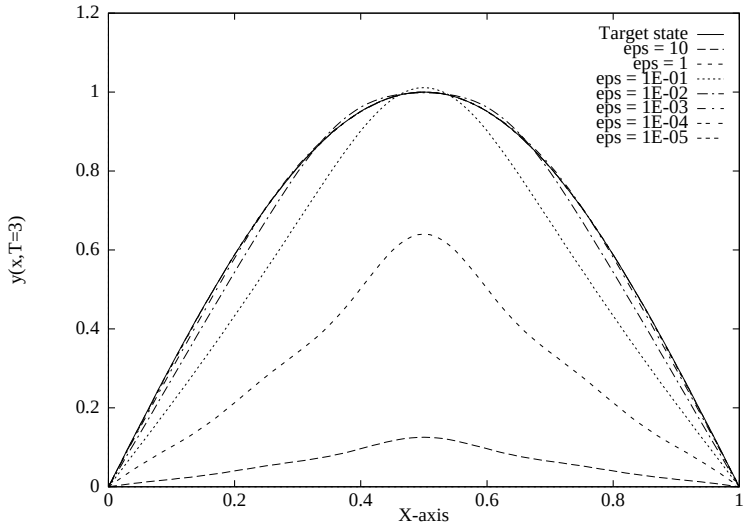


Figure 6: State $y(x,T)$, at $t = T$, when ε varies from 1 to 10^{-5}.

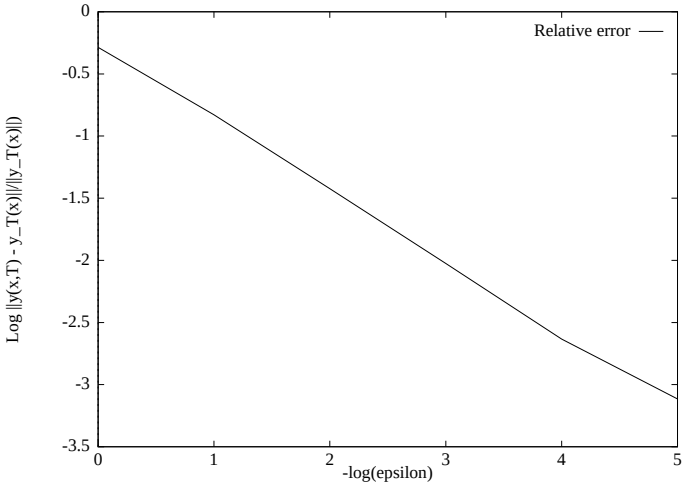


Figure 7: log of the relative error $\|y(x,T) - y_T(x)\|_{L^2}/\|y_T(x)\|_{L^2}$ at $t = T$ as a function of $-\log(\varepsilon)$. The plot indicates a linear trend as ε becomes smaller.

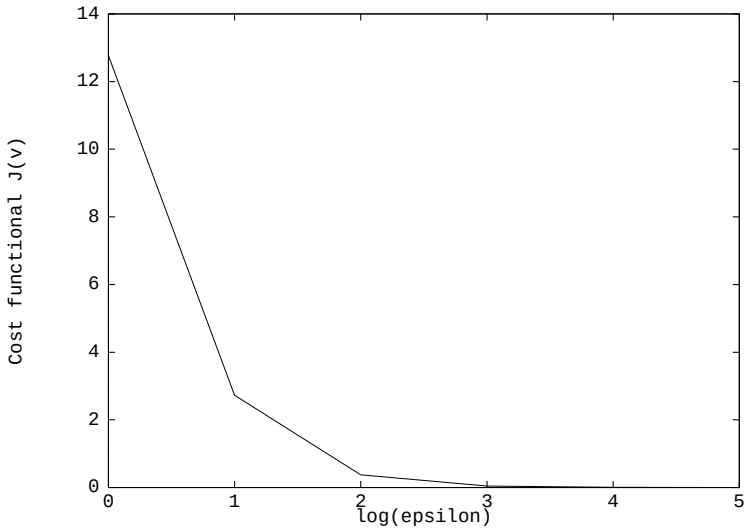


Figure 8: Cost functional $J(v)$, as a function of ε.

6 A Genetic approach

6.1 Essential points about Genetic Algorithms (GAs)

Genetic Algorithms are search algorithms based on the mechanisms of natural selection. They lay on one of the most important principles of Darwin : survival of the fittest.

Globally, we make use of a population, submitted to many transformations. After some generations, the population evolves no more ; the best individual represents the optimal solution.

Genetic algorithms are different from more normal optimization and search procedures in four ways :

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point. They can treat simultaneously several individuals. This notion of parallelism is a source of robustness.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

GAs are more robust than direct search methods. They work from a rich database of points simultaneously (a population of strings), climbing many peaks in parallel ; thus, the probability of finding a false peak is reduced over methods that go point to point.

Unlike many methods, GAs use probabilistic transition rules to guide their search. The fact that they use a random choice is a tool to guide a search toward regions of the search space with likely improvement.

The data used by GAs are generally coded in a binary finite-length *string* representation. These strings are called *individuals* or *chromosomes*. Each bit, 1 or 0, is called a *gene*. Each chromosome represents a *potential solution* of the problem to be solved.

However, there are various domains where the binary-coded genetic algorithms are no longer performent (see for example [3] or [5]). When we deal with optimum design problems in aerodynamics, there are several parameters taking values in different intervals that are to be optimized with a high accuracy. In this context, another crucial point emerges : the time needed for each fitness evaluation which demands a long time to be computed. We then use real-coded genetic algorithms (see [6], [7]).

In our actual 1D problem, we use a binary-coded genetic algorithms but it was interesting to note that GAs should be either binary-coded or real-coded.

6.2 Dynamics of GAs

An initial population is chosen randomly. At each step, called *generation*, all the individuals (or chromosomes) of the population are evaluated from their *fitness values* or *selective values*. In accordance with this evaluation, a new population more fitted is obtained by many manipulations applied to the chromosomes. We call them *genetic operators*, and there are three of them (see for more details [2] and [5] for example) :

1. *Reproduction* : it is the most important because it allows the individuals to survive, reproduct or die. The future of each chromosome is determined by its fitness value : individuals with a higher value are assigned a higher probability of contributing one or more offsprings in the next generation, and individuals with a lower value have very poor chances to survive. To realize this, some authors use a "roulette-wheel selection", others choose a "tournament" selection.

   - Roulette-wheel selection :
     To implement algorithmicly the reproduction operator, we create a biased roulette wheel where each individual in the population has a roulette wheel slot sized in proportion to its fitness value. We spin the weighted roulette wheel n times, where n is the number of individuals in one population. The individuals with high fitness will have many chances to survive. A scheme is given in figure 9, based on an example of a population with 4 individuals.
   - Tournament selection :
     In a tournament selection, a set of individuals is randomly chosen from the current population and the best of this subset is chosen to be represented in the next population. This allows the best individuals to have a high probability of being represented in future generations. Also, by adjusting the size of the tournament, we can exert some control over the amount of selection pressure and convergence speed. The tournament of smallest size, that is two (or "binary tournament") exhibits a convergence slower than any tournament of larger size.

2. *Crossover* : as reproduction does not create new individuals, the crossover operator is needed to increase diversity among the population. It is required that the population evolves. Crossover proceeds in two steps. First, strings of the newly reproduced population are mated at random ; this crossover operation is made

with a certain probability p_c. Second, each pair of strings undergoes crossing over as follows : an integer position k along the string is selected uniformly at random in $[1, l-1]$, where l is the strig length. Two new strings are created by swapping all characters between positions $k+1$ and l inclusively. A skematic of the crossover operation is given in figure 10.

3. *Mutation* : it is needed because, reproduction and crossover can occasionally loose some potentially useful genetic material (1's or 0's at particular locations). Mutation is then an insurance policy against premature loss of important notions. In the binary coding, this simply means changing a 1 to a 0 and vice versa, randomly, with a small probability p_m. Figure 11 gives a scheme of mutation.

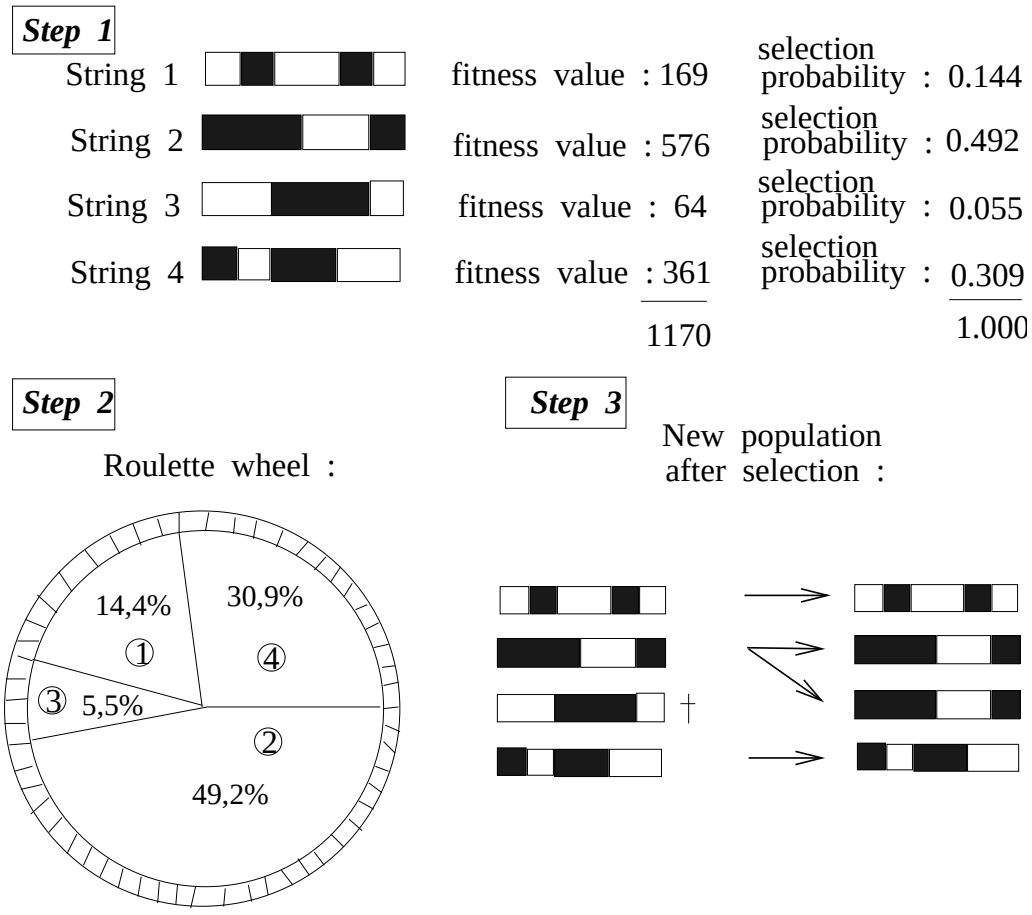


Figure 9: An example of roulette-wheel reproduction for the case of 4 strings : first we evaluate each individual, secondly, we spin 4 times the roulette wheel and finally we obtain a new population.

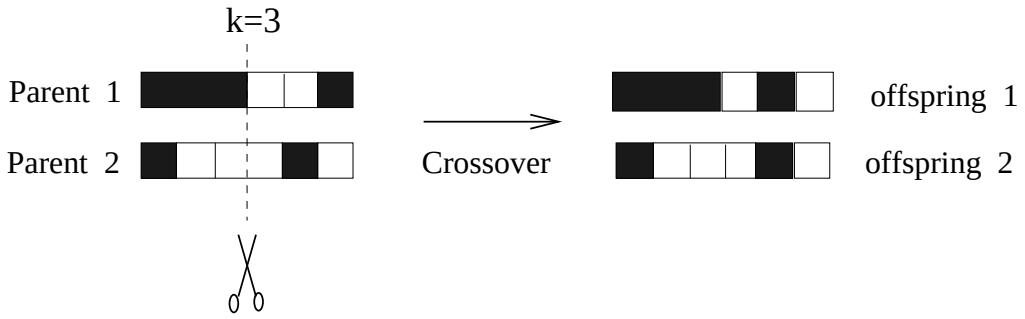


Figure 10: An example of crossover between two parents.

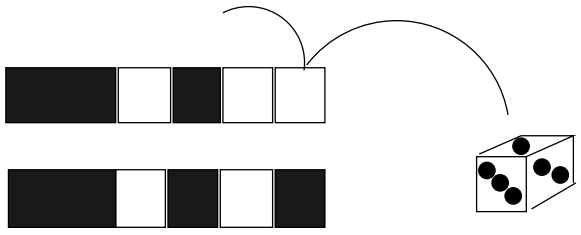


Figure 11: A schematic of simple mutation.

6.3 Numerical experiments

A standard genetic algorithm defined below was employed in these experiments.

BEGIN

$gen \leftarrow 0$ (initialization of the generation)
initialize population P_0
REPEAT (main iteration loop)
evaluate P_{gen}
apply crossover and mutation on P_{gen}
$gen \leftarrow gen + 1$
UNTIL $gen > gen_{max}$

END

The control variable $v(t)$ is represented by 20 parameters discretizing this function at 20 different time locations :

$$v(t) = (v(t_1), v(t_2), \cdots, v(t_{20}))$$

$t_1, \cdots, t_{20} \in [0, 3]$:

$$\{t_i\} = \{0., 0.25, 0.5, 0.75, 1., 1.25, 1.5, 1.75, 2., 2.25, 2.5, 2.6, 2.65, 2.7, 2.75, 2.8, 2.85, 2.9, 2.95, 3.\}$$

Each parameter $v(t_i)$ is coded on 16 bits. Then, one individual (a control variable) is composed of $20 \times 16 = 320$ genes.
We have worked with a population of 20 individuals.
We choosed $p_c = 0.6$ and $p_m = 0.001$.
We stop the genetic algorithm when $gen_{max} = 400$.
We have used tournament selection of size two.

First of all, we have initialized randomly the population of controls. We have obtained, for several values of ε, a final control, a final fitness value and a final state associated with the Genetic Algorithm. We compare these results with the solutions obtained with the Riccati method.
We note that when ε is not small enough, the Genetic Algorithm produces better functional values than the Riccati method (see Table 1). Moreover, for $\varepsilon \leq 10^{-2}$, Riccati solution gives better cost functionals (see Table 2). The controls obtained with the Genetic Algorithm are shown on Figures 13.

ε	$J(v) - Riccati$	$J(v) - Genetic$
		with random initialization
10	20.5	12.03
1	7.11	2.2
10^{-1}	6.9610^{-1}	3.9310^{-1}

Table 1 : Riccati solution vs Genetic Algorithm with random initialization, when ε is not small.

ε	$J(v) - Riccati$	$J(v) - Genetic$
		with random initialization
10^{-2}	5.8510^{-2}	7.7110^{-2}
10^{-3}	4.9810^{-3}	3.910^{-2}
10^{-4}	4.5110^{-4}	1.4910^{-2}
10^{-5}	4.8510^{-5}	1.2610^{-2}

Table 2 : Riccati solution vs Genetic Algorithm with random initialization, when ε is small.

The optimized "genetic" controls $v(t)$, for the different values of ε are observed to be very noisy. While the crossover and mutation operators are necessary to increase the algorithm robustness by spanning the search space more completely, these operators are responsible for the introduction of noise. Despite the noise, the final states $y(x,T)$ converge toward the target state, when ε becomes smaller. For greater ε ($\varepsilon = 10, 1$), the target state is not reached but the same behaviour is observed for the Riccati method. The final states obtained with the Genetic Algorithm are presented on Figure 14.

Secondly, we have initialized the population of controls with the optimized control given by the Riccati method. For all the values of ε, the obtained cost functionals are better (i.e less) than those obtained with the random initialization.
By comparing with the Riccati solution, the cost functionals, for $\varepsilon \geq 10^{-2}$, are better with the genetic approach (see Table 3), but when ε becomes smaller, the Riccati method gives better results (see Table 4). The different controls obtained with the Genetic Algorithm with Riccati initialization are presented on Figure 15, for the different values of ε. The different final states are presented on Figure 16.

ε	$J(v) - Riccati$	$J(v) - Genetic$
		with Riccati initialization
10	20.5	11.9
1	7.11	2.1
10^{-1}	6.9610^{-1}	3.410^{-1}
10^{-2}	5.8510^{-2}	4.2910^{-2}

Table 3 : Riccati solution vs Genetic Algorithm with Riccati initialization, when ε is not small.

ε	$J(v) - Riccati$	$J(v) - Genetic$
		with Riccati initialization
10^{-3}	4.9810^{-3}	6.810^{-3}
10^{-4}	4.5110^{-4}	1.5910^{-3}
10^{-5}	4.8510^{-5}	2.710^{-4}

Table 4 : Riccati solution vs Genetic Algorithm with Riccati initialization, when ε is small.

The final controls $v(t)$ obtained by the G.A, when ε is not small ($\varepsilon = 10, 1, 10^{-1}$) still present some irregularities, but their general trend agree with the solution of the

Riccati method.

Remark : The cost functional J is convex and we may wonder why we find a better cost functional with the Genetic Algorithm (for some values of ε), whereas the Riccati solution seems to be exact... In fact, if V^* is the continuous optimal control and if $J(V^*)$ is the optimal cost functional, when we solve the control problem with Riccati method, we do not obtain V^* but V_h^*, sought as a projection of V^* ($V_h^* = P_h V^*$) in a discrete approximation subspace. This means that we minimize $(MinJ(V))_h = V_h^*$.

By solving with the Genetic Algorithm, we discretize V in V_h, and we get, at the different generations, some $V_h^1, V_h^2, \cdots, V_h^n$ (indices 1, 2, $\cdots n$ denote the number of the current generation) and finally, the optimal control is V_h^∞. This means that we minimize $MinJ(V_h) = V_h^\infty$.

$V_h^* - V^*$ is an approximation error, and $J(V_h^*) - J(V^*)$ is an indirect measure of this approximation.
$J(V_h^\infty) - J(V^*)$ is a defect due to optimizing in an incomplete subspace. A skematic of this remark is given in Figure 12. Thus it is possible that in some cases $J(V_h^\infty) < J(V_h^*)$, as observed in our experiments for large ε.

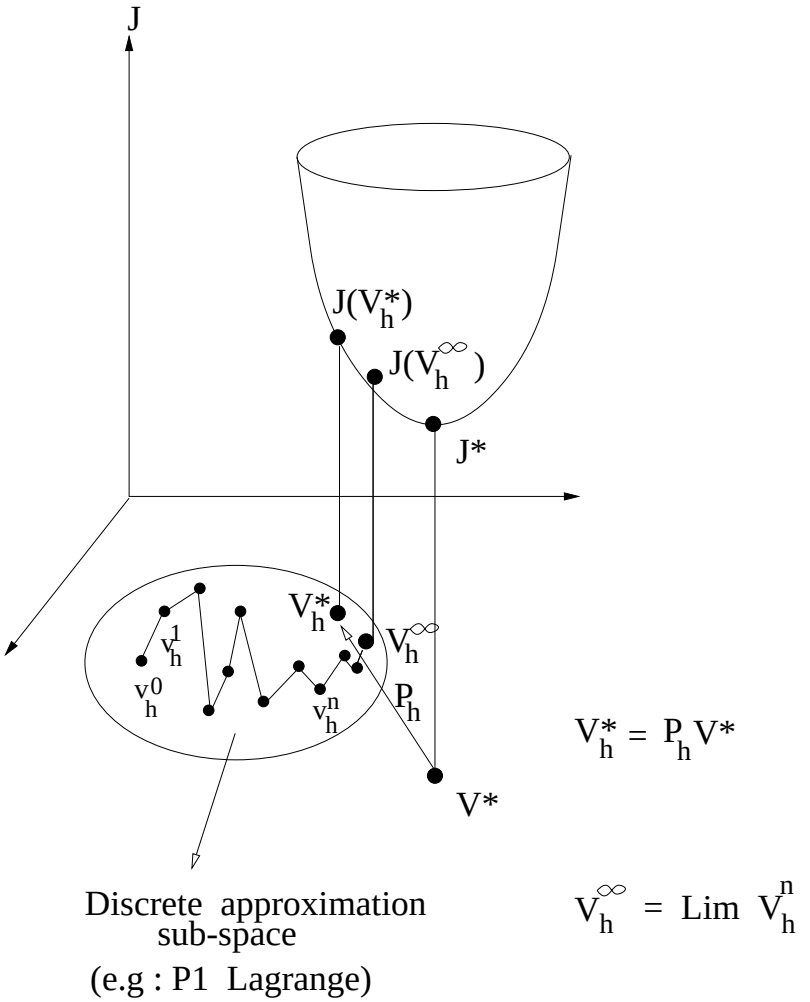


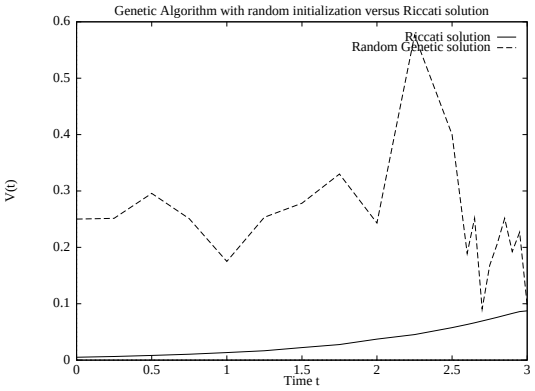
Figure 12: Minimization with Riccati method and Genetic method.

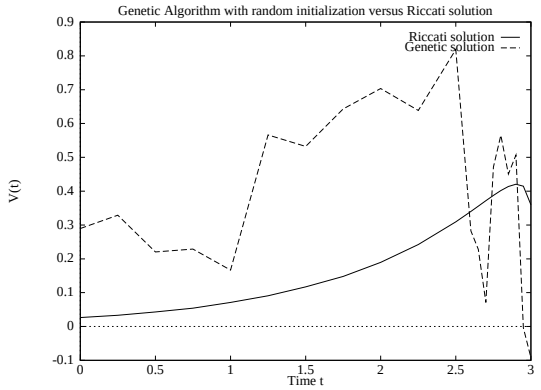
The termination criterion of the optimization, was to stop when the number of generations had reached $gen_{max} = 400$ or when the functional had become constant. In Figures 14 and 16, the number of generations used to get the final states, for different ε, are noted.

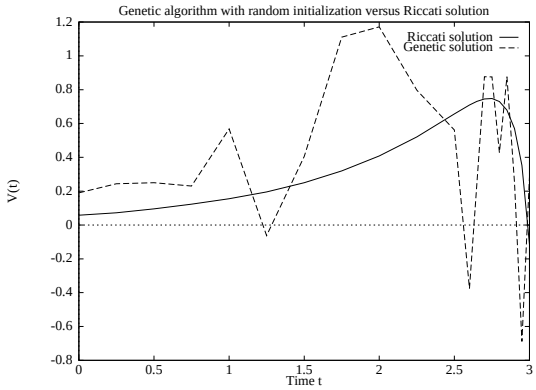
Our investigations were performed on a SUN Station SS10. 400 generations were done in about $14h30$ CPU, 250 generations were done in about $9h30$ CPU.

7 Concluding remarks

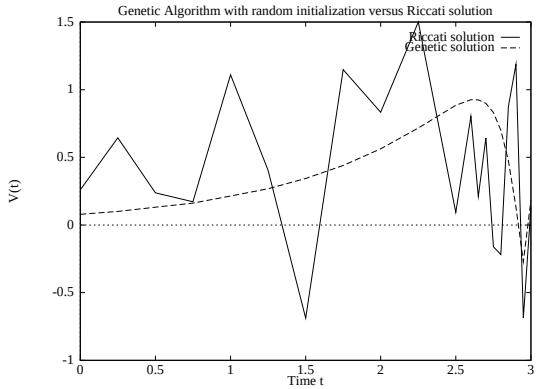
We have verified that the Genetic Algorithm allowed to solve the optimization problem. However, costs are higher. When ε is smaller, the stiffness of the problem is such that the Genetic Algorithm reproduces the known optimal solution (by Riccati equation) in a resonable CPU time only if the initial condition is near the optimal solution. This difficulty would deserve more investigation since robustness is the major expected quality of GA's. However, we must moderate this remark because the problem studied although simple to implement is a singular perturbation problem that becomes very stiff numerically as ε tends to 0.

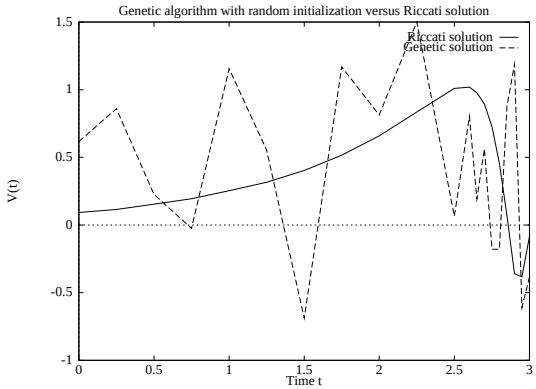




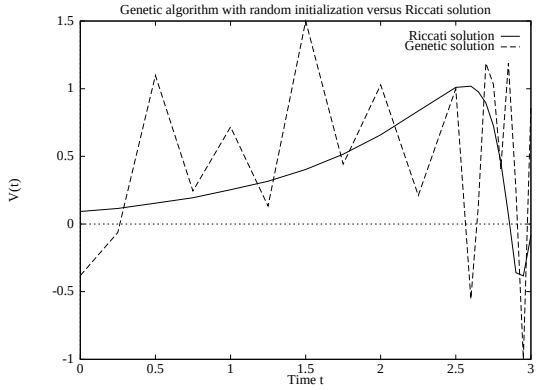


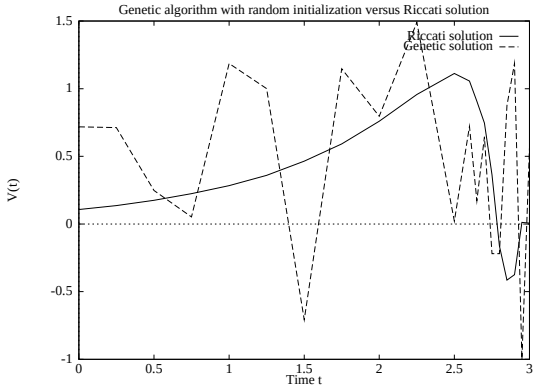
Genetic Algorithm with random initialization, when $\varepsilon = 10, 1$ and 10^{-1}.





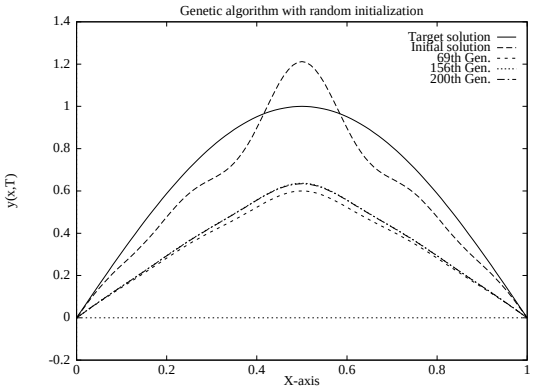
Genetic Algorithm with random initialization, when $\varepsilon = 10^{-2}$ and 10^{-3}.

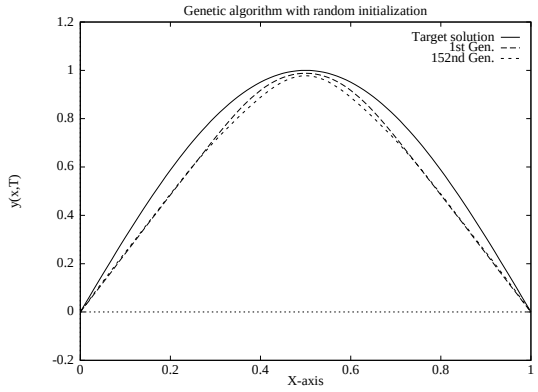


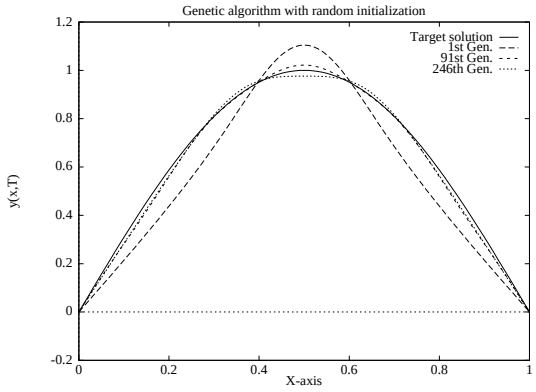


Genetic Algorithm with random initialization, when $\varepsilon = 10^{-4}$ and 10^{-5}.

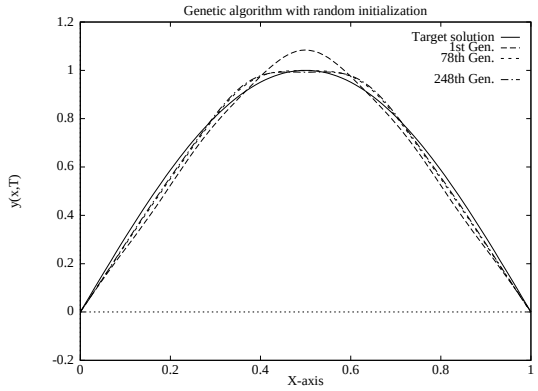
Figure 13: Optimized controls obtained by the Genetic Algorithm with random initialization versus optimized control obtained by Riccati solution, for different values of ε.

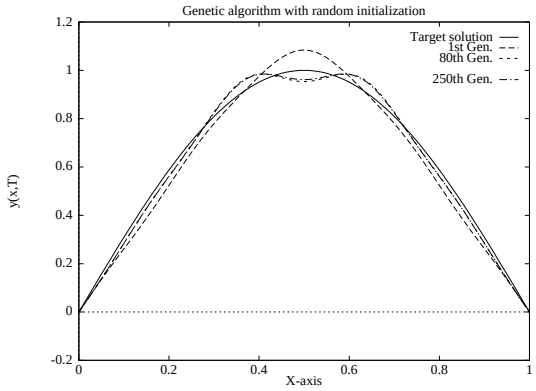




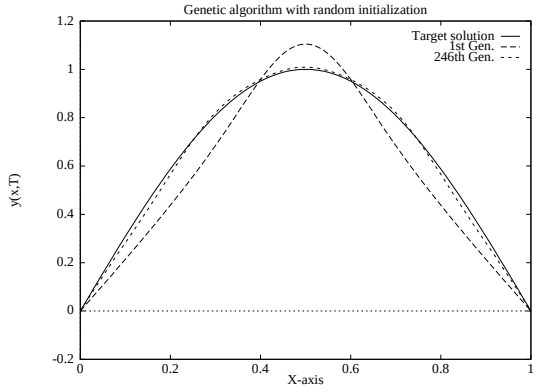


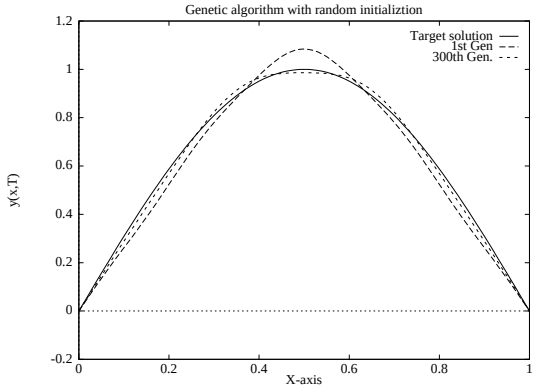
Genetic Algorithm with random initialization, when $\varepsilon = 10, 1$ and 10^{-1}.





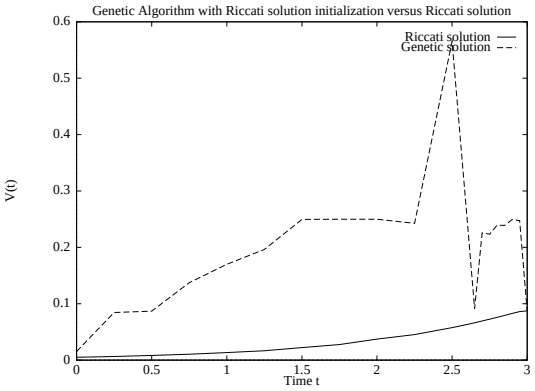
Genetic Algorithm with random initialization, when $\varepsilon = 10^{-2}$ and 10^{-3}.

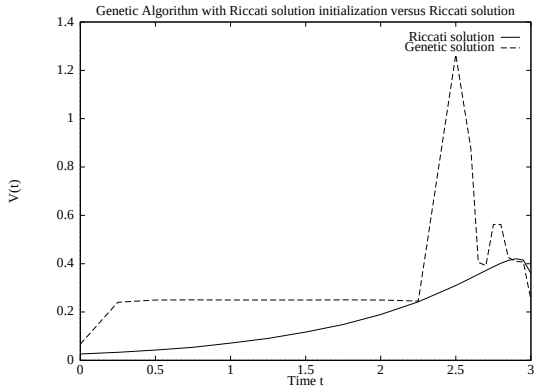


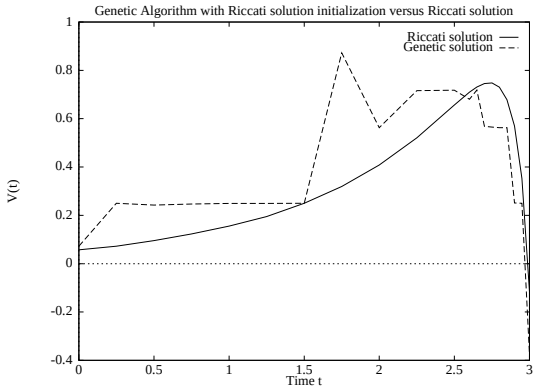


Genetic Algorithm with random initialization, when $\varepsilon = 10^{-4}$ and 10^{-5}.

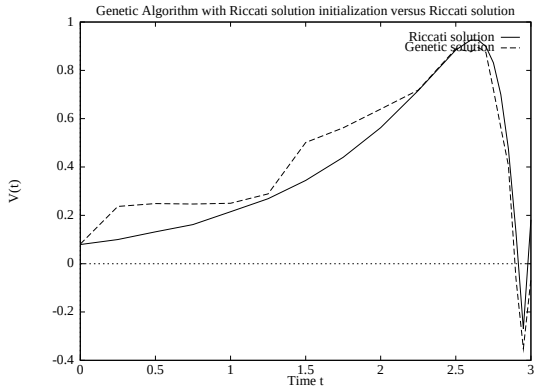
Figure 14: Final states obtained by the Genetic Algorithm with random initialization versus the final states obtained with the Riccati solution, for different values of ε.

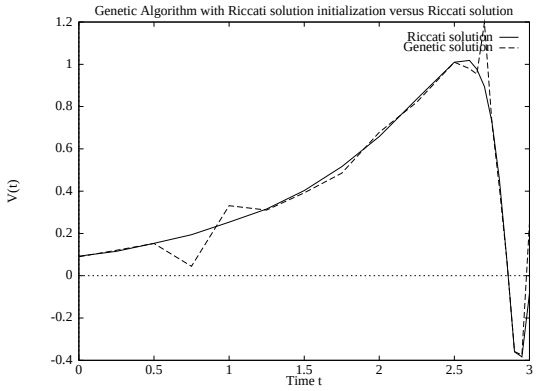




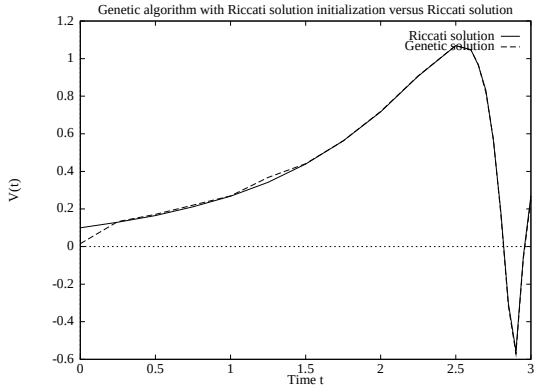


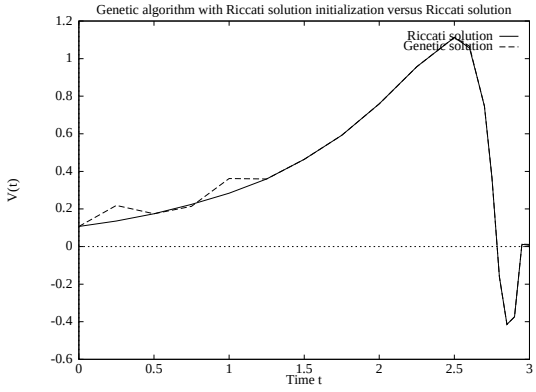
Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10, 1$ and 10^{-1}.





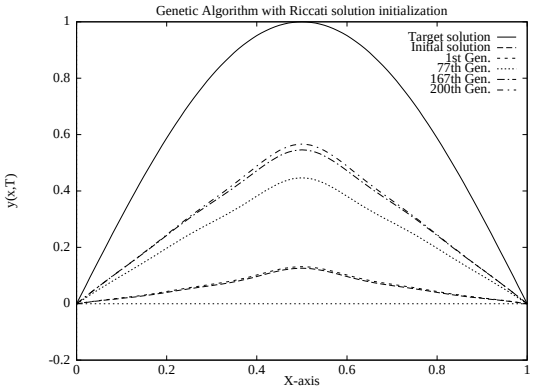
Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10^{-2}$ and 10^{-3}.

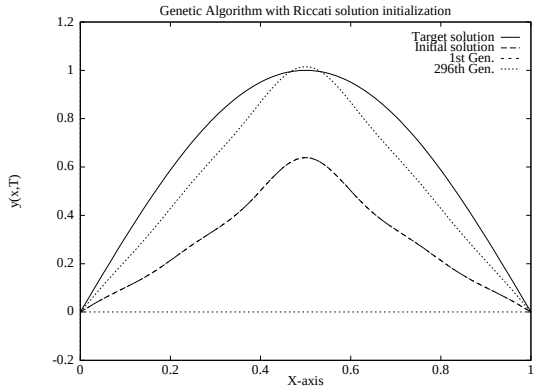


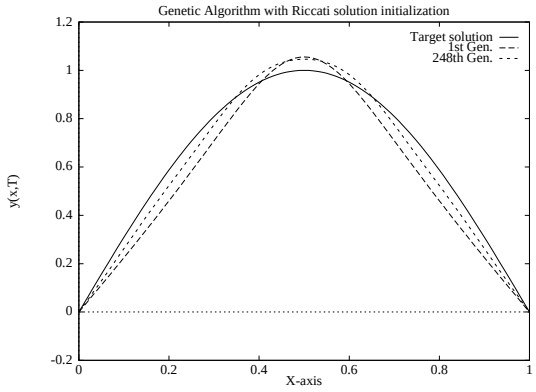


Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10^{-4}$ and 10^{-5}.

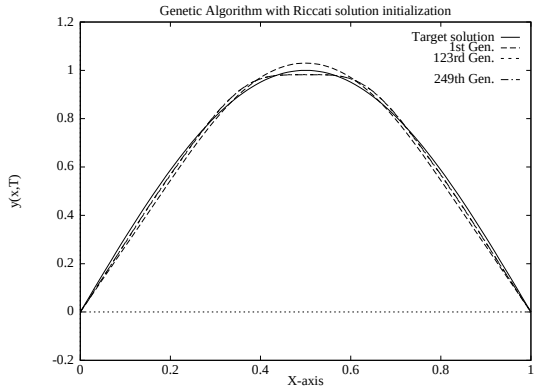
Figure 15: Optimized controls obtained by the Genetic Algorithm with initialization by the Riccati optimal solution versus optimized control obtained by Riccati solution, for different values of ε.

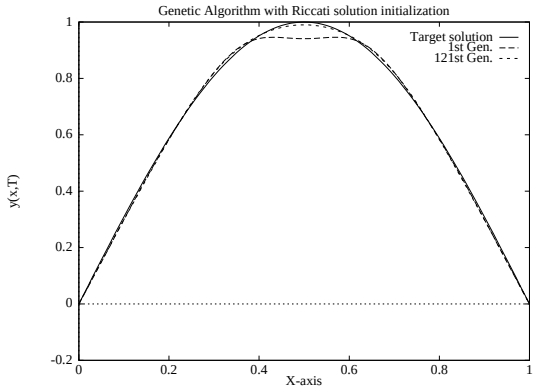




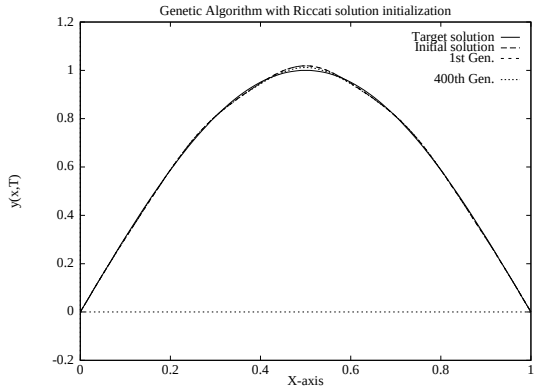


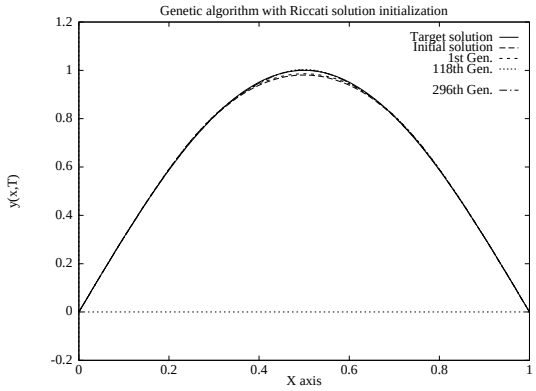
Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10, 1$ and 10^{-1}.





Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10^{-2}$ and 10^{-3}.





Genetic Algorithm with initialization by the Riccati optimal solution, when $\varepsilon = 10^{-4}$ and 10^{-5}.

Figure 16: Final states obtained by the Genetic Algorithm with initialization by the Riccati optimal solution, for different values of ε.

References

[1] A.E. BRYSON and Y.C. HO. *Applied Optimal Control.* Blaisdell Publishing Company, Waltham (Massachussets), 1969.

[2] D.E. GOLDBERG. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Company INC., 1989.

[3] C.Z. JANIKOW and Z. MICHALEWICZ. An experimental comparison of binary and floating point representations in genetic algorithms. In R. Belew and L. Booker, editors, *Proceedings of the Fourthth International Conference on Genetic Algorithms*, Los Altos (CA), 1991. Morgan Kaufmann.

[4] J.L. LIONS. *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles.* Dunod Gauthier-Villars, Paris, 1968. Etudes Mathématiques, collection dirigée par P. Lelong.

[5] Z. MICHALEWICZ. *Genetic algorithms + data structures = evolution programs.* Springer-Verlag, New-York, 1992. J. Périaux and G. Winter editors, Artificial Intelligence. Genetic Algorithms in Engineering and Computer Science.

[6] J. PERIAUX, M. SEFRIOU, B. STOUFFLET, B. MANTEL, and E. LAPORTE. Robust Genetic Algorithms for Optimization Problems in Aerodynamic Design. In M. Galàn G. Winter, J. Périaux and P. Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*, pages 371–396. John Wiley and Sons, 1995.

[7] D. QUAGLIARELLA. Genetic Algorithms Applications in Computational Fluid Dynamics. In M. Galàn G. Winter, J. Périaux and P. Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*, pages 417–442. John Wiley and Sons, 1995.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399