TOWARDS AN EXPERT SYSTEM IN STOCHASTIC CONTROL : OPTIMIZATION IN THE CLASS OF LOCAL FEEDBACKS

C.GOMEZ - J.P. QUADRAT - A.SULEM

I INTRODUCTION

Stochastic control problems can be solved completely or approximatively by different kind of approaches :

- dynamic programming
- decoupling technique
- stochastic gradient
- perturbation method.

The set of these methods are described in THEOSYS [11] for example.

For each approach we are designing a generator of program able to write automatically fortram program solving the problem.

In Gomez-Quadrat-Sulem [10] we have described a set of automatic tools to solve the problem by the dynamic programming approach.

In this paper we explain the decoupling approach, discuss the possibility of the corresponding generator. Then we give an example of generated program and the numerical results obtained by this generated program.

The plan is the following :

We want solve the stochastic control problem for diffusion processes that is

$$\underset{u}{\text{Min}}\ E \int_0^T C(t,X_t,U_t)dt$$

where U_t is the control and X_t is a diffusion process satisfying the stochastic differential equation

$$dX_t = b(t,X_t,U_t)dt + \sigma(t,X_t)dW_t$$

where W_t denotes a brownian motion b and σ are given functions.

When X_t belongs to $\mathbb{R}^n$ n large perhaps larger than 3 or 4 the traditional dynamic programming approach cannot be used practically. We have to apply other methods which do not give the optimal feedback but a good one or the optimum in a subsclass of the general feedback class.

In the next paragraph we explain the way of computing the optimal local feedback that is we suppose that each control is associated to a subsystem described by a subset I_i of the component of X_t and depends only of the corresponding components of the state.

$$\underset{i}{\mathbb{U}} : (X_j,\ j \in I_i) \to R$$

$\underset{i}{U}\, I_i = \{1,\ldots,n\}$ where n is the dimension of X.

II. OPTIMIZATION IN THE CLASS OF LOCAL FEEDBACKS.

In this paragraph we give the optimality conditions in the class of local feedbacks, and show that it is more difficult to solve these conditions than to compute the solution of the Hamilton-Jacobi equation. Then we study two particular cases :

- the case of the uncoupled dynamics,
- the case of systems having the product form property.

In these cases only it is possible to compute the optimal local feedbacks for large systems. Finally we discuss briefly the decoupling point of view.

2.1. *The general situation.*

Given I the indexes of the subsystems $I = \{1,2,\dots,k\}$ n_i, [resp-m_i] denotes the dimension of the states [resp.the controls] of the subsystem $i \in I$. The local feedback S_i is a mapping of $\mathbb{R}^+ \times \mathbb{R}^{n_i}$ in $\mathcal{U}_i \subset \mathbb{R}^{m_i}$ the set of the admissible values of the control i. $\mathcal{S}_L$ denotes the class of local feedbacks $\mathcal{S}_L = \{S = (S_1,\dots,S_k)\}$. Given the drift term of the system :

$$\begin{array}{ccccccc} b : & \mathbb{R}^+ & \times \mathbb{R}^n & \times \mathcal{U} & \rightarrow \mathbb{R}^n \\ & t & x & u & b(t,x,u) \end{array}$$

with $$n = \sum_{i\in I} n_i, \mathcal{U} = \prod_{i\in I} \mathcal{U}_i,$$

- the diffusion term :

$$\begin{array}{ccccc} \sigma : & \mathbb{R}^+ & \times \mathbb{R}^n & \rightarrow M_n \\ & t & x & \sigma(t,x) \end{array}$$

with M_n the set of matrices (n,n) and $a = \frac{1}{2}\sigma\sigma^*$ where $*$ denotes the transposition

- the instantaneous cost :

$$\begin{array}{ccccccc} c = & \mathbb{R}^+ & \times \mathbb{R}^n & \times \mathcal{U} & \rightarrow \mathbb{R}^+ \\ & t & x & u & c(t,x,u) \end{array}$$

then boS [resp coS] denotes the functions $\mathbb{R}^+ \times \mathbb{R}^n \rightarrow \mathbb{R}^n$

$[\text{resp } \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^+] \quad b(t,x,S(t,x)) \quad \text{resp } c(t,x,S(t,x))$

Then if X^S denotes the diffusion $(b\circ S,a)$ (drift $b\circ S$, and diffusion term σ) and P^S_μ its measure defined on $\Omega = C(\mathbb{R}^+, \mathbb{R}^n)$ with μ the law of the initial condition we want to solve

$$\underset{S\in\mathcal{S}_L}{\text{Min}} \ \mathbb{E}_{P^S_\mu} \int_0^T C\circ S(t,\omega_t)dt$$

where $\omega \in \Omega$, T denotes the time horizon. We have here a team of I players working to optimize a single criterion.

A simple way to obtain the optimality conditions is to consider another formulation of this problem : the control of the Fokker Planck equation that is :

$$\underset{S\in\mathcal{S}_L}{\text{Min}} \quad J^S = \int_Q C\circ S(t,x)p^S(t,x)dt\ dx$$

with p solution of

$$\mathcal{L}^*_S\, p^S = 0$$

$$p^S(0,.) = \mu$$

with $\quad Q = [0,T] \times \mathcal{O} \quad$ and $\mathcal{O} = R^n$

$$\mathcal{L}_S = \frac{\partial}{\partial t} + \sum_j b_j\circ S \frac{\partial}{\partial x_j} + \sum_{i,j} a_{ij} \frac{\partial^2}{\partial x_i \partial x_j}$$

μ the law of the initial condition.

Than we have :

<u>Theorem 1</u>

<u>A N.S.C. for $J^R \geq J^S$, R $\quad S \in \mathcal{S}_L$, is that :</u>

(1) $\quad \underline{H(t,R,p^R,V^S) \geq H(t,S,p^R,V^S) \quad \text{pp in } t}$

with

$$(2)\quad \begin{cases} \underline{H(t,R,p,V)} = \int_{\mathcal{O}} \underline{[CoR(t,x) + \sum_i b_i oR(t,x)} \frac{\partial V}{\partial x_i} \underline{(t,x)]\ p(t,x)dx} \\ \underline{\mathcal{L}^*_R p^R = 0 \ p^R(0,.) = \mu \ ; \mathcal{L}_S \ V^S + CoS = 0,\ V^S(T,.) = 0} \end{cases}$$

Remark 1. From this theorem the Pontriaguine condition can be obtained, that is a necessary condition of optimality of the strategy S is that : p,V,S satisfy

$$H(t,S,p^S,V^S) = \underset{R \in \mathcal{S}_L}{\mathrm{Min}} \ H(t,R,p^S,V^S) \ ;$$

$$(3)\quad \begin{cases} \mathcal{L}^*_S p^S = 0 \quad , \quad p(0,.) = \mu \ ; \\ \mathcal{L}_S V^S + CoS = 0 \quad , \quad V^S(T,.) = 0. \end{cases}$$

A proof is given in J.L. Lions [8].

Remark 2. This theorem give an algorithm to improve a given strategy R that is :

Step 1 : compute p^R

Step 2 : solve backward simultaneously

$$(4)\quad \begin{cases} \mathcal{L}_S V^S + CoS = 0 \qquad V^S(T,.) = 0 \\ S \in \mathrm{Arg}\ \underset{Z}{\mathrm{Min}}\ H(t,Z,p^R,V^S) \end{cases}$$

By this way we improve the strategy.

A fixed point of the application $R \to S$ will satisfy the conditions (3).

We see that one iteration (4) of this algorithm is more expensive than the computation cost of the solution of the H.J.B. equation.

2.2. *Uncoupled dynamic systems.*

This is the particular case where b_i is a function of x_i and u_i, $\forall i \in I$

$$\begin{array}{ccccccc} b_i : \mathbb{R}^+ & \times & \mathbb{R}^{n_i} & \times & \mathcal{V}_i & \to & \mathbb{R}^{n_i} \\ t & & x_i & & u_i & & b_i(t,x_i,u_i) \end{array}$$

and the noises are not coupled between the subsystems that is :

$$\begin{array}{ccc} \sigma_i : \mathbb{R}^+ \times \mathbb{R}^{n_i} & \to & M_{n_i} \\ t \quad x_i & & \sigma_i(t,x_i) \end{array}$$

In this situation we have

$$p^R = \prod_{i\in I} p_i^{R_i}$$

with $p_i^{R_i}$ solution of

(5) $$\mathscr{L}^*_{i,R_i}\, p_i^{R_i} = 0 \quad p_i^{R_i}(0..) = \mu_i \qquad \text{with} \quad = \prod_{i\in I} \mu_i$$

and

$$\mathscr{L}_{i,R_i} = \frac{\partial}{\partial t} + \sum_{k\in I_i} b_k o R_i(t,X) \frac{\partial}{\partial X_k} + \sum_{k,\ell\in I_i} a_{k\ell} \frac{\partial^2}{\partial X_k \partial X_\ell}$$

with $I_i = \{ \sum_{j<i} n_j < k \leq \sum_{j<i+1} n_j \}.$

Let us denote by

(6) $$\begin{array}{ccc} C_i^R o R_i : \mathbb{R}^+ \times \mathbb{R}^{n_i} & \to & \mathbb{R}^+ \\ t \quad x_i & & \int CoR(t,x) \prod_{j\neq i} p_j^{R_j}(t,x_j)dx_j \end{array}$$

That is the conditional expectation of the instantaneous cost knowing the information only on the local subsystem i.

We have the following sufficient conditions to be optimal player by player :

<u>Theorem 2.</u> <u>A sufficient condition for a strategy S to be optimal player by player is that the following conditions are satisfied</u> :

(7) $$\underline{\operatorname*{Min}_{R_i} [\mathscr{L}_{i,R_i} V_i + C_i^R o R_i] = 0, \quad i \in I}\ ;$$

<u>with $C_i^R o R_i$ defined by (6) and (5)</u>

The optimal cost is $\mu_1(V_1)\ldots = \mu_I(V_I)$ with $\mu_i(V_i) = \int_{\mathbb{R}^{n_i}} \mu_i(dx_i)V_i(o,x_i)$

Remark 3. The theorem 3 gives an algorithm to compute a feedback optimal player by player

given $\varepsilon,\ \nu \in \mathbb{R}^+$

Step 1) Choose $i \in I$

Solve (7)

if : $\mu_i(V_i) \leq \nu - \varepsilon$ than $\nu : \mu_i(V_i)$

$$R_i := \operatorname*{Arg\,Min}_{R_i} \{\mathscr{L}_{i,R_i} V_i + C_i^R o R_i\}$$

if not choose another $i \in I$ until

$$\mu_i(V_i) \geq \nu - \varepsilon, \forall i \in I.$$

Step 2) When $\mu_i(V_i) \geq \nu - \varepsilon$, $\forall i \in I$, than $\varepsilon := \frac{\varepsilon}{2}$, go to step 1.

By this algorithm we obtain a decreasing sequence $\nu^{(n)}$ which converges to a cost optimal player by player.

For a proof of a discrete version of this algorithm see Quadrat-Viot [1].

Remark 4. The interpretation of $V_i(t,X_i)$ $i \in I$ in terms of the variables of theorem 1 is :

$$V_i(t,x_i) = \int V(t,x) \prod_{j \neq i} p^{R_j}(t,X_j)dX_j$$

Remark 5. In this situation we have to solve a coupled system of P.D.E. but each of them is defined on a space of small dimension. By this way we can optimize, in the class of local feedback, systems which are not reachable by H.J.B. equation. An application to hydropower systems is given in Delebecque-Quadrat [2].

2.3. *Systems having the product form property.*

The property that a system has its dynamic uncoupled is very restrictive in this paragraph, we show systems which have their invariant measure uncoupled, they are limit of network of queues of Jackson type. This property can be used to apply to them the results of 2.2. for the corresponding ergodic control problem that is :

$$\underset{S}{\mathrm{Min}} \lim_{T\to\infty} \frac{1}{T} \int_0^T \mathrm{CoS}(\omega_t)\,dt$$

Given B a generator of a Markov chain defined on $E = \{1,2,\dots,n\}$, a function $\begin{array}{l} E \times R \to R \\ (i,x) \;\; u_i(x) \end{array}$ a matrix $\sigma \in M_n$, $A = \frac{1}{2}\sigma\sigma^*$, D a diagonal matrix satisfying :

(8) $$DB^* + BD + 2A = 0$$

Theorem 3.

The invariant measure of probability p of the diffusion ($b = Bu$, $a=A$) such that (8) is true has the product form property that is :

(9) $$p(x) = C \prod_{i=1}^{n} p_i(x_i) \;, \quad i \in E$$

(10) $$p_i(x_i) = \exp - \frac{1}{d_{ii}} \int_0^{x_i} u_i(s)\,ds$$

where C is a constant of normalization.

Demonstration : The Fokker-Planck equation can be written :

(11) $$-\operatorname{div}[bp] + \operatorname{div}[A \operatorname{grad} p] = 0$$

Let us make the change of variables $p = \exp V$ in (11), we obtain

$$(\operatorname{grad} V, b - A \operatorname{grad} V) + \operatorname{div}(b - A \operatorname{grad} V) = 0$$

Using (10), we have :

(12) $$(D^{-1}u, (B + AD^{-1})u) + \operatorname{tr}[(B + AD^{-1}) \operatorname{grad} u] = 0$$

The quadratic part in (u) of (12) is equal to 0 if and only if :

$$D^{-1}B + B^*D^{-1} + 2D^{-1}AD^{-1} = 0$$

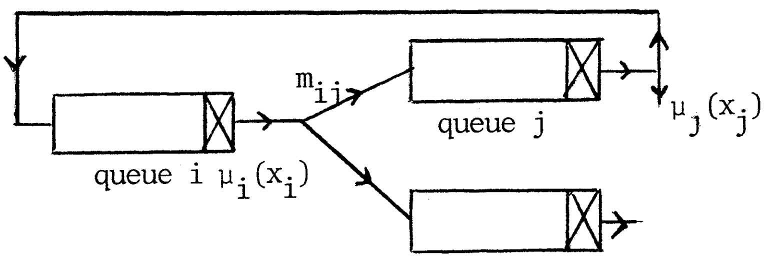
which can be written :

$$BD + DB^* + 2A = 0$$

which is (8).

We have also $\operatorname{tr}[B + AD^{-1}] \operatorname{grad} u = 0$. Indeed grad u is diagonal because u_i is a function of x_i only and the coefficient of $\frac{\partial u_i}{\partial x_i}$ is $b_{ii} + a_{ii} / d_{ii}$ which is equal to zero by (8).

Remark 6. This class of diffusion processes are quite natural if we see them as the limit process when $N \to \infty$, obtained from Jackson network of queues by the scaling $x \to \frac{x}{N}$, $t \to \frac{t}{N^2}$.



where $\mu_i(x_i)$ is the output rate of the queue i, m_{ij} is the probability of a customer leaving the queue i to go to the queue j.

The correlation of the noise given by (8) corresponds to system for which the noise satisfies a conservation law (for example the total number of customer in a closed network of queues).

Remark 7. We can now apply the result of 2.2 to compute the optimal local feedback for systems having the product form property and an ergodic criterion. Indeed :

$$\operatorname{Min} \frac{1}{T} \int_0^T \operatorname{CoS}(\omega_t)dt = \int \operatorname{CoS}(x)\, p(x)dx$$

$$p(x) = \prod_{i=1}^{n} p_i(x_i)$$

and p_i satisfies :

$$- \frac{\partial}{\partial x_i}[u_i p_i] + \frac{\partial^2}{\partial x_i^2}[d_{ii}\, p_i] = 0, \qquad i \in E$$

$$\int p_i(x_i)dx_i = 1$$

2.4. *Remarks on decoupling.*

Another way to use the results of 2.2 when the dynamic is coupled is to do a change of feedback lest us consider the simpler case

$$\begin{array}{ccccc} b : \mathbb{R}^n & \times & U \to \mathbb{R}^n & & \text{with } u \in \mathbb{R}^n \\ x & & u \quad b(x,u) & & \end{array}$$

we use the feedback transformation $v = b(x,u)$ to decouple the drift terms. Now v is the control and we can apply the results of 2.2 to compute the best local feedback $v_i = S_i(x_i)$.

Then the solution in u of

$$b(x,u) = S(x) \tag{13}$$

gives the best feedback among the class that we can call "decoupling feedbacks".

One difficulty with this approach is for example the constraints on the control: the image by b of an hypercube is not in general an hypercube and if we take for constraints on the new control $v \in V(x) \subset b(x,\mathcal{U})$ with $V(x)$ an hypercube of $\mathbb{R}^n$, the loss of optimality can become unacceptable.

This approach is well studied for deterministic linear and non linear systems Wonham [3], Isidori [4] and in the dynamic programming litterature Larson [5], Claude [6], Levine [7].

III. A GENERATOR OF PROGRAM FOR COMPUTING THE OPTIMAL LOCAL FEEDBACK.

It is difficult to write an efficient and general program to solve problem described in 2.2. Indeed each subsystem can have a special structure, special boundary conditions. Moreover each subsystem can have different space dimensions.

We have written in MACSYMA a program able to generate automatically a large class of such problems. Where MACSYMA is a language developped at MIT for formal calculus purpose.

The class is precisely described by the following grammar where we use a kind of Backus-Naur notation ("|" for the or, and "<Word>for a non terminal word)..

<local-feedback>::=<criterium>,
Σ(<initial-condition>,<subsystems>)
type

<criterium>::=ψ,d

ψ is the coupling function function from R to R
d(t) is a demand function of time

<subsystems>::=m,<subsystem-type>

$m \in \mathbb{N}$ denotes the number of subsystems having its structure described by <subsystem-type>

<subsystem-type>::=<domain>,<inside -condition>,
<boundary-condition>

<domain>::= $[0,1]^n \times [0,T]$
$n \in \mathbb{N}$

<boundary-condition>::= Σ <boundary-condition>,<boundary-element>
<boundary-element>

<boundary-element>::=$<X_i>=0$ | $<X_i>=1$ | t=T

$<X_i>::=X_1|X_2|\dots|X_n$
$<y>::=(X_1,X_2,\dots X_n,t)$

<boundary-condition>::=$V=f\,|\,\frac{\partial V}{\partial n}=f$

<inside-condition>::=<dynamic>,<local-cost>

<dynamic>::= $\frac{\partial V}{\partial t} + \sum_{i=1}^{n} b_i(\text{<y>}) \frac{\partial}{\partial X_i} V + a_i(\text{<y>}) \frac{\partial^2 V}{\partial X_j^2}$ |

$\frac{\partial V}{\partial t} + \sum_{i=1}^{n} b_i(\text{<y>},\text{<u>}) \frac{\partial}{\partial X_i} V + \sum_{i=1}^{n} a_i(\text{<y>},\text{<u>}) \frac{\partial^2 V}{\partial X_i^2}$,

<constraints>

<u>::=$(u_1,u_2,\dots,u_p)$

<constraints>::= $[\alpha,\beta]^p$

$\alpha \in \mathbb{R}$
$\beta \in \mathbb{R}$
$p \in \mathbb{N}$ the dimension of the control

<local-cost>::= $\phi(\text{<y>})$ | $\phi(\text{<y>},\text{<u>})$

<initial-condition>::= p(<y>)

$p : \mathbb{R}^n \to R$ such that $\int_{\text{<domain>}} p(dx)dx = 1$

Moreover we have to specify to the generator the method of discretization in time: explicit or implicit, in space, the method of optimization : newton, gradient, gradient with projection and so on.

With these informations the generator is able to write a Fortran program solving the problem. An example is given in the following chapter.

In the future we shall extend the class of systems that the generator is able to solve by generalizing :

- the structure of ψ,

- extending the method to ergodic and static problem,

- generalizing the structure of the control space,

- improving the numerical method of integration

For the classical HJB equation a more general generator exists and is described in Gomez-Quadrat-Sulem [10].

IV. AN EXAMPLE

Let us consider the following stochastic control problem which models a water storage management for electricity generation.

The dynamics of the water stocks are the following :

$$dX_t^i = (a_t^i - u_t^i)dt + \sigma_i\, dw_t^i - d\xi_1^i + d\xi_2^i$$

$$0 \leq u_t^i \leq \sqrt{X_t^i}$$

where

- t denotes the time,

- i denotes index of a dam $i \in \{1,2,3\}$,

- X_t^i denotes the amount of water in the stock,

- a_t^i denotes the average imput of water,

- $\sigma_i dw_t^i$ the stochastic perturbation on the imput of water,

- ξ_1^i is an increasing process strictly increasing only when $X_t^i = 1$ denoting the cumulated overflowing water,

- ξ_2^i is an increasing process strictly increasing only when $X_t^i = 0$, denoting the cumulate water that we have to add to the imput in such way that X_t^i be always positive. It can be seen as a model correction indeed $a^i dt + \sigma_i dw_t^i$ is not almost surely positive but $a_t^i dt + \sigma_i dw_t^i + d\xi_2^i$ will be always positive when $X_t^i = 0$. By this way X_t^i is always positive.

The influence of ξ_2^i is small because $a_t^i > 0$ and $u_t^i = 0$ if $X_t^i = 0$,

The criterion is

$$E \int_0^T \psi(z_t - \sum_{i=1}^{3} u_t^i)$$

where

- z_t denotes the demand in electricity

- the function $\psi : x \to x^2$ denotes the generation cost of thermal means. Indeed $z_t - \sum_{i=1}^{z} u_t^i$ can be seen as the thermal electricity generation to be produced.

The following annex and figures show :

- the macsyma program which specifies the problem and calls the generator of fortran program

- the program generated,

- the main program calling the subroutine generated,

- the optimal price of water obtained by the local feedback method,

- the optimal price obtained by solving the complete HJB equation.

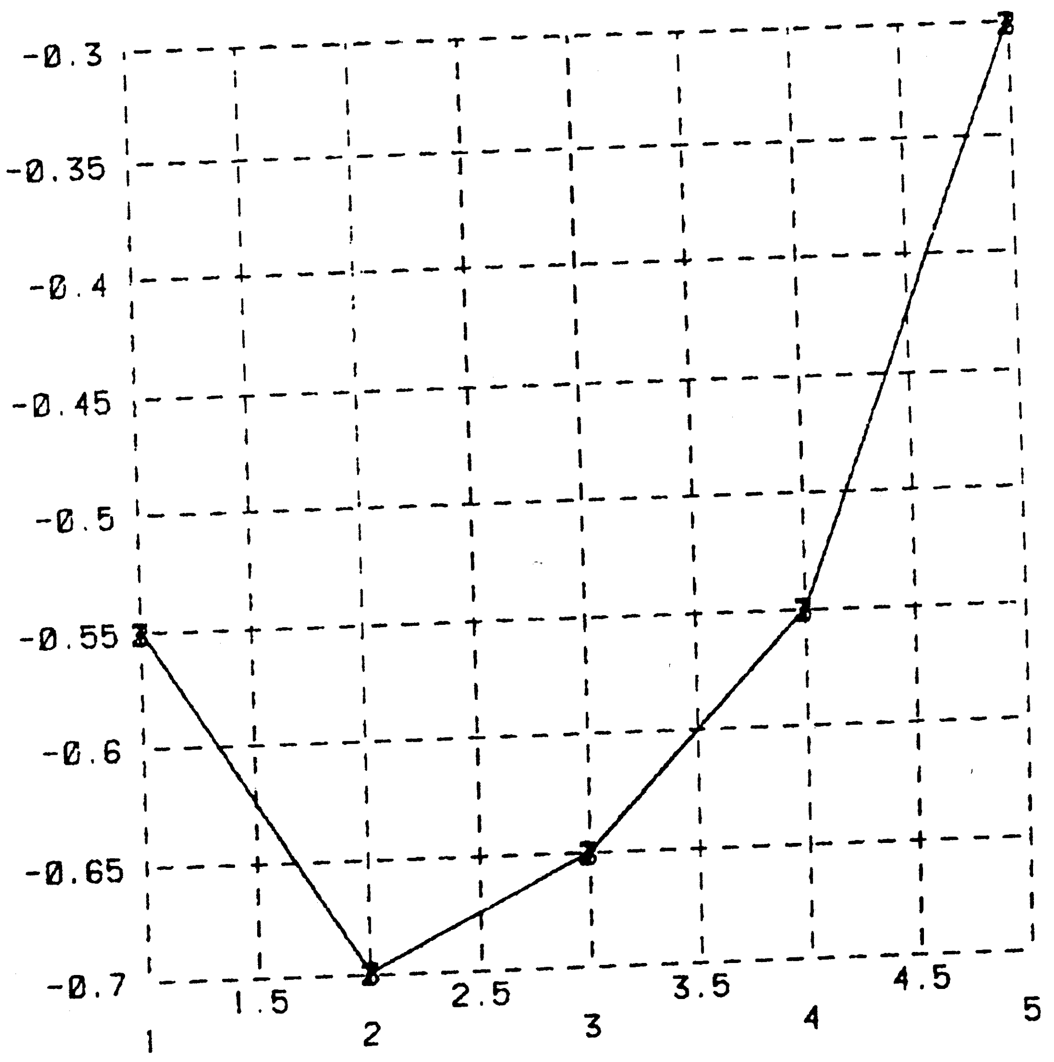


Figure 1

Minus the price of water $\frac{\partial v_1}{\partial x_1}$ as a function of the water level for one dam.

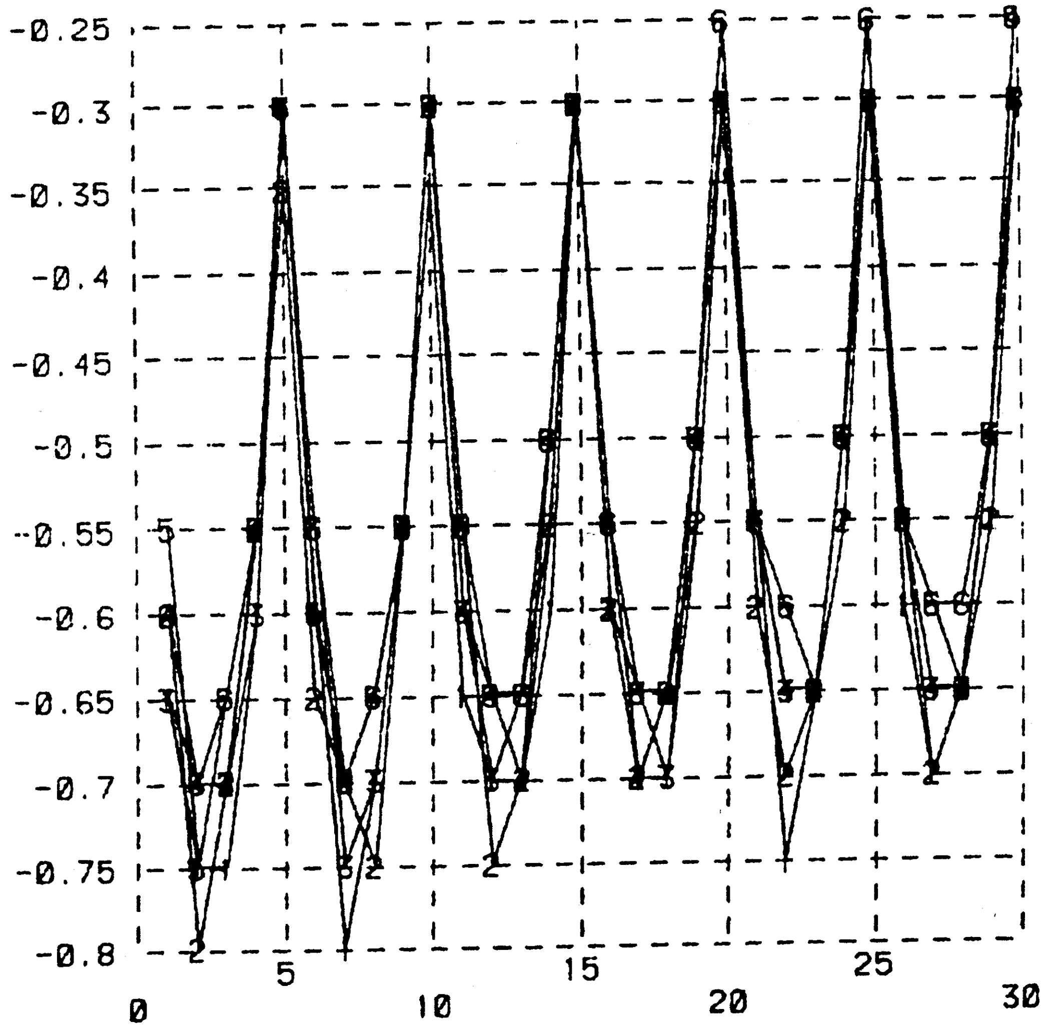


Figure 2

Minus the price of water $(\frac{\partial v}{\partial x})$ as a function of the three dimensional space obtained by solving the complete HJB equation. $X_2 = 0.1$ is represented by the abscisse 1 to 5 the section $X_2 = 0,3$ by the abscisse 5 to 10 etc...

ANNEX 1

Subroutine in macsyma specifying the control problem by the list "syst" and calling the generator of fortran program here "feedloc". In the future we shall use a semi-natural language interface to specify the problem.

```
appel():=(
     cline("dl belman.fortran"),
     cdl0:((nat,0)),
     cdl1:((nat,0)),
     ap:(1.0+cos(zf+44.0*x0/7.0))/2.0,
     demm:5.0+3.0*cos(44.0*x0/7.0)/2.0,
     hm:(ap-u1)*p1-u1,
     type:(1,parab,exp,0.0,pasecriture,pasmoy,condlim,cdl0,cdl1,parae,difu,
         derive,(0),plus,dif,(za),belm,1,newto,gradproj,hm,((0.0,zu*x^(1/2))),
          param,(zu,za,zf)),
     psi(x):=x^2,
     syst:(psi,demm,(1.0),(3,type)),
     feedloc(syst) )$
```

ANNEX 2

Subroutines fortran, automatically generated, solving the problem

```
      subroutine primal1(n1,n0,h0,v,u,eps,nmax,ymoen,variance,zu,za,zf,r
     1   og)
       dimension v(n1,n0),u(1,n1,n0),ymoen(n0),variance(n0)
c
c       Resolution de 1 equation de Bellman dans le cas ou:
c             Les parametres sont zu za zf
c             L etats-temps est:  x1 x0
c             La dynamique du systeme est decrite par 1 operateur
c                           p1 cos(zf + 6.2857142 x0)
c       plus( q1 za , Minu( ------------------------- - p1 u1
c                                       2
c
c                                      2                                 2
c   (- u1 + ymoen(i0) + 2 variance(i0))    (- u1 + ymoen(i0) + variance(i0))
c+ ------------------------------------ + ----------------------------------
c                  16                                     4
c
c                                    2                                   2
c   (- u1 + ymoen(i0) - variance(i0))    (- u1 + ymoen(i0) - 2 variance(i0))
c+ ---------------------------------- + ------------------------------------
c                 4                                      16
c
c                   2
c   3 (ymoen(i0) - u1)    p1
c+ ------------------- + -- ) )
c           8             2
c             ou v(..) et w designe  le cout optimal
c             ou pi designe sa derivee premiere par rapport a xi
c             ou qi designe sa derivee seconde par rapport a xi
c             Le probleme est parabolique
c             Le temps note x0 appartient a (0,(n0-1)*h0)
c             le cout sur l'etat final 0.0
c             Les conditions aux limites sont:
c                  x1 = 0    -p1 = 0
c                  x1 = 1    p1 = 0
c       Les nombres de points de discretisation sont: n1 n0
c                  x1 = 1 correspond a i1 = n1 - 1
c                  x1 = 0 correspond a i1 = 2
```

```
c      Le schema de discretisation en temps est explicite
c      p1 est discretise par difference divise  symetrique
c      Minimisation par la methode de  gradient avec projection
c
c                                                                          de l'Hamiltonien:
c            p1 cos(zf + 6.2857142 x0)
c            ------------------------- - p1 u1
c                        2
c
c                                        2                                   2
c  (- u1 + ymoen(i0) + 2 variance(i0))    (- u1 + ymoen(i0) + variance(i0))
c+ ------------------------------------ + ---------------------------------
c                   16                                  4
c
c                                      2                                     2
c  (- u1 + ymoen(i0) - variance(i0))    (- u1 + ymoen(i0) - 2 variance(i0))
c+ ---------------------------------- + ------------------------------------
c                  4                                   16
c
c                    2
c  3 (ymoen(i0) - u1)    p1
c+ ------------------ + --
c          8             2
c      contraintes sur le controle:
c            0.0 =< u1 =< sqrt(x1) zu
c      nmax designe le nombre maxi d iteration de la methode de
c                                                                    gradient avec projection
c      eps designe l erreur de convergence de la methode de
c                                                                    gradient avec projection
c
      h1 = 0.999999/(n1-3)
      u1 = u(1,1,1)
      hih1 = h1**2
      h21 = 2*h1
      nm0 = n0-1
      nm1 = n1-1
      do  111   i1 = 1 , n1 , 1
      x1 = h1*(i1-2)
      v(i1,n0) = 0.0
  111 continue
      do  100   ii0 = 1 , nm0 , 1
      i0 = n0-ii0
      x0 = h0*(i0-1)
      v(n1,i0+1) = v(n1-1,i0+1)
      v(1,i0+1) = v(2,i0+1)
  110 continue
      do  109   i1 = 2 , nm1 , 1
      x1 = h1*(i1-2)
      q1 = (v(i1+1,i0+1)-2*v(i1,i0+1)+v(i1-1,i0+1))/hih1
      p1 = (v(i1+1,i0+1)-v(i1-1,i0+1))/h21
      niter = 0
      w0 = -1.0e+20
  101 continue
      niter = niter+1
      if ( niter - nmax )  102 , 102 , 103
  103 continue
      write(8,901)i1,i0
  901 format(' descente n a pas converge', 2 i3)
      goto  104
  102 continue
      un1 = (1-2*rog)*u1+(p1+2*ymoen(i0))*rog
      u1 = un1
      u1 = amax1(u1,0.0)
      u1 = amin1(u1,sqrt(x1)*zu)
      ww = p1*cos(zf+6.2857142*x0)/2.0-p1*u1+(-u1+ymoen(i0)+2*variance(i
     1  0))**2/16.0+(-u1+ymoen(i0)+variance(i0))**2/4.0+(-u1+ymoen(i0)-
     2  variance(i0))**2/4.0+(-u1+ymoen(i0)-2*variance(i0))**2/16.0+3.0
     3  *(ymoen(i0)-u1)**2/8.0+p1/2.0
      er = abs(ww-w0)
      if ( er - eps )  104 , 104 , 105
  105 continue
      w0 = ww
      goto  101
```

```
104 continue
    u(1,i1,i0) = u1
    w0 = ww
    w1 = q1*za
    w0 = w1+w0
    vnew = h0*w0+v(i1,i0+1)
    v(i1,i0) = vnew
109 continue
100 continue
    return
    end
```

```
      subroutine dual1(n1,n0,h0,v,variance,ymoen,u,zu,za,zf)
      dimension variance(n0),ymoen(n0),v(n1,n0),u(1,n1,n0)
c
c      Resolution de l equation de Fokker_Planck dans le cas ou:
c            Les parametres sont zu za zf
c            L etats-temps est:  x1 x0
c            La dynamique du systeme est decrite par l operateur
c        2
c       d             d       cos(zf + 6.2857142 x0)        1
c     ---- (v za) - --- (v (---------------------- - u1 + -))
c        2          dx1                 2                   2
c     dx1
c            ou v(..) et w designe  la densite de probabilite
c            Le probleme est parabolique
c            Le temps note x0 appartient a (0,(n0-1)*h0)
c            la condition initiale 1.0
c            variance designe la variance de - u1
c            ymoen designe la moyenne de - u1
c            Les conditions aux limites sont:
c                           d              cos(zf + 6.2857142 x0)        1
c                 x1 = 0   --- (v za) - v (---------------------- - u1 + -) = 0
c                          dx1                        2                   2
c
c
c                            cos(zf + 6.2857142 x0)        1     d
c                 x1 = 1   v (---------------------- - u1 + -) - --- (v za) = 0
c                                        2                   2    dx1
c
c
c      Les nombres de points de discretisation sont: n1 n0
c                 x1 = 1 correspond a i1 = n1 - 1
c                 x1 = 0 correspond a i1 = 2
c      Le schema de discretisation en temps est explicite
c
      h1 = 0.999999/(n1-3)
      hih1 = h1**2
      nm0 = n0-1
      nm1 = n1-1
      x0 = 0
      do  106   i1 = 1 , n1 , 1
      x1 = h1*(i1-2)
      v(i1,1) = 1.0
  106 continue
      do  100   i0 = 2 , n0 , 1
      x0 = h0*(i0-1)
      v(n1,i0-1) = v(n1-1,i0-1)*(h1*(cos(zf+6.2857142*x0)/2.0-u(1,n1-1,i
     1   0-1)+1.0/2.0)/2.0+za)
      v(1,i0-1) = v(2,i0-1)*(za-h1*(cos(zf+6.2857142*x0)/2.0-u(1,2,i0-1)
     1   +1.0/2.0)/2.0)
  103 continue
      do  102   i1 = 2 , nm1 , 1
      x1 = h1*(i1-2)
      p1 = v(i1+1,i0-1)*(za/hih1-(cos(zf+6.2857142*x0)/2.0-u(1,i1+1,i0-1
     1   )+1.0/2.0)/h1/2.0)+v(i1-1,i0-1)*((cos(zf+6.2857142*x0)/2.0-u(1,
     2   i1-1,i0-1)+1.0/2.0)/h1/2.0+za/hih1)-2*v(i1,i0-1)*za/hih1
        if (i1.eq.nm1) p1 = p1-v(n1,i0-1)*(-0.5*(0.5*cos(zf+6.2857142*x0
     1   )-u(1,n1,i0-1)+0.5)/h1+za/hih1-1/hih1)
        if (i1.eq.2) p1 = v(1,i0-1)*(-0.5*(0.5*cos(zf+6.2857142*x0)-u(1,
     1   1,i0-1)+0.5)/h1-za/hih1+1/hih1)+p1
      w0 = p1
      vnew = h0*w0+v(i1,i0-1)
      v(i1,i0) = vnew
  102 continue
      ymo1 = 0.0
      ymo2 = 0.0
      do  104   i1 = 2 , n1 - 1 , 1
      x1 = h1*(i1-2)
      ymo1 = ymo1-u(1,i1,i0-1)*v(i1,i0-1)/(n1-2)
      ymo2 = ymo2+u(1,i1,i0-1)**2*v(i1,i0-1)/(n1-2)
  104 continue
      ymoen(i0-1) = ymo1
      variance(i0-1) = ymo2-ymo1**2
  100 continue
      return
      end
      subroutine fedloc(n11,n0,ymoen,variance,dem,vdem,h0,dem1,vdem1,u1,
     1   vv1,v1,pr1,nflmax,epsilon,epsimp,impmax,eps,nmax,ro1,rog1)
      common/parametre/zu(3),za(3),zf(3)
      dimension ymoen(n0),variance(n0),dem(n0),vdem(n0),dem1(3,n0),vdem1
     1   (3,n0),u1(1,n11,n0),vv1(3,n11,n0),v1(n11,n0),pr1(n11,n0)
c
c         Optimisation dans la classe des feedbacks locaux d'un
c       systeme compose de sous-systemes a dynamiques decouplees
c       mais couples par le critere
c         Il y a 1 types de sous-systemes
c           - 3 sous systeme de type 1
c         Les sous systemes sont decrits precisement dans les
c       commentaires des sous-programmes primaux et duaux corres
c       pondants
c                                   2
c         Le critere s'ecrit: (p + d)
c         d designe la demande : 1.5 cos(6.2857142 x0) + 5.0
c         p la production somme des productions locales pi
c         pi designe la production d'un sous-ysteme de type i
c           p1 = - u1
c         Le critere est evalue pour la condition initiale:
c           v1 = 1.0
c       ou vi designe la densite de probabilite initiale des
c       sous-systeme de type i
c         La methode de resolution est une methode de relaxation
c         Les parametres d'appel sont:
c           -epsilon l'erreur de convergence de la relaxation entre
c       sous systeme
c           -nflmax le nbre maxi d'iterations correspondantes
c           -epsimp l'erreur de cvgce pour les systemes implicites
c           -impmax le nbre d'iteration maxi correspondante
c           -rogi controle la conver. de la meth. de desc. du syst i
c           -roi controle la convergence du syst implicite i
c           -eps l'erreur de convergence dans la methode de Newton
c           -nmax le nbre maxi d'iterations correspondantes
c         Le temps note x0 appartient a (0,(n0-1)*h0)
c         nij designe le nbre de pts de discretisation de la
c       composante i d'un sous systeme de type j
```

```
c          Les sorties sont:
c            -vvi(j,...) designe le cout vu par le j-eme sous-systeme
c        de type i
c            -demi(j,..) la production moyenne correspondante
c            -vdemi(j,..) la variance de la production correspondante
c          Les autres parametres ne servent que pour avoir des
c        dimensions variables dans le sous programmes
c          Les parametres  (zu, za, zf) doivent etre passes dans le
c        common parametre
c
      do  100   i0 = 1 , n0 , 1
      x0 = h0*(i0-1)
      dem(i0) = 1.5*cos(6.2857142*x0)+5.0
  100 continue
      do  101   j = 1 , 3 , 1
      call dual1(n11,n0,h0,pr1,variance,ymoen,u1,zu(j),za(j),zf(j))
      do  102   i0 = 1 , n0 , 1
      dem1(j,i0) = ymoen(i0)
      dem(i0) = ymoen(i0)+dem(i0)
      vdem1(j,i0) = variance(i0)
      vdem(i0) = vdem(i0)+variance(i0)
  102 continue
  101 continue
      coutv = 10000000000
      nitfl = 0
      write(8,901)
  901 format('  converg:(')
  113 continue
      nitfl = nitfl+1
      do  103   j = 1 , 3 , 1
      do  104   i0 = 1 , n0 , 1
      dem(i0) = dem(i0)-dem1(j,i0)
      vdem(i0) = vdem(i0)-vdem1(j,i0)
  104 continue
      call primal1(n11,n0,h0,v1,u1,eps,nmax,dem,vdem,zu(j),za(j),zf(j),r
     1   og1)
      do  105   i1 = 1 , n11 , 1
      do  105   i0 = 1 , n0 , 1
      vv1(j,i1,i0) = v1(i1,i0)
  105 continue
```

```
      coutneuf = 0
      do  106   i1 = 2 , n11 - 1 , 1
      x1 = 0.999999*(i1-2)/(n11-3)
      coutneuf = v1(i1,1)/(n11-2)+coutneuf
  106 continue
      write(8,902)coutneuf
  902 format('  ',e14.7,',')
      call dual1(n11,n0,h0,pr1,variance,ymoen,u1,zu(j),za(j),zf(j))
      do  107   i0 = 1 , n0 , 1
      dem1(j,i0) = ymoen(i0)
      dem(i0) = ymoen(i0)+dem(i0)
      vdem1(j,i0) = variance(i0)
      vdem(i0) = vdem(i0)+variance(i0)
  107 continue
  103 continue
      if ( nflmax - nitfl )  110 , 109 , 109
  110 continue
      write(8,900)
  900 format(' feedloc n a pas converge')
      goto  112
  109 continue
      if ( - epsilon + coutv - coutneuf )  112 , 112 , 111
  111 continue
      coutv = coutneuf
      goto  113
  112 continue
      write(8,903)
  903 format(' ())$')
      return
      end
```

ANNEX 3

Main program to write by hand to call the subroutine feedloc which solves the problem

```
      dimension dem1(3,61),vdem1(3,61),vv1(3,13,61)
      dimension ymoen(61),variance(61),dem(61),vdem(61),v1(13,61),
     1 u1(1,13,61),pr1(13,61)
      common /parametre/zu(3),za(3),zf(3)
      do 100 i=1,61
      do 100 j=1,3
      u1(1,j,i)=1.0
100   continue
      do 101 j=1,3
      za(j)=0.18
      zf(j)=1.57
      zu(j)=3.0
101   continue
      call fedloc(13,61,ymoen,variance,dem,vdem,0.009,dem1,vdem1,u1,
     1 vv1,v1,pr1,10,.01,0.01,100,0.01,20,0.01,0.5)
      write (9,200)
200   format (" v:(")
      do 202 jj=1,6
      j=1+10*(jj-1)
      write (9,201)((vv1(k,i,j),i=1,13),k=1,3)
201   format(" (",38(f4.2,","),f4.2,"),")
202   continue
      write (9,203)
203   format(" ())$")
      stop
      end
```

REFERENCES.

[1] QUADRAT - VIOT : Product form and optimal local feedback for a multiindex Markov Chain, 18th Allerton Conference, October 1980.

[2] DELEBECQUE - QUADRAT : Contribution of stochastic control singular perturbation team theories to an example of large scale system : management of hydropower production,IEEE AC, April 1978, pp. 209-222.

[3] WONHAM : Linear system : geometric approach, Springer Verlag, 1974.

[4] ISIDORI : The geometric approach to non linear feedback control : a survey, 5th Conference on "Analyse et Optimisation des Systèmes", Versailles, 1982, Lecture Notes in Control and Information Sciences n°44, Springer Verlag.

[5] LARSON - KORSAK : A dynamic programming successive approximations : technique with convergence proofs Part I & II, Automatica, 1969.

[6] CLAUDE : Linéarisation par difféomorphisme et immersion des systèmes, 6th Conference "Analyse et Optimisation des Systèmes", Nice, June 1984, Springer Verlag, Lecture Notes in Control and Information Sciences

[7] GEROMEL - LEVINE - WILLIS : A fast algorithm for systems decoupling using formal calculus, 6th Conference "Analyse et Optimisation des Systèmes", Nice, Juin 1984, Springer Verlag, Lect. Notes in Control and Information Sciences.

[8] J.L. LIONS : Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles, Paris, Dunod 1968.

[9] Mit Mathlab Group : MACSYMA, Manual, Mit Press.

[10] GOMEZ - QUADRAT - SULEM : Vers un système expert en contrôle stochastique, 6th Conference "Analyse et Optimisation des Systèmes", Nice, Juin 1984, Springer Verlag, Lecture Notes in Control and Information Sciences.

[11] THEOSYS, Commande Optimale de systèmes stochastiques, RAIRO Automatique, à paraître.