An anti-diffusive scheme for viability problems

Olivier Bokanowski, Sophie Martin, Remi Munos, Hasnaa Zidani

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An anti-diffusive scheme for viability problems

Olivier BOKANOWSKI — Sophie MARTIN — Remi MUNOS — Hasnaa ZIDANI

N° 5431

Décembre 2004

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5431--FR+ENG

An anti-diffusive scheme for viability problems

Olivier BOKANOWSKI* , Sophie MARTIN† , Remi MUNOS‡ , Hasnaa ZIDANI§

Thème NUM — Systèmes numériques
Projet SYDOCO

Rapport de recherche n° 5431 — Décembre 2004 — 20 pages

Abstract: This paper is concerned with numerical approximation of viability kernels. We use a characterization of the viability kernel by the value function of an optimal control problem. Since this value function is discontinuous, usual discretization schemes (such as finite differences) provide poor approximation quality because of numerical diffusion.

We investigate the use of the *ultra-bee scheme* for its anti-diffusive property in the transport of discontinuous functions. Numerical experiments, compared with the viability algorithm [8], show the relevance of this scheme for computing viability kernels and capture basins on several benchmark problems.

Key-words: Viability kernel, Capture basin, Ultrabee scheme

* Lab. Jacques-Louis Lions, Université Pierre et Marie Curie, 175 Rue Chevaleret 75013 Paris, France. Email: boka@math.jussieu.fr

† Lab. d'ingénierie des systèmes complexes, Cemagref, 24 av. des Landais, BP 50085, 63172 Aubière Cedex, France. Email: sophie.martin@clermont.cemagref.fr

‡ Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France

§ ENSTA, UMA, 32 Boulevard Victor, 75739 Paris Cedex 15, France. Email: Hasnaa.Zidani@ensta.fr. Also at Projet Sydoco, INRIA Rocquencourt, BP. 105, 78153 Le Chesnay Cedex.

Un schéma anti-diffusif pour les problèmes de viabilité

Résumé : Dans ce travail, nous nous interessons au calcul numérique des noyaux de viabilité. Nous utilisons une caractérisation du noyau de viabilité à l'aide de la fonction valeur d'un problème de contrôle optimal avec contraintes sur l'état. Cette fonction valeur étant discontinue, son approximation par les schémas de discrétisation classiques n'est pas satisfaisante à cause des diffusions numériques.

Ici, nous utilisons le schéma *ultra-bee* connu pour ses propriétés anti-diffusives pour l'approximation des équations de transport avec données discontinues. Nous comparons ce schéma avec l'algorithme de viabilité [8] sur plusieurs exemples.

Mots-clés : Noyau de viabilité, Bassin de capture, schéma ultrabee

1 Introduction

We consider a control system, defined by the dynamics

$$\dot{y} = f(y, u), \quad u \in U(y) \tag{1a}$$
$$y(0) = x_0 \tag{1b}$$

where $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $U(y)$ is a compact subset in $\mathbb{R}^m$. We write F the set-valued map $F(x) = \{f(x, u), u \in U(x)\}$.

Following Aubin [2], we say that a trajectory $y(t)$ is *viable* in a *constrained set* $K \in \mathbb{R}^n$ if it remains in K forever:

$$\exists u(\cdot), \forall t \geq 0, y(t) \in K \tag{2}$$

The *Viability Kernel* of K under F, denoted by $\text{Viab}_F(K)$, is defined as the set of points x_0 from which can start a viable solution, i.e.,

$$\text{Viab}_F(K) := \{x_0 \in K,\ \exists u(\cdot), \forall t \geq 0, y(t) \in K\} \tag{3}$$

The viability kernel may be characterized in diverse ways through tangential conditions thanks to the viability theorems (under usual assumption, e.g. that F is a Marchaud Map or F is Lipschitz [2]). We recall that F is said to be Marchaud if it is upper semi-continuous, convex compact valued, and if $\exists c \geq 0$, $\forall x, u$, $||f(x, u)|| \leq c(||x|| + 1)$.

The viability kernel algorithm, proposed by Saint-Pierre [8] computes for a given grid G_h, a discrete viability kernel (of the discretized $K \cap G_h$ under a time-discretized and augmented dynamics) that converges to the viability kernel $\text{Viab}_F(K)$ when the grid resolution h tends to 0.

We first characterize the viability kernel by the infinite time limit of the value function of an evolutionary control problem.

This value function being discontinuous, usual discretization schemes such as those based on interpolation techniques (as Semi-Lagrangian, finite differences) fail to provide accurate approximations because of numerical diffusion.

Here, we propose to use the anti-diffusive *Ultra-bee scheme* extended to the resolution of Hamilton-Jacobi-Bellman equations [3], which we believe is particularly relevant to the specific shape of the value functions derived from viability problems.

So far, no convergence proof for this scheme is available. However, the numerical experiments tested on several benchmark problems and compared to the viability algorithm are very encouraging, in terms of the approximation error.

Extension to the computation of the *capture basin* of a target $C \subset K$ (defined as the set of initial states $x_0 \in K$ such that C is reached in finite time before possibly leaving K by at least one trajectory $y(\cdot)$) is also treated and illustrated.

The paper is organized as follows. In section 2, we define value functions related to the above problems, and give Hamilton-Jacobi-Bellman (HJB) equations satisfied by these value functions. In section 3, we recall the Ultra-bee scheme adapted to treat HJB equations, and also define an Ultra-bee scheme for computing a capture basin. In section 4 we compare on various examples the numerical results given by the Ultra-Bee scheme and by the viability algorithm (we also thank deeply P. Saint-Pierre for allowing us to use his code).

For sake of completness we also recall the viability algorithm in an appendix.

2 Statement of the problem and basic results

Let K be a compact subset of $\mathbb{R}^n$ and U a compact subset of $\mathbb{R}^p$. Let $f : \mathbb{R}^n \times U \longrightarrow \mathbb{R}^n$ be a bounded continuous map. In the sequel, we suppose that f is locally lipshitz continuous with respect to x uniformly in u.

2.1 Viability kernel

For any measurable control, we denote $y_x(t)$ the trajectory satisfying

$$\dot{y}_x = f(y_x, u) \quad u \in U(y) \tag{4a}$$

$$y_x(0) = x \tag{4b}$$

In view of the constraint given by (2), we set the following optimal control problem

$$(\mathcal{P}_x) \qquad \min\{0,\ \exists u(\cdot),\ y_x(t) \in K \text{ for all } t > 0,\}$$

We define the value function associated to this problem by

$$\mathcal{V}(x) := \mathrm{Inf}(\mathcal{P}_x)$$

where the value of $\mathcal{V}(x)$ is supposed to be $+\infty$ if the set of constraints is empty. Of course we thus have that $\mathcal{V}(x) = +\infty$ if $x \notin K$. Also, we have

$$\mathcal{V}(x) = \begin{cases} 0 & \text{if } \exists u(\cdot), \forall t > 0,\, y_x(t) \in K \\ +\infty & \text{otherwise} \end{cases}$$

Then the Viability Kernel is given by

$$\mathrm{Viab}(K) = \{x,\ \mathcal{V}(x) = 0\} = \{x,\ \exists u(\cdot),\ y_x(t) \in K\ \forall t \geq 0\}.$$

For each $T > 0$, we introduce now the following optimal control problem:

$$(\mathcal{P}_{\tau,x}) \qquad \min\{0,\ \ y_{\tau,x}(t) \in K \text{ for all } t \in [\tau, T]\}$$

where $y_{\tau,x}$ is a solution of

$$\dot{y}_{\tau,x} = f(y_{\tau,x}, u) \quad u \in U(y) \tag{5a}$$

$$y_{\tau,x}(\tau) = x \tag{5b}$$

and we define also the value function associated to this problem by

$$V(T-\tau, x) = \min(\mathcal{P}_{\tau,x}).$$

We see that

$$V(T,x) = \begin{cases} 0 & \text{if } \exists u(\cdot),\ \forall t \in [0,T],\ y_{\tau,x}(t) \in K \\ +\infty & \text{otherwise} \end{cases}$$

Lemma 2.1. *For every $x \in K$, $V(T,x)$ converges towards $\mathcal{V}(x)$ as $T \to +\infty$.*

Proof. We first remark that $T \to V(T,x)$ is non-decreasing. Indeed, suppose $T < T'$. If $V(T,x) = \infty$, for any control $u(\cdot)$, there exists $t \leq T$ such that $y_x(t) \notin K$. Since also $t \leq T'$, we obtain $V(T',x) = \infty$. If $V(T,x) = 0$, there is nothing to prove. Hence we have $V(T,x) \leq V(T',x)$ in all cases.

Now if $\mathcal{V}(x) = \infty$, there exists $T_1 > 0$ such that $y_x(T_1) \notin K$. Hence, $V(T_1,x) = \infty$, and this implies that $V(T,x) \overset{T\to\infty}{\to} \infty$ using the non-decreasing property. If otherwise $\mathcal{V}(x) = 0$, then $\exists u(\cdot)$, $\forall t \geq 0$, $y_x(t) \in K$, and $V(T,x) = 0$ for all $T > 0$. Hence also $V(T,x) \overset{T\to\infty}{\to} 0$. $\square$

We then have the following (which proof is left to the reader):

Lemma 2.2. *Let $\Omega_T := \{x,\ V(T,x) = 0\}$. Then we have*
(i) $\Omega_{T'} \subset \Omega_T \subset K$, for every $T' \geq T \geq 0$,
(ii) $\cap_{T\to+\infty}\Omega_T = \mathrm{Viab}_K$.

Now we propose to compute Ω_T for T large enough, and to approximate Viab_K by Ω_T. We thus look for an approximation for $V(\cdot,T)$ in K. We know that the function V satisfies an HJB equation of the following form:

$$V_t - \min_{u\in U}\left(f(x,u)\cdot\nabla_x V\right) = 0, \quad t > 0,\ x \in K; \tag{6a}$$

$$V(0,x) = 0, \quad x \in K; \tag{6b}$$

$$V(t,x) = +\infty, \quad t > 0,\ x \notin K. \tag{6c}$$

If one of the two following assumptions are satisfied

(i) $\exists \alpha > 0$, $\forall x \in \partial K$, $\exists u \in U$, $\eta_x \cdot f(x,u) < -\alpha$

(ii) For all $x \in K$, $A(x) := \{u,\ y_x \in K\} \neq \emptyset$

then V is a continuous viscosity solution of (6). Here, however, the assumptions (i) or (ii) are not necessarily satisfied, since it would imply that $\mathrm{Viab}_F(K) = K$. Hence in general the function V is not continuous. It is shown to be a solution of (6) in a particular sense given by Frankowska in [5]. In this paper, we propose to compute the function V by using the so called "Ultra-bee" scheme [4, 3] for the discretisation of (6).

The discretisation of equation (6) will be studied in section 3.

2.2 Capture basin

Let the target C be a subset of $\mathbb{R}^n$. The subset of initial states $x \in K$ such that C is reached in finite time before possibly leaving K by at least one trajectory $y_x(\cdot)$ is called the **capture basin** of C in K and denoted $\mathrm{Capt}_F(C)$.

Let us introduce the set-valued map F_C which coincides with F outside C, equals to 0 inside C and equals to the convex hull of $\{0\} \cup F(x)$ on ∂C. If K is a repeller for F (i.e. $\mathrm{Viab}_F(K) = \emptyset$) then $\mathrm{Capt}_F(C) = \mathrm{Viab}_{F_C}(K)$. Otherwise we have in general $\mathrm{Viab}_{F_C}(K) = \mathrm{Capt}_F(C) \cup \mathrm{Viab}_F(K)$.

For our purpose, let $T > 0$. Let χ_C be the caracteristic function of C, i.e. defined by $\chi_C(x) := 0$ if $x \in C$ and $\chi_C(x) = +\infty$ otherwise. Let $\vartheta_T(t,x)$ and $\hat{\vartheta}_T(t,x)$ be defined by

$$\vartheta_T(t,x) := \min_{\tau \in [t,T]} \{\chi_C(y_{t,x}(\tau)),\ \dot{y}_{t,x}(s) \in F(y_{t,x}(s)) \text{ and } y_{t,x}(s) \in K \text{ for } s \in [t,\tau]\},$$

and

$$\hat{\vartheta}_T(t,x) := \min\{\chi_C(y_{t,x}(T));\ \dot{y}_{t,x}(s) \in F_C(y_{t,x}(s)), \text{ and } y_{t,x}(s) \in K \text{ for all } s \in [t,T]\}.$$

Now we define the **capture basin before time** T by

$$\mathrm{Capt}_F(C;T) := \{x \in K, \quad \vartheta_T(0,x) = 0\}.$$

In particular, $T \to \mathrm{Capt}_F(C;T)$ is increasing for the inclusion, and also we find the usual capture basin as $T \to \infty$:

$$\lim_{T \to \infty} \mathrm{Capt}_F(C;T) = \mathrm{Capt}_F(C).$$

It is not difficult to obtain (assuming F is Marchaud, and in particular that $F(x,U)$ convex for all x), the identity of the previous value functions:

$$\vartheta_T(t,x) = \hat{\vartheta}_T(t,x).$$

Hence we have also

$$\mathrm{Capt}_F(C;T) := \{x \in K, \quad \hat{\vartheta}_T(0,x) = 0\}.$$

The function $\hat{\vartheta}$ is a value function of an optimal control problem (more precisely, a *Rendez-Vous* problem with state constraints). In particular we can state a dynamic programming principle for $\hat{\vartheta}$ (for $t + \Delta t \leq T$)

$$\hat{\vartheta}_T(t,x) = \min_{y_{t,x}} \hat{\vartheta}_T(t + \Delta t, y_{t,x}(t + \Delta t)), \tag{7a}$$

and

$$\hat{\vartheta}_T(T,x) = \chi_C(x). \tag{7b}$$

where $y_{t,x}$ are solutions of $\dot{y}_{t,x} \in F_C(y_{t,x})$ on $[t, t + \Delta t]$. Under some technical asumptions, $\hat{\vartheta}$ satisfies also an HJB equation in a generalized sense (see [5]). In order to approximate $\mathrm{Capt}_F(C;T)$, the last equations (7a)-(7b) will be discretized in the next section.

3 Ultra-bee scheme

We shall present the Ultra-bee (UB) scheme for the HJB equation (6) in space dimension 2, using three steps: we first present the UB scheme for linear advection in 1d, then in 2d, and finaly in 2d for the HJB equation. For practical purpose, the $+\infty$ value can be replaced by $+1$, and in particular the condition (6c) can be replaced by

$$V(t,x) = +1 \quad \text{if } x \notin K. \tag{8}$$

UB scheme for 1d linear advection. We consider the discretisation of

$$\begin{cases} v_t + f(x)\, v_x = 0, & t > 0,\ x \in \mathbb{R} \\ v(0,x) = v_0(x) \end{cases} \tag{9}$$

where $x \to f(x)$ is lipschitz-continous, and the initial condition v_0 is assumed in $L^1_{loc}(\mathbb{R})$. Let (x_j) such that $x_{j+1} - x_j = \Delta x$ and $t_n = n\Delta t$ be uniform space and time discretisations, where $\Delta x, \Delta t$ are the mesh sizes. Let V_j^n denotes a numerical approximation to the solution $v(t_n, x_j)$, The UB scheme for (9) takes the following form:

$$\frac{V_j^{n+1} - V_j^n}{\Delta t} + f(x_j)\frac{V^{n,L}_{j+\frac12} - V^{n,R}_{j-\frac12}}{\Delta x} = 0, \tag{10}$$

with the initialization:

$$V_j^0 := \frac{1}{\Delta x}\int_{x_{j-\frac12}}^{x_{j+\frac12}} v_0(x)dx, \tag{11}$$

where $x_{j+\frac12} = x_j + \frac{\Delta x}{2}$. Here $V^{n,L}_{j+\frac12}$ and $V^{n,R}_{j+\frac12}$ are numerical *fluxes* that will be defined below. We write (10) in the equivalent non-conservative form:

$$V_j^{n+1} = V_j^n - \nu_j\left(V^{n,L}_{j+\frac12} - V^{n,R}_{j-\frac12}\right), \tag{12}$$

where

$$\nu_j := \frac{\Delta t}{\Delta x} f(x_j)$$

is a "local CFL" number. We assume that $|\nu_j| \le 1\ \forall j$. In the case $\nu_j = 0$, we thus consider $V_j^{n+1} = V_j^n$, and the fluxes $V^{n,R/L}_{j+\frac12}$ need not to be defined.
We first set

$$\text{if } \nu_j > 0, \quad \begin{cases} b_j^+ := \max(V_j^n, V_{j-1}^n) + \frac{1}{\nu_j}(V_j^n - \max(V_j^n, V_{j-1}^n)), \\ B_j^+ := \min(V_j^n, V_{j-1}^n) + \frac{1}{\nu_j}(V_j^n - \min(V_j^n, V_{j-1}^n)). \end{cases} \tag{13}$$

$$\text{if } \nu_j < 0, \quad \begin{cases} b_j^- := \max(V_j^n, V_{j+1}^n) + \frac{1}{|\nu_j|}(V_j^n - \max(V_j^n, V_{j+1}^n)), \\ B_j^- := \min(V_j^n, V_{j+1}^n) + \frac{1}{|\nu_j|}(V_j^n - \min(V_j^n, V_{j+1}^n)), \end{cases} \tag{14}$$

Now, we define the "fluxes" $V^{n,L}_{j+\frac{1}{2}}$ and $V^{n,R}_{j+\frac{1}{2}}$ as follows (see [3])

- If $\nu_j > 0$ then define $V^{n,L}_{j+1/2} := \min(\max(V^n_{j+1}, b^+_j), B^+_j)$;
- If $\nu_j < 0$ then define $V^{n,R}_{j-1/2} := \min(\max(V^n_{j-1}, b^-_j), B^-_j)$.
- If $\nu_j \leq 0$ and $\nu_{j+1} \geq 0$, then define

$$V^{n,R}_{j+\frac{1}{2}} := V^n_{j+1} \quad \text{and} \quad V^{n,L}_{j+\frac{1}{2}} := V^n_j. \tag{15}$$

- If $\nu_j \nu_{j+1} > 0$, then define $V^{n,R}_{j+\frac{1}{2}} := V^{n,L}_{j+\frac{1}{2}}$ (if $\nu_j > 0$) or $V^{n,L}_{j+\frac{1}{2}} := V^{n,R}_{j+\frac{1}{2}}$ (if $\nu_{j+1} < 0$).

For stability and convergence properties of this scheme, we refer to [3]. Note that in the case $\nu_j > 0$ $\forall j$, we have $V^{n,R}_{j+\frac{1}{2}} = V^{n,L}_{j+\frac{1}{2}}$ and thus, denoting $V^n_{j+\frac{1}{2}} = V^{n,L}_{j+\frac{1}{2}}$, the scheme (12) takes the more simple form

$$V^{n+1}_j = V^n_j - \nu_j \left(V^n_{j+\frac{1}{2}} - V^n_{j-\frac{1}{2}} \right).$$

UB scheme for 2d linear advection. Now we consider the equation

$$v_t + f_1(x,y)v_x + f_2(x,y)v_y = 0. \tag{16}$$

$$v(0,x,y) = v_0(x,y) \tag{17}$$

We consider a cartesian mesh (x_j, y_k) with constant mesh sizes $x_{i+1} - x_i = \Delta x$ and $y_{j+1} - y_j = \Delta y$, and assume the CFL condition

$$\max_{i,j} \left(max\left(|f_1(x_i,y_j)| \frac{\Delta x}{\Delta t}, |f_2(x_i,y_j)| \frac{\Delta y}{\Delta t} \right) \right) \leq 1. \tag{18}$$

The UB scheme (12) is extended to (16) by using simply a Trotter splitting (see [6]). The initialization step is

$$V^0_{i,j} = \frac{1}{\Delta x \Delta y} \int_{I_i \times J_j} v_0(x,y)\; dx\; dy \tag{19}$$

where $I_i = [x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}]$ and $J_j = [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}]$. Then we first compute $(V^{n,1}_{i,j})$ from $(V^n_{i,j})$ by solving one time step of the UB scheme in the x direction for

$$v_t + f_1(x,y)v_x = 0$$

for each given $y = y_j$. Finaly we obtain $(V^{n+1}_{i,j})$ from $(V^{n,1}_{i,j})$ by solving one time step of the UB scheme in the y direction for

$$v_t + f_2(x,y)v_y = 0$$

for each given $x = x_i$. The CFL condition (18) is natural because here we consider a Trotter splitting. Also, for boundary conditions we choose the value $V^n_{i,j} = 1$ if $(x_i, y_j) \notin K$ as in (8).

F. Lagoutière [6] proved the very interesting property that the UB scheme advects exactly a particular class of step functions, in the case of constant advection. For instance, for 2-dimensional problems, let u_0 such that V^0_{ij} initialized as in (19) belongs to the following space $\mathcal{S}$:

$$\mathcal{S} := \{(V_{i,j}),\ \forall (a,b) \in \{0,1,2\},\ V_{3i+a,3j+b} = V_{3i,3j}\}\,.$$

Consider the UB scheme for $v_t + f \cdot \nabla v = 0$ where $f = (f_1, f_2) = \mathit{const}$ is a constant advection vector of $\mathbb{R}^2$. Then, assuming the CFL condition $\max(|f_1|\frac{\Delta t}{\Delta x}, |f_2|\frac{\Delta t}{\Delta y}) \le 1$, we have $\forall i, j$ and $n \ge 0$:

$$V^n_{i,j} = \frac{1}{\Delta x \Delta y} \int_{I_i \times J_j} v(t,x)\ dx$$

where $v(t,x) = v_0(x - ft)$ is the exact solution (see also [6] for more general functions that are exactly advected).

It is this exact tranportation property, which corresponds to an "antidissipative" behavior of the UB scheme, which motivates us for using it in front propagation problems such as (6). It is also numerically observed that if $V^0_{i,j}$ does not belong to the class S, then $V^n_{i,j}$ tends to be very close to a function space such as $\mathcal{S}$ after a few time steps. We refer to Désprès and Lagoutière[4] for other interesting properties of the UB scheme.

HJB-UB scheme. We now consider the discretisation of an HJB equation of the form:

$$v_t - \min_{u \in U(x,y)} (f(x,y,u) \cdot \nabla v) = 0, \quad t > 0,\ (x,y) \in K. \tag{20}$$

We assume the CFL condition (18). The initialization of $V^0_{i,j}$ is done as in (19). At time $t = t_n$, for a given (x_i, y_j) we consider $(u_k)_{k=1,\dots,N}$ a given discretization of the admissible set $U(x_i, y_j)$. We denote by $V^{n+1,UB}_{i,j}(u)$ the UB scheme obtained from $(V^n_{i,j})$ by using the advection $f(\cdot, u)$, i.e., one time step of the UB scheme for

$$v_t - f(x,y,u) \cdot \nabla v = 0.$$

Then the HJB scheme is given by

$$V^{n+1}_{i,j} = \min_{u_k} \left(V^{n+1,UB}_{i,j}(u_k) \right). \tag{21}$$

We refer to [3] for first applications of the "Ultra-bee" scheme to the resolution of HJB equations with discontinous initial data. The scheme seems well adapted to treat discontinuous solutions and in particular when the value function takes only two values (0 and 1). However, presently we do not have a convergence proof of the HJB-UB scheme for (20).

HJB-UB scheme for the computation of a capture basin before time T

The algorithm for computing a capture bassin $\mathrm{Capt}_F(C;T)$ before time T, for a given target C, is the following in the 2d setting. Here we assume that $x \in \mathbb{R}^2$. Our aim is to discretise, for a given $T > 0$, the function $V(t,x) := \hat{\vartheta}_T(T-t,x)$ where $\hat{\vartheta}_T$ obeys eqs (7). The boundary condition is

$$V(t,x) = 1, \quad (x) \notin K,$$

and the initial condition is

$$V(0,x) = 1_{K\backslash C}(x), \quad (x) \in K,$$

i.e. the value fonction is 0 on C only.
We first remark that any absolutely continuous solution of the differential inclusion $\dot{y} \in F_C(y)$ corresponds also to a solution of $\dot{y} = f_C(y,u,v)$ for a given $(u,v) \in L^\infty\big([0,T], U \times \{0,1\}\big)$, where

$$f_C(x,u,v) := \begin{cases} f(x,u) & \text{if } x \in K\backslash C, \text{ and for } u \in U, \\ vf(x,u) & \text{for } x \in \partial C, \ u \in U, \text{ and } v \in \{0,1\}, \\ 0 & \text{otherwise.} \end{cases}$$

In order to discretise the dynamic f_ρ, we define a radius $\rho := (\Delta x^2 + \Delta y^2)^{1/2}$ where Δx and Δy are the mesh sizes, and $C_\rho := \{x \in K, \ d((x,y), K\backslash C) \geq \rho\}$, where d is the Euclidian distance (C_ρ is a subset of C). Then we consider for $u \in U(x)$ and $v \in \{0,1\}$ the following approximated dynamic f_ρ:

$$f_\rho(x,u,v) := \begin{cases} f(x,u) & \text{if } x \in K\backslash C, \\ vf(x,u) & \text{if } x \in C\backslash C_\rho, \\ 0 & \text{if } x \in C_\rho. \end{cases} \tag{22}$$

Note that we deduce from (7a) a dynamic programming principle for V:

$$V(t+\Delta t, x) = \min_{(u,v)\in U\times\{0,1\}} V(t, y_{t,x}(t+\Delta t)), \tag{23}$$

where $y = y_{t,x}$ is the solution of $y(t) = x$ and $\dot{y} = f_C(y, u(s), v(s))$ on $[t, t+\Delta t]$. In the case we have only one given control $(u(s), v(s))$ for $s \in [t, t+\Delta t]$, the solution is $V(t+\Delta t, x) = V(t, y_{t,x}(t+\Delta t))$ at time $t_{n+1} = t + \Delta t$. It is thus approximated by the UB scheme by a value denoted $V_{i,j}^{n+1,UB}(u,v)$, obtained from the values $V_{i,j}^n$ at time $t = t_n$ (in the cell centered in $x_{i,j}$). Hence we propose the following scheme:

$$V_{i,j}^{n+1} = \min_{u_k, \ v\in\{0,1\}} \Big(V_{i,j}^{n+1}(u_k, v) \Big). \tag{24}$$

(where $(u_k)_{k=1,\dots,N}$ is a given discretization of the set $U(x_{i,j})$).

Note in fact that this scheme corresponds exactly to the HJB-UB scheme applied to the following "formal" HJB equation:

$$v_t - \min_{u\in U(x), \ v\in\{0,1\}} (f_\rho(x,u,v)\cdot\nabla v) = 0, \quad t > 0, \ x \in K. \tag{25}$$

This is why we shall also refer to this scheme as an HJB-UB scheme.

Stopping criteria. For the computation of a viability kernel or a capture basin using the UB scheme, the principle is first to evolve in time and compute some approximation of $V(t,x)$ using the HJB-UB algorithm, (using a time step $\Delta t > 0$ satisfying the CFL condition 18). Then we decide to stop the scheme when the values $V_{i,j}^n$ are numerically converging. This means in particular that the capture basin $\mathrm{Capt}_F(C)$ is approximated by $\mathrm{Capt}_F(C;T)$ for some $T > 0$. In practice, for the first two tests of the following section, the UB scheme is stopped when the quantity $||V^n - V^{n-1}||_{L^1} := \Delta x \Delta y \sum_{i,j} |V_{i,j}^n - V_{i,j}^{n-1}|$ satisfies:

$$||V^n - V^{n-1}||_{L^1} \leq 10^{-4}.$$

4 Numerical tests

In the following numerical tests, for the viability algorithm, we have used the basic version as presented in [8].

Example 1 (consommation problem) We consider the problem of computing the viability kernel for:

$$\dot{x}(t) = x(t) - y(t), \tag{26a}$$

$$\dot{y}(t) \in [-c, c], \tag{26b}$$

with $c = 1/2$, and the constraints $x(t) \in [0,2]$ and $y(t) \in [0,3]$. This corresponds to a consommation problem [8, 2]. Hence here $K := [0,2] \times [0,3]$ and the corresponding time dependant 2d HJB problem is

$$\begin{aligned}
&V_t + \max_{u=\pm c}(-f(x,y,u)\cdot\nabla v) = 0, \quad \forall t > 0, \forall (x,y) \in K,\\
&V(0,x,y) = 0, \quad \forall (x,y) \in K,\\
&V(t,x,y) = 1, \quad \forall (x,y) \notin K,\ t \geq 0,
\end{aligned}$$

where $f(x,y,u) = \begin{pmatrix} x-y \\ u \end{pmatrix}$. We have replaced the $+\infty$ value by 1 for commodity, and still have $\Omega_t = \{x,\ V(t,x) = 0\}$.

We have plotted in Fig. 1 the results given by the viability algorithm and by the HJB-UB scheme, for various mesh size ($Px = Py = 50$ and 100). For the UB scheme we have used time steps $\Delta t \simeq 0.013$ and 0.007 respectively, and stopped the computation at time $t_n = n\Delta t = 5$ approximatly. We have also used $N_u = 2$ ($u \in \{-c, c\}$). The black lines delimit the border of the exact solution.

Note that the viability algorithm computes values 0 or 1. In our algorithm, we compute values which are 0 or 1, or some intermediary value. The intermediary values are observed to be always on a "frontier" which bandwidth is about one or two mesh size. The error on this frontier is not diffused by the scheme (to the contrary to most numerical methods as Semi-Lagrangian or finite difference methods), but stays well localised in a small bandwidth.

In Fig. 1 and the following, the small black square regions represent the computed viability kernel (or capture basin).

For the UB scheme, the black square regions are associated with the points where $0 \leq V_{ij}^n \leq \epsilon$ with $\epsilon = 10^{-10}$ (the points from which we should be able to reach the target in time lesser than or equal to t_n); the grey points represents the mesh box with an intermediary value of $V_{i,j}^n$ between 0 (black) and 1 (white). More precisely, these boxes are represented if $\epsilon \leq V_{i,j}^n \leq 1 - \epsilon$. This correspond to mesh boxes where the discontinuity is detected.

Example 2 (Zermelo problem). In this example, we compute the capture basin for a "Zermelo Problem":

$$\dot{x}(t) = 1 - ay(t)^2 + u\cos(\theta), \tag{27a}$$

$$\dot{y}(t) = u\sin(\theta) \tag{27b}$$

in the domain $(x, y) \in K := [-6, 2] \times [-2, 2]$, and for controls $0 \leq u \leq u_{max} := 0.44$, $\theta \in [0, 2\pi[$, with $a = 0.1$. The target is chosen here as the ball $C := B(0, r)$ with $r = 0.44$.

The viability algorithm (see [9] in this case) and the UB scheme are compared in Fig.2, with $Px = Py = 100$. For the UB scheme, we have used $N_u = 20$ points and $dt \simeq 0.019$. the stopping criteria was $||V^n - V^{n-1}||_{L^1} \leq 10^{-4}$, which gave a stopping time $t \simeq 7$.

The circle delimits the border of the target, and the black lines also delimit the exact capture basin (we have computed the limit trajectories by using the Pontryagin Principle, see Bryson and Ho [1]).

Note that a good preliminary approximation is also obtained by the UB scheme even with a small number of mesh points. For instance in Fig.3 we have used $P_x = P_y = 20$ (with same number of controls, and $\Delta t \simeq 0.27$).

Example 3. In this example, we compute the "capture basin" for the following 2d rotational dynamic:

$$f(x, y, u) = \begin{pmatrix} y \\ -x \end{pmatrix}$$

on the domain $K = [-1, 1]^2$. The target is the ball centered in $(0.5, 0)$ and of radius 0.2, i.e.,

$$C := \{(x, y) \in K,\ (x - 0.5)^2 + y^2 \leq 0.2^2\}.$$

Note that in the dynamic f there is no dependency over a control u; however, the UB scheme does use a dynamic that depend of a control v as in (22) (hence in practice we have $N_u = 2$).

In Fig.4 we compare the viability algorithm and the UB scheme at time $T = \pi$ (half a turn), with $P_x = P_y = 100$ (and $\Delta t = 0.02$) and $P_x = P_y = 200$ (and $\Delta t = 0.01$). The small circle delimits the target and the black line reprensents the border of the exact solution.

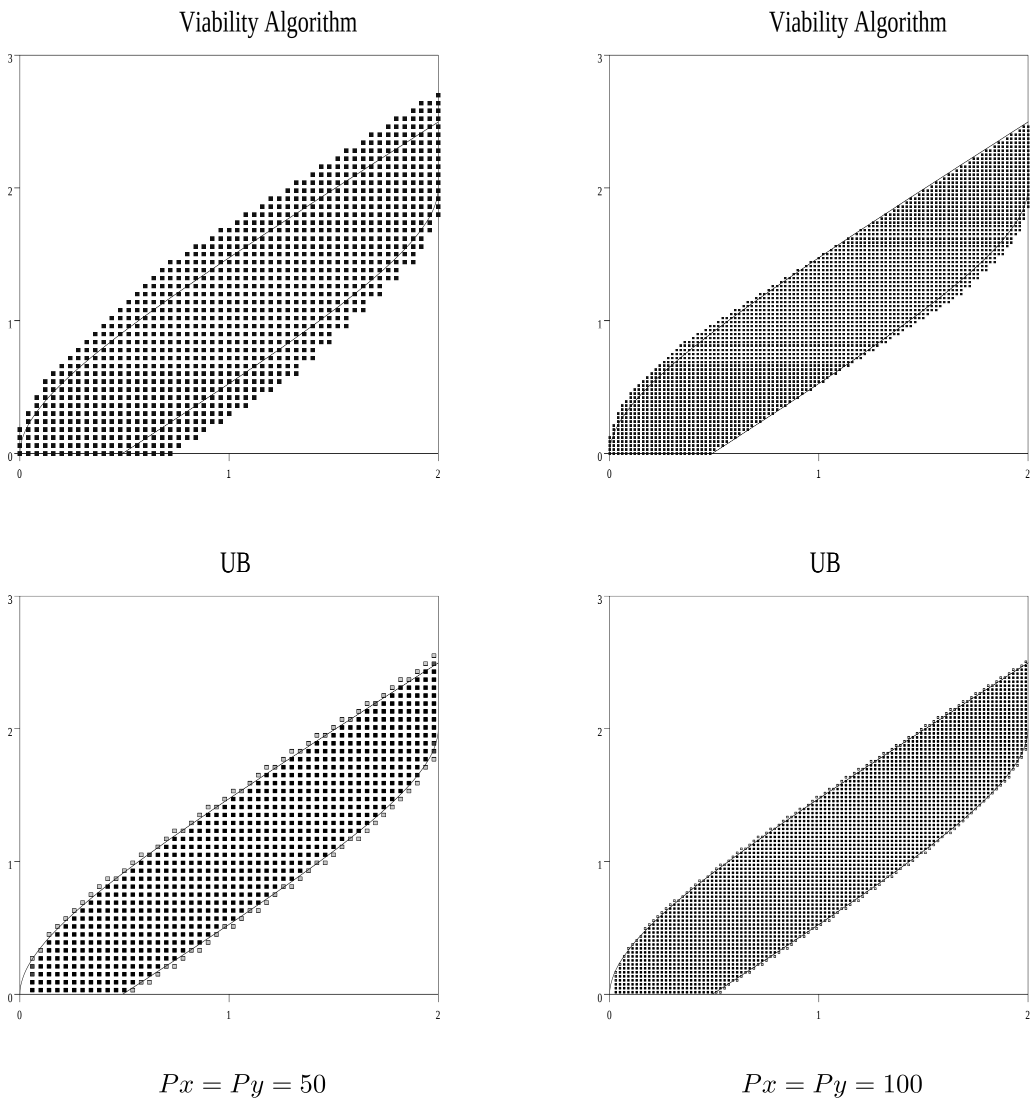


Figure 1: Comparison of the viability algorithm and the UB scheme for the consommation problem

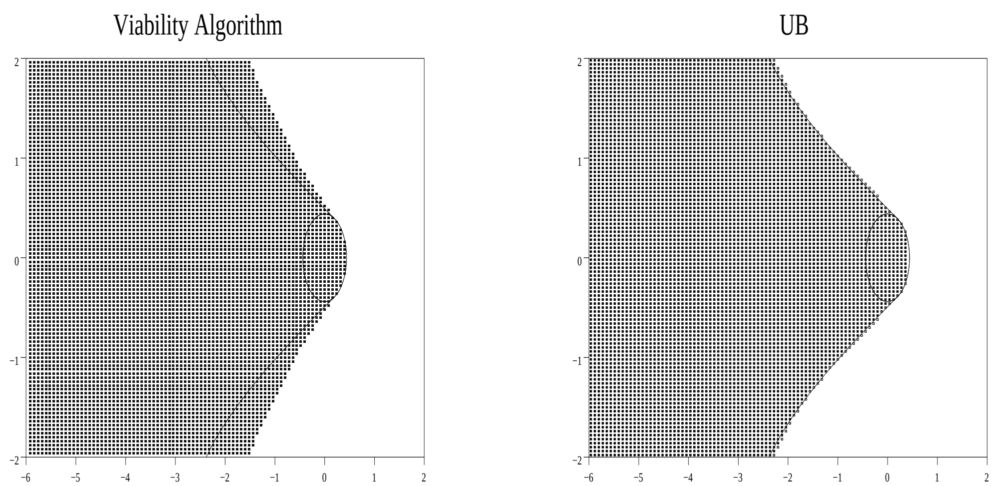


Figure 2: Approximation of the capture basin for the zermelo problem, $P_x = P_y = 100$

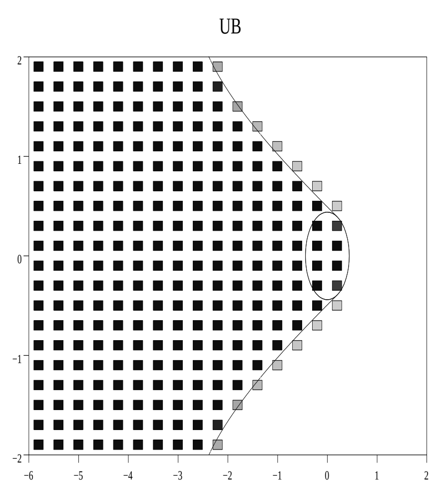


Figure 3: Zermelo problem, $P_x = P_y = 20$

In Fig.5 we show the results with the UB scheme at time $T = 10\pi$ (five turns), with $P_x = P_y = 50$ and $P_x = P_y = 100$ ($\Delta t = 0.04$ and $\Delta t = 0.02$ resp.). We see that the UB scheme present no visible diffusion, even on a long time period, whereas the viability algorithm - not shown for the case $T = 10\pi$ - has a tendency to diffuse more and more with time.

This example well illustrates the problem of diffusion of some schemes. A diffusive algorithm is going to create more and more errors as we go far from the target (or as time goes on). However the anti-diffusive scheme well approximates the capture basin even for long time as illustrated in Fig.5. (We have used $P_x = P_y = 25, 50$ and 100 with $\Delta t = 0.077, 0.038$ and 0.019 resp.)

Example 4. In this example we compute the capture basin for the following target problem

$$f(x, y, u) = \begin{pmatrix} y \\ u \end{pmatrix},$$

on a domain $K = [-1, 1]^2$, with control $u \in [-1, 1]$. The target is a "thin target" $C := (0, 0)$. Numerically, the mesh for the UB scheme is chosen so that $(0, 0)$ be the center of a mesh box of size $(\Delta x, \Delta y)$, and the initial data is $V_{i,j} = 0$ if $(x_i, y_j) = (0, 0)$ and $V_{i,j} = 1$ otherwise (this corresponds to take $v_0(x, y) = 1_{|x| \leq \Delta x/2, |y| \leq \Delta y/2})$. Here the problem is discretised with three controls $u \in \{-1, 0, 1\}$.

The results are given in Fig.6. As before, we obtain a small error and a small diffusion with the UB scheme, compared with the viability algorithm.

Acknowledgments. The authors wish to thank P. Saint Pierre for useful discussions and for help using his code.

A The viability Algorithm

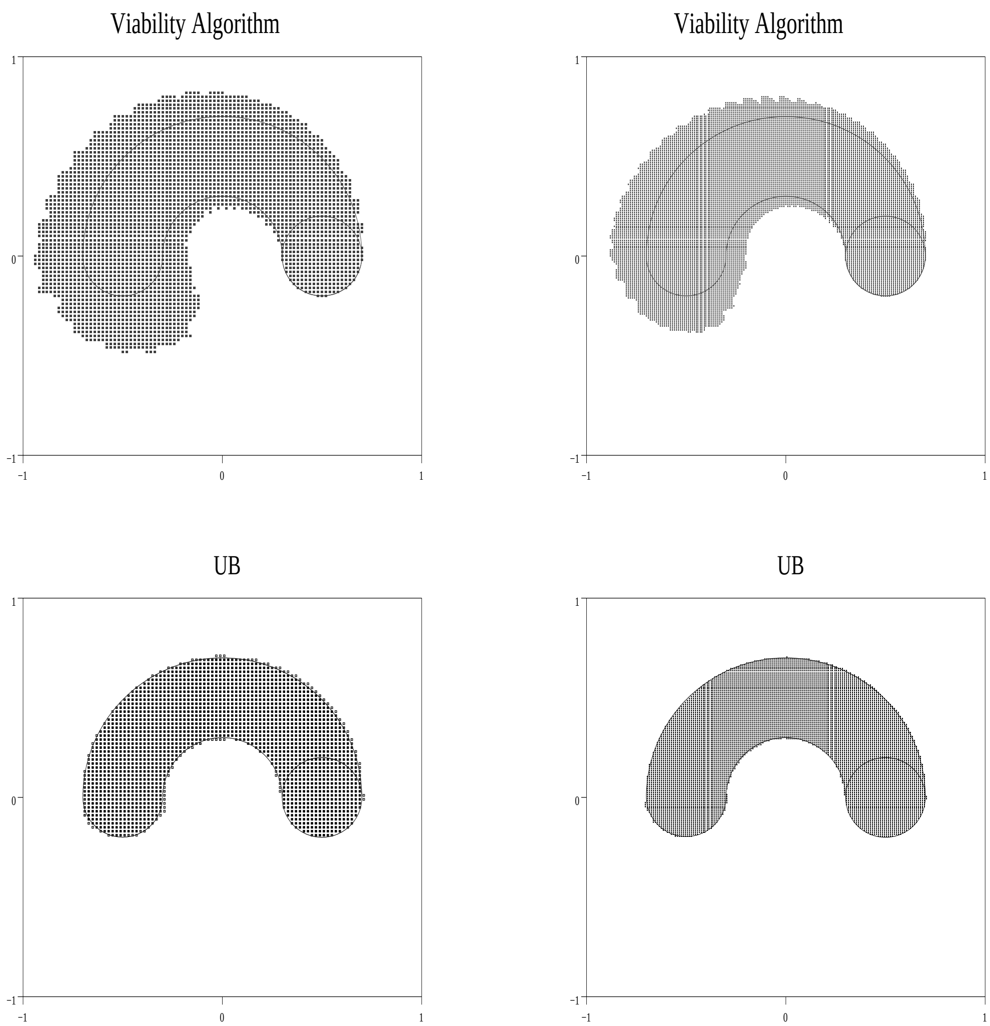
The approach followed in Saint-Pierre [8] is to determine the viability kernel in a constructive way by using discrete approximation. Also, Quincampoix and Saint-Pierre studied the case of a Hölderian differential inclusion [7]. In this case, the kernel is approximated by kernels of discrete dynamical systems and then by finite kernels of finite discrete dynamical systems.

A.1 Introduction and Notations.

Let X a finite-dimensional vector space and let K be a compact subset of X. We consider the differential inclusion:

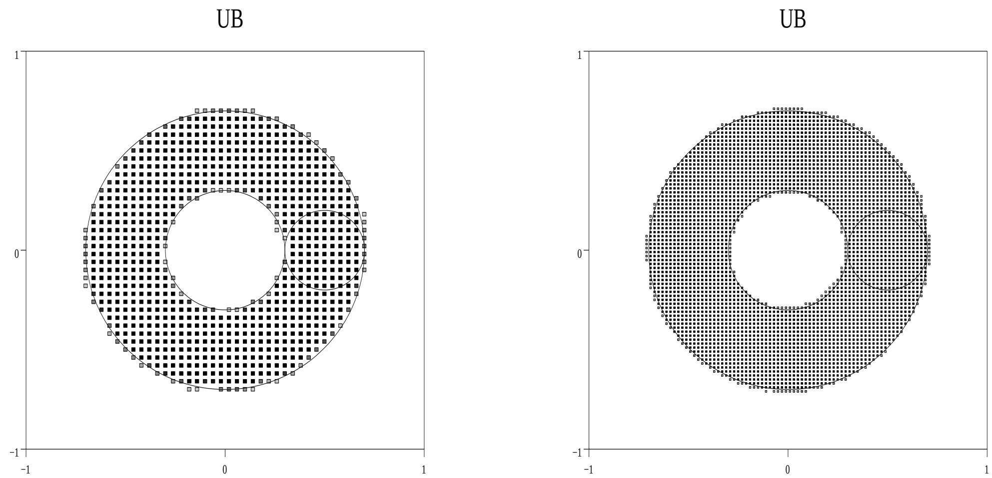
$$\begin{cases} x'(t) \in F(x) & \text{for almost all } t \geq 0, \\ x(0) = x_0 \in K, \end{cases} \tag{28}$$

where F is a Marchaud map defined from X to X.



$Px = Py = 100$ $Px = Py = 200$

Figure 4: points that can reach the target in time $t \leq \pi$



$Px = Py = 50$ $\qquad\qquad$ $Px = Py = 100$

Figure 5: points that can reach the target in time $t \leq 10\pi$ (UB scheme)

With this inclusion, for a fixed $\rho > 0$, we associate the discrete explicit scheme:

$$\begin{cases} \frac{x^{n+1}-x^n}{\rho} \in F(x^n) \quad \text{for all } n \geq 1, \\ x^0 = x_0 \in K. \end{cases} \tag{29}$$

We denote by G_ρ the set-valued map $G_\rho = 1 + \rho F$ and the system (29) can be rewritten as follows:

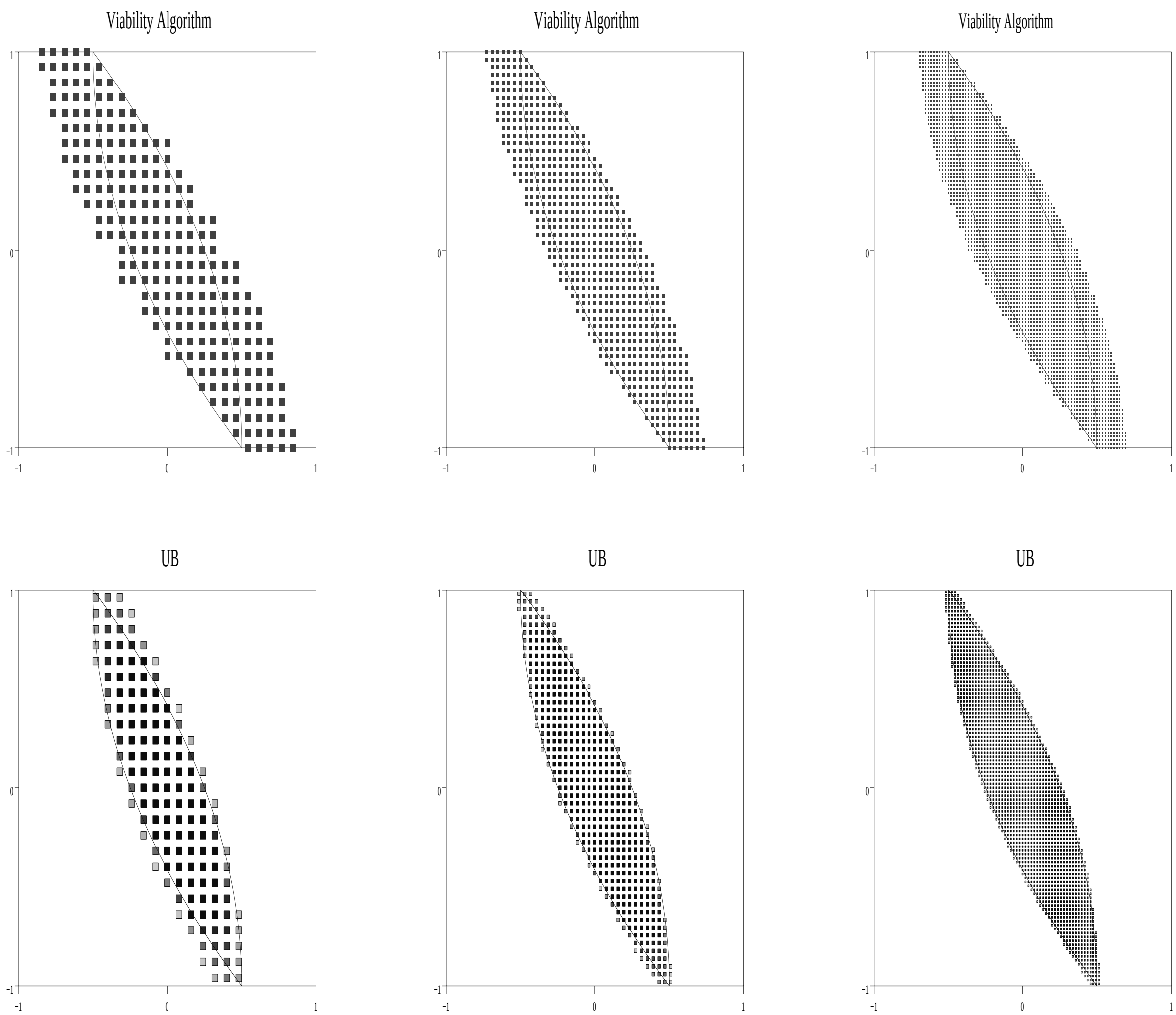
$$x^{n+1} \in G_\rho(x^n) \text{ for all } n \geq 0. \tag{30}$$

The viability kernel of K under F is the subset of all elements $x_0 \in K$ such that at least a viable solution starting at x_0 exists [2]. We denote it $\mathrm{Viab}_F(K)$. As far as the discrete dynamical system associated with G is concerned, we denote the discrete viability kernel of K under G $\mathrm{Viab}_G(K)$.

A.2 Approximation by Kernels of Discrete Dynamical Systems

Saint-Pierre [8] first addresses the problem of the approximation of kernels of discrete dynamical systems. Under some assumptions, Saint-Pierre [8] proves that, if the sequence $(K^n)_n$ (with $K^0 = K$) is defined as follows:

$$K^{n+1} := \{x \in K^n \text{ such that: } G(x) \bigcap K^n \neq \emptyset\} \tag{31}$$



$Px = Py = 26$ $Px = Py = 52$ $Px = Py = 104$

Figure 6: Cible problem, points that can reach the target in time $t \leq 1$

then $\text{Viab}_G(K) = \bigcap_{n=0}^{+\infty} K^n$.

He next proves the convergence of the following approximation process:

Theorem A.1. *Let F be a Marchaud and ℓ-Lipschitz set-valued map and K a closed subset of X such that $M := sup_{x\in K} sup_{y\in F(x)} \|y\| < +\infty$.*
Consider $F_\rho := F + \frac{M\ell}{2}\rho B$ and $\Gamma_\rho := 1 + F_\rho$. Then

$$\lim_{\rho\to 0} Viab_{\Gamma_\rho}(K) = Viab_F(K). \tag{32}$$

A.3 Approximation by Finite Set-Valued Maps

With any $h \in R$ we associate X_h a countable subset of X for instance a grid with step h. Let $G_h : X_h \to X_h$ a finite set-valued map and a subset $K_h \subset Dom(K_h)$. The finite dynamical system associated with G_h is

$$x_h^{n+1} \in G_h(x_h^n) \text{ for all } n \geq 0. \tag{33}$$

Saint-Pierre first remarks that, if the sequence $(K_h^n)_n$ (with $K_h^0 = K_h$) is defined as follows:

$$K_h^{n+1} := \{x \in K_h^n \text{ such that: } G_h(x) \bigcap K_h^n \neq \emptyset\} \tag{34}$$

then $\text{Viab}_{G_h}(K_h) = \bigcap_{n=0}^{+\infty} K_h^n$. Moreover, there exist p finite such that $\text{Viab}_{G_h}(K_h) = K_h^p$.

Let $G^r : X \to X$ such that $\forall x \in X$, $G^r(x) = G(x) + rB$. The following proposition links finite discrete viability kernels and discrete viability kernels:

Proposition A.1. *Let $G : X \to X$ be an upper semicontinuous set-valued map with closed values and K a closed subset of $Dom(G)$. Let $r > 0$ be such that for all $x \in Dom(G^r) \bigcap X_h$, $G^r(x) \bigcap X_h \neq \emptyset$, then*

$$Viab_{G^r}(K_h) \subset Viab_{G^r}(K) \bigcap X_h$$

.

Furthermore, for a good choice of r, these sets coincide.

Gathering the preceding results Saint-Pierre proves the following convergence properties of approximations of viability kernel of K under F with finite viability kernels computable in a finite number of steps:

$$\limsup_{\rho,h\to 0} Viab_{G_{\rho h}^{2M\ell\rho^2}}(K_h^{Ml\rho^2}) = Viab_F(K)$$

with $G_\rho^{2M\ell\rho^2} : X \to X$ and $G_{\rho h}^{2M\ell\rho^2} : X_h \to X_h$ defined as follows :

$$G_\rho^{2M\ell\rho^2} := x + \rho F(x) + 2M\ell\rho^2\mathcal{B}$$

$$G_{\rho h}^{2M\ell\rho^2} := G_{\rho h}^{2M\ell\rho^2} \cap X_h.$$

$$K_h^{M\ell\rho^2} := (K + M\ell\rho^2) \cap X_h.$$

This viability kernel algorithm allows to compute the exact discrete and finite viability kernel of the associated discrete problem defined on a finite grid.

References

[1] Bryson A.E. and Ho Y.C. *Applied optimal control.* Hemisphere Publishing, New-York, 1975.

[2] J.P. Aubin. *Viability theory.* Birkhäuser, 1991.

[3] O. Bokanowski and H. Zidani. Anti-dissipative schemes for advection and application to hamilton-jacobi-bellman equations. *Inria Report*, RR-5337, 2004.

[4] B. Després and F. Lagoutière. Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *J. Sci. Comput.*, 16:479–524, 2001.

[5] Hélène Frankowska and Sławomir Plaskacz. Semicontinuous solutions of Hamilton-Jacobi-Bellman equations with degenerate state constraints. *J. Math. Anal. Appl.*, 251(2):818–838, 2000.

[6] F. Lagoutière. PhD thesis, University of Paris VI, Paris, 2000.

[7] M. Quincampoix and P. Saint-Pierre. An algorithm for viability kernels in hölderian case: approximation by discrete dynamical systems. *Journal of Mathematical Systems, Estimation, and Control*, 5:1–13, 1995.

[8] P. Saint-Pierre. Approximation of viability kernel. *Appl. Math. Optim.*, 29:187–209, 1994.

[9] Patrick Saint-Pierre. Viable capture basin for studying differential and hybrid games: application to finance. *Int. Game Theory Rev.*, 6(1):109–136, 2004.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399