A method for optimal control problems

An adaptative antidissipative method for optimal control problems

Olivier Bokanowski*, Nadia Megdich**, Hasnaa Zidani**

* Lab. Jacques-Louis Lions, Université Pierre et Marie Curie, 175 Rue Chevaleret 75013 Paris, France. Email: boka@math.jussieu.fr
** ENSTA, UMA, 32 Boulevard Victor, 75739 Paris Cedex 15, France.
Emails: {Nadia.Megdich, Hasnaa.Zidani}@ensta.fr.

RÉSUMÉ. On étudie une méthode numérique pour les équations HJB provenant des problèmes de contrôle optimal avec contraintes sur l'état. Plus précisément on présente un schéma antidissipatif sur une grille adaptative. La grille adaptative est générée en utilisant la structure des quadtree linéaires. Cette technique facilite le stockage et la maniabilité des mailles.

ABSTRACT. We deal with a numerical method for HJB equations coming from optimal control problems with state constraints.
More precisely, we present here an antidissipative scheme applied on an adaptative grid. The adaptative grid is generated using linear quadtree structure. This technique of adaptation facilitates stocking data and dealing with large numerical systems.

MOTS-CLÉS : problèmes de contrôle optimal, équations HJB, schéma antidissipatif, quadtree linéaire.

KEYWORDS : optimal control problems, HJB equations, antidissipative scheme, linear quadtree.

Volume 5 – 2006, pages 256 à 271 – **ARIMA**

1. Introduction

In this paper, we deal with an optimal control problem $(\mathcal{P}_{s,x})$ with state constraint :

$$\left\{\begin{array}{l} \min \varphi(y_{x,s}(T)), \\ \dot{y}_{x,s}(t) = f(y_{x,s}(t), a(t)) \ \forall t \in [s,T] \\ y_{x,s}(s) = x, \ x \in \boldsymbol{R}^n, \\ a(t) \in \mathcal{A} \text{ a.e }, \ \forall t \in [s,T], \\ y_{x,s}(t) \in \mathcal{K} \ \ \forall t \in [s,T]. \end{array}\right. \tag{1}$$

The set of controls $\mathcal{A}$ is a compact of $\boldsymbol{R}^m$, $\varphi : \boldsymbol{R}^n \to \boldsymbol{R} \cup \{+\infty\}$ is lower semi continuous (l.s.c), and T is a fixed final time. The set $\mathcal{K} \neq \emptyset$ is a compact convex set of $\boldsymbol{R}^n$. The dynamics $f : \boldsymbol{R}^n \times \mathcal{A} \to \boldsymbol{R}^n$ is assumed to be Lipschitz and bounded.

Let $v : [0,T] \times \boldsymbol{R}^n \to \boldsymbol{R} \cup \{+\infty\}$ be the value function defined by $v(s,x) = \inf(\mathcal{P}_{s,x})$. For every $s \in [0,T]$ and $x \notin \mathcal{K}$, $v(s,x) = +\infty$ and for $x \in \mathcal{K}$, $v(T,x) = \varphi(x)$.
It is known that the value function v satisfies the *Dynamic Programming Principle* (DPP) :

$$v(s,x) = \inf_{a(.) \in A(s,\tau;x)} v(\tau, y_{x,s}(\tau)), \qquad \forall \tau \in]s,T], \ \forall x \in \mathcal{K}, \tag{2}$$

where $A(s,\tau;x) := \{a : [0,+\infty[\to \mathcal{A} \text{ measurable}, \ y_{x,s}(t) \in \mathcal{K}, \ \forall t \in [s,\tau]\}$. In the case when the final cost function φ is continuous and $\mathcal{K} = \boldsymbol{R}^n$, the value function is the unique continuous "viscosity" solution [B-CD, B, CD-L] of the *Hamilton-Jacobi-Bellman* (HJB) equation :

$$\left\{\begin{array}{ll} -v_t(t,x) - \min_{a \in \mathcal{A}} f(x,a).v_x(t,x) = 0, & (t,x) \in [0,T] \times \mathcal{K}, \\ v(T,x) = \varphi(x), & x \in \mathcal{K}. \end{array}\right. \tag{3}$$

Here, we are interested in the case when φ is given by

$$\varphi(x) = \left\{\begin{array}{ll} 0 & \text{if } x \in \mathcal{C}, \\ +\infty & \text{otherwise,} \end{array}\right. \tag{4}$$

where $\mathcal{C} \neq \emptyset$ is a compact convex set of $\boldsymbol{R}^n$, $\mathcal{C} \subset \mathcal{K}$ and $\mathcal{K} \neq \boldsymbol{R}^n$. In section 4, we will see that this case modelizes several control problems (target problem, Rendez-Vous problem, viability kernels,...). Here, the value function v may clearly be discontinuous and takes its values in $\{0, +\infty\}$. It still satisfies equation (3) in a sense given by Frankowska and her co-authors, see [FV, FP] and the references therein for all the details.

Several numerical schemes have been studied for discretizing (3). The most popular are the Semi-Lagrangian schemes [FF, FG, Gr] and the finite differences schemes [S, CL]. These schemes provide a good approximation for a continuous value function. However, they all use interpolation techniques at some level and are no more suitable

for the approximation of discontinuous value functions. Indeed, the interpolation steps produce more or less numerical diffusion, which causes an increasing loss of precision mainly around the discontinuities. To our knowledge, the only scheme which doesn't use any interpolation technique is the one based on the viability algorithm developed by P. Saint Pierre and his co-authors [SP]. But, as already shown in [BMMZ] this scheme still diffuses.

The approximation method we study here is a mixture of the antidissipative *UltraBee* (UB) scheme [DL, BZ] and of an adaptative gridding technique. The UltraBee scheme has been studied by B. Désprès and F. Lagoutière [DL] for solving the transport equation with positive constant velocity. It has been extended by O. Bokanowski and H. Zidani [BZ] for the transport equation with a changing sign velocity and applied for the resolution of Hamilton Jacobi equations (3) on a regular grid.
In our case, the value function takes only values 0 and 1 (the value 1 coding in fact the $+\infty$ value). In this special situation, the UltraBee scheme has a nice property : it is able to localize accurately the discontinuity of v corresponding to the interface Γ_t separating the region where $v(t,.)$ takes the value 1 from the region where it takes the value 0. This property allows us to design a simple method for adaptative gridding. Moreover, the real calculations at every time $t^n = n\Delta t$ (Δt being the time step) have only to be done on a small neighborhood of the interface Γ_{t^n}. Hence adaptative gridding is particularly interesting in our case. Moreover, we use linear quadtrees which provide a good way to handle easily adaptative grids and to achieve a significant save of memory.

Adaptative gridding for solving HJB equations has already been studied in the case of a continuous value function [CY1, CY2, Gr]. In [Gr] for example, L.Grune has handled the Semi-Lagrangian scheme to solve (3) and explained the criteria he used for the refinement and coarsening steps. These criteria are based on a fixed tolerance for the interpolation error. The presence of discontinuities in our case makes these criteria no more suitable.

The paper is organized as follows. In section 2 we give the formulation of the UltraBee scheme and some of its properties. In section 3 we present the adaptative technique that we use and explain the steps of the proposed method. Finally in section 4, we give several numerical simulations in 2 dimensions coming from control problems and propagating front problems.

2. The UltraBee scheme

Notice that when we deal with only one control, the HJB equation (3) becomes a transport equation. Hence we will first present the UltraBee scheme in this simple case in one space dimension ($n = 1$).

2.1. Transport equation

Let $f : \boldsymbol{R} \to \boldsymbol{R}$ be Lipschitz and bounded and $u_0 : \boldsymbol{R} \to \boldsymbol{R}$ be lower semi continuous. We consider the transport problem :

$$\left\{ \begin{array}{l} u_t(t,x) + f(x)u_x(t,x) = 0,\ x \in \boldsymbol{R},\ t \geq 0, \\ u(0,x) = u_0(x),\ x \in \boldsymbol{R}. \end{array} \right. \tag{5}$$

In all the sequel, we will use the following notations : Δt denotes the time step, Δx is the space step of a regular grid $\mathcal{G}$ of $\boldsymbol{R}$ and ν_j is the local CFL number at cell M_j defined by :

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

Consider the following scheme of finite volumes type :

$$\left\{ \begin{array}{l} \dfrac{U_j^{n+1} - U_j^n}{\Delta t} + f(x_j)\dfrac{U_{j+\frac{1}{2}}^{n,L} - U_{j-\frac{1}{2}}^{n,R}}{\Delta x} = 0,\ \forall j \in \boldsymbol{Z},\ \forall n \in \boldsymbol{N}, \\ U_j^0 = \dfrac{1}{\Delta x}\displaystyle\int_{M_j} u_0(x)dx,\ \forall j \in \boldsymbol{Z}. \end{array} \right. \tag{6}$$

where x_j is the middle point of cell $M_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, U_j^n is an approximation of the mean value $\dfrac{1}{\Delta x}\displaystyle\int_{M_j} u(t^n,x)dx$ of u on cell M_j at t^n, and $U_{j+\frac{1}{2}}^{n,L},\ U_{j+\frac{1}{2}}^{n,R}$ are fluxes respectively on the left and on the right of the interface of cells M_j and M_{j+1} at time t^n. For the UltraBee scheme, these fluxes are defined in the following way.

• In the case when the velocity $f(.) \equiv f$ is a positive constant, the fluxes $U_{j+\frac{1}{2}}^{n,L}$ and $U_{j+\frac{1}{2}}^{n,R}$ coincide and we have $U_{j+\frac{1}{2}}^{n,L} = U_{j+\frac{1}{2}}^{n,R} =: U_{j+\frac{1}{2}}^n$. The scheme becomes :

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + f(x_j)\frac{U_{j+\frac{1}{2}}^n - U_{j-\frac{1}{2}}^n}{\Delta x} = 0.$$

The UltraBee scheme, as defined in [DL], is a downwind choice of the fluxes under some stability conditions. This choice replaces the classical Upwind flux which is stable but dissipative. More precisely, the flux $U_{j+\frac{1}{2}}^n$ is given by solving the minimization problem : $\min_{b_j^{n,+} \leq U \leq B_j^{n,+}} |U - U_{j+1}^n|$ where $b_j^{n,+}$ and $B_j^{n,+}$ are defined by :

$$b_j^{n,+} = \frac{1}{\nu_j}(U_j^n - \max(U_j^n, U_{j-1}^n)) + \max(U_j^n, U_{j-1}^n),$$

$$B_j^{n,+} = \frac{1}{\nu_j}(U_j^n - \min(U_j^n, U_{j-1}^n)) + \min(U_j^n, U_{j-1}^n).$$

It follows that

$$U_{j+\frac{1}{2}}^n = \min(\max(U_{j+1}^n, b_j^{n,+}), B_j^{n,+}). \tag{7}$$

• In the case when f is of changing sign, the *UltraBee generalized scheme* (UB-G) [BZ] is defined as follows

a) if $\nu_j > 0$, $U^{n,L}_{j+\frac{1}{2}} = \min(\max(U^n_{j+1}, b^{n,+}_j), B^{n,+}_j)$ as proposed in (7).

b) if $\nu_j < 0$, we define symetrically $U^{n,R}_{j-\frac{1}{2}} = \min(\max(U^n_{j-1}, b^{n,-}_j), B^{n,-}_j)$ with $b^{n,-}_j = \frac{1}{|\nu_j|}(U^n_j - \max(U^n_j, U^n_{j+1})) + \max(U^n_j, U^n_{j+1})$, and $B^{n,-}_j = \frac{1}{|\nu_j|}(U^n_j - \min(U^n_j, U^n_{j+1})) + \min(U^n_j, U^n_{j+1})$.

c) if $\nu_j \leq 0$ and $\nu_{j+1} \geq 0$, $U^{n,L}_{j+\frac{1}{2}} = U^n_j$, $U^{n,R}_{j+\frac{1}{2}} = U^n_{j+1}$.

d) if $\nu_j\nu_{j+1} > 0$, $U^{n,R}_{j+\frac{1}{2}} = U^{n,L}_{j+\frac{1}{2}}$ (if $\nu_j > 0$) or $U^{n,L}_{j+\frac{1}{2}} = U^{n,R}_{j+\frac{1}{2}}$ (if $\nu_{j+1} < 0$).

When the velocity is constant, and under the CFL condition,

$$|\nu_j| \leq 1 \; \forall j \in \mathbb{Z}, \tag{8}$$

one interesting property of the UltraBee scheme is an exact advection [DL, Theorem 3] for a class of step functions defined by : $\exists k^0 \in [0,1[$ such that $\forall j \in \mathbb{Z}$,

$$U^0_{3j+1} = U^0_{3j}, \quad U^0_{3j+2} = k^0 U^0_{3j+1} + (1-k^0)U^0_{3j+3}. \tag{9}$$

Exact advection means that the computed value U^n_j is the exact mean value,

$$U^n_j = \frac{1}{\Delta x}\int_{M_j} u(t^n, x)dx,$$

where u is the exact solution of the advection problem. For the convergence proofs of the UltraBee scheme, we refer to [DL, BZ].

2.2. HJB equation

Here, we are still in dimension 1. First, we consider the simple change of variable,

$$\hat{v}(t,x) = v(T-t,x), \; \forall t \in [0,T], \; \forall x \in \boldsymbol{R}.$$

Then the function $\hat{v}$ satisfies

$$\begin{cases} \hat{v}_t(t,x) - \min\limits_{a\in\mathcal{A}} f(x,a).\hat{v}_x(t,x) = 0, \; \forall (t,x) \in [0,T]\times\mathcal{K}, \\ \hat{v}(0,x) = \varphi(x), \; \forall x \in \mathcal{K}. \end{cases} \tag{10}$$

The application of UB-G to the HJB equation (10) consists, on a regular grid $\mathcal{G}$ of $\mathcal{K}$, in the following steps (UB-HJB) :

• Step 1 : We compute the discrete initial condition

$$V^0_j = \frac{1}{\Delta x}\int_{M_j} \varphi(x)dx, \; \forall j \in J := \{j \in \mathbb{Z}, \; M_j \in \mathcal{G}\}.$$

• Step 2 : We discretize the set of controls $\mathcal{A}$ into N_a controls, $a_1, a_2, ..., a_{N_a}$.

• Step 3 : For $n \geq 1$, knowing the approximation $(V_j^n)_{j\in J}$ of $\hat{v}(t^n, .)$

- We compute, for each $i = 1...N_a$, $(U_j^{n+1}(a_i))_{j\in J}$ given by the UB-G scheme :

$$\begin{cases} U_j^{n+1}(a_i) = U_j^n(a_i) - \dfrac{f(x_j, a_i)\Delta t}{\Delta x}(U_{j+\frac{1}{2}}^{n,L}(a_i) - U_{j-\frac{1}{2}}^{n,R}(a_i)), \\ U_j^n(a_i) = V_j^n, \quad \forall j \in J. \end{cases}$$

- We take $V_j^{n+1} := \min_{i=1..N_a} U_j^{n+1}(a_i),\ \forall j \in J$. This defines the numerical approximation of $\hat{v}$ at t^{n+1}.

In dimension 2, we apply the UB-HJB using the classical Trotter splitting : the numerical solution evolves during a time step in the x^1-direction and then during another time step in the x^2-direction. The resolution using this splitting technique is stable under the CFL condition,

$$\max(|\frac{f_1(x_j, a_i)\Delta t}{\Delta x^1}|, |\frac{f_2(x_j, a_i)\Delta t}{\Delta x^2}|) \leq 1,\ \forall j \in J,\ \forall i = 1, .., N_a. \tag{11}$$

Here, the dynamics f is defined by $f := (f_1, f_2)$, Δx^1 and Δx^2 are the space steps respectively in the x^1-direction and in the x^2-direction, and $x_j \in \boldsymbol{R}^2$ is the center of the cell M_j.

In [BMZ], under some suitable assumptions, we prove in one space dimension the convergence of the UB-HJB scheme towards the value function for any initial condition φ which is C^1-piecewise regular with compact support.

Notice that, at the first step of UB-HJB scheme, when we compute the average values $(V_j^0)_{j\in J}$, the only components whose values are strictly between 0 and 1 are those corresponding to the cells containing the front Γ_0 (we recall that Γ_0 is the interface separating 0-values of $\hat{v}(t = 0, .)$ and its 1-values). In dimension 1, we prove in [BMZ] that, for every $n \geq 0$, the interface Γ_{t^n} is localized on no more than one cell. In dimension 2, we shall verify numerically that Γ_{t^n} is still well localized by the UB-HJB scheme, but we don't have yet any precise theoretical result to claim.

3. The adaptative method

We explain in this section the details of the method that we propose. For sake of simplicity, we take $n = 2$ and $\mathcal{K} = [X^1_{\min}, X^1_{\max}] \times [X^2_{\min}, X^2_{\max}]$. Before dealing with details, we start by presenting the linear quadtrees technique that we use for stocking data.

3.1. Linear quadtrees

As we deal with adaptative grids, we look for a technique that facilitates stocking and finding data relative to each cell of the grid. This technique is explained by I. Gargantini in [Ga] and uses the notion of *linear quadtree*. If we represent our final adapted grid by a tree, each cell is a leaf (final node of the tree) and the initial quadrant (all the domain $\mathcal{K}$ before adaptation) is the root of the tree. The method for stocking data using quadtrees is based on coding each leaf of the tree with a quaternary function. This code representation is implicitly the path from the root to the concerned leaf.
Every code is composed of 0, 1, 2, 3. When dividing a cell into four subcells, the NW quadrant is indexed by 0, the NE by 1, the SW by 2 and the SE by 3. The code of each subcell is the concatenation of the code of the mother cell with the index of the subcell (as shown in figure 3.1). Here cells 20, 21, 22 and 23 are sisters and 2 is the mother cell.

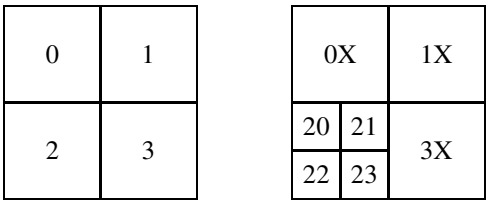


Figure 1. *Refinement of a cell by quadtrees*

Notice that when coding the grid using a tree, all intermediate cells have to be memorized, for example we memorize cells 2, 20, 21, 22, 23. However, in a linear quadtree, we have to stock only final cells of the grid, i.e. cells 20, 21, 22, 23. Then, to find the intermediate cells, we have just to truncate the codes. Furthermore the use of linear quadtrees allows to manage efficiently the adapted grid. In fact, thanks to fast algorithms, operations like encoding a cell into its quaternary code and finding adjacencies of a cell are run in logarithmic time [Ga].

3.2. Algorithm of the method

Our contribution consists in finding a suitable criterion to adapt the computational domain. This criterion must be compatible with the fact that we deal with mean values on each cell and that our value function is discontinuous. Let $L_{\max}$ be a fixed integer that corresponds to the maximal level of refinement. We set $\Delta X^1_{\min} = \frac{|X^1_{\max} - X^1_{\min}|}{2^{L_{\max}}}$ and $\Delta X^2_{\min} = \frac{|X^2_{\max} - X^2_{\min}|}{2^{L_{\max}}}$. Then $(\Delta X^1_{\min}, \Delta X^2_{\min})$ is the minimal cell size. The maximal level $L_{\max}$ is chosen such that the following CFL condition holds :

$$\max(|\frac{f_1(x_j, a_i)\Delta t}{\Delta X^1_{\min}}|, |\frac{f_2(x_j, a_i)\Delta t}{\Delta X^2_{\min}}|) \leq 1, \ \forall j \in J, \ \forall i = 1, .., N_a. \qquad (12)$$

In all the sequel, for every $n \geq 0$, we denote by $\mathcal{G}^n$ the adaptative grid at time $t_n = n\Delta t$. We also use the notation $\mathcal{G}_l$, $l = 1, \cdots, L_{\max}$, for the regular grid with mesh steps : $\Delta_l X^1 = \frac{|X^1_{\max} - X^1_{\min}|}{2^l}$, $\Delta_l X^2 = \frac{|X^2_{\max} - X^2_{\min}|}{2^l}$.

We give now the algorithm of the method which begins by an initialization step. First, we handle refinement steps in order to localize the discontinuity. At every refinement level, a cell which is surrounded by cells not having the same mean value is refined. After these refinement steps, the grid we obtain may contain sister cells having the same mean value (0 or 1) as their immediate neighboring cells. These cells do not contain any discontinuity and have to be coarsened : this is the coarsening steps. Finally, we get the grid $\mathcal{G}^0$ where a cell of minimal size either contains a discontinuity or is a neighbor of a cell containing a discontinuity or is a sister of such a cell.

Step 1 : The construction of the grid $\mathcal{G}^0$.

– Step 1.1 : Take $\mathcal{G}^{0,0} = \mathcal{K}$ (one cell), and define $\mathcal{G}^{0,1}$ as the domain $\mathcal{K}$ splitted into four cells. Set $l = 1$.

– Step 1.2 : For $1 \leq l \leq L_{\max} - 1$. For all cells $M_j \in \mathcal{G}^{0,l} \setminus \mathcal{G}^{0,l-1}$, compare the value on M_j with its neighboring values. If the values are different, then refine cell M_j. We obtain a new grid denoted $\mathcal{G}^{0,l+1}$. Set $l = l + 1$ and go to Step 1.2.
Otherwise, set $l = L_{\max}$ and go to Step 1.3.

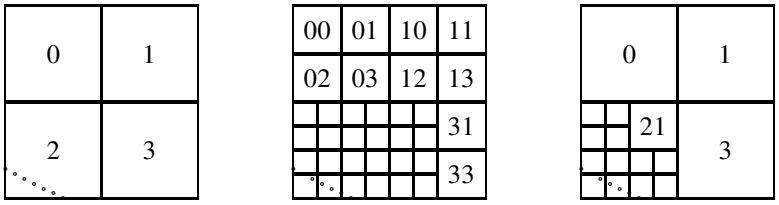
– Step 1.3 : Set $\widetilde{\mathcal{G}}^{0,L_{\max}} = \mathcal{G}^{0,L_{\max}}$.

– Step 1.4 : For $l = L_{\max}, \cdots, 2$, for every cell $M_j \in \widetilde{\mathcal{G}}^{0,l} \cap \mathcal{G}_l$, if the sisters of M_j have the same mean value (0 or 1) and if the neighboring cells of the four sisters have also the same value, then coarsen M_j with its sisters. We obtain a new grid denoted $\widetilde{\mathcal{G}}^{0,l-1}$. Set $l = l - 1$, and go to Step 1.4. Otherwise, set $l = 1$, and go to Step 1.5.

– Step 1.5 : Set $\mathcal{G}^0 := \widetilde{\mathcal{G}}^{0,1}$ and define V^0 on $\mathcal{G}^0$.

For example, the construction of the adapted grid $\mathcal{G}^0$ follows the refinement steps explicited in figure 3.2 for $L_{max} = 3$. The value function here takes value 1 below the discontinuity and value 0 beyond. At the second level of refinement, cell 2 is refined as its mean value is in $]0, 1[$. Cells 0, 1, 3 are refined too because their value 0 differs from the value of their neighbor cell 2. Notice that, at level 3, $\mathcal{G}^{0,3}$ is such that all cells containing the discontinuity are of minimal size as well as their neighboring cells. After refinement, we carry out a coarsening step. Following the test of the algorithm, we coarsen subcells of 21, 30, 32 and then subcells of 0, 1 and 3.

Now, for $n \geq 0$, we have the adapted grid $\mathcal{G}^n$ (at time t^n) and the numerical solution V^n on $\mathcal{G}^n$. The discontinuity at t^n lies in the region of $\mathcal{G}^n$ where the cells are of minimal size. Because of the CFL condition (12), we know that the discontinuity is still in this region at t^{n+1} (as already explained the discontinuity does not evolve of more than one cell of size $(\Delta X^1_{\min}, \Delta X^2_{\min})$ during a time step). We conclude that $V^{n+1}_j = V^n_j$ (=0 or 1) whenever $M_j \in \mathcal{G}^n$ is not of minimal size, and the only computations which remain to be done correspond to the cells of minimal size.



$\mathcal{G}^{0,1}$ $\mathcal{G}^{0,3}$ $\mathcal{G}^{0}$

Figure 2. *Construction of $\mathcal{G}^0$, the discontinuity is plotted with dots.*

Step 2 : UB-HJB computation and construction of $\mathcal{G}^{n+1}$.

– Step 2.0 : Do an iteration of UB-HJB scheme on the cells of minimal size $(\Delta X^1_{\min}, \Delta X^2_{\min})$ of $\mathcal{G}^n$. We obtain V^{n+1} on $\mathcal{G}^n$.

– Step 2.1 : Define $\mathcal{G}^{n+1,0} := \mathcal{G}^n$, and $V^{n+1,0} := V^{n+1}$ on $\mathcal{G}^n$.

– Step 2.2 : For $1 \leq l \leq L_{\max} - 1$, for all cells $M_j \in \mathcal{G}^{n+1,l} \cap \mathcal{G}_l$, compare the value[1] $V_j^{n+1,l}$ on M_j with its neighboring values. If the values are different, then refine cell M_j, attribute to the daugther cells of M_j the same value $V_j^{n+1,l}$. This defines a new grid $\mathcal{G}^{n+1,l+1}$. Set $l = l + 1$, and go to Step 2.2.
Otherwise, set $l = L_{max}$ and go to step 2.3.

– Step 2.3 : Set $\widetilde{\mathcal{G}}^{n+1,L_{\max}} := \mathcal{G}^{n+1,L_{\max}}$ and $\widetilde{V}^{n+1,L_{\max}} := V^{n+1,L_{\max}}$.

– Step 2.4 : For $l = L_{\max}, \cdots, 2$, do a corsening step following the same idea as in Step 1.4. If there is no coarsening to do, then set $l = 1$, and go to Step 2.5.

– Step 2.5 : Set $\mathcal{G}^{n+1} := \widetilde{\mathcal{G}}^{n+1,1}$, and $V^{n+1} := \widetilde{V}^{n+1,1}$. This corresponds to the approximation on $\mathcal{G}^{n+1}$ of the solution $\hat{v}$ of (10) at $t = t^{n+1}$.

By construction, we have the following equivalence result :
Theorem : Let $L_{\max}$ be a fixed integer. Under the CFL condition (12), the approximation of (10) using the UB-HJB scheme on an adaptative grid gives the same numerical solution as the resolution using the UB-HJB scheme on a regular grid $\mathcal{G}_{L_{\max}}$.

4. Numerical simulations

In the graphics through all this section, we use the black color for cells with mean value strictly between 0 and 1, white for cells with value 0 and light gray for cells with value 1. We also use the notation $\mathcal{B}(c_0, r)$ for the ball centered in c_0 and with radius r.

Example 1 : A propagating front problem
We first start with a propagating fronts problem. The initial condition is here two sources

1. Recall that for every $M_j \in \mathcal{G}^{n+1,l} \cap \mathcal{G}_l$, with $l < L_{\max}$, the mean value $V_j^{n+1,l}$ is equal to 0 or 1.

from which a fire spreads. Let φ be a function that modelizes the burnt region at $t = 0$, and defined as :

$$\varphi(x) = \begin{cases} 0 & \text{if } x \in \mathcal{B}(c_1, 0.1) \cup \mathcal{B}(c_2, 0.1), \\ 1 & \text{otherwise,} \end{cases}$$

with $c_1 = (0.4, 0.4)$ and $c_2 = (0.6, 0.6)$. Let $\mathcal{K}$ denote the domain $[-0.5, 2.5]\times[-1.5, 1.5]$. We associate to this problem the function $\hat{v}$ which takes value 0 in $(t, x) \in [0, T] \times \mathcal{K}$ if the flame front has already reached x at t, and 1 otherwise. In fact, $\hat{v}$ satisfies the Eikonal equation,

$$\begin{cases} \hat{v}_t(t, x) + ||\nabla \hat{v}(t, x)|| + (-x_2, x_1)^t . \nabla \hat{v}(t, x) = 0, \ \forall t \in [0, T], \ \forall x = (x_1, x_2) \in \mathcal{K}, \\ \hat{v}(0, x) = \varphi(x), \ \forall x \in \mathcal{K}. \end{cases} \tag{13}$$

The discontinuity of $\hat{v}$ at time t is the position of the flame front at time t. Hence the set $\{x \in \mathcal{K}, \hat{v}(t, x) = 0\}$ represents the burnt zone at time t.
Although this problem comes from front propagation, it takes place in the formalism we study. Indeed, the Eikonal equation (13) can be written as an HJB equation :

$$\begin{cases} \hat{v}_t(t, x) - \min_{a \in \mathcal{A}} f(x, a) . \nabla \hat{v}(t, x) = 0, \ \forall t \in [0, T], \ \forall x \in \mathcal{K}, \\ \hat{v}(0, x) = \varphi(x), \ \forall x \in \mathcal{K}, \end{cases}$$

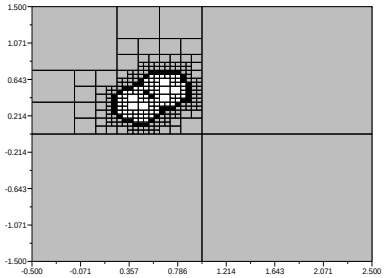
where the set of controls is $\mathcal{A} = [0, 2\pi]$, and the dynamics is given by :

$$f(x, a) = (x_2 - \cos(a), -x_1 - \sin(a))^t, \ \forall x \in \boldsymbol{R}^2, \ \forall a \in \mathcal{A}.$$

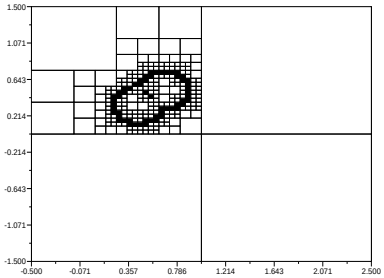
In the numerical tests, we discretize $\mathcal{A}$ into $N_a = 8$ controls, and we choose $L_{max} = 6$ as maximal level of refinement. We visualize the computed solution and the error which is defined on each cell M_j, for $j \in J$, by $\varepsilon_j^n = |V_j^n - \widetilde{V}_j^n|$. Here $\widetilde{V}_j^n$ is the average value of the exact solution $\hat{v}$ on cell M_j at time t^n.

We display graphics at $T = 0.11$ (figure 3) when the two fronts meet, and then at $T = 0.87$ (figure 4) when we get only one front which is already far from the sources of fire. Notice that the error is localized in a thin region around the discontinuity of bandwidth of no more than twice the size of a minimal cell. This is the antidissipative behavior of the scheme. Notice also that the approximation quality isn't distorted when the discontinuity evolves in time. This is another feature of the antidissipative behavior.

Table 1 summerizes the gain we obtain in terms of number of cells when we apply adaptation by comparison to an equivalent regular grid. We can notice that, as expected, we obtain exactly the same error on an adaptative grid and on a regular one. Notice that when we increase the refinement level by 1, the gain is multiplied by 2 and the error is multiplied by $\frac{2}{3}$. This reflects optimization in the management of cells. We can also notice at $T = 0.11$ that when we fix $L_{max} = 10$ we handle almost 4000 cells on an adaptative grid, as much cells as if we did calculations on a regular grid corresponding to $L_{max} = 6$.

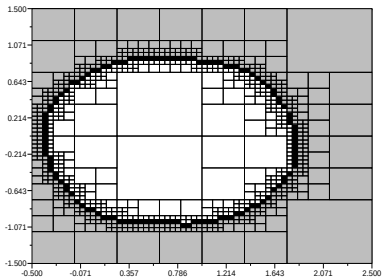


(a) computed solution

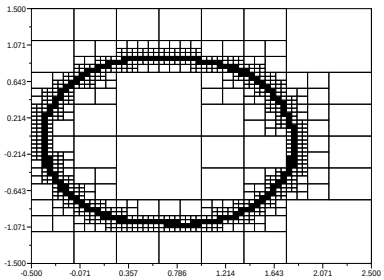


(b) error $(\varepsilon_j^n)_{j\in J}$

Figure 3. *Computed solution and error at T=0.11, # cells=244,* $L_{max} = 6$.



(a) computed solution



(b) error $(\varepsilon_j^n)_{j\in J}$

Figure 4. *Computed solution and error at T=0.87, # cells=817,* $L_{max} = 6$.

Hence when we adapt the grid we improve the precision 3 times without spending any additional memory cost. For refinement levels bigger than 10, it is no more possible to handle calculations on a regular grid. Hence we can not have better precision on a regular grid : this reflects the gain of precision achieved by the use of the adaptative algorithm.

Example 2 : A capture basin problem (Zermelo problem)
Let $\mathcal{K} := [-6, 2] \times [-2, 2]$ and $\mathcal{C} := \mathcal{B}(c_0, r)$ with $c_0 = (0, 0)$ and $r = 0.44$. We define the dynamics $f : \boldsymbol{R}^2 \times \mathcal{A} \to \boldsymbol{R}^2$,

$$f(x, a, \theta) = (1 - \beta x_2^2 + a\cos(\theta), a\sin(\theta)),$$

where the constant $\beta = 0.1$, and $\mathcal{A}$ denotes the set $[0, 0.44] \times [0, 2\pi[$.

Our aim is to approximate the capture basin of $\mathcal{C}$ which is the subset of initial states $x \in \mathcal{K}$ for which exists an admissible control $(a, \theta) \in L^\infty([0, +\infty[; \mathcal{A})$ and a finite time

Grid	L_{max}	L^1 error	# cells	gain
adapt	6	8.8E-3	244	16.78
reg	6	8.8E-3	4096	-
adapt	7	5.67E-3	547	29.95
reg	7	5.67E-3	16384	-
adapt	8	4E-3	1108	59.14
reg	8	4E-3	65536	-
adapt	9	3E-3	1939	135.195
reg	9	3E-3	262144	-
adapt	10	2.46E-3	3940	266.14
reg	10	-	1048576	-

Tableau 1. *Gain relative to each refinement level at T=0.11.*

$t \geq 0$ such that the trajectory $y_{x,0}(.)$ evolving with the dynamics f under (a, θ) lives in $\mathcal{K}$ and reaches $\mathcal{C}$ at time t :

$$\text{Capt}_f(\mathcal{C}) := \{x \in \mathcal{K},\ \exists t \geq 0,\ \exists (a,\theta) \in L^\infty(\boldsymbol{R}^+; \mathcal{A}),\ y_{x,0}(\tau) \in \mathcal{K}\ \forall \tau \in [0,t],\ y_{x,0}(t) \in \mathcal{C}\}.$$

We consider the capture basin of $\mathcal{C}$ before time T :

$$\text{Capt}_f(T, \mathcal{C}) := \{x \in \mathcal{K},\ \exists t \in [0,T],\ \exists (a,\theta) \in L^\infty([0,T]; \mathcal{A}),\ y_{x,0}(.) \in \mathcal{K},\ y_{x,0}(t) \in \mathcal{C}\}.$$

It is clear that $T \mapsto \text{Capt}_f(T, \mathcal{C})$ is increasing for inclusion. Moreover, we can prove [BMMZ] that $\lim_{T \to +\infty} \text{Capt}_f(T, \mathcal{C}) = \text{Capt}_f(\mathcal{C})$. Let us set

$$\varphi(x) = 0 \text{ if } x \in \mathcal{C}, \quad \text{and } 1 \text{ otherwise,}$$

and consider the set-valued map defined by

$$\Lambda(x) = \begin{cases} 0 & \text{if } x \in \overset{\circ}{\mathcal{C}}, \\ [0,1] & \text{if } x \in \partial\mathcal{C}, \\ \{1\} & \text{if } x \in \mathcal{K} \setminus \mathcal{C}. \end{cases}$$

Let v_T be the value function of the following control problem :

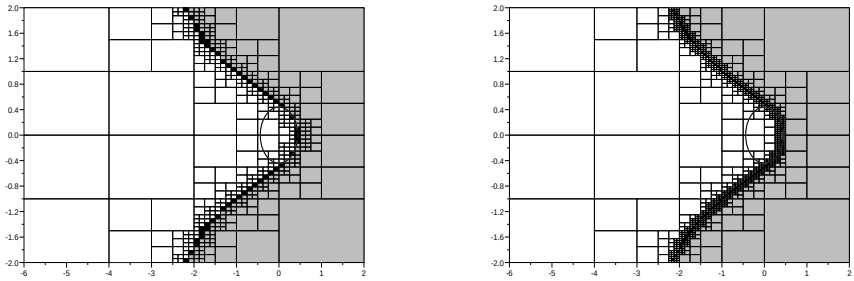
$$\begin{aligned} &\min\{\varphi(y_{x,s}(T)), \\ &\dot{y}_{x,s}(t) = \lambda(t) f(y_{x,s}(t), a(t), \theta(t)), \quad y_{x,s}(s) = x, \\ &(a(t), \theta(t)) \in \mathcal{A} \ \ \& \ \ \lambda(t) \in \Lambda(y_{x,s}(t)) \quad \text{for a.e. } t \in (0,T), \\ &y_{x,s}(t) \in \mathcal{K}\ \forall t \in [0,T]. \end{aligned}$$

We use the classical change of variable : $\hat{v}(t,x) = v_T(T-t,x),\ \forall t \in [0,T],\ \forall x \in \mathcal{K}$. Then, following [BMMZ], we have :

$$\text{Capt}_f(T, \mathcal{C}) = \{x \in \mathcal{K},\ v_T(0,x) = 0\} = \{x \in \mathcal{K},\ \hat{v}(T,x) = 0\}.$$

Then as in [BMMZ], in order to approximate $\mathrm{Capt}_f(\mathcal{C})$, we compute an approximation V^n of $\hat{v}(t^n, \cdot)$, with $t^n := n\Delta t$, for n large enough and satisfying the stopping test

$$||V^n - V^{n-1}||_{L^1} := \sum_j \Delta x^1_{min} \Delta x^2_{min} |V^n_j - V^{n-1}_j| \leq 10^{-4}. \tag{14}$$



(a) L_{max}=6, $T = 12.96$, # cells=478 (b) L_{max}=7, $T = 7.18$, # cells=952

Figure 5. *Capture basin of a Zermelo problem.*

In figure 5, we show the graphics we obtain for maximal level of refinement $L_{max} = 6$ (figure 5 (a)) and $L_{max} = 7$ (figure 5 (b)). In the graphics, the black circle is the border of the target $\mathcal{C}$. We give in the following table the gain obtained for each refinement level, the stopping time (i.e time for which the stopping test (14) is fulfilled) and the value of the residual $||V^n - V^{n-1}||_{L^1}$. Notice that when we increase the refinement level L_{max},

Grid	L_{max}	# cells	gain	stopping time	$\|\|V^n - V^{n-1}\|\|_{L^1}$
adapt	5	232	4.41	26.125	3.89 E-7
reg	5	1024	-	26.125	3.89 E-7
adapt	6	478	8.56	12.96	1.82 E-8
reg	6	4096	-	12.96	1.82 E-8
adapt	7	952	17.21	7.187	2.38 E-5
reg	7	16384	-	7.187	2.38 E-5

Tableau 2. *Gain relative to each refinement level with the value of the residual and the stopping time.*

the precision required in the stopping test is reached faster. For example, with $L_{max} = 7$ we obtain a good solution on the adaptative grid at $T = 7.187$ using 952 cells, whereas with $L_{max} = 5$, the stopping test is fulfilled only after $T = 26.125$ and the corresponding regular grid contains 1024 cells. Hence adaptative gridding allows not only to have a better precision using almost the same number of cells but also to handle less calculations

and to save time.

Example3 : A viability kernel problem (consumption problem)
Let $\mathcal{K} = [0,2] \times [0,3]$, $\mathcal{A} = [-\frac{1}{2}, \frac{1}{2}]$ and $f(x,a) = (x_1 - x_2, a),\ \forall x \in \boldsymbol{R}^2,\ \forall a \in \mathcal{A}$.

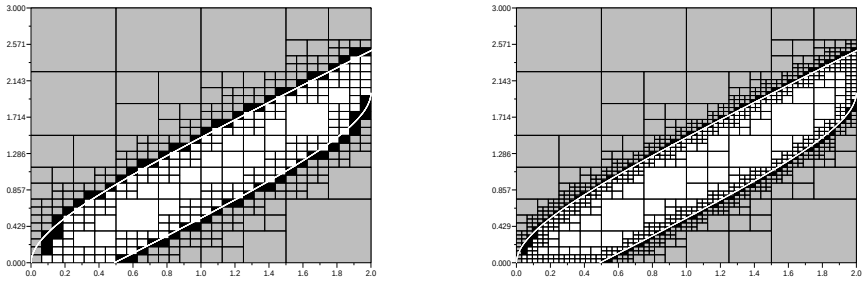
We define the viability kernel associated to $\mathcal{K}$ as the set of initial states $x \in \mathcal{K}$ such that exists a control $a \in L^\infty(\boldsymbol{R}^+; \mathcal{A})$ and a trajectory $y_{x,0}(.)$ (evolving under the control a) which never leaves $\mathcal{K}$,

$$\text{Viab}(\mathcal{K}) := \{x \in \mathcal{K}, \exists a \in L^\infty(\boldsymbol{R}^+; \mathcal{A}),\ y_{x,0}(t) \in \mathcal{K}\ \forall t \geq 0\}.$$

We define also $\text{Viab}(T, \mathcal{K}) := \{x \in \mathcal{K}, \exists a \in L^\infty([0,T], \mathcal{A}),\ y_{x,0}(t) \in \mathcal{K}\ \forall t \in [0,T]\}$. Let $\varphi(x) = 0$ if $x \in \mathcal{K},$ and 1 otherwise. Consider the value function v_T of the control problem (1) associated to the dynamics f, the final cost φ and the set $\mathcal{K}$.

Let $\hat{v}(t,x) = v_T(T-t,x)$. From [BMMZ], we have : $\text{Viab}(T, \mathcal{K}) = \{x \in \mathcal{K},\ \hat{v}(T,x) = 0\}$, and $\text{Viab}(T, \mathcal{K}) \longrightarrow_{T \to +\infty} \text{Viab}(\mathcal{K})$. As in the previous example, in order to approximate $\text{Viab}(\mathcal{K})$, we compute an approximation V^n of $\hat{v}(t^n, \cdot)$ for n satisfying the same stopping test (14).

We show the graphics we obtain for maximal level $L_{max} = 5$ (figure 6 (a)) and $L_{max} = 6$ (figure 6 (b)). In these figures, the white line is the border of the exact viability kernel. Here, the set of controls $\mathcal{A}$ is discretized into $N_a = 2$ controls ($a \in \{-\frac{1}{2}, \frac{1}{2}\}$).



(a)L_{max}=5, $T \approx 256$, # cells=421 (b)L_{max}=6, $T \approx 5$, # cells=916

Figure 6. *Viability kernel of the consumption problem.*

In a previous work [BMMZ], the UB-HJB scheme has been compared to the viability algorithm [SP]. Many numerical examples have been handled on a regular grid (among others the Zermelo problem and the consumption problem) to prove the relevance of UB-HJB in this kind of problems. In fact, UB-HJB provides a much better approximation of these sets (viability kernels, capture basins). Here we continue in the same direction and

improve results given by UB-HJB scheme. The use of the adaptative method allows us to reach a better precision by optimizing the management of the memory.

5. Bibliographie

[B-CD] M. BARDI, I. CAPUZZO-DOLCETTA, « Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations », *Systems and Control : Foundations and Applications. Birkhäuser, Boston, (1997).*

[B] G. BARLES, « Solutions de viscosité des équations de Hamilton-Jacobi », *Mathématiques et applications. Springer, Paris, (1994).*

[BZ] O. BOKANOWSKI, H. ZIDANI, « Anti-dissipative schemes for advection and application to Hamilton-Jacobi-Bellmann equations », *to appear in J. Sci. Comp., (2005).*

[BMMZ] O. BOKANOWSKI, S. MARTIN, R. MUNOS, H. ZIDANI, « An Anti-diffusive scheme for viability problems », *to appear in Appl. Num. Math..*

[BMZ] O. BOKANOWSKI, N. MEGDICH, H. ZIDANI, « On the convergence of a non-monotone scheme for HJB equations », *in preparation.*

[CD-L] I. CAPUZZO-DOLCETTA, P. L. LIONS, « Hamilton-Jacobi equations with state constraints », *Trans. American Math. Soc., 318(1990), 643-685.*

[CL] M. G. CRANDALL, P. L. LIONS, « Viscosity solutions of Hamilton-Jacobi equations », *Trans. American Math. Soc., 277(1983), 1-42.*

[CY1] B. COCKBURN, B. YENIKAYA, « An adaptive method with rigorous error control for the Hamilton-Jacobi equations. Part II : the two-dimensional steady-state case », *App. Num. Math., 52(2005), 175-195.*

[CY2] B. COCKBURN, B. YENIKAYA, « An adaptive method with rigorous error control for the Hamilton-Jacobi equations. Part II : the two-dimensional steady-state case », *J. Comput. ph., 209(2005), 391-405.*

[DL] B. DÉSPRÈS, F. LAGOUTIÈRE, « Contact discontinuity capturing schemes for linear advection and compressible gas dynamics », *J. Sci. Comput., 16(2001), 479-524.*

[FF] M. FALCONE, R. FERRETTI, « Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods », *J. Comput. Phy., 175(2002), 559-575.*

[FG] M. FALCONE, T. GIORGI, « An approximation scheme for evolutive Hamilton-Jacobi equations », *Stochastic Analysis, Control, Optimization and Applications, Birkhäuser, Boston (1999), 289-303.*

[FV] H. FRANKOWSKA, R. B. VINTER, « Existence of neighboring feasible trajectories : applications to dynamic programming for state-constraint optimal control problems », *J. Optim. Theo. Appl., 104(2000), 27-40.*

[FP] H. FRANKOWSKA, S. PLASKACZ, « Semicontinuous solutions of Hamilton-Jacobi-Bellman equations with degenerate state constraints », *J. M. A. A., 251(2000), 818-838.*

[Ga] I. GARGANTINI, « An effective way to represent quadtrees », *Communications of the ACM, 25-12(1982), 905-910*.

[Gr] L. GRUNE, « Adaptative grid generation for evolutive Hamilton-Jacobi-Bellman equations », *Numerical methods for viscosity solutions and applications, World scientific, (2001), 153-172*.

[SP] P. SAINT-PIERRE, « Approximation of viability kernel. », *Appl. Math. Optim., 29(1994), 187-209*.

[S] P. E. SOUGANIDIS, « Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. », *Journal of differential equations, 59(1985), 1-43*.