

Some convergence results for Howard's algorithm

Olivier Bokanowski — Stefania Maroso — Hasnaa Zidani

N° ????

Octobre 2007

Thème NUM



*Rapport
de recherche*

Some convergence results for Howard's algorithm

Olivier Bokanowski ^{*}, Stefania Maroso [†], Hasnaa Zidani [‡]

Thème NUM — Systèmes numériques
Projets Commands

Rapport de recherche n° ???? — Octobre 2007 — 29 pages

Abstract: This paper deals with convergence results of Howard's algorithm for the resolution of $\min_{a \in \mathcal{A}} (B^a x - b^a) = 0$ where B^a is a matrix, b^a is a vector (possibly of infinite dimension), and \mathcal{A} is a compact set. We show a global super-linear convergence result, under a monotonicity assumption on the matrices B^a .

In the particular case of an obstacle problem of the form $\min(Ax - b, x - g) = 0$ where A is an $N \times N$ matrix satisfying a monotonicity assumption, we show the convergence of Howard's algorithm in no more than N iterations, instead of the usual 2^N bound. Still in the case of obstacle problem, we establish the equivalence between Howard's algorithm and a primal-dual active set algorithm (M. Hintermüller et al., *SIAM J. Optim.*, Vol 13, 2002, pp. 865-888). We also propose an Howard-type algorithm for a "double-obstacle" problem of the form $\max(\min(Ax - b, x - g), x - h) = 0$.

We finally illustrate the algorithms on the discretization of nonlinear PDE's arising in the context of mathematical finance (American option, and Merton's portfolio problem), and for the double-obstacle problem.

Key-words: Howard's algorithm (policy iterations), primal-dual active set algorithm, semi-smooth Newton's method, super-linear convergence, double-obstacle problem, slant differentiability.

^{*} Lab. Jacques-Louis Lions, Université Paris 6 & 7, CP 7012, 175 rue du Chevaleret, 75013 Paris, boka@math.jussieu.fr

[†] Projet Tosca, INRIA Sophia-Antipolis, 2004 route des lucioles, BP 93, 06902 Sophia Antipolis Cedex, Stefania.Maroso@inria.fr

[‡] UMA - ENSTA, 32 Bd Victor, 75015 Paris. Also at projet Commands, CMAP - INRIA Futurs, Ecole Polytechnique, 91128 Palaiseau, Hasnaa.Zidani@ensta.fr

Quelques résultats de convergence pour l'algorithme de Howard

Résumé : Nous étudions des résultats de convergence pour l'algorithme de Howard appliqué à la résolution du problème $\min_{a \in \mathcal{A}} (B^a x - b^a) = 0$ où B^a est une matrice, b^a est un vecteur (éventuellement de dimension infinie), et \mathcal{A} est un ensemble compact. Nous obtenons une convergence surlinéaire, sous une hypothèse de monotonie des matrices B^a .

Dans le cas particulier d'un problème d'obstacle de la forme: $\min(Ax - b, x - g) = 0$ où A est une matrice $N \times N$ monotone, nous prouvons que l'algorithme de Howard converge en N itérations. De plus, nous établissons l'équivalence entre l'algorithme de Howard et la méthode primal-dual (M. Hintermüller et al., *SIAM J. Optim.*, Vol 13, 2002, pp. 865-888). Nous étendons aussi l'algorithme de Howard à la résolution du problème de double obstacle de la forme: $\max(\min(Ax - b, x - g), x - h) = 0$.

Des tests numériques sont aussi réalisés pour l'approximation d'équations aux dérivées partielles provenant de la finance (option américaine, optimisation de portefeuille), et d'un problème de double obstacle.

Mots-clés : algorithme d'Howard (itérations sur les politiques), algorithme primal-dual, méthode de Newton semi-lisse, convergence surlinéaire, problème de double obstacle

1 Introduction

The aim of this paper is to study the convergence of Howard's algorithm (also known as the "policy iteration algorithm") for solving the following problem:

$$\text{Find } x \in X, \quad \min_{a \in \mathcal{A}} (B^a x - b^a) = 0, \quad (1)$$

where \mathcal{A} is a non empty compact set, and for every $a \in \mathcal{A}$, B^a is a monotone matrix and $b^a \in X$ (here either $X = \mathbb{R}^N$ for some $N \geq 1$, or $X = \mathbb{R}^{\mathbb{N}^*}$).

Problems such as (1) come typically from the discretization of Hamilton-Jacobi-Bellman equations [9], or from the discretization of obstacle problems.

In general, two methods are used to solve (1). The first one is a fixed point method called "value iterations algorithm". Each iteration of this method is cheap. However, the convergence is linear and one needs a large number of iterations to get a reasonable error. The second method, called policy iteration algorithm, or Howard's algorithm, was developed by Bellman [4, 5] and Howard [11].

Here, we prove that (when \mathcal{A} is a compact set) Howard's algorithm converges super-linearly. To obtain this result, we use an equivalence with a semi-smooth Newton's method.

Puterman and Brumelle [20] were among the first to analyze the convergence properties for policy iterations with continuous state and control spaces. They showed that the algorithm is equivalent to a Newton's method, but under very restrictive assumptions which are not easily verifiable. Recently, Rust and Santos [22] obtained local and global super-linear convergence results for Howard algorithm applied to a class of stationary infinite-horizon Markovian decision process (under additional regularity assumptions, the authors even obtain up to quadratic convergence; see also [21]).

In the first part of this paper, we give a simple proof for the global super-linear convergence. We recall also that when $X = \mathbb{R}^N$ (for some $N \geq 1$), and when \mathcal{A} is finite, then Howard's algorithm converges in at most $(\text{Card}(\mathcal{A}))^N$ iterations.

In the second part of the paper, we focalize our attention to the finite dimensional obstacle problem. More precisely, we are interested by the following problem (for $N \geq 1$):

$$\text{find } x \in \mathbb{R}^N, \quad \min(Ax - b, x - g) = 0, \quad (2)$$

where A is a monotone $N \times N$ matrix (see Definition (2.2)), b and g in \mathbb{R}^N . This problem can be written in the form of (1), with $\text{Card}(\mathcal{A}) = 2$. We prove that a specific formulation of Howard's algorithm for problem (2) converges in no more than N iterations, instead of the expected 2^N bound. We also show that Howard's algorithm is equivalent to the primal-dual active set method for (2), studied in [10, 6, 12]. Using the equivalence of the algorithms, we also conclude that the primal-dual active set method converges in no more than N iterations.

Let us briefly outline the structure of the paper. In Section 2 some notations and preliminary results are given. In Section 3 we prove the equivalence between Howard's algorithm and a semi-smooth Newton's method, and then the super-linear convergence. Section 4 is devoted to the analysis of the obstacle problem, and the equivalence between

Howard's algorithm and the primal-dual active set method. In section 5 we study a policy iteration algorithm for the following double obstacle problem (for $N \geq 1$):

$$\text{find } x \in \mathbb{R}^N, \quad \max(\min(Ax - b, x - g), x - h) = 0.$$

and give also a convergence result in finite number of iterations.

Finally in Section 6 we present some numerical applications. In the context of mathematical finance, the above algorithms are applied to the resolution of implicit schemes for non-linear PDE's, in the case of the American option and for Merton's portfolio problem. The convergence for a double-obstacle problem is also illustrated and compared with a more classical "Projected Successive Over Relaxation" algorithm.

2 Notations and preliminaries

In all the sequel, X denotes either the finite dimensional vector space \mathbb{R}^N for some $N \geq 1$, or the infinite dimensional vector space $\mathbb{R}^{\mathbb{N}^*}$. We consider the set of matrices \mathcal{M} defined as follows: $\mathcal{M} := \mathbb{R}^{N \times N}$ (if $X = \mathbb{R}^N$), or $\mathcal{M} := \mathbb{R}^{\mathbb{N}^* \times \mathbb{N}^*}$ (if $X = \mathbb{R}^{\mathbb{N}^*}$). We will also use the notation $\mathbb{I} := \{1, \dots, N\}$ (if $X = \mathbb{R}^N$), or $\mathbb{I} := \mathbb{N}^*$ (if $X = \mathbb{R}^{\mathbb{N}^*}$).

Let (\mathcal{A}, d) be a compact (metric) set. For each $a \in \mathcal{A}$, the notations b^a and B^a will refer respectively to a vector in X and a matrix in \mathcal{M} associated to the variable a .

Throughout all the paper, we fix two real numbers $p, q \in [1, \infty]$ such that $\frac{1}{p} + \frac{1}{q} = 1$, and use the following norms:

- For $x \in X$, $\|x\|_p := \left(\sum_{i \in \mathbb{I}} |x_i|^p \right)^{1/p}$ (if $p < \infty$) and $\|x\|_\infty := \sup_{i \in \mathbb{I}} |x_i|$.
- For $A \in \mathcal{M}$, we denote

$$\|A\|_{q,\infty} := \max_{i \in \mathbb{I}} \left(\sum_{j \in \mathbb{I}} |A_{ij}|^q \right)^{1/q} \quad (\text{if } q < \infty), \quad \|A\|_{\infty,\infty} := \sup_{i,j \in \mathbb{I}} |A_{ij}|,$$

$$\|A\|_{1,p} := \left(\sum_{i \in \mathbb{I}} \left(\sum_{j \in \mathbb{I}} |A_{ij}| \right)^p \right)^{1/p} \quad (\text{if } p < \infty).$$

We also denote

$$\ell^p := \{x \in X, \|x\|_p < \infty\},$$

and

$$\mathcal{M}_{q,\infty} := \{A \in \mathcal{M}, \|A\|_{q,\infty} < \infty\}.$$

For every $x, y \in X$, we use the notation $y \geq x$ if $y_i \geq x_i, \forall i \in \mathbb{I}$. We also denote $x \geq 0$ if $x_i \geq 0, \forall i \in \mathbb{I}$, and $\min(x, y)$ (resp. $\max(x, y)$) denotes the vector with components $\min(x_i, y_i)$ (resp. $\max(x_i, y_i)$).

Also if $(x^a)_{a \in \mathcal{A}}$ is a bounded subset of X , we denote $\min_{a \in \mathcal{A}}(x^a)$ (resp. $\max_{a \in \mathcal{A}}(x^a)$) the vector of components $\min_{a \in \mathcal{A}}(x_i^a)$ (resp. $\max_{a \in \mathcal{A}}(x_i^a)$).

We consider the problem to find $x \in X$, solution of equation (1):

$$\min_{a \in \mathcal{A}} (B^a x - b^a) = 0.$$

Let

$$\mathcal{A}_\infty := \mathcal{A}^N \text{ if } X = \mathbb{R}^N, \text{ or } \mathcal{A}_\infty := \mathcal{A}^{\mathbb{N}^*} \text{ if } X = \mathbb{R}^{\mathbb{N}^*},$$

endowed with the topology such that $\alpha^k \xrightarrow{k \rightarrow +\infty} \alpha$ in \mathcal{A}_∞ , if $\alpha_i^k \xrightarrow{k \rightarrow +\infty} \alpha_i$ for all $i \in \mathbb{I}$.

Remark 2.1 For $\alpha = (a_i)_{i \in \mathbb{I}} \in \mathcal{A}_\infty$, we denote $B(\alpha)$ and $b(\alpha)$ the matrice and vector such that $B_{i,j}(\alpha) := B_{i,j}^{\alpha_i}$ and $b_i(\alpha) = b^{\alpha_i}$. Then we notice that (1) is equivalent to the following equation:

$$\text{find } x \in X, \quad \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha)) = 0, \quad (3)$$

because in (3) the i th row is a minimisation over a_i (the i th component of α) only.

We consider the following assumption:

(H1) $\alpha \in \mathcal{A}_\infty \mapsto B(\alpha) \in \mathcal{M}_{q,\infty}$ and $\alpha \in \mathcal{A}_\infty \mapsto b(\alpha) \in \ell_\infty$ are continuous.

In order to give a general existence result, we use here the concept of "monotone" matrices.

Definition 2.2 We say that a matrix $A \in \mathcal{M}$ is monotone if A has a continuous left inverse¹ and

$$\forall x \in X, \quad Ax \geq 0 \Rightarrow x \geq 0.$$

In the finite dimensional case, A is monotone if and only if A is invertible and $A^{-1} \geq 0$ (componentwise).

In this paper, we also assume that:

(H2) $\forall \alpha \in \mathcal{A}_\infty$, $B(\alpha)$ is monotone.

(H3) In the case when $X = \mathbb{R}^{\mathbb{N}^*}$, we have:

$$\varepsilon_J := \sup_{\alpha \in \mathcal{A}_\infty} \max_{i \geq 1} \left(\sum_{|j-i| \geq J} |B_{ij}(\alpha)|^q \right)^{1/q} \xrightarrow{J \rightarrow +\infty} 0, \quad \text{if } q < \infty,$$

$$\text{or } \varepsilon_J := \sup_{\alpha \in \mathcal{A}_\infty} \max_{i \geq 1, |j-i| \geq J} |B_{ij}(\alpha)| \xrightarrow{J \rightarrow \infty} 0, \quad \text{if } q = \infty.$$

In the finite dimensional case (i.e, $X = \mathbb{R}^N$ for some $N \geq 1$), assumption (H3) is not needed. On the other hand, when $q < \infty$, (H3) is equivalent to $\sup_{a \in \mathcal{A}} \max_{i \geq 1} \left(\sum_{|j-i| \geq J} |B_{ij}^a|^q \right)^{1/q} \xrightarrow{J \rightarrow \infty} 0$.

¹That is, there exists $C \geq 0$, for any $y \in \ell^\infty$, there exists $x \in \ell^p$ such that $Ax = y$ and $\|x\|_p \leq C\|y\|_\infty$.

Remark 2.3 Assumption (H3) is obviously satisfied whenever there exists some $K \geq 0$ such that for every $\alpha \in \mathcal{A}_\infty$, $B(\alpha)$ is a K -band matrix (i.e., $\forall \alpha, B_{ij}(\alpha) = 0$ if $|j - i| > K$).

Remark 2.4 Also we notice that assumption (H1) is satisfied if (for instance) we have:

- (i) for every $i, j \in \mathbb{I}$, the function $a \in \mathcal{A} \mapsto B_{ij}^a$ is continuous,
- (ii) In the case $X = \mathbb{R}^{\mathbb{N}^*}$, assumption (H3) holds, and

$$\eta_I := \sup_{a \in \mathcal{A}} \max_{i \geq I, j \geq I} |B_{ij}^a| \xrightarrow{I \rightarrow +\infty} 0.$$

Howard's algorithm (Ho-1) Howard's algorithm for (3) can be defined as follows:

- Initialize α^0 in \mathcal{A}_∞ ,
- Iterate for $k \geq 0$:
 - (i) find $x^k \in X$ solution of $B(\alpha^k)x^k = b(\alpha^k)$.
If $k \geq 1$ and $x^k = x^{k-1}$, then stop. Otherwise go to (ii).
 - (ii) $\alpha^{k+1} := \operatorname{argmin}_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x^k - b(\alpha))$.
Set $k := k + 1$ and go to (i).

Under assumption (H2), for every $\alpha \in \mathcal{A}_\infty$, the matrix $B(\alpha)$ is monotone, thus the linear system in iteration (i) of Howard's algorithm is well defined and has a unique solution $x^k \in X$.

It is known that Howard's algorithm is convergent [8]. In our setting, we have the following result.

Theorem 2.5 Assume that (H1)-(H3) hold. Then there exists a unique x^* in ℓ^p solution of (3). Moreover, the sequence (x^k) given by Howard's algorithm (Ho-1) satisfies

- (i) $x^k \leq x^{k+1}$ for all $k \geq 0$.
- (ii) $x^k \rightarrow x^*$ pointwisely. (i.e., $\forall i \in \mathbb{I}, \lim_{k \rightarrow \infty} x_i^k = x_i^*$).
- (iii) If \mathcal{A} is finite and $X = \mathbb{R}^N$, (Ho-1) converges in at most $(\operatorname{Card}(\mathcal{A}))^N$ iterations (ie, x^k is a solution for some $k \leq (\operatorname{Card}(\mathcal{A}))^N$).

We shall indeed prove in the next section that we have the stronger global convergence result (Theorem 3.4):

$$\lim_{k \rightarrow \infty} \|x^k - x^*\|_p = 0, \quad \text{and} \quad \lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

Proof. We start by checking uniqueness. Let x and y two solutions of (3), and let $\bar{\alpha} \in \mathcal{A}_\infty$ be a minimizer associated to y :

$$\bar{\alpha} := \operatorname{argmin}_{\alpha \in \mathcal{A}_\infty} (B(\alpha)y - b(\alpha)).$$

Then

$$\begin{aligned} B(\bar{\alpha})y - b(\bar{\alpha}) &= \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)y - b(\alpha)) = 0 \\ &= \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha)) \\ &\leq B(\bar{\alpha})x - b(\bar{\alpha}). \end{aligned}$$

Hence $B(\bar{\alpha})(y - x) \leq 0$, and by using (H2), we get that $y \leq x$. With the same arguments, we prove also that $y \geq x$. Therefore, we have $y = x$ which proves the uniqueness result.

To prove that $(x^k)_k$ is an increasing sequence, we use:

$$\begin{aligned} B(\alpha^{k+1})x^k - b(\alpha^{k+1}) &= \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x^k - b(\alpha)) \\ &\leq B(\alpha^k)x^k - b(\alpha^k) \\ &= 0 \\ &= B(\alpha^{k+1})x^{k+1} - b(\alpha^{k+1}). \end{aligned}$$

Then by the monotonicity assumption (H2) we obtain the result.

We now prove that x^k converges pointwisely towards some $x^* \in \ell^p$. By the step (i) of the Howard algorithm we have

$$\begin{aligned} \|x^k\|_p &\leq \|B^{-1}(\alpha^k)b(\alpha^k)\|_p \\ &\leq \max_{\alpha \in \mathcal{A}_\infty} \|B^{-1}(\alpha)\|_{1,p} \|b(\alpha)\|_\infty. \end{aligned} \quad (4)$$

Since $\alpha \mapsto B(\alpha)$ is continuous and for every $\alpha \in \mathcal{A}_\infty$, $B(\alpha)$ is invertible, then $\alpha \mapsto B^{-1}(\alpha)$ is also continuous. Moreover, $\alpha \mapsto b(\alpha)$ is continuous, and \mathcal{A}_∞ is a compact. This implies, with the inequality (4), that $(x^k)_k$ is bounded in ℓ^p . Hence x^k converges pointwisely toward some $x^* \in \ell^p$. Let us show that x^* is the solution of (3).

Define the function F for $x \in X$ by: $F(x) := \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha))$, and let $F_i(x)$ be the i th component of $F(x)$, i.e.

$$F_i(x) = \min_{\alpha \in \mathcal{A}_\infty} [B(\alpha)x - b(\alpha)]_i.$$

In the case when $X = \mathbb{R}^N$ it is obvious that $\lim_{k \rightarrow +\infty} F_i(x^k) = F_i(x^*)$. Now consider the case when $X = \mathbb{R}^{\mathbb{N}^*}$ and assume that $q < \infty$ (the same following arguments hold also for $q = +\infty$). For a given $i \geq 1$, and for any $J \geq 1$, we have:

$$\begin{aligned} |F_i(x^k) - F_i(x^*)| &\leq \max_{\alpha \in \mathcal{A}} \sum_{j \geq 1} |B_{ij}(\alpha)| |x_j^k - x_j^*| \\ &\leq \max_{\alpha \in \mathcal{A}} \left(\sum_{|j-i| \geq J} |B_{ij}(\alpha)|^q \right)^{1/q} \left(\sum_{|j-i| \geq J} |x_j^k - x_j^*|^p \right)^{1/p} \\ &\quad + \max_{\alpha \in \mathcal{A}} \sum_{|j-i| < J} |B_{ij}(\alpha)| |x_j^k - x_j^*| \\ &\leq \varepsilon_J \|x^k - x^*\|_p + \varepsilon_1 \sum_{|j-i| < J} |x_j^k - x_j^*|, \end{aligned}$$

where ε_J , ε_1 are defined as in (H3). Hence

$$\limsup_{k \rightarrow \infty} |F_i(x^k) - F_i(x^*)| \leq \varepsilon_J (\max_k \|x^k\|_p + \|x^*\|_p), \quad (5)$$

for all $J \geq 0$. By using (H3), we conclude that:

$$\lim_{k \rightarrow +\infty} F_i(x^k) = F_i(x^*), \quad \forall i \geq 1.$$

By the compactness of \mathcal{A} and using a diagonal extraction argument, there exists a subsequence of $(\alpha^k)_k$ denoted α^{ϕ_k} , that converges toward some $\alpha \in \mathcal{A}_\infty$. Furthermore, we have:

$$|[B(\alpha^k)x^k]_i - [B(\alpha)x^*]_i| \leq 2\varepsilon_J + \sum_{|j-i| < J} |B(\alpha^k)_{ij}x_j^k - B(\alpha)_{ij}x_j^*|,$$

and with assumption (H3), it yields: $\lim_{k \rightarrow \infty} (B(\alpha^k)x^k)_i - (B(\alpha)x^*)_i = 0$. Passing to the limit in $(B(\alpha^{\phi_k})x^{\phi_k} - b(\alpha^{\phi_k}))_i = 0$, we deduce that $(B(\alpha)x^* - b(\alpha))_i = 0$, for all $i \in \mathbb{I}$. On the other hand we have also

$$\begin{aligned} F_i(x^*) &= \lim_{k \rightarrow \infty} F_i(x^{\phi_k-1}) \\ &= \lim_{k \rightarrow +\infty} (B(\alpha^{\phi_k})x^{\phi_k-1} - b(\alpha^{\phi_k}))_i \\ &= [B(\alpha)x^* - b(\alpha)]_i. \end{aligned}$$

Hence $F_i(x^*) = 0$, which concludes the proof of (ii) and implies also that x^* is a solution of (3).

To prove (iii) we first notice that there is at most $(\text{Card}(\mathcal{A})^N)$ different variables $\alpha \in \mathcal{A}_\infty$. Then there exist two indices k, ℓ such that $0 \leq k < \ell \leq (\text{Card}(\mathcal{A})^N)$ and $\alpha^k = \alpha^\ell$ (α^k and α^ℓ being respectively the k th and ℓ th iterate of the Howard's algorithm). Hence $x^\ell = x^k$, and since $x^k \leq x^{k+1} \leq \dots \leq x^\ell$ we obtain also that $x^k = x^{k+1}$. Therefore, Howard's algorithm stops at the $(k+1)$ th iteration, and $x^{k+1} = x^k$ is the solution of (3). \square

3 Superlinear convergence

First let us prove that Howard's algorithm for problem (3) is a semi-smooth Newton's method applied to find the zero of the function $F : \ell^p \rightarrow \ell^\infty$ defined by

$$F(x) := \min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha)). \quad (6)$$

For $k \geq 0$, by definitions of α^{k+1} and x^k (the k th iterate of the Howard's algorithm), we have

$$B(\alpha^{k+1})x^k - b(\alpha^{k+1}) = F(x^k), \quad \text{and} \quad B(\alpha^{k+1})x^{k+1} - b(\alpha^{k+1}) = 0.$$

Therefore, $B(\alpha^{k+1})(x^k - x^{k+1}) = F(x^k)$, and thus

$$x^{k+1} = x^k - B(\alpha^{k+1})^{-1}F(x^k). \quad (7)$$

The equation (7) can be interpreted as an iteration of a semi-smooth Newton's method, where $B(\alpha^{k+1})$ plays the role of a derivative of F at point x^k .

In order to prove the superlinear convergence, we shall prove that F is slantly differentiable in the sense of [10, Definition 1].

Definition 3.1 (Slant differentiability) *Let Y and Z be two Banach spaces. A function $\mathcal{F} : Y \rightarrow Z$ is said slantly differentiable in an open set $U \subset Y$ if there exists a family of mappings $G : U \rightarrow \mathcal{L}(Y, Z)$ such that*

$$\mathcal{F}(x+h) = \mathcal{F}(x) + G(x+h)h + o(h)$$

as $h \rightarrow 0$, $\forall x \in U$. G is called a slanting function for \mathcal{F} in U .

Let us denote $\alpha(x)$ an optimal minimizer associated to $F(x)$, i.e.,

$$\alpha(x) := \operatorname{argmin}_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha)),$$

and let \mathcal{A}_∞^x be the set of the optimal minimizers associated to $F(x)$, i.e.,

$$\mathcal{A}_\infty^x := \left\{ \alpha \in \mathcal{A}_\infty, B(\alpha)x - b(\alpha) = B(\alpha(x))x - b(\alpha(x)) \right\}.$$

For every $i \in \mathbb{I}$, we also define the set \mathcal{A}_i^x of the optimal minimizers associated to the i th component of $\min_{\alpha \in \mathcal{A}_\infty} (B(\alpha)x - b(\alpha))$, i.e.,

$$\mathcal{A}_i^x := \left\{ a \in \mathcal{A}, [B^a x - b^a]_i = [B(\alpha(x))x - b(\alpha(x))]_i \right\}.$$

With these notations, it is clear that $\mathcal{A}_\infty^x = \prod_{i \in \mathbb{I}} \mathcal{A}_i^x$ (see remark 2.1).

We first prove the following Lemma.

Lemma 3.2 *Assume that (H1) holds. For every $x \in X$ and for all $i \in \mathbb{I}$,*

$$d(\alpha_i(x+h), \mathcal{A}_i^x) \rightarrow 0 \quad \text{as } h \in \ell^p \text{ and } \|h\|_p \rightarrow 0,$$

where $\alpha_i(x+h)$ denotes the i th component of an optimal minimizer $\alpha(x+h)$.

Proof. Suppose on the contrary that there exists some $\delta > 0$ and a subsequence $h_n \in \ell^p$, with $\|h_n\|_p \rightarrow 0$, such that $d(\alpha_i(x+h_n), \mathcal{A}_i^x) \geq \delta$, $\forall n \geq 0$. Let $K_\delta := \{a \in \mathcal{A}, d(a, \mathcal{A}_i^x) \geq \delta\}$, the function $f : \mathcal{A} \rightarrow \mathbb{R}$ defined by $f(a) := [B^a x - b^a]_i$, and

$$m_\delta := \inf_{a \in K_\delta} f(a).$$

We note that K_δ is a compact set, hence $m_\delta = f(\bar{a})$ for some $\bar{a} \in K_\delta$. In particular $\bar{a} \notin \mathcal{A}_i^x$ and thus $m_\delta = f(\bar{a}) > f(\alpha_i(x))$. On the other hand, $\alpha_i(x+h_n) \in K_\delta$. Thus

$$f(\alpha_i(x+h_n)) - f(\alpha_i(x)) \geq f(\bar{a}) - f(\alpha_i(x)) > 0. \quad (8)$$

Let $C := \max_{\alpha \in \mathcal{A}_\infty} \|B(\alpha)\|_{q,\infty}$. We have that $\|F(y) - F(z)\|_\infty \leq C\|y - z\|_p$ for every $y, z \in \ell^p$. Moreover, $(F(x+h) - F(x))_i = f(\alpha_i(x+h)) - f(\alpha_i(x)) + (B(\alpha(x+h))h)_i$. Hence $f(\alpha_i(x+h)) - f(\alpha_i(x)) \leq 2C\|h\|_p$. Taking $h := h_n \rightarrow 0$ we obtain a contradiction with (8). \square

The previous Lemma leads to the following result.

Theorem 3.3 *We assume that (H1)-(H3) hold. The function F defined in (6) is slantly differentiable on ℓ^p , with slanting function $G(x) = B(\alpha(x))$.*

Proof. In the case when $X = \mathbb{R}^N$ (for some $N \geq 1$), the theorem is a simple consequence from Lemma 3.2. To prove the assertion when $X = \mathbb{R}^{\mathbb{N}^*}$, we proceed as follows. Let x, h be in ℓ^p .

From the definition of F , we have for any $\alpha \in \mathcal{A}_\infty^x$:

$$F(x) + B(\alpha(x+h))h \leq F(x+h) \leq F(x) + B(\alpha)h,$$

and thus

$$\begin{aligned} 0 \leq F(x+h) - F(x) - B(\alpha(x+h))h &\leq (B(\alpha) - B(\alpha(x+h)))h \\ &\leq \|B(\alpha) - B(\alpha(x+h))\|_{q,\infty} \|h\|_p. \end{aligned} \quad (9)$$

Suppose that $q < \infty$ (the arguments are similar when $q = +\infty$), and let $J \geq 1$. We have:

$$\|B(\alpha) - B(\alpha(x+h))\|_{q,\infty} \leq \max_{i \geq 1} \left(\sum_{|j-i| < J} |B_{ij}(\alpha) - B_{ij}(\alpha(x+h))|^q \right)^{1/q} + 2\varepsilon_J.$$

By Lemma 3.2 there exist $\alpha^h \in \mathcal{A}_\infty^x$ such that $d(\alpha_j^h, \alpha_j(x+h)) \rightarrow 0$ as $\|h\|_p \rightarrow 0$, for all $j \leq J$. Thus, by continuity,

$$\lim_{\|h\|_p \rightarrow 0} \max_i \left(\sum_{|j-i| < J} |B_{ij}(\alpha^h) - B_{ij}(\alpha(x+h))|^q \right)^{1/q} = 0.$$

Hence

$$\limsup_{h \rightarrow 0} \|B(\alpha^h) - B(\alpha(x+h))\|_{q,\infty} \leq 2\varepsilon_J.$$

Since the result is true for any $J \geq 1$, letting $J \rightarrow \infty$ and using assumption (H3), we obtain

$$\lim_{h \rightarrow 0} \|B(\alpha^h) - B(\alpha(x+h))\|_{q,\infty} = 0, \quad (10)$$

and this concludes the proof. \square

The slant differentiability of F is useful to obtain the *local* super-linear convergence of the semi-smooth Newton's method defined by $x^{k+1} = x^k - G(x^k)^{-1}F(x^k)$ (see [10, Theorem 1.1]). We will also prove the *global* superlinear convergence of Howard's algorithm by using the concavity of F .

Theorem 3.4 *Assume that (H1)-(H3) are satisfied. Then (3) has a unique solution $x^* \in \ell^p$, and for any initial iterate $\alpha^0 \in \mathcal{A}_\infty$, Howard's algorithm converges globally super-linearly, i.e., $\lim_{k \rightarrow \infty} \|x^k - x^*\|_p = 0$ and*

$$\|x^{k+1} - x^*\|_p = o(\|x^k - x^*\|_p), \quad \text{as } k \rightarrow \infty.$$

Proof. Existence and uniqueness of the solution, as well as the increasing property of the sequence x^k have already been proved in Theorem 2.5.

There remains to show the super-linear convergence. We consider $h_k := x^k - x^*$ and denote $\alpha^{k+1} := \alpha(x^k) = \alpha(x^* + h_k)$. As in the proof of Lemma 3.2, for all $k \geq 0$, we can find $\alpha^{k,*} \in \mathcal{A}^{x^*}$ such that

$$B(\alpha^{k+1}) - B(\alpha^{k,*}) \rightarrow 0 \text{ as } k \rightarrow \infty. \quad (11)$$

Using the concavity of the function F and the fact that $F(x^*) = 0$ we obtain $F(x^k) \leq B(\alpha^{k+1})(x^k - x^*)$ (indeed for any $\alpha \in \mathcal{A}^{x^*}$, i.e. an optimal minimizer associated to x^* , we have $F(x) \leq F(x^*) + B(\alpha)(x - x^*) = B(\alpha)(x - x^*)$). Hence, by monotonicity of $B(\alpha^{k+1})$,

$$\begin{aligned} x^{k+1} &= x^k - B(\alpha^{k+1})^{-1}F(x^k) \\ &\geq x^k - B(\alpha^{k+1})^{-1}B(\alpha^{k,*})(x^k - x^*), \end{aligned}$$

and thus

$$0 \geq x^{k+1} - x^* \geq (I - B(\alpha^{k+1})^{-1}B(\alpha^{k,*}))(x^k - x^*). \quad (12)$$

By Theorem 3.3 and (H1), we obtain $I - B(\alpha^{k+1})^{-1}B(\alpha^{k,*}) \xrightarrow{k \rightarrow +\infty} 0$ (we also use the fact that $B^{-1}(\alpha)$ is bounded on \mathcal{A}). Hence

$$\exists K \geq 0, \quad \forall k \geq K, \quad \|I - B(\alpha^{k+1})^{-1}B(\alpha^{k,*})\| \leq \frac{1}{2}.$$

We first deduce from (12) that $\|x^{k+1} - x^*\|_p \leq \frac{1}{2}\|x^k - x^*\|_p$ and the global convergence. Then using again (12), we obtain

$$0 \geq x^{k+1} - x^* \geq o(x^k - x^*),$$

and this concludes the proof of super-linear convergence. \square

Remark 3.5 *The proof of the super-linear convergence is strongly dependent on the fact that F is concave.*

Remark 3.6 *The assertions of the theorems 2.5 and 3.4 also hold for the problem*

$$\text{find } x \in X, \quad \min_{\alpha \in \prod_{i \in I} \mathcal{A}_i} (B(\alpha)x - b(\alpha)) = 0.$$

where \mathcal{A}_i are non empty compact sets.

Remark 3.7 (quadratic convergence in the case $X = \mathbb{R}^N$) Note that stronger convergence results can be obtained under an additional assumption on the dependance of $f(x, \alpha) := B(\alpha)x - b(\alpha)$ with respect to α (as in [22]). For instance assume that $X = \mathbb{R}^N$, \mathcal{A} is a compact interval of \mathbb{R} , and that for all $1 \leq i \leq N$, $f_i(x, \alpha) = r_i(x)\alpha_i^2 + s_i(x)\alpha_i + t_i(x)$ (i.e. quadratic in α_i), with $r_i(x) > 0$, $\forall x \in \mathbb{R}^N$, and with $r_i(x)$ and $s_i(x)$ Lipschitz functions of x (see the exemple of Section 6.2). In this case, for every $x \in X$, a minimizeer α^x is given by $\alpha_i^x := \operatorname{argmin}_{\alpha_i \in \mathcal{A}} f_i(x, \alpha) = P_{\mathcal{A}}(-\frac{s_i(x)}{2r_i(x)})$ where $P_{\mathcal{A}}$ denotes the projection on the interval \mathcal{A} . Hence in the neighborhood of the solution x^* , we obtain that $\|\alpha^x - \alpha^{x^*}\| \leq C\|x - x^*\|$ (for some $C > 0$). This implies also that $\|B(\alpha^x) - B(\alpha^{x^*})\| \leq C\|x - x^*\|$, and using (12) this leads to a global quadratic convergence result.

4 The obstacle problem, and link with the primal-dual active set strategy

We consider the following equation, called "obstacle problem" (for $N \geq 1$):

$$\text{find } x \in \mathbb{R}^N, \quad \min(Ax - b, x - g) = 0, \quad (13)$$

where $A \in \mathbb{R}^{N \times N}$ and $b, g \in \mathbb{R}^N$ are given.

Equation (13) is a particular case of (1), with $\mathcal{A} = \{0, 1\}$, $B^0 = A$, $B^1 = I_N$ (I_N being the $N \times N$ identity matrix), $b^0 = b$, and $b^1 = g$. Also it is a particular case of (3), with

$$B_{ij}(\alpha) := \begin{cases} A_{ij} & \text{if } \alpha_i = 0 \\ \delta_{ij} & \text{if } \alpha_i = 1, \end{cases} \quad \text{and} \quad b_i(\alpha) := \begin{cases} b_i & \text{if } \alpha_i = 0 \\ g_i & \text{if } \alpha_i = 1, \end{cases} \quad (14)$$

where $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ if $i \neq j$.

Remark 4.1 With the notations (14), assumption (H1) is obviously satisfied since \mathcal{A} is finite, the Howard's algorithm converges in at most 2^N iterations under assumption (H2).

Remark 4.2 In the particular case when $A = \text{tridiag}(-1, 2, -1)$, A is monotone and one can show that (H2) is satisfied. Also, if A is a matrix such that any submatrix of the form $(A_{ij})_{i,j \in I}$ where $I \subset \{1, \dots, N\}$ is monotone, and with $A_{ij} \leq 0 \forall i \neq j$, then (H2) also holds.

Also, if A is a strictly dominant M-matrix of $\mathbb{R}^{N \times N}$ (i.e., $A_{ij} \leq 0 \forall j \neq i$, and $\exists \delta > 0, \forall i, A_{ii} \geq \delta + \sum_{j \neq i} |A_{ij}|$) then (H2) is true (since all $B(\alpha)$ are all strictly dominant M-matrices, and are thus monotone).

We now consider the following specific Howard's algorithm for equation (13):

Algorithm (Ho-2)

Initialize α^0 in $\mathcal{A}^N := \{0, 1\}^N$.

Iterate for $k \geq 0$:

(i) Find $x^k \in \mathbb{R}^N$ s.t. $B(\alpha^k)x^k = b(\alpha^k)$.

If $k \geq 1$ and $x^k = x^{k-1}$ then stop. Otherwise go to (ii).

(ii) For every $i = 1, \dots, N$, take $\alpha_i^{k+1} := \begin{cases} 0 & \text{if } (Ax^k - b)_i \leq (x^k - g)_i \\ 1 & \text{otherwise.} \end{cases}$ Set

$k := k + 1$ and go to (i).

Let us remark that in the case when $(Ax^k - b)_i = (x^k - g)_i$, we make the choice $\alpha_i^{k+1} = 0$. In the following we show that this choice leads to a drastic improvement in the convergence of Howard's algorithm.

Theorem 4.3 *Assume that the matrices $B(\cdot)$ defined in (14) satisfy (H2). Then Howard's algorithm (Ho-2) applied to (13) converges in at most $N + 1$ iterations (i.e, $x^k = x^{k+1}$ for some $k \leq N + 1$). In the case we start with $\alpha_i^0 = 1, \forall i = 1, \dots, N$, the convergence holds in at most N iterations (i.e, $x^k = x^{k+1}$ for some $k \leq N$).*

Note that taking $\alpha_i^0 = 1$, for every $1 \leq i \leq N$, implies that $x^0 = g$ and hence the step $k = 0$ has no cost. The cost of the N iterations really means a cost of N resolutions of linear systems (from $k = 1$ to $k = N$).

Proof of Theorem 4.3. First, remark that $x^1 \geq g$. Indeed,

- if $\alpha_i^1 = 1$, then we have $(x^1 - g)_i = 0$ (by definition of x^1).
- if $\alpha_i^1 = 0$, then $(Ax^0 - b)_i \leq (x^0 - g)_i$. Furthermore one of the two terms $(Ax^0 - b)_i$ or $(x^0 - g)_i$ is zero, by definition of x^0 . Hence $(x^0 - g)_i \geq 0$, and $(x^1 - g)_i \geq 0$.

We also know, by theorem 2.5, that $x^{k+1} \geq x^k$. This concludes to:

$$\forall k \geq 1, \quad x^k \geq g. \quad (15)$$

Now if $\alpha_i^k = 0$ for some $k \geq 1$ then $(Ax^k - b)_i = 0$ and by (15) we deduce that $\alpha_i^{k+1} = 0$. This proves that the sequence $(\alpha^k)_{k \geq 1}$ is decreasing in \mathcal{A}^N . Also, it implies that the set of points $I^k := \{i, (Ax^k - b)_i \leq (x^k - g)_i\}$ satisfies

$$I^k \subset I^{k+1}, \quad \text{for } k \geq 0.$$

Since $\text{Card}(I^k) \leq N$, there exists a first index $k \in [0, N]$ such that $I^k = I^{k+1}$, and we have $\alpha^{k+1} = \alpha^{k+2}$. In particular, $F(x^{k+1}) = B(\alpha^{k+2})x^{k+1} - b(\alpha^{k+2}) = B(\alpha^{k+1})x^{k+1} - b(\alpha^{k+1}) = 0$, and thus x^{k+1} is the solution for some $k \leq N$. This makes at most $N + 1$ iterations.

In the case $\alpha_i^0 = 1, \forall i$, we obtain that $(\alpha^k)_{k \geq 0}$ is decreasing. Hence there exists a first index $k \in [0, N]$ such that $\alpha^k = \alpha^{k+1}$, and we obtain $F(x^k) = 0$. This is the desired result. \square

Now, we consider the algorithm (Ho-2') which is a variant of Howard's algorithm (Ho-2) and defined as follows: we start from a given x^0 , and then compute α^1 (by step (ii)) and x^1 (by step (i)), then α^2 and x^2 , and so on, until $x^k = x^{k-1}$. For (Ho-2'), we have a specific result, that will be useful when studying the approximation for American options in Section 6.

Theorem 4.4 Assume that the matrices $B(\cdot)$ defined in (14) satisfy (H2). Assume that x^0 satisfies,

$$\forall i, \quad x_i^0 > g_i \Rightarrow (Bx^0 - b)_i \leq 0. \quad (16)$$

Then the iterates of (Ho-2') satisfy $x^{k+1} \geq x^k$ for all $k \geq 0$, and the convergence is in at most N iterations (i.e., x^k for some $k \geq N$ is solution).

Remark 4.5 Note that the assumption (16) is satisfied in the following cases:

- (i) x^0 is such that $\min(Bx^0 - b^0, x^0 - g) = 0$ for some $b^0 \leq b$.
- (ii) $x^0 \leq g$.

Proof of Theorem 4.4 The only difficulty is to prove that $x^1 \geq x^0$, otherwise the proof is the same as in Theorem 4.3. First in the case $\alpha_i^1 = 1$, we have $x_i^1 = g_i$ and $(Bx^0 - b)_i \geq (x^0 - g)_i$. If $(x^0 - g)_i > 0$ then $(Bx^0 - b)_i > 0$, which contradicts the assumption on x^0 . Hence $x_i^0 \leq g_i$ and thus $x_i^0 \leq x_i^1$. Otherwise in the case $\alpha_i^1 = 0$, we have $(Bx^0 - b)_i < (x^0 - g)_i$ and $(Bx^1 - b)_i = 0$. If $(Bx^0 - b)_i > 0$ then $(x^0 - g)_i > 0$, which contradicts the assumption on x^0 . Hence $(Bx^0 - b)_i \leq 0$. In particular, $(Bx^0)_i \leq (Bx^1)_i$. In conclusion, we have the vector inequality $B(\alpha^1)x^0 \leq B(\alpha^1)x^1$. This implies that $x^0 \leq x^1$ using the monotonicity of $B(\alpha^1)$. \square

Remark 4.6 A similar algorithm as (Ho-2) can be built for the equivalent problem

$$\min(Ax - b, C(x - g)) = 0,$$

where $C = \text{diag}(c_1, \dots, c_N)$ is a diagonal matrix with $c_i > 0$, for $i = 1, \dots, N$.

Moreover, in the particular case when $c_i = c > 0$ for $i = 1, \dots, N$, the Howard's algorithm is equivalent to the primal-dual active set method of Hintermüller, Ito and Kunisch [10] as explained below.

It is well known that, for a given $c > 0$, x is a solution of (3) if and only if there exists $\lambda \in \mathbb{R}_+^N$ such that

$$\begin{cases} Ax - \lambda &= b \\ \mathcal{C}(x, \lambda) &= 0, \end{cases} \quad (17)$$

where

$$\mathcal{C}(x, \lambda) := \min(\lambda, c(x - g)) = \lambda - \max(0, \lambda - c(x - g)).$$

(note also that $\lambda = \mathcal{P}_{[0, +\infty)}(\lambda - c(x - g))$). The idea developed in [10] is to use $\mathcal{C}(x, \lambda) = 0$ as a prediction strategy as follows.

Primal-dual active set algorithm

Initialize $x^0, \lambda^0 \in \mathbb{R}^N$.

Iterate for $k \geq 0$:

(i) Set $\mathcal{I}_k = \{i : \lambda_i^k \leq c(x^k - g)_i\}$, $\mathcal{AC}_k = \{i : \lambda_i^k > c(x^k - g)_i\}$.
If $k \geq 1$ and $\mathcal{I}_k = \mathcal{I}_{k-1}$ then stop. Otherwise go to (ii).

(ii) Solve

$$\begin{aligned} Ax^{k+1} - \lambda^{k+1} &= b, \\ x^{k+1} &= g \text{ on } \mathcal{AC}_k, \quad \lambda^{k+1} = 0 \text{ on } \mathcal{I}_k. \end{aligned}$$

set $k = k + 1$ and return to (i).

The sets \mathcal{I}_k and \mathcal{AC}_k are called respectively the inactive and active sets. Note that the algorithm satisfies at each step, $\lambda^{k+1} = Ax^{k+1} - b$, and $\lambda_i^{k+1} = (Ax^{k+1} - b)_i = 0$ for $i \in \mathcal{I}_k$, and $(x^{k+1} - g)_i = 0$ for $i \in \mathcal{AC}_k$. This is equivalent to say that

$$\tilde{B}x^{k+1} - \tilde{b} = 0, \tag{18}$$

where

$$\begin{aligned} \tilde{B}_{i,\cdot} &:= \begin{cases} A_{i,\cdot} & \text{if } (Ax^k - b)_i \leq c(x^k - g)_i \\ cI_{N,i,\cdot} & \text{otherwise,} \end{cases} \\ \tilde{b}_i &:= \begin{cases} b_i & \text{if } (Ax^k - b)_i \leq c(x^k - g)_i \\ cg_i & \text{otherwise.} \end{cases} \end{aligned}$$

(I_N denotes the $N \times N$ identity matrix.)

Let us consider the equivalent formulation $\min(Ax - b, Cx - Cg) = 0$, with $C = \text{diag}(c, \dots, c)$ (by Remark 4.6), and let $B(\cdot)$ and $b(\cdot)$ be defined as in (14). For $k \geq 0$, if we set $\alpha_i^{k+1} := 0$ for $i \in \mathcal{I}_k$ and $\alpha_i^{k+1} := 1$ otherwise, then we find that α^{k+1} is defined from x^k , as in Howard's algorithm, and we have $\tilde{B} = B(\alpha^{k+1})$, $\tilde{b} = b(\alpha^{k+1})$. Therefore, (18) is equivalent to

$$B(\alpha^{k+1})x^{k+1} - b(\alpha^{k+1}) = 0,$$

and x^{k+1} is defined as in Howard's algorithm applied to $\min(Ax - b, Cx - Cg) = 0$.

Theorem 4.7 *Howard's algorithm (Ho-2) and primal-dual active set algorithm for the obstacle problem are equivalent: if we choose $\lambda^0 := Ax^0 - b$ initially then the sequences (x^k) generated by both algorithms are the same for $k \geq 0$.*

Remark 4.8 *By Theorem 4.3, the primal-dual active set strategy converges also in no more than N iterations (taking $x^0 = g$). This analysis gives an other justification for the small number of iterations needed in the computations done by Ito and Kunisch in [13].*

In the same way, one can see that the Front-Traking algorithm proposed in [1, Chapter 6.5.3] for a particular obstacle problem and based on the primal-dual active set method, is equivalent to the Howard's algorithm.

5 Double obstacle problem

In this section we consider the problem to find $x \in \mathbb{R}^N$ solution of the following "double obstacle problem":

$$\max(\min(Ax - b, x - g), x - h) = 0, \quad (19)$$

(the equation must hold for each component) where A is a given matrix of $\mathbb{R}^{N \times N}$, and b, g, h are in \mathbb{R}^N . We aim to solve (19) by a policy iteration algorithm converging in at most N^2 resolutions of linear systems.

Define the functions F and G on \mathbb{R}^N by:

$$F(x) := \min(Ax - b, x - g), \quad \text{and} \quad G(x) := \max(F(x), x - h) \quad \text{for } x \in \mathbb{R}^N.$$

The problem (19) is then equivalent to find $x \in \mathbb{R}^N$ solution of $G(x) = 0$.

We first re-write the function F as follows:

$$F(x) := \min_{\alpha \in \mathcal{A}^N} (B(\alpha)x - b(\alpha))$$

where $\mathcal{A} = \{0, 1\}$, and $B(\alpha)$ is defined as in (14):

$$B_{ij}(\alpha) := \begin{cases} A_{ij} & \text{if } \alpha_i = 0 \\ \delta_{ij} & \text{if } \alpha_i = 1, \end{cases} \quad \text{and} \quad b_i(\alpha) := \begin{cases} b_i & \text{if } \alpha_i = 0 \\ g_i & \text{if } \alpha_i = 1, \end{cases} \quad (20)$$

with $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. In the sequel, we consider that the matrices $B(\alpha)$ satisfy the assumption (H2) (see remark 4.2). Also we define, for every $\beta \in \mathcal{A}^N$, $F^\beta(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ by

$$F^\beta(x)_i := \begin{cases} F(x)_i, & \text{if } \beta_i = 0 \\ (x - h)_i, & \text{if } \beta_i = 1 \end{cases}, \quad \text{for } x \in \mathbb{R}^N.$$

We obtain easily that for every $x \in \mathbb{R}^N$, we have $G(x) = \max_{\beta \in \mathcal{A}^N} F^\beta(x)$, and the problem (19) is equivalent to:

$$\text{find } x \in \mathbb{R}^N, \quad \max_{\beta \in \mathcal{A}^N} F^\beta(x) = 0. \quad (21)$$

To solve this problem, we consider the following policy iteration algorithm:

Algorithm (Ho-3) Initialize $\beta^0 \in \mathcal{A}^N := \{0, 1\}^N$.

Iterate for $k \geq 0$:

- (i) Find x^k such that $F^{\beta^k}(x^k) = 0$ (resolution at fixed β^k).
- If $k \geq 1$ and $x^k = x^{k-1}$ then stop. Otherwise go to (ii).
- (ii) For every $i = 1, \dots, N$, set

$$\beta_i^{k+1} := \begin{cases} 0 & \text{if } F^{\beta^k}(x^k)_i \geq (x^k - h)_i, \\ 1 & \text{otherwise.} \end{cases}$$

Set $k := k + 1$ and go to (i).

Note that, for every $k \geq 0$, the equation $F^{\beta^k}(x) = 0$ is a simple obstacle problem in the form of (13). The resolution in step (i) of the above algorithm could be performed with the Howard's algorithm (Ho-2) in at most N resolution of linear systems.

On other hand, the step (ii) corresponds to the computation of a specific element of $\operatorname{argmax}_{\beta_i \in \mathcal{A}} (F^\beta(x^k)_i)$.

In order to study this algorithm, we start with the following preliminary results.

Proposition 5.1 *Assume that the matrices $B(\alpha)$ satisfy the assumption (H2). Then the following assertions hold:*

- (i) F is monotone². Moreover, for every $\beta \in \mathcal{A}^N$, F^β is monotone.
- (ii) G is also monotone.

Proof. Through this part, for $x \in \mathbb{R}^N$ we denote $\alpha^x \in \mathcal{A}^N$ a minimizer associated to $F(x)$ defined by:

$$\alpha_i^x := \begin{cases} 0 & \text{if } (Ax - b)_i \leq (x - g)_i \\ 1 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, N.$$

- (i) Let x, y be in \mathbb{R}^N and such that $F(x) \leq F(y)$. We have:

$$B(\alpha^x)x - b(\alpha^x) \leq F(y) \leq B(\alpha^x)y - b(\alpha^x).$$

Taking into account the monotonicity of $B(\alpha^x)$, we get $x \leq y$.

We now prove the monotonicity of F^β , for $\beta \in \mathcal{A}^N$. We assume that $F^\beta(x) \leq F^\beta(y)$.

- For $i \in \{1, \dots, N\}$ such that $\beta_i = 0$, we have $F_i^\beta(x) \leq F_i^\beta(y)$ and thus

$$(B(\alpha^x)x - b(\alpha^x))_i \leq (B(\alpha^y)y - b(\alpha^y))_i \leq (B(\alpha^x)y - b(\alpha^x))_i.$$

Hence $(B(\alpha^x)x)_i \leq (B(\alpha^x)y)_i$.

- On the other hand, for i such that $\beta_i = 1$, we have $(x - h)_i \leq (y - h)_i$ and thus $x_i \leq y_i$.
- Now let $\bar{\alpha} \in \mathcal{A}^N$ be defined by

$$\bar{\alpha}_i := \begin{cases} 0 & \text{if } \beta_i = 0 \text{ and } \alpha_i^x = 0 \\ 1 & \text{otherwise.} \end{cases}$$

We see that if $\beta_i = 0$ and $\alpha_i^x = 0$ then

$$(B(\bar{\alpha})x)_i = (B(\alpha^x)x)_i \leq (B(\alpha^x)y)_i = (B(\bar{\alpha})y)_i,$$

and otherwise,

$$(B(\bar{\alpha})x)_i = x_i \leq y_i = (B(\bar{\alpha})y)_i.$$

Hence $B(\bar{\alpha})x \leq B(\bar{\alpha})y$ and thus $x \leq y$ using the monotonicity of $B(\bar{\alpha})$.

²We say that a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is *monotone* if

$$\forall x, y \in \mathbb{R}^N, \quad f(x) \leq f(y) \Rightarrow x \leq y.$$

(ii) We assume $G(x) \leq G(y)$. In view of (21), let β_y be such that $G(y) = F^{\beta_y}(y)$. In particular,

$$F^{\beta_y}(x) \leq G(y) = F^{\beta_y}(y)$$

Using the monotonicity of F^{β_y} , we obtain $x \leq y$. \square

Now we can prove the main result of this section

Theorem 5.2 *Assume (H2'). Then there exists a unique solution of (19), and algorithm (Ho-3) converges in at most $N + 1$ iterations. It converges in at most N iterations if we start with $\beta_i^0 = 1, \forall i$.*

Proof. We proceed as for the proof of Theorem 4.3. The uniqueness of the solution is obtained by the monotonicity of the function G .

Now we analyse the algorithm. First we note that $F^{\beta^{k+1}}(x^{k+1}) = 0 = F^{\beta^k}(x^k) \leq G(x^k) = F^{\beta^{k+1}}(x^k)$ hence $x^{k+1} \leq x^k$ by monotonicity of $F^{\beta^{k+1}}$.

Then we prove that

$$\forall k \geq 1, \quad x^k \leq h. \quad (22)$$

It suffices to prove that $x^1 - h \leq 0$. In the case $\beta_i^1 = 1$, we have $(x^1 - h)_i = 0$ (by definition of x^1). In the case $\beta_i^1 = 0$, we have $F(x^0)_i \geq (x^0 - h)_i$. Furthermore one of the two terms is zero, by definition of x^0 . Hence $(x^0 - h)_i \leq 0$, and $(x^1 - h)_i \leq 0$. This concludes to (22).

Now if $\beta_i^k = 0$ for some $k \geq 1$ then $F_i(x^k) = 0$ and using that $(x^k - h)_i \leq 0$ we deduce that $\beta_i^{k+1} = 0$. This proves that the sequence $(\beta^k)_{k \geq 1}$ is decreasing in \mathcal{A}^N . The set of points $I^k := \{i, (x^k - h)_i \leq F_i(x^k)\}$ is thus increasing for $k \geq 0$. Since $\text{Card}(I^k) \leq N$, there exists a first index $k \in [0, N]$ such that $I^k = I^{k+1}$, and we have $\beta^{k+1} = \beta^{k+2}$. In particular, $G(x^{k+1}) = F^{\beta^{k+2}}(x^{k+1}) = F^{\beta^{k+1}}(x^{k+1}) = 0$, which means that x^{k+1} is the solution. This makes a maximum of $k + 1$ iterations, bounded by $N + 1$.

In the case $\beta_i^0 = 1, \forall i$, we obtain that $(\beta^k)_{k \geq 0}$ is decreasing. Hence there exists a first index $k \in [0, N]$ such that $\beta^k = \beta^{k+1}$, and we obtain that $G(x^k) = 0$ and a maximum of N iterations.

Finally, this also implies the existence of a solution of $G(x) = 0$, which concludes the proof. \square

Remark 5.3 *Since each resolution of $F^{\beta^k}(x) = 0$ can be solved by Howard's algorithm using at most N iterations (i.e, resolution of linear systems), this means that the global algorithm converges in at most N^2 resolution of linear systems.*

We conclude this section by an extension to a more general max-min problem.

Let $B^{a,b}$ be a set of $N \times N$ real matrices and $c^{a,b}$ a set of vectors of \mathbb{R}^N , and we consider the problem of finding $x \in \mathbb{R}^N$ solution of

$$G(x) := \max_{b \in \mathcal{B}} \left(\min_{a \in \mathcal{A}} (B^{a,b} x - c^{a,b}) \right) = 0, \quad (23)$$

where \mathcal{A} and \mathcal{B} denote two non empty compact sets. We denote by $B(\alpha, \beta)$ the matrices such that $B(\alpha, \beta)_{ij} := B_{ij}^{\alpha_i, \beta_i}$ for all $\alpha = (\alpha_i) \in \mathcal{A}^N$ and $\beta = (\beta_i) \in \mathcal{B}^N$, and $c(\alpha, \beta)_i := c^{\alpha_i, \beta_i}$.

Let $F^\beta(x) := \min_{\alpha \in \mathcal{A}^N} B(\alpha, \beta)x - c(\alpha, \beta)$. Then we can consider the following Howard's algorithm:

Algorithm (Ho-4). Consider a given $\beta^0 \in \mathcal{B}^N$. Iterate for $k \geq 0$:

- (i) Find x^k such that $F^{\beta^k}(x^k) = 0$ (resolution at fixed β^k).
- If $k \geq 1$ and $x^k = x^{k-1}$ then stop. Otherwise, go to (ii).
- (ii) For $i = 1, \dots, N$, set $\beta_i^{k+1} = \operatorname{argmax}_{\beta_i \in \mathcal{A}} (F^{\beta^k}(x^k)_i)$.
- Set $k := k + 1$ and go to (i).

The following result can be obtained using the same argument as used in the paper (the proof is left to the reader).

Theorem 5.4 *Assume that $(\alpha, \beta) \rightarrow B(\alpha, \beta)$ is continuous, and that all matrices $B(\alpha, \beta)$ are monotone. Then the min-max problem (23) admits a unique solution. Furthermore, the algorithm (Ho-4) satisfies $x^k \geq x^{k+1}$. If \mathcal{B} is finite, then the convergence is obtained in at most $\operatorname{Card}(\mathcal{B})^N$ iterations of the algorithm (Ho-4) (i.e., x^k is a solution for some $k \leq \operatorname{Card}(\mathcal{B})^N$).*

6 Applications

6.1 An American option

In this section we consider the problem of computing the price of an American put option in mathematical finance. The price $u(t, s)$ for $t \in [0, T]$ and $s \in [0, S_{\max}]$ is known to satisfy the following non-linear P.D.E (see [15] or [18] for existence and unicity results in the viscosity framework):

$$\min \left(\partial_t u - \frac{1}{2} \sigma^2 s^2 \partial_{ss} u - r s \partial_s u + r u, u - \varphi(x) \right) = 0, \quad t \in [0, T], \quad s \in (0, S_{\max}), \quad (24a)$$

$$u(t, S_{\max}) = 0, \quad t \in [0, T], \quad (24b)$$

$$u(0, s) = \varphi(s), \quad x \in (0, S_{\max}). \quad (24c)$$

where $\sigma > 0$ represents a volatility, $r > 0$ is the interest rate, $\varphi(s) := \max(K - s, 0)$ is the "Payoff" function (where $K > 0$, is the "strike"). The bound S_{\max} should be $S_{\max} = \infty$, for numerical purpose we consider $S_{\max} > 0$, large, but finite.

Let $s_j = jh$ with $h = S_{\max}/N_s$ and $t_n = n\delta t$ with $\delta t = T/N$, where $N \geq 1$ and $N_s \geq 1$ are two integers. Suppose we want to implement the following simple Implicit Euler (IE)

scheme with unknown (U_j^n) :

$$\begin{aligned} \min \left(\frac{U_j^{n+1} - U_j^n}{\delta t} - \frac{1}{2} \sigma^2 s_j^2 \frac{(D^2 U^{n+1})_j}{h^2} - r s_j \frac{D^+ U_j^{n+1}}{h} + r U_j^{n+1}; \right. \\ \left. U_j^{n+1} - g_j \right) = 0, \quad j = 0, \dots, N_s - 1, \quad n = 0, \dots, N - 1, \\ U_{N_s}^{n+1} = 0, \quad n = 0, \dots, N - 1, \\ U_j^0 = g_j, \quad j = 0, \dots, N_s - 1 \end{aligned}$$

where $(D^2 U)_j$ and $(D^+ U)_j$ are the finite differences defined by

$$(D^2 U)_j := U_{j-1} - 2U_j + U_{j+1}, \quad (D^+ U)_j := U_{j+1} - U_j,$$

and with $g_j := \varphi(s_j)$. Note that for $j = 0$ the scheme is simply

$$\min \left(\frac{U_0^{n+1} - U_0^n}{\delta t} + r U_0^{n+1}, U_0^{n+1} - g_0 \right) = 0.$$

(More accurate schemes can be considered, here we focus on the simple (IE) scheme for illustrative purpose mainly.)

It is known that the (IE) scheme converges to the viscosity solution of (24) when $\delta t, h \rightarrow 0$ (one may use for instance the Barles-Souganidis [3] abstract convergence result and monotonicity properties of the scheme). The motivation for using an implicit scheme is for unconditional stability.³

Now for $n \geq 0$, we set $b := U^n$. Thus, the problem to find $x = U^{n+1} \in \mathbb{R}^{N_s}$ (i.e. $x = (U_0^{n+1}, \dots, U_{N_s-1}^{n+1})^T$) can be written equivalently as $\min(Bx - b, x - g) = 0$, where $B = I + \delta t A$ and A is the matrix of \mathbb{R}^{N_s} such that for all $j = 0, \dots, N_s - 1$:

$$(AU)_j = -\frac{1}{2} \sigma^2 s_j^2 \frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} - r s_j \frac{U_{j+1} - U_j}{h} + r U_j,$$

(assuming $U_{N_s} = 0$). Then one can see that B is an M -matrix, hence (H2) is satisfied and we can apply Howard's algorithm and generate a sequence of approximations (x^k) (for a given step n of the (IE) scheme). We choose to apply the algorithm (Ho-2') with starting point $x^0 := U^n$.

For each $n \in [0, \dots, N - 1]$, Howard's algorithm may need up to N_s iterations and the total number of Howard's iterations (for the (IE) scheme) is thus bounded by $N \times N_s$. This can be improved as follows.

Proposition 6.1 *The total number of Howard's iterates (using algorithm (Ho-2')) from $n = 0$ up to $n = N - 1$, is bounded by N_s . On other words, the resolution of (IE) scheme uses no more than N_s resolution of linear systems.*

³An explicit scheme would need a condition of the form $\frac{\delta t}{h^2} \leq \text{const}$ in order to be stable.

Proof. Step 1. First let us notice that for each fixed n , $x = U^{n+1}$ is the solution of $\min(Bx - b, x - g)$ where $b = U^n$. Furthermore, let us show that $U^{n+1} \geq U^n$ by recursion. For $n = 0$, we have $U^1 \geq g = U^0$. For $n \geq 1$, let us assume that $U^n \geq U^{n-1}$. We know that $BU^{n+1} - b \geq 0$, hence $BU^{n+1} \geq U^n \geq U^{n-1}$. On the other hand, $U^{n+1} \geq g$. Hence $x = U^{n+1}$ satisfies $\min(Bx - U^{n-1}, x - g) \geq 0$. This means that U^{n+1} is a super-solution of $\min(Bx - U^{n-1}, x - g) = 0$ (whose solution is U^n). Hence $U^{n+1} \geq U^n$.

Step 2. Then for each fixed n , the iterates (x^k) of the algorithm (Ho-2') applied to $\min(Bx - b, x - g) = 0$ (where $b = U^n$) and starting from $x^0 := U^n$ form an increasing sequence. This is a consequence of Theorem 4.4 applied with $b^0 := U^{n-1}$.

Step 3. Now for each n , the number of Howard's iterations is bounded by $p_n := \text{Card}(j, U_j^{n+1} > g_j) - \text{Card}(j, U_j^n > g_j)$. To see this, let $m_k := \text{Card}(j, x_j^k > g_j) - \text{Card}(j, x_j^0 > g_j)$ for $k \geq 0$ and n fixed. Since $(x^k) \nearrow$, we have also $(m_k) \nearrow$, bounded by p_n . However in the case $m_{k+1} = m_k$, and since $(x^k) \nearrow$, then $\{i, x_i^{k+1} > g_i\} = \{i, x_i^k > g_i\}$. In particular $\alpha^{k+1} = \alpha^k$ and $x^{k+1} = x^k$, and the algorithm has converged at iteration $k \leq p_n$.

As a consequence, the total number of Howard's iterates is bounded by $\sum_{n=0, \dots, N-1} p_n = \text{Card}(j, U_j^N > g_j) - \text{Card}(j, U_j^0 > g_j)$, which is bounded by N_s . \square

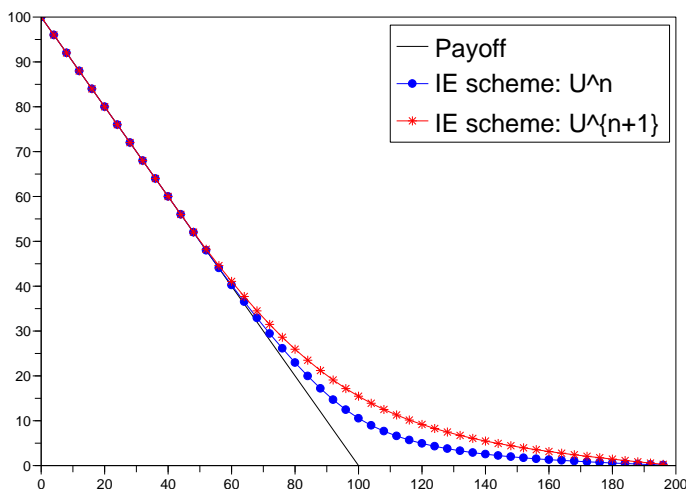


Figure 1: Plot of U_j^n and U_j^{n+1} (time $t_n = 0.2$) with respect to s_j . Parameters: $K = 100$, $\sigma = 1$, $r = 0.1$, $T = 1$, $S_{\max} = 200$, $N_s = 50$, $N = 10$.

This simple remark is particularly interesting for more complex American options, such as American options on two assets (involving a P.D.E in two space dimension). It can be

used to prove that the number of iterates of an Howard algorithm for an implicit scheme can be bounded (in some situations) by the number of points which are inactive in the final solution and thus by the total number of spatial grid points used in the scheme.

We finally mention that in [14] (see also [1]), the Primal-Dual active set algorithm is used for the approximation of an American option (in a Finite element setting), and convergence in a finite number of iterations is also observed even though the matrices involved are not necessarily monotonous matrices.

6.2 Compact control set: Merton's problem

Howard's algorithm can be useful for the computation of the value functions of optimal control problems. We consider the following example, also known as Merton's problem in mathematical finance [17] (see for instance [16] or [7] for recent applications of Howard's algorithm to solve non-linear equations arising in finance). The problem is to find the optimal value for a dynamic portofolio. At a given time t , the holder may invest a part of the portfolio into a risky asset with interest rate $\mu > 0$ and volatility $\sigma > 0$, and the other part into a secure asset with interest rate $r > 0$. The value $u = u(t, s)$, for $t \in [0, T]$ and $s \in [0, S_{\max}]$, is a solution of the following PDE:

$$\min_{\alpha \in \mathcal{A}} \left(\partial_t u - \frac{1}{2} \sigma^2 \alpha^2 s^2 \partial_{ss} u - (\alpha \mu + (1 - \alpha)r) x \partial_s u \right) = 0, \quad t \in [0, T], \quad s \in (0, S_{\max}), \quad (25a)$$

$$u(0, s) = \varphi(s), \quad s \in (0, S_{\max}). \quad (25b)$$

where $\mathcal{A} := [a_{\min}, a_{\max}]$, $S_{\max} = +\infty$, and $K > 0$ is the strike (see for instance Oksendal [17]). Existence and uniqueness results can be obtained using [19].

In general the exact solution is not known. For testing purposes, we consider here the particular case of $\varphi(x) = x^p$ for some $p \in (0, 1)$. In this case, the analytic solution (when $S_{\max} = +\infty$) is known to be $u(t, x) := e^{\alpha_{opt} t} \varphi(x)$ where

$$\alpha_{opt} := \max_{\alpha \in \mathcal{A}} \left(-\frac{1}{2} p(1-p) \alpha^2 + (\alpha \mu + (1-\alpha)r) p \right)$$

(the constant α_{opt} can be computed explicitly since the functional is quadratic in α). For numerical purpose we consider the problem on a bounded interval $[0, S_{\max}]$ with a finite, large, S_{\max} . In this case we need to add a boundary condition at $s = S_{\max}$ for completeness of the problem. Since $\frac{(x^p)'}{x^p} = \frac{p}{x}$, we consider the following mixed boundary condition:

$$\partial_x u(t, S_{\max}) = \frac{p}{S_{\max}} u(t, S_{\max}), \quad t \in [0, T]. \quad (26)$$

A natural IE scheme for the set of equations (25)-(26) is:

$$\min_{\alpha \in \mathcal{A}} \left(\frac{U_j^{n+1} - U_j^n}{\delta t} - \frac{1}{2} \sigma^2 s_j^2 \alpha^2 \frac{U_{j-1}^{n+1} - 2U_j^{n+1} + U_{j+1}^{n+1}}{h^2} - (\alpha\mu + (1-\alpha)r) s_j \frac{U_{j+1}^{n+1} - U_j^{n+1}}{h} \right) = 0,$$

$$j = 0, \dots, N_s, \quad n = 0, \dots, N-1,$$

$$\frac{U_{N_s}^{n+1} - U_{N_s-1}^{n+1}}{h} = \frac{p}{S_{\max}} U_{N_s}^{n+1}, \quad n = 0, \dots, N-1,$$

$$U_j^0 = \varphi(s_j), \quad j = 0, \dots, N_s.$$

Note that for $j = 0$ the scheme simply reads $\frac{U_0^{n+1} - U_0^n}{\delta t} = 0$.

Now for $b := U^n$ given (and for a given time iteration $n \geq 0$), the computation of $x = U^{n+1} \in \mathbb{R}^{N_s+1}$ (i.e, $x = (U_0^{n+1}, \dots, U_{N_s}^{n+1})^T$) is equivalent to solve

$$\min_{\alpha \in \mathcal{A}} (B_\alpha x - b) = 0,$$

where $B_\alpha := I + \delta t A_\alpha$ and A_α is the matrix of $\mathbb{R}^{(N_s+1) \times (N_s+1)}$ such that, for all $j = 0, \dots, N_s - 1$,

$$(A_\alpha U)_j = -\frac{1}{2} \sigma^2 s_j^2 \alpha^2 \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} - (\alpha\mu + (1-\alpha)r) s_j \frac{U_{j+1} - U_j}{h},$$

and

$$(A_\alpha U)_{N_s} = \frac{1}{2} \sigma^2 x_{N_s}^2 \frac{\alpha^2}{h^2} \left(-U_{N_s-1} + \left(1 - \frac{\frac{hp}{S_{\max}}}{1 - \frac{hp}{S_{\max}}}\right) U_{N_s} \right) - (\alpha\mu + (1-\alpha)r) x_{N_s} \frac{p}{S_{\max}} U_{N_s}.$$

We obtain the monotonicity of the matrices B_α under a CFL-like condition on $\delta t, h$.⁴

Remark 6.2 Here the CFL condition comes from the mixed boundary condition and only plays a role on the last row of the matrices B_α (for monotonicity of B_α). Note also that it is of the form $\frac{\delta t}{h} \leq \text{const}$ (when h is small), which is less restrictive than the CFL condition we could obtain with an explicit scheme (it would be of the form $\frac{\delta t}{h^2} \leq \text{const}$).

⁴Chose $h > 0$ and $\delta t > 0$ such that, for h small:

$$\frac{\delta t}{h} \max_{\alpha} \left(\frac{1}{2} \sigma^2 x_{N_s}^2 \alpha^2 \frac{\frac{p}{S_{\max}}}{1 - \frac{hp}{S_{\max}}} + \frac{h^2 p}{S_{\max}} (\alpha\mu + (1-\alpha)r) \right) \leq 1.$$

In view of the expression of $(B_\alpha x)_j$ for a given x , which is quadratic in α , the second step of Howard's algorithm can always be performed analytically (otherwise a minimizing procedure should be considered). This improves considerably the speed for finding the optimal control α (this step has a negligible cputime with respect to the first step of Howard's algorithm where a linear system must be solved).

Remark 6.3 Here we obtain that $\alpha_j := \operatorname{argmin}_{\alpha \in \mathcal{A}} (B_\alpha x)_j$ is defined by

$$\alpha_j = P_{\mathcal{A}} \left(\frac{(\mu - r)d_j^{(1)}}{\sigma^2 s_j d_j^{(2)}} \right),$$

for $0 < j < N_s$, where $d_j^{(1)} := \frac{U_{j+1} - U_j}{h}$, $d_j^{(2)} := \frac{-U_{j-1} + 2U_j - U_{j+1}}{h^2}$ (in the case $d_j^{(2)} \neq 0$), and where $P_{\mathcal{A}}$ is the projection on the interval \mathcal{A} .

We give in Fig. 2 the approximated solution and the associated numerical control obtained with the IE scheme. In this example, the exact optimal control is $\alpha_{opt} = 5$ (constant).

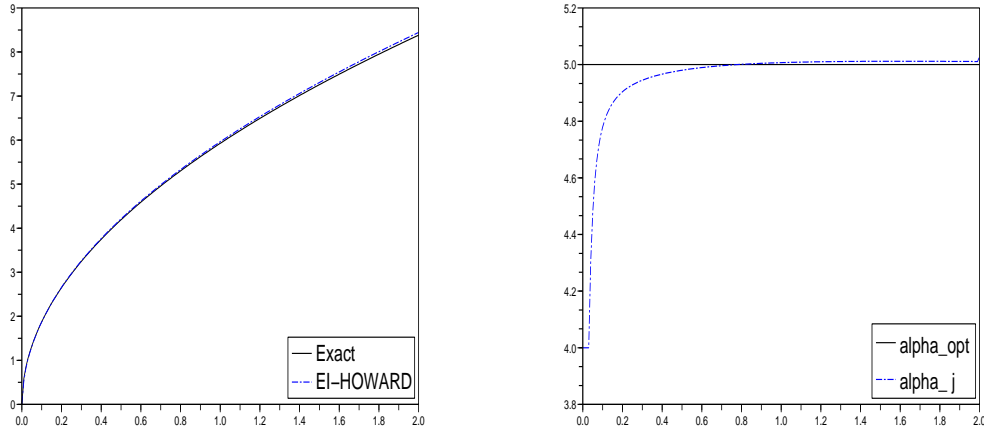


Figure 2: Plot of (U_j^N) (left) and of the discrete optimal control (α_j) at time $t_N = 1$ (right), with respect to s_j . Parameters: $S_{\max} = 2$, $\mathcal{A} = [0.4, 0.6]$, $p = \frac{1}{2}$, $\sigma = 0.2$, $r = 0.1$, $\mu = 0.2$, $T = 1$, and $N_s = 200$, $N = 20$.

Remark 6.4 In general, Howard's algorithm is used on a finite, discretised control set which is an approximation of \mathcal{A} . It thus gives convergence in a finite number of iterations,

yet with an approximated control set. Here we can work directly with the control set \mathcal{A} and take advantage of super linear convergence (at each time step). Furthermore, using Remark 3.7 the convergence can be shown to be quadratic.

Remark 6.5 At each step we have to solve a sparse linear system. Here the system is tridiagonal and thus can be solved in $O(N)$ elementary operations. More generally, multigrid methods can be considered (see for instance [2])

6.3 Double obstacle problem

We consider the following discrete double obstacle problem: find $U = (U_i)_{1 \leq i \leq N}$ in \mathbb{R}^N such that

$$\begin{cases} \max \left(\min \left(-\frac{U_{i-1} - 2U_i + U_{i+1}}{\Delta s^2} + m(s_i), U_i - g(s_i) \right), \right. \\ \qquad \qquad \qquad \left. U_i - h(s_i) \right) = 0, \quad i = 1, \dots, N \\ U_0 = u_\ell, \quad U_{N+1} = u_r, \end{cases} \quad (27)$$

with $u_\ell = 1$, $u_r = 0.8$ (left and right border values), $\Delta s = \frac{1}{N+1}$ and $s_i = i\Delta s$, $m(s) \equiv 0$, $g(s) := \max(0, 1.2 - ((s - 0.6)/0.1)^2)$ and $h(s) := \min(2, 0.3 + ((s - 0.2)/0.1)^2)$.

This problem can easily be reformulated as (19) and with A , a tridiagonal $N \times N$ matrix.

Note that this problem comes from the approximation by finite differences of the following continuous double-obstacle problem: find $u(s)$ such that

$$\begin{cases} \min \left(\left(-u''(s) + m(s), u(s) - g(s) \right), u(s) - h(s) \right) = 0, \quad s \in (0, 1), \\ u(0) = u_g, \quad u(1) = u_d. \end{cases}$$

We compare two algorithms. The first one is an analog of the PSOR algorithm for obstacle problem (see for instance [1]) and that we recall in the appendix. The second one is the Howard algorithm (Ho-3). Note that at each step of (Ho-3), the Howard's algorithm (Ho-2) is also used to compute x^k such that $F^{\beta^k}(x^k) = 0$.

The results are shown in Fig. 3. The left figure shows x^k at iteration $k = 200$ for the PSOR algorithm, where convergence is not yet obtained, and the right figure shows Howard's algorithm at iteration $k = 14$, where convergence has been reached (here this corresponds to a total number of 88 resolutions of linear systems).

Acknowledment. The Authors deeply thank Professor F. Bonnans for interesting discussions and comments on the manuscript.

References

- [1] Y. Achdou and O. Pironneau. *Computational methods for option pricing*. Frontiers in applied mathematics. SIAM, 2005.

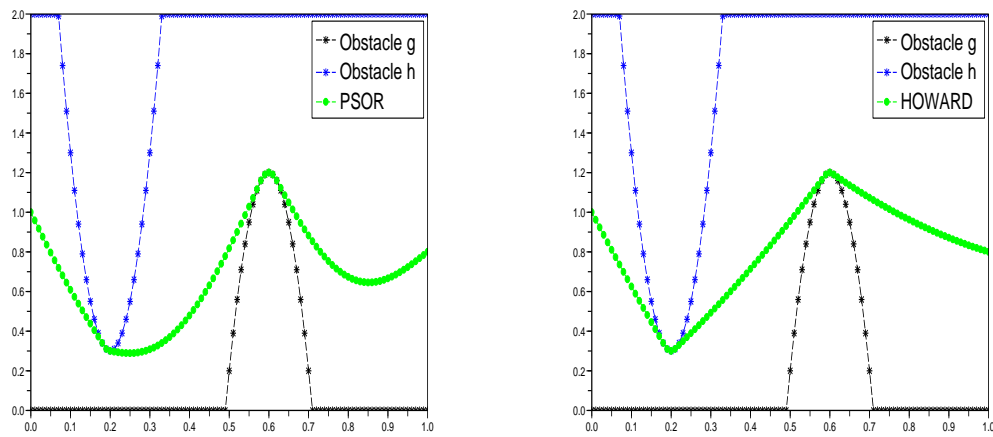


Figure 3: PSOR (left, with $k = 200$ iterations) and Howard's algorithm (right, with $k = 14$ iterations) for the double obstacle problem with $N = 99$. Values U_j^n are plotted with respect to s_j .

-
- [2] M. Akian, J. Menaldi, and A. Sulem. On an investment-consumption model with transaction costs. *SIAM J. Control Optim.*, 34(1):329–364, 1996.
- [3] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully non-linear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.
- [4] R. Bellman. Functional equations in the theory of dynamic programming. v. positivity and quasi-linearity. *Proc. Nat. Acad. Sci. USA*, 41:743–746, 1955.
- [5] R. Bellman. *Dynamic programming*. Princeton University Press, Princeton, 1961.
- [6] M. Bergounioux, M. Hintermüller, and K. Kunisch. Primal dual strategy for constrained optimal control problems. *SIAM J. Control Optim.*, 37:1176–1194, 1999.
- [7] O. Bokanowski, B. Bruder, S. Maroso, and H. Zidani. Numerical approximation of a superreplication problem under gamma constraints. in preparation, 2007.
- [8] J.F. Bonnans and P. Rouchon. *Commande et optimisation de systèmes dynamiques*. Les éditions de l'école polytechnique, Palaiseau, France, 2005.
- [9] J.-Ph. Chancelier, B. Øksendal, and A. Sulem. Combined stochastic control and optimal stopping, and application to numerical approximation of combined stochastic and impulse control. *Tr. Mat. Inst. Steklova*, 237(Stokhast. Finans. Mat.):149–172, 2002.
- [10] M. Hintermüller, K. Ito, and K. Kunish. The primal-dual active set strategy as a semismooth newton method. *SIAM J. Optim.*, 13:865–888, 2002.
- [11] R. A. Howard. *Dynamic programming and Markov processes*. The Technology Press of the MIT. J. Wiley. Cambridge MA, New York, 1960.
- [12] K. Ito and K. Kunish. Optimal control of elliptic variational inequalities. *Appl. Math. Optim.*, 41:343–364, 2000.
- [13] K. Ito and K. Kunish. Semi-smooth newton methods for variational inequalities of the first kind. *M2AN*, 37(1):41–62, 2003.
- [14] K. Ito and K. Kunish. Parabolic variational inequalities: the lagrange multiplier approach. *J. Math. Pures Appl.*, 85(3):415–449, 2006.
- [15] D. Lamberton and B. Lapeyre. *Introduction au calcul stochastique appliqué à la finance*. Ellipses, Paris, second edition, 1997.
- [16] M. Mnif and A. Sulem. Optimal risk control and dividend policies under excess of loss reinsurance. *Stochastics*, 77(5):455–476, 2005.
- [17] B. Øksendal. *Stochastic differential equations*. Universitext. Springer-Verlag, Berlin, sixth edition, 2003.

- [18] H. Pham. Optimal stopping, free boundary, and American option in a jump-diffusion model. *Appl. Math. Optim.*, 35(2):145–164, 1997.
- [19] H. Pham. *Optimisation et Contrôle Stochastique Appliqués à la Finance*. Springer Verlag, to appear.
- [20] M.L. Puterman and S.L. Brumelle. On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4:60–69, 1979.
- [21] M.S. Santos. Accuracy of numerical solutions using the Euler equation residuals. *Econometrica*, 68(6):1377–1402, 2000.
- [22] M.S. Santos and J. Rust. Convergence properties of policy iteration. *SIAM J. Control Optim.*, 42:2094–2115, 2004.

A PSOR algorithm for double-obstacle problem

In the case A is a triangular inferior matrix with $A_{ii} > 0$, the problem has a unique solution given by:

$$\begin{aligned} x_1 &= \min(\max(b_1/A_{11}, g_1), h_1), \\ x_2 &= \min(\max((b_2 - A_{21}x_1)/A_{22}, g_2), h_2), \\ &\vdots \\ x_N &= \min(\max((b_N - A_{N1}x_1 \cdots - A_{N,N-1}x_{N-1})/A_{NN}, g_N), h_N). \end{aligned}$$

Let us denote by $x = q_L(b)$ the result of the above algorithm. In the general case, we can consider the following iterative algorithm (which is a modification of the PSOR algorithm): we first decompose $A = L + U$ where U is the strictly triangular superior part of A .

- start with a given x^0 in \mathbb{R}^N ,
- iterate for $k \geq 0$: $x^{k+1} := q_L(b - Ux^k)$.

The second step is equivalent to find x^{k+1} such that:

$$\max(\min(Lx^{k+1} - (b - Ux^k), x^{k+1} - g), x^{k+1} - h) = 0.$$

Then we have:

Theorem A.1 *Suppose that A is strictly diagonal dominant ($|A_{ii}| > \sum_{j \neq i} |A_{ij}|, \forall i$) and with positive diagonal coefficients ($A_{ii} \geq 0, \forall i$). There exists a unique solution to problem (19), and the adapted PSOR algorithm converges to the solution. Furthermore the convergence is linear.*

Proof. The proof is similar to the proof of the PSOR algorithm (see for instance [1]). One can show that $x \rightarrow q_L(b - Ux)$ is Lipschitz continuous with lipschitz constant bounded by $\rho := \max_i \left(\frac{\sum_{j>i} |A_{ij}|}{A_{ii} - \sum_{i<j} |A_{ij}|} \right)$. Since A is diagonal dominant we get $\rho < 1$. Details are left to the reader. □

□

Contents

1	Introduction	3
2	Notations and preliminaries	4
3	Superlinear convergence	8
4	The obstacle problem, and link with the primal-dual active set strategy	12
5	Double obstacle problem	16
6	Applications	19
6.1	An American option	19
6.2	Compact control set: Merton's problem	22
6.3	Double obstacle problem	25
A	PSOR algorithm for double-obstacle problem	28



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 Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (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