SIAM J. NUMER. ANAL.
Vol. 47, No. 4, pp. 3001–3026

SOME CONVERGENCE RESULTS FOR HOWARD'S ALGORITHM*

OLIVIER BOKANOWSKI[†], STEFANIA MAROSO[‡], AND HASNAA ZIDANI[§]

Abstract. This paper deals with convergence results of Howard's algorithm for the resolution of $\min_{a\in\mathcal{A}}(B^a x - c^a) = 0$, where B^a is a matrix, c^a is a vector, and $\mathcal{A}$ is a compact set. We show a global superlinear convergence result, under a monotonicity assumption on the matrices B^a. Extensions of Howard's algorithm for a max-min problem of the form $\max_{b\in\mathcal{B}}\min_{a\in\mathcal{A}}(B^{a,b}x - c^{a,b}) = 0$ are also proposed. 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 an obstacle problem, we establish the equivalence between Howard's algorithm and a primal-dual active set algorithm [M. Hintermüller, K. Ito, and K. Kunisch, *SIAM J. Optim.*, 13 (2002), pp. 865–888]. The algorithms are illustrated on the discretization of nonlinear PDEs arising in the context of mathematical finance (American option and Merton's portfolio problem), of front propagation problems, and for the double-obstacle problem.

Key words. Howard's algorithm (policy iterations), primal-dual active set algorithm, semi-smooth Newton's method, superlinear convergence, double-obstacle problem, min-max problem

AMS subject classifications. 49M15, 65K10, 90C47, 91B28

DOI. 10.1137/08073041X

1. Introduction.

The main goal of this paper is to study new convergence results of Howard's algorithm for solving the nonlinear problem (for $N \geq 1$)

$$\text{find } x \in \mathbb{R}^N, \quad \min_{\alpha\in\mathcal{A}^N}(B(\alpha)x - c(\alpha)) = 0, \tag{1.1}$$

where $\mathcal{A}$ is a nonempty compact set and, for every $\alpha \in \mathcal{A}^N$, $B(\alpha)$ is an $N \times N$ monotone matrix and $c(\alpha) \in \mathbb{R}^N$ (for $x, y \in \mathbb{R}^N$ the notation $\min(x, y)$ denotes the vector of components $\min(x_i, y_i)$). Problems such as (1.1) come from the discretization of Hamilton–Jacobi–Bellman equations [11, 12, 22, 26], from optimal control problems, and from many other applications. A simple example is the well known obstacle problem

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

where the $N \times N$ matrix A and the vectors b and g are fixed.

In general, two methods are used to solve (1.1). The first one is a fixed point method called the "value iteration 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 Howard's algorithm or the *policy iteration* algorithm, was developed by Bellman [5, 6] and Howard [15] for solving steady infinite-horizon Markovian dynamic programming (MDP) problems. For the problem

*Received by the editors July 16, 2008; accepted for publication (in revised form) June 5, 2009; published electronically August 28, 2009.
http://www.siam.org/journals/sinum/47-4/73041.html
[†]Laboratoire Jacques-Louis Lions, Université Paris 6, 75252 Paris Cedex 05, France, and UFR de Mathématiques, Université Paris 7, Case 7012, 75251 Paris Cedex 05, France (boka@math.jussieu.fr).
[‡]Projet Tosca, INRIA Sophia-Antipolis, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France (Stefania.Maroso@inria.fr).
[§]UMA - ENSTA, 32 Bd Victor, 75015 Paris, France, and Projet Commands, CMAP - INRIA Saclay, Ecole Polytechnique, 91128 Palaiseau, France (Hasnaa.Zidani@ensta.fr).

(1.1), the policy iteration algorithm computes the solution x^* via a sequence of trial values $\{x^k\}$ and policies $\{\alpha^k\}$ under an alternating sequence of policy improvement and policy evaluation steps

$$\alpha^{k+1} = \underset{\alpha \in \mathcal{A}^N}{\arg\min} \left[B(\alpha)x^k - c(\alpha)\right] \qquad \text{(policy improvement)},$$
$$x^{k+1} \in \mathbb{R}^N \text{ is a solution of } B(\alpha^{k+1})x - c(\alpha^{k+1}) = 0 \qquad \text{(policy evaluation)}.$$

The alternating process corresponds roughly to a splitting of the minimum operator and the linear system.

Puterman and Brumelle [33] were among the first to analyze the convergence properties for policy iteration for MDP problems with continuous state and control spaces. In this framework, they showed that the algorithm is equivalent to Newton's method but under very restrictive assumptions which are not easily verifiable. Recently, Santos and Rust [35] obtained local and global superlinear convergence results for Howard's algorithm applied to a class of stationary infinite-horizon MDP problems (under additional regularity assumptions, the authors even obtain up to quadratic convergence; see also [34]).

In the first part of this paper (sections 2 and 3), we analyze Howard's algorithm for a general class of finite dimensional problems in the form of (1.1).

When $\mathcal{A}$ is finite (finite actions), Howard's algorithm converges in at most $(\text{Card}(\mathcal{A}))^N$ iterations. The idea behind this result is that policy iteration has some similarities to the simplex algorithm of linear programming (LP). Just as the simplex algorithm generates a sequence of improving trial solutions to the LP problem, policy iteration generates an improving sequence of decision rules to the nonlinear problem (1.1). Analogous to LP, where the number of vertices is finite, the number of possible policies is also finite and is bounded by $(\text{Card}(\mathcal{A}))^N$. This, together with the fact that the policy iteration method is monotone, imply that the algorithm does not cycle and it converges in a finite number of steps.

Under a monotonicity assumption on the matrices $B(\alpha)$, we prove that when $\mathcal{A}$ is an infinite compact set, Howard's algorithm converges superlinearly. This result is connected to an interpretation of Howard's algorithm as a semismooth Newton method for solving $F(x) = 0$, where $F(x) = \min_{\alpha \in \mathcal{A}^N}(B(\alpha)x - c(\alpha))$.

The concept of semismoothness was introduced by Mifflin [24] for real-valued functions defined on finite dimensional spaces. Qi and Sun [31, 32] extended this notion to mappings between finite dimensional spaces and showed that, although the underlying mapping is in general nonsmooth, Newton's method can be generalized to semismooth equations and converges locally with a superlinear rate to a regular solution (see also [20, 21, 10]).

In the second part of the paper (section 4), we focus the study on the obstacle problem. More precisely, we are interested in the following problem (for $N \geq 1$):

$$\text{(1.2)} \qquad \text{find } x \in \mathbb{R}^N, \quad \min(Ax - b, x - g) = 0,$$

where A is a monotone[1] $N \times N$ matrix and b and g are in $\mathbb{R}^N$. This problem can be written in the form of (1.1), with $\text{Card}(\mathcal{A}) = 2$. We prove that a specific formulation of Howard's algorithm for problem (1.2) converges in no more than N iterations, instead of the expected 2^N bound. Also, we prove that, starting from a particular initial iterate, the number of iterations can be less than N (see Theorem 4.2). Even

[1]A matrix A is said to be *monotone* if it is invertible and $A^{-1} \geq 0$ (componentwise).

though in numerical tests one can observe that Howard's algorithm converges in a few iterations, the number of iterations needed to get the solution is in general dependent on the dimension N. We refer to [34] for a counterexample given by Tsitsiklis, showing that the number of policy iteration steps cannot be bounded by a constant that is independent of N.

On the other hand, we prove that, in the context of (1.2), Howard's algorithm is equivalent to the primal-dual active set method studied in [14, 7, 16]. We also illustrate the theoretical convergence on two numerical applications coming from finance (pricing of American options and Merton's portfolio problem).

In section 5 we study a generalization of the policy iteration algorithm to max-min problems stated as follows (for $N \geq 1$):

$$\text{find } x \in \mathbb{R}^N, \quad \max_{\beta \in \mathcal{B}^N} \min_{\alpha \in \mathcal{A}^N} (B(\alpha, \beta)x - c(\alpha, \beta)) = 0 \tag{1.3}$$

(for assumptions on B and c, see section 5.1). This equation comes also from the discretization of some optimal control problems and especially from the discretization of Isaac's equations associated with the control of differential games. In the context of (1.3), a straightforward policy iteration method would be as follows:

(i) $(\beta^{k+1}, \alpha^{k+1}) := \arg \max_{\beta \in \mathcal{B}^N} \min_{\alpha \in \mathcal{A}^N} \left(B(\alpha, \beta)x^k - c(\alpha, \beta)\right)$.

(ii) Compute $x^{k+1} \in \mathbb{R}^N$, solution of $B(\alpha^{k+1}, \beta^{k+1})x - c(\alpha^{k+1}, \beta^{k+1}) = 0$.

This algorithm can be shown to be equivalent to a Newton-like method for solving (1.3). However, since the max-min operator is neither convex nor concave, this algorithm may not converge (see Remark 5.8).

In this paper, we give other extensions of the policy iteration method which are globally convergent (see Algorithms Ho-3 and Ho-4), are computationally feasible, and are not Newton-like methods. Numerical tests are performed to solve a max-min problem coming from front propagation with mean curvature motion [19]. We also give some numerical comments on the double-obstacle problem (of the form $\max(\min(Ax - b, x - g), x - h) = 0$).

2. Notations and preliminaries. Let N be a fixed integer. We denote by $\mathcal{M}$ the set of real-valued $N \times N$ matrices, and $\mathbb{I}$ the set $\{1, \ldots, N\}$. Throughout the paper, for $x \in \mathbb{R}^N$ and $A \in \mathcal{M}$, we use the following usual norms:

$$\|x\| = \sup_{i \in \mathbb{I}} |x_i|, \qquad \|A\| := \sup_{i \in \mathbb{I}} \sum_{j \in \mathbb{I}} |A_{ij}|.$$

For every $x, y \in \mathbb{R}^N$, the notation $y \geq x$ means that $y_i \geq x_i\ \forall i \in \mathbb{I}$. We also denote by $\min(x, y)$ (resp., $\max(x, y)$) the vector with components $\min(x_i, y_i)$ (resp., $\max(x_i, y_i)$). Moreover, if $(x^a)_{a \in \mathcal{A}}$ is a bounded subset of $\mathbb{R}^N$, we denote by $\min_{a \in \mathcal{A}}(x^a)$ (resp., $\max_{a \in \mathcal{A}}(x^a)$) the vector of components $\min_{a \in \mathcal{A}}(x_i^a)$ (resp., $\max_{a \in \mathcal{A}}(x_i^a)$).

Let $(\mathcal{A}, d)$ be a nonempty compact metric set. For each $\alpha \in \mathcal{A}^N$, the notations $c(\alpha)$ and $B(\alpha)$ will refer, respectively, to a vector in $\mathbb{R}^N$ and a matrix in $\mathcal{M}$ associated with the variable α.

We consider the problem of finding $x \in \mathbb{R}^N$ (for a fixed $N \geq 1$), solution of (1.1); i.e.,

$$\text{find } x \in \mathbb{R}^N, \quad \min_{\alpha \in \mathcal{A}^N} \left(B(\alpha)x - c(\alpha)\right) = 0.$$

Throughout the paper, it is furthermore assumed that

$$\text{for all } \alpha = (\alpha_1, \dots, \alpha_N) \in \mathcal{A}^N \ \forall i, j, \quad B_{ij}(\alpha) \text{ depends only on } \alpha_i.$$

Let us point out that the min operation in (1.1) is understood componentwise. Hence by setting, for every $a \in \mathcal{A}$,

$$\alpha^a := (a, \dots, a)^\mathsf{T} \in \mathcal{A}^N, \quad B^a := B(\alpha^a), \quad \text{and } c^a = c(\alpha^a),$$

we get an equivalent reformulation of (1.1) as

$$\min_{a \in \mathcal{A}} \left(B^a x - c^a \right) = 0. \tag{2.1}$$

(Conversely, if $(B^a)_{a \in \mathcal{A}}$ is given, for $\alpha = (\alpha_1, \dots, \alpha_N)^\mathsf{T} \in \mathcal{A}^N$, we define $B(\alpha)$ by setting $B(\alpha)_{i,j} := B^{\alpha_i}_{i,j}$).

Throughout the paper, we use an important "monotonicity" assumption on the matrices (recall that a matrix $A \in \mathcal{M}$ is said to be *monotone* if and only if A is invertible and $A^{-1} \geq 0$ (componentwise)). More precisely, we assume the following:

(H1) For every $\alpha \in \mathcal{A}^N$, the matrix $B(\alpha)$ is monotone.

(H2) When $\mathcal{A}$ is an *infinite* compact set, the functions $\alpha \in \mathcal{A}^N \longmapsto B(\alpha) \in \mathcal{M}$ and $\alpha \in \mathcal{A}^N \longmapsto c(\alpha) \in \mathbb{R}^N$ are continuous.

Assumption (H2) is satisfied whenever the functions $a \in \mathcal{A} \longmapsto B^a_{ij}$ are continuous for every $i, j \in \mathbb{I}$.

Example 1. A simple example of problem (1.1) is the well known obstacle problem

$$\text{find } x \in \mathbb{R}^N, \quad \min(Ax - b, x - g) = 0, \tag{2.2}$$

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

$$\mathcal{A} = \{0, 1\}, \ B^0 = A, \ B^1 = I_N, \ c^0 = b, \ c^1 = g$$

(I_N being the $N \times N$ identity matrix), it is clear that (2.2) is a particular case of (2.1). Also, it can be set in the form of $\min_{\alpha \in \mathcal{A}^N}(B(\alpha)x - c(\alpha)) = 0$, with

$$B_{ij}(\alpha) := \begin{cases} A_{ij} & \text{if } \alpha_i = 0, \\ \delta_{ij} & \text{if } \alpha_i = 1 \end{cases} \qquad \text{and} \quad c_i(\alpha) := \begin{cases} b_i & \text{if } \alpha_i = 0, \\ g_i & \text{if } \alpha_i = 1, \end{cases} \tag{2.3}$$

where $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ if $i \neq j$. In the particular case when $A =$ *tridiag*$(-1, 2, -1)$, A is monotone and one can show that (H1) 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 $A_{ij} \leq 0 \ \forall i \neq j$, then assumption (H1) holds.

When A is a strictly dominant M-matrix of $\mathbb{R}^{N \times N}$ (i.e., $A_{ij} \leq 0 \ \forall j \neq i$, and $\exists \delta > 0 \ \forall i, \ A_{ii} \geq \delta + \Sigma_{j \neq i} |A_{ij}|$), then (H1) is true (since all $B(\alpha)$ are all strictly dominant M-matrices and are thus monotone).

Example 2. Let us consider a controlled Markov chain on a discrete state space $\mathcal{G}$. Let $\{X^q, q \geq 0\}$ be the states of the Markov chain at time q, with given transition probabilities denoted by $p(x, y|a)$, where $p(x, y|a)$ is the probability to move from x to y under the action of a ($a \in \mathcal{A}$ is the canonical control variable). Let h be the time step and $\mathbb{E}^u_x$ be the conditional expectation given that $X^0 = x$ and an admissible control u is used.

We consider the optimal control problem (see [22])

$$\vartheta(x_k) = \min_{u=(u_q),\ u_q\in\mathcal{A}} h\mathbb{E}^u_{x_k}\left[\sum_{q\geq 0}\ell(X^q,u_q)(1+\lambda h)^{-q-1}\right], \tag{2.4}$$

where $x_k = X^0$ is the first state and $(u_q)_{q\geq 0}$ represents the control action for the chain at discrete time q. Then the dynamic programming equation for the controlled chain $\{X^q, q\geq 0\}$ and the cost (2.4) yields

$$\vartheta(x_k) = (1+\lambda h)^{-1}\min_{a\in\mathcal{A}}\left[h\ell(x_k,a) + \sum_{x_l\in\mathcal{G}} p(x_k,x_l|a)\vartheta(x_l)\right] \tag{2.5}$$

for $x_k \in \mathcal{G}$. This can be rewritten in the form

$$(1+\lambda h)V = \min_{a\in\mathcal{A}}\left[C(a) + \sum_{x_l\in\mathcal{G}}\mathbb{P}(a)V\right], \tag{2.6}$$

where $V := (\vartheta(x_k))_{x_k\in\mathcal{G}}$. Similar discrete control problems can also be obtained by discretization of the dynamic programming principle for continuous stochastic control problems; see, for instance, [28, 22, 12, 26, 8].

HOWARD'S ALGORITHM Ho-1. For problem (1.1), the algorithm is as follows:

Initialize α^0 in $\mathcal{A}^N$,

Iterate for $k\geq 0$:

(i) find $x^k\in\mathbb{R}^N$ solution of $B(\alpha^k)x^k = c(\alpha^k)$.

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

(ii) $\alpha^{k+1} := \operatorname{argmin}_{\alpha\in\mathcal{A}^N}\big(B(\alpha)x^k - c(\alpha)\big)$.

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

Under assumption (H1), for every $\alpha\in\mathcal{A}^N$, the matrix $B(\alpha)$ is monotone, and thus the linear system in iteration (i) of Howard's algorithm is well defined and has a unique solution $x^k\in\mathbb{R}^N$.

In order to compute α^{k+1} (according to step (ii) of Howard's algorithm Ho-1) in a fast way, it is useful to use the equivalence of the two formulations (1.1) and (2.1) and then remark that α^{k+1} can also be given by

$$\alpha^{k+1} = \operatorname*{argmin}_{a\in\mathcal{A}}(B^a x^k - c^a).$$

For instance, in the case of obstacle problem (2.2), step (ii) of the algorithm amounts to taking

$$\begin{aligned}&\alpha_i^{k+1} = 0 \quad\text{if}\quad [Ax^k - b]_i < x_i^k - g_i, \qquad \alpha_i^{k+1} = 1 \quad\text{if}\quad [Ax^k - b]_i > x_i^k - g_i,\\ &\alpha_i^{k+1}\in\{0,1\}\quad\text{if}\quad [Ax^k-b]_i = x_i^k - g_i.\end{aligned}$$

THEOREM 2.1. *Assume that* (H1)–(H2) *hold. Then there exists a unique x^* in $\mathbb{R}^N$ solution of* (1.1). *Moreover, the sequence* (x^k) *given by Howard's algorithm Ho*-1 *satisfies the following:*

(i) $x^k\leq x^{k+1}$ *for all* $k\geq 0$.

(ii) *When $\mathcal{A}$ is finite, the algorithm converges in at most* $(\mathrm{Card}(\mathcal{A}))^N$ *iterations (i.e., $x^k = x^*$ for some $k\leq(\mathrm{Card}(\mathcal{A}))^N$).*

(iii) *If $\mathcal{A}$ is an infinite compact set, then $x^k\to x^*$ when k tends to $+\infty$.*

Actually, we shall prove in the next section that we have the stronger convergence result (Theorem 3.4):

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0 \qquad \text{for any } x^0 \in \mathbb{R}^N.$$

The convergence of Howard's algorithm is well known; see, for instance, [9]. The statement (i) of Theorem 2.1 is proved in the literature under strong assumptions; see, for instance, [35]. We give here a very simple proof based on the monotonicity assumption (H1).

Proof of Theorem 2.1. We first start by proving the uniqueness of x^*, while the existence of x^* will be shown separately by proving (ii) and (iii).

Consider $x, y \in \mathbb{R}^N$ two solutions of (2.1), and let $\alpha \in \mathcal{A}^N$ be a minimizer associated with y:

$$\alpha^y := \arg\min_{\alpha\in\mathcal{A}^N} \left(B(\alpha)y - c(\alpha)\right).$$

Then we have

$$\begin{aligned} B(\alpha^y)y - c(\alpha^y) = \min_{\alpha\in\mathcal{A}^N} (B(\alpha)y - c(\alpha)) &= 0 \\ &= \min_{\alpha\in\mathcal{A}^N} (B(\alpha)x - c(\alpha)) \\ &\le B(\alpha^y)x - c(\alpha^y), \end{aligned} \tag{2.7}$$

which gives that $B(\alpha^y)(y - x) \le 0$, and since $B(\alpha^y)$ is monotone, we get that $y \le x$. In the same way, we also have $y \ge x$. Therefore, y and x coincide.

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

$$\begin{aligned} B(\alpha^{k+1})x^k - c(\alpha^{k+1}) &= \min_{\alpha\in\mathcal{A}^N} (B(\alpha)x^k - c(\alpha)) \\ &\le B(\alpha^k)x^k - c(\alpha^k) \\ &= 0 \\ &= B(\alpha^{k+1})x^{k+1} - c(\alpha^{k+1}). \end{aligned}$$

Then by the monotonicity assumption (H1) we obtain the result.

(ii) Assume that $\mathcal{A}$ is finite. Hence there are at most $(\mathrm{Card}(\mathcal{A})^N)$ different variables $\alpha \in \mathcal{A}^N$. Then there exist two indices k, ℓ such that $0 \le k < \ell \le (\mathrm{Card}(\mathcal{A})^N)$ and $\alpha^k = \alpha^\ell$ (α^k and α^ℓ being, respectively, the kth and ℓth iterate of Howard's algorithm). Hence $x^\ell = x^k$, and since $x^k \le x^{k+1} \le \cdots \le x^\ell$, we obtain also that $x^k = x^{k+1}$. Therefore, Howard's algorithm stops at the $(k+1)$th iteration. This proves that $x^{k+1} = x^k$ is a solution of (1.1), since $F(x^k) = B(\alpha^{k+1})x^k - c(\alpha^{k+1}) = B(\alpha^{k+1})x^{k+1} - c(\alpha^{k+1}) = 0$.

(iii) We now consider the case when $\mathcal{A}$ is an infinite compact set. By the step (i) of the algorithm we have

$$\begin{aligned} \|x^k\| &\le \|B^{-1}(\alpha^k)c(\alpha^k)\| \\ &\le \max_{\alpha\in\mathcal{A}^N} \|B^{-1}(\alpha)\|\, \|c(\alpha)\|. \end{aligned} \tag{2.8}$$

By assumptions (H1)–(H2), the function $B^{-1}(\cdot)$ is continuous on $\mathcal{A}^N$. Moreover, $c(\cdot)$ is continuous, and $\mathcal{A}^N$ is a compact set. Therefore, from the inequality (2.8), we deduce that $(x^k)_k$ is bounded in $\mathbb{R}^N$. Hence x^k converges towards some $x^* \in \mathbb{R}^N$. Let us show that x^* is the solution of (1.1).

Define the function F for $x \in X$ by

$$F(x) := \min_{\alpha \in \mathcal{A}^N} (B(\alpha)x - c(\alpha)), \tag{2.9}$$

and let $F_i(x)$ be the ith component of $F(x)$, i.e.,

$$F_i(x) = \min_{\alpha \in \mathcal{A}^N} \left[B(\alpha)x - c(\alpha)\right]_i.$$

It is obvious that $\lim_{k\to+\infty} F_i(x^k) = F_i(x^*)$. By the compactness of $\mathcal{A}$ and using a diagonal extraction argument, there exists a subsequence of $(\alpha^k)_k$ denoted by α^{ϕ_k} that converges toward some $\alpha \in \mathcal{A}^N$. Furthermore, with assumption (H2), we have $\lim_{k\to\infty}(B(\alpha^{\phi_k})x^k)_i - (B(\alpha)x^*)_i = 0$.

Passing to the limit in $(B(\alpha^{\phi_k})x^{\phi_k} - c(\alpha^{\phi_k}))_i = 0$, we deduce that $(B(\alpha)x^* - c(\alpha))_i = 0$ for all $i \in \mathbb{I}$. On the other hand, we also have

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

Hence $F_i(x^*) = 0$, which concludes the proof of (iii) and implies also that x^* is a solution of (1.1). □

3. Superlinear convergence. In this section, we consider the case when $\mathcal{A}$ is an infinite compact set. First, let us rewrite Howard's algorithm for problem (1.1) as a Newton-like method applied to find the zero of the function $F : \mathbb{R}^N \to \mathbb{R}^N$ defined by (2.9). For $k \geq 0$, by the definitions of α^{k+1} and x^k (the kth iterate of Howard's algorithm), we have

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

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

$$x^{k+1} = x^k - B(\alpha^{k+1})^{-1}F(x^k). \tag{3.1}$$

Equation (3.1) can be interpreted as an iteration of a semismooth Newton method, where $B(\alpha^{k+1})$ plays the role of *a derivative* of F at point x^k. It is well known that Newton's method exhibits locally a quadratic rate of convergence provided that the functional derivative satisfies a certain regular Lipschitz condition which requires that the set of minimizers α^k is unique. This uniqueness is not necessarily satisfied in our context. However, we can prove that the function F is differentiable in a generalized sense, called *slant differentiability.* This notion was introduced by [31, 32] and allows us to extend Newton's method to a more general class of finite or infinite dimensional problems (see also [13, 14]).

Here we first prove the superlinear convergence by using direct arguments. Then we prove that this result is also connected to the fact that the function F is slantly differentiable and that $B(\alpha^{k+1})$ is *a slant derivative* of F at point x^k.

Remark 3.1. All the convergence results proved here for finite dimensional problems can be extended, up to some additional technical assumptions, to the infinite countable spaces: find $x \in \mathbb{R}^{\mathbb{N}^*}$ such that $\min_{\alpha\in\mathcal{A}^{\mathbb{N}^*}}(B(\alpha)x - c(\alpha)) = 0$; see [8] for an example.

For every $x \in \mathbb{R}^N$, let us denote $\mathcal{A}_x$ by the set of minimizers associated with $F(x)$, i.e.,

$$\mathcal{A}_x := \Big\{\alpha \in \mathcal{A}^N,\ B(\alpha)x - c(\alpha) = F(x)\Big\}. \tag{3.2a}$$

For every $i \in \mathbb{I}$, we also define the set $\mathcal{A}_{x,i}$ of minimizers associated with the ith component of $F(x)$, i.e.,

$$\mathcal{A}_{x,i} := \Big\{a \in \mathcal{A},\ \big[B^a x - c^a\big]_i = \big[F(x)\big]_i\Big\}. \tag{3.2b}$$

With these notations, it is clear that $\mathcal{A}_x = \prod_{i\in\mathbb{I}} \mathcal{A}_{x,i}$.

LEMMA 3.2. *Assume that* (H1)–(H2) *hold. For every $x \in X$ and for all $i \in \mathbb{I}$,*

$$d(\alpha_i^{x+h},\ \mathcal{A}_{x,i}) \to 0 \quad \text{as} \quad h \in \mathbb{R}^N \text{ and } \|h\| \to 0,$$

with $\alpha_i^{x+h} \in \mathcal{A}_{x+h,i}$.

Remark 3.3. Of course, one readily sees that α_i^{x+h} does not converge necessarily to a minimizer α_i^x when h tends to 0. However, by Lemma 3.2, the set-valued application $x \longmapsto \mathcal{A}_{x,i}$ is upper semicontinuous on $\mathbb{R}^N$ for every $i \in \mathbb{I}$.

The proof of the previous lemma is given in Appendix A. Now we claim the following convergence result.

THEOREM 3.4. *Assume that* (H1)–(H2) *are satisfied. Then* (1.1) *has a unique solution $x^* \in \mathbb{R}^N$, and for any initial iterate $\alpha^0 \in \mathcal{A}^N$, Howard's algorithm converges globally* ($\lim_{k\to\infty} \|x^k - x^*\| = 0$) *and superlinearly, i.e.,*

$$\|x^{k+1} - x^*\| = o\left(\|x^k - x^*\|\right) \quad \text{as } k \to \infty.$$

Proof. Existence and uniqueness of the solution as well as the increasing property of the sequence x^k have already been proved in Theorem 2.1.

It remains to show the superlinear 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 Theorem 3.8, for all $k \geq 0$, we can find $\alpha^{k,*} \in \mathcal{A}_{x^*}$ such that

$$B(\alpha^{k+1}) - B(\alpha^{k,*}) \to 0 \text{ as } k \to \infty. \tag{3.3}$$

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., a minimizer associated with 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

$$0 \geq x^{k+1} - x^* \geq (I - B(\alpha^{k+1})^{-1}B(\alpha^{k,*}))(x^k - x^*). \tag{3.4}$$

By (3.3) and (H1), we obtain $I - B(\alpha^{k+1})^{-1}B(\alpha^{k,*}) \xrightarrow{k\to+\infty} 0$ (we also use the fact that $B^{-1}(\cdot)$ is continuous on the compact set $\mathcal{A}$). Then, from (3.4), we obtain

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

and this concludes the proof of superlinear convergence. ☐

Remark 3.5 (quadratic convergence). Note that stronger convergence results can be obtained under an additional assumption on the dependence of $f(x,\alpha) := B(\alpha)x - c(\alpha)$ with respect to α (as in [35]). For instance, assume that $\mathcal{A}$ is a compact interval of $\mathbb{R}$ and that for all $1 \le i \le N$, $f_i(x,\alpha) = r_i(x)\alpha_i^2 + s_i(x)\alpha_i + t_i(x)$ (i.e., quadratic in α_i), where $r_i > 0$, $r_i(\cdot)$, $s_i(\cdot)$, and $t_i(\cdot)$ are Lipschitz continuous functions on $\mathbb{R}^N$ (see the example of section 4.3). In this case, for every $x \in \mathbb{R}^N$, a minimizer α^x is given by $\alpha_i^x := \mathrm{argmin}_{\alpha_i \in \mathcal{A}} f_i(x,\alpha) = P_{\mathcal{A}}(-\frac{s_i(x)}{2r_i})$, where $P_{\mathcal{A}}$ denotes the projection on the interval $\mathcal{A}$. Hence, in the neighborhood of the solution x^*, we obtain that $\|\alpha^x - \alpha^{x^*}\| \le C\|x - x^*\|$ (for some $C > 0$). This implies that $\|B(\alpha^x) - B(\alpha^{x^*})\| \le C\|x - x^*\|$, and using (3.4) this leads to a global quadratic convergence result.

Remark 3.6. The assertions of Theorems 2.1 and 3.4 also hold for the problem

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

where $\mathcal{A}_i$ are nonempty compact sets.

Although our proof of Theorem 3.4 is based on direct arguments, the superlinear convergence may also be obtained as a consequence of the *slant differentiability* of the function F and the interpretation of Howard's algorithm as Newton's method [14, 32]. Let us recall the notion of slant differentiability.

DEFINITION 3.7 (slant differentiability [14, Definition 1]). *Let Y and Z be two Banach spaces. A function $\mathcal{F} : Y \to Z$ is said to be* slantly differentiable *in an open set $U \subset Y$ if there exists a family of mappings $\mathcal{G} : U \to \mathcal{L}(Y,Z)$ such that*

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

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

THEOREM 3.8. *We assume that* (H1)–(H2) *hold. The function F defined in* (2.9) *is slantly differentiable on $\mathbb{R}^N$, with slanting function $G(x) = B(\alpha^x)$ for $\alpha^x \in \mathcal{A}_x$.*

Proof. Let x, h be in $\mathbb{R}^N$. From the definition of F, for any $\alpha \in \mathcal{A}_x$ and any $\alpha^{x+h} \in \mathcal{A}_{x+h}$, we have

$$F(x) + B(\alpha^{x+h})h \le F(x+h) \le F(x) + B(\alpha)h,$$

and thus

$$\begin{aligned} 0 \le F(x+h) - F(x) - B(\alpha^{x+h})h &\le (B(\alpha) - B(\alpha^{x+h}))h \\ &\le \|B(\alpha) - B(\alpha^{x+h})\|\|h\|. \end{aligned} \tag{3.5}$$

By Lemma 3.2 there exists $\alpha^h \in \mathcal{A}_x$ such that $d(\alpha_i^h, \alpha_i^{x+h}) \to 0$ as $\|h\| \to 0$ for all $j \in \mathbb{I}$. Thus, by continuity assumption (H2), we obtain

$$\lim_{h\to 0} \|B(\alpha^h) - B(\alpha^{x+h})\| = 0, \tag{3.6}$$

and this concludes the proof. $\square$

4. Some applications.

4.1. Obstacle problem and link with the primal-dual active set strategy.

We focus now on the obstacle problem introduced in (2.2). We assume that A is such that assumption (H1) is satisfied.

As said in Example 1, here we have $\mathcal{A} = \{0, 1\}$, and Howard's algorithm converges in at most 2^N iterations under assumption (H1).

Now we consider the following specific Howard's algorithm for (2.2).

ALGORITHM Ho-2.

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

Iterate for $k \geq 0$:

(i) Find $x^k \in \mathbb{R}^N$ such that $B(\alpha^k)x^k = c(\alpha^k)$.

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

(ii) For every $i = 1, \ldots, N$,

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 emphasize the fact that when $(Ax^k - b)_i = (x^k - g)_i$, we make the choice $\alpha_i^{k+1} = 0$. Although this is not necessary to obtain the convergence, in the following we show that this specific choice leads to a drastic decrease of the number of iterations needed to get the convergence of Howard's algorithm.

THEOREM 4.1. *We assume that the matrices $B(\cdot)$ defined in* (2.3) *satisfy* (H1). *Then Howard's algorithm (Algorithm Ho-2) 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 $\alpha_i^0 = 1$ $\forall i = 1, \ldots, N$, 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 the cost of solving N linear systems (from $k = 1$ to $k = N$).

Proof of Theorem 4.1. First, note 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 2.1, that $x^{k+1} \geq x^k$. This concludes that

$$\forall k \geq 1, \qquad x^k \geq g. \tag{4.1}$$

Now if $\alpha_i^k = 0$ for some $k \geq 1$, then $(Ax^k - b)_i = 0$, and by (4.1) 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 $\mathrm{Card}(I^k) \leq N$, 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, $F(x^{k+1}) = B(\alpha^{k+2})x^{k+1} - c(\alpha^{k+2}) = B(\alpha^{k+1})x^{k+1} - c(\alpha^{k+1}) = 0$, 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. □

Next, we consider Algorithm Ho-2' which is a variant of Howard's algorithm (Algorithm Ho-2) and defined as follows.

ALGORITHM Ho-2'.

Start from a given x^0. Then compute α^1 (by step (ii)) and x^1 (by step (i)), then α^2 and x^2, and so on, until $F(x^k) = 0$.

For Algorithm Ho-2', we have a specific result, that will be useful when studying the approximation of American options in section 4.2.

THEOREM 4.2. *Assume that the matrices* $B(\cdot)$ *defined in* (2.3) *satisfy* (H1).

(i) *Assume that* x^0 *satisfies*

$$\forall i \in \mathbb{I}, \quad x_i^0 > g_i \ \Rightarrow \ (Ax^0 - b)_i \leq 0 \tag{4.2}$$

(or, equivalently, $\min(Ax^0 - b,\ x^0 - g) \leq 0$*). Then the iterates of Algorithm Ho-2' satisfy* $x^{k+1} \geq x^k$ *for all* $k \geq 0$.

(ii) *If furthermore* $x^0 \geq g$ *and if* x^* *denotes the solution of* (1.1), *then the convergence is in at most* p *iterations, with*

$$p := \mathrm{Card}\{i \in \mathbb{I},\ x_i^* > g_i\} - \mathrm{Card}\{i \in \mathbb{I},\ x_i^0 > g_i\}.$$

In other words, (ii) means that "the number of iterations is bounded by the number of points that take off the boundary g, between x^0 and x^*." The proof of the above theorem is given in Appendix B.

Now we turn to showing the relationship with the primal-dual active set algorithm proposed by Hintermüller, Ito, and Kunisch [14] (see also [17, 18]). It is well known that x is a solution of (2.2) if and only if there exists $\lambda \in \mathbb{R}_+^N$ such that

$$\begin{cases} Ax - \lambda &= b, \\ \mathcal{C}(x, \lambda) &= 0, \end{cases} \tag{4.3}$$

where

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

(note also that $\lambda = \mathcal{P}_{[0,+\infty)}(\lambda - (x - g))$). The idea developed in [14] 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 $\mathcal{I}_k := \{i : \lambda_i^k \leq (x^k - g)_i\}$, $\mathcal{AC}_k := \{i : \lambda_i^k > (x^k - g)_i\}$.
If $k \geq 1$ and $\mathcal{I}_k = \mathcal{I}_{k-1}$ then stop. Otherwise go to (ii).

(ii) Solve

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

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 $\lambda^{k+1} = Ax^{k+1} - b$, $\lambda_i^{k+1} = (Ax^{k+1} - b)_i = 0$ for $i \in \mathcal{I}_k$, and $(x^{k+1} - g)_i = 0$ for $i \in \mathcal{AC}_k$. This is equivalent to saying that

$$\widetilde{B} x^{k+1} - \widetilde{c} = 0, \tag{4.4}$$

where

$$\widetilde{B}_{i,.} := \begin{cases} A_i, & \text{if } (Ax^k - b)_i \leq (x^k - g)_i, \\ (I_N)_i, & \text{otherwise}, \end{cases}$$

$$\widetilde{c}_i := \begin{cases} b_i & \text{if } (Ax^k - b)_i \leq (x^k - g)_i, \\ g_i & \text{otherwise}, \end{cases}$$

where I_N denotes the $N \times N$ identity matrix.

Let us consider the equivalent formulation $\min(Ax-b, x-g)=0$, and let $B(.)$ and $c(.)$ be defined as in (2.3). 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 $\tilde{B} = B(\alpha^{k+1})$, $\tilde{c} = c(\alpha^{k+1})$. Therefore, (4.4) is equivalent to

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

and x^{k+1} is defined from x^k as in Howard's algorithm applied to $\min(Ax-b, x-g)=0$. (Note also that similar considerations apply if we replace $x-g$ by $d(x-g)$ for a given constant $d>0$.)

THEOREM 4.3. *Howard's algorithm (Algorithm Ho-2) and the 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 4.4. By Theorem 4.1, the primal-dual active set strategy converges in no more that N iterations (taking $x^0 = g$). We refer to [14] for similar observations.

Remark 4.5. In the same way, one can see that the front-tracking algorithm proposed in [1, Chapter 6.5.3] for a particular obstacle problem (mainly, the American put option) and based on the primal-dual active set method is equivalent to Howard's algorithm.

4.2. 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 nonlinear PDE (see [23] or [29] for existence and uniqueness results in the viscosity framework):

$$\min\left(\partial_t u - \frac{1}{2}\sigma^2 s^2 \partial_{ss} u - rs\partial_s u + ru,\ u - \varphi(s)\right) = 0,$$

$$t \in [0,T],\ s \in (0, S_{\max}), \tag{4.5a}$$

$$u(t, S_{\max}) = 0, \quad t \in [0,T], \tag{4.5b}$$

$$u(0,s) = \varphi(s), \quad s \in (0, S_{\max}), \tag{4.5c}$$

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 purposes 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 $\delta t = T/M$, where $M \geq 1$ 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):

$$\min\left(\frac{U_j^{n+1} - U_j^n}{\delta t} - \frac{1}{2}\sigma^2 s_j^2 \frac{(D^2 U^{n+1})_j}{h^2} - rs_j \frac{D^+ U_j^{n+1}}{h} + rU_j^{n+1};\right.$$
$$\left. U_j^{n+1} - g_j\right) = 0, \quad j = 0, \dots, N_s - 1,\ n = 0, \dots, M-1,$$

$$U_{N_s}^{n+1} = 0, \quad n = 0, \dots, M-1,$$
$$U_j^0 = g_j, \quad j = 0, \dots, N_s - 1,$$

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

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

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

$$\min\left(\frac{U_0^{n+1} - U_0^n}{\delta t} + rU_0^{n+1},\ U_0^{n+1} - g_0\right) = 0.$$

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

It is known that the IE scheme converges to the viscosity solution of (4.5) when $\delta t, h \to 0$ (one may use, for instance, the Barles–Souganidis [4] abstract convergence result and monotonicity properties of the scheme). The motivation for using an implicit scheme is for unconditional stability.[2]

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

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

assuming $U_{N_s} = 0$. Since A is an M-matrix, assumption (H1) is satisfied and, for each step n of the IE scheme, Howard's algorithm generates a sequence of approximations (x^k) of U^{n+1}. We choose to apply Algorithm Ho-2' with starting point $x^0 := U^n$ (see Figure 4.1). Thus, for each $n \in [0, \dots, M-1]$, Howard's algorithm may need up to N_s iterations, and the total number of Howard's iterations (for the IE scheme) is a priori bounded by $M \times N_s$. Actually, in the present example, if we choose to apply Algorithm Ho-2' in each step n with starting point $x^0 := U^n$, then the number of Howard's iterations can be improved.

PROPOSITION 4.6. *The total number of Howard's iterations (using Algorithm Ho-2') from $n = 0$ up to $n = M - 1$ is bounded by N_s. In other words, solving the IE scheme requires us to solve at most N_s linear systems.*

Proof. Let us first 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 $x = U^{n+1}$ is a solution of

$$\min(Ax - U^n, x - g) = 0. \tag{4.6}$$

Therefore, $\min(Ax - U^{n-1}, x - g) \geq 0$. This means that U^{n+1} is a supersolution of $\min(Ax - U^{n-1}, x - g) = 0$ (whose solution is U^n). By the monotonicity of the matrices $B(\alpha)$, we deduce that $U^{n+1} \geq U^n$.

Then, for each $n \geq 0$ and with $x^0 := U^n$, we have $\min(Ax^0 - U^n, x^0 - g) \leq \min(Ax^0 - U^{n-1}, x^0 - g) = 0$. Hence, by Theorem 4.2 and using Algorithm Ho-2' (initialized with $x^0 = U^n$) to solve (4.6), we obtain that U^{n+1} can be computed in no more than p_n iterations, where $p_n := \text{Card}\{j,\ U_j^{n+1} > g_j\} - \text{Card}\{j,\ U_j^n > g_j\}$. Henceforth, the total number of Howard's iterations to compute $\{U^1, \dots, U^M\}$ is bounded by

$$\sum_{n=0,\dots,M-1} p_n = \text{Card}\{j,\ U_j^M > g_j\} - \text{Card}\{j,\ U_j^0 > g_j\},$$

which is bounded by N_s. □

The result of the above proposition can hold also for the monotone implicit scheme for more complex American options (such as American options on two assets involving a PDE in two space dimensions).

We finally mention that in [18] (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 involved matrices are not necessarily monotonous.

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

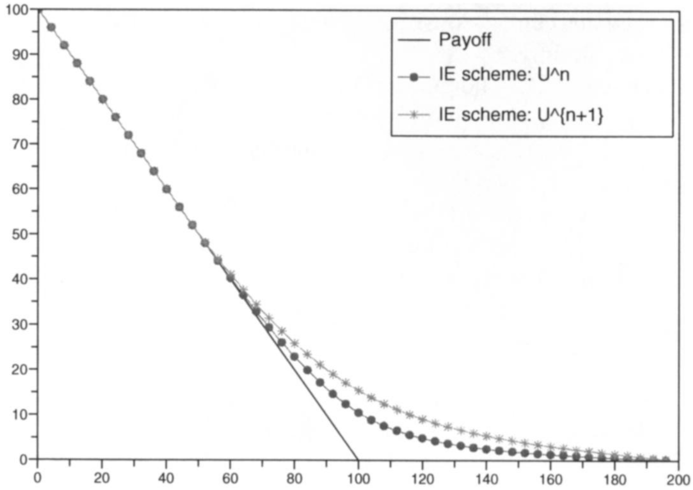

FIG. 4.1. *Plot of* U_j^n *and* U_j^{n+1} *(time* $t_n = 0.2$*) with respect to* s_j*. Parameters:* $K = 100$, $\sigma = 1$, $r = 0.1$, $T = 1$, $S_{\max} = 200$, $N_s = 50$, *and* $M = 10$.

4.3. Application to Merton's problem and with a compact control set. 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 [27] (see, for instance, [25] or [8] for recent applications of Howard's algorithm to solve nonlinear equations arising in finance). The problem is to find the optimal value for a dynamic portfolio. 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:

$$\min_{\alpha \in \mathcal{A}} \left(\partial_t u - \frac{1}{2}\sigma^2 \alpha^2 s^2 \partial_{ss} u - (\alpha\mu + (1-\alpha)r) s \partial_s u \right) = 0, \tag{4.7a}$$
$$t \in [0, T],\ s \in (0, S_{\max}),$$

$$u(0, s) = \varphi(s), \quad s \in (0, S_{\max}), \tag{4.7b}$$

where $\mathcal{A} := [a_{\min}, a_{\max}]$, $S_{\max} = +\infty$, and φ is a given payoff function (see, for instance, Øksendal [27]). Existence and uniqueness results can be obtained using [30].

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

$$\partial_x u(t, S_{\max}) = \frac{p}{S_{\max}} u(t, S_{\max}), \quad t \in [0, T]. \tag{4.8}$$

A natural IE scheme for the set of equations (4.7)–(4.8) is

$$\min_{\alpha\in\mathcal{A}}\left(\frac{U_j^{n+1}-U_j^n}{\delta t}-\frac{1}{2}\sigma^2 s_j^2\alpha^2\frac{U_{j-1}^{n+1}-2U_j^{n+1}+U_{j+1}^{n+1}}{h^2}\right.$$
$$\left.-(\alpha\mu+(1-\alpha)r)s_j\frac{U_{j+1}^{n+1}-U_j^{n+1}}{h}\right)=0,$$
$$j=0,\dots,N_s,\quad n=0,\dots,M-1,$$
$$\frac{U_{N_s+1}^{n+1}-U_{N_s}^{n+1}}{h}=\frac{p}{S_{\max}}U_{N_s}^{n+1},\quad n=0,\dots,M-1,$$
$$U_j^0=\varphi(s_j),\quad j=0,\dots,N_s.$$

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 solving

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

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

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

and

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

We obtain the monotonicity of the matrices B^α under a CFL-like condition on $\delta t, h$.[3]

Remark 4.7. 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 $\frac{\delta t}{h}\leq const$, which is less restrictive than the CFL condition we would obtain with an explicit scheme (which is $\frac{\delta t}{h^2}\leq const$).

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 CPU time with respect to the first step of Howard's algorithm where a linear system must be solved.

Remark 4.8. It is clear 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 $P_{\mathcal{A}}$ is the projection on the interval $\mathcal{A}$.

[3]Choose $h>0$ and $\delta t>0$ such that $\frac{\delta t}{h}\max_{\alpha\in\mathcal{A}}\left(\frac{1}{2}\sigma^2\alpha^2 pS_{\max}+hp(\alpha\mu+(1-\alpha)r)\right)\leq 1$.

In Figure 4.2, we compare the approximated and exact solutions, and the associated numerical control obtained with the IE scheme. In this example, the exact optimal control is $\alpha_{opt} = 0.625$ (constant). Table 4.1 illustrates the quadratic convergence behavior of the error for solving one time step by Howard's algorithm. We also observed that, in order to reach convergence, the number of Howard's iterates stays small (about 3), even for large N_s values.

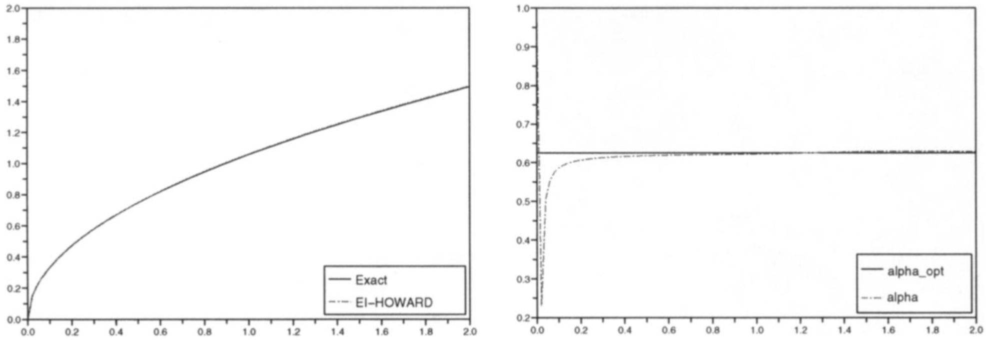

FIG. 4.2. *Plot of* (U_j^M) *(left) and of the discrete optimal control* (α_j) *at time* $t_M = 1$ *(right) with respect to* s_j. *Parameters:* $S_{\max} = 2$, $\mathcal{A} = [0, 1]$, $p = \frac{1}{2}$, $\sigma = 0.4$, $r = 0.1$, $\mu = 0.15$, $T = 1$, $N_s = 200$, *and* $M = 20$.

TABLE 4.1
Howard's algorithm for Merton problem: iteration number and error (example at time step $t_n = 0.2$*) for solving one time step evolution of the IE scheme. Quadratic convergence is observed.*

Iteration	Error $\|\|x^k - x^{k-1}\|\|$
$k = 1$	$1.4\ 10^{-4}$
$k = 2$	$3.8\ 10^{-9}$
$k = 3$	$2.0\ 10^{-15}$

Remark 4.9. At each step we have to solve a sparse linear system. Here the system is tridiagonal and thus can be solved in $O(N_s)$ elementary operations. More generally, multigrid methods can be considered [2] (see also [3]).

5. Max-min problems. In this section, our aim is to generalize Howard's algorithm for a max-min problem

$$\text{find } x \in \mathbb{R}^N, \qquad \max_{b \in \mathcal{B}^N} \left(\min_{a \in \mathcal{A}^N} \left(B^{a,b}x - c^{a,b} \right) \right) = 0. \tag{5.1}$$

Such a nonlinear equation arises in game theory and in optimal control problems. Here we propose computationally feasible algorithms based on policy iterations. We prove convergence results and analyze a numerical test coming from a two-player game problem (see section 5.2).

Specific convergence properties will also be studied in the context of the double-obstacle problem.

5.1. A general setting. Let $\mathcal{A}$ and $\mathcal{B}$ be two compact sets. For every $(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}$, $B^{a,b}$ denotes an $N \times N$ real matrix and $c^{a,b}$ is a vector of $\mathbb{R}^N$. As in section 2, we introduce the matrices $B(\alpha, \beta)$ and vectors $c(\alpha, \beta)$ defined by

$$B_{ij}(\alpha, \beta) := B_{ij}^{\alpha_i, \beta_i} \quad \text{and} \quad c_i(\alpha, \beta) := c_i^{\alpha_i, \beta_i}$$

for all $\alpha = (\alpha_i)_{1\le i\le N} \in \mathcal{A}^N$ and $\beta = (\beta_i)_{1\le i\le N} \in \mathcal{B} := \mathcal{B}^N$. We also introduce the functions $G : \mathbb{R}^N \to \mathbb{R}$ and $F^\beta : \mathbb{R}^N \to \mathbb{R}$ (for $\beta \in \mathcal{B}^N$) defined by

$$F^\beta(x) := \min_{\alpha\in\mathcal{A}^N} \left(B(\alpha,\beta)x - c(\alpha,\beta)\right), \quad G(x) := \max_{\beta\in\mathcal{B}^N} F^\beta(x). \tag{5.2}$$

The min and max operators are understood componentwise. With these notations, problem (5.1) can be rewritten as $G(x) := \max_{\beta\in\mathcal{B}^N} F^\beta(x) = 0$.

A natural generalization of Howard's algorithm for solving

$$\text{find } x \in \mathbb{R}^N, \quad \max_{\beta\in\mathcal{B}^N} F^\beta(x) = 0 \tag{5.3}$$

can be stated as follows.

ALGORITHM Ho-3. Consider a given $\beta^0 \in \mathcal{B}^N$. Iterate for $k \ge 0$:

(i) Find x^k such that $F^{\beta^k}(x^k) = 0$ (resolution at fixed β^k).

(ii) Compute $\beta^{k+1} := \operatorname{argmax}_{\beta\in\mathcal{B}^N} F^\beta(x^k)$.

If $G(x^k) = F^{\beta^{k+1}}(x^k) = 0$ then stop;

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

With the same tools used in the previous sections, we obtain the following theorem.

THEOREM 5.1. *Let $\mathcal{B}$ be a metric compact set. For all $1 \le i \le N$ and $\beta_i \in \mathcal{B}$, we consider a function $F_i^{\beta_i} : \mathbb{R}^N \to \mathbb{R}$. For $\beta = (\beta_1, \dots, \beta_N) \in \mathcal{B}^N$, we also denote*

$$F^\beta(x) := (F_1^{\beta_1}(x), \dots, F_N^{\beta_N}(x))^\mathsf{T}.$$

We assume that the following assumptions hold:

(i) *for all $\beta \in \mathcal{B}^N$, F^β is monotone (i.e., $\forall x, y \in \mathbb{R}^N$, $F^\beta(x) \le F^\beta(y) \Rightarrow x \le y$);*

(ii) *for all $\beta \in \mathcal{B}^N$, there exist x such that $F^\beta(x) = 0$;*

(iii) *$\{x \in \mathbb{R}^N \mid \exists\beta \in \mathcal{B}^N, F^\beta(x) = 0\}$ is a bounded set;*

(iv) *$x \to F^\beta(x)$ is a uniformly continuous function with respect to $\beta \in \mathcal{B}^N$.*

Then problem (5.3) *admits a unique solution x^*. Moreover, the iterates given by Algorithm Ho-*3 *satisfy $x^k \ge x^{k+1}$, and x^k converges to the solution x^* as $k \to \infty$.*

When $\mathcal{B}$ is finite, the convergence is obtained in at most $\mathrm{Card}(\mathcal{B})^N$ *iterations (i.e., x^k is a solution for some $k \le \mathrm{Card}(\mathcal{B})^N$).*

Proof. First, we claim that the monotonicity and continuity of the functions F^β imply that G is also monotone. Indeed, suppose that $G(x) \le G(y)$ for $x, y \in \mathbb{R}^N$. Let $\beta_y \in \mathcal{B}^N$ such that $G(y) = F^{\beta_y}(y)$. Then we have $F^{\beta_y}(x) \le G(y) = F^{\beta_y}(y)$. Using the monotonicity of F^{β_y}, we obtain $x \le y$. Therefore, G is monotone, and problem (5.3) admits at most one solution.

With exactly the same arguments as in the proof of Theorem 2.1(ii), we derive the convergence of x^k to the solution x^* of (5.3) as $k \to \infty$ (for this, uniform continuity of $F^\beta(\cdot)$ with respect to $\beta \in \mathcal{B}$ is required). □

The convergence result of Theorem 5.1 is stated in a general framework with abstract function F^β, not necessarily defined as in (5.2). In the particular case of (5.1), Algorithm Ho-3 can be used under monotonicity conditions on matrices $B(\alpha,\beta)$. More precisely, we have the following.

THEOREM 5.2. *Assume that $(\alpha,\beta) \in \mathcal{A}^N \times \mathcal{B}^N \to B(\alpha,\beta)$ and $(\alpha,\beta) \in \mathcal{A}^N \times \mathcal{B}^N \to c(\alpha,\beta)$ are continuous and that all matrices $B(\alpha,\beta)$ are monotone. Then $F^\beta(x)$ defined as in* (5.2) *satisfies the assumptions of Theorem* 5.1 *(and thus we have the convergence of Howard's algorithm).*

We shall see in the next subsection an example where this convergence result can be applied.

Proof of Theorem 5.2. Suppose that F^β is defined as in (5.2) and that monotonicity and continuity assumptions on $B(\alpha,\beta)$ hold. By sections 2 and 3, for every $\beta \in \mathcal{B}$, equation $F^\beta(x)=0$ admits a unique solution x satisfying

$$\|x\| \leq \max_{\alpha\in\mathcal{A}^N,\beta\in\mathcal{B}^N} \|B(\alpha,\beta)^{-1}\|\|c(\alpha,\beta)\|.$$

Moreover, straightforward arguments show that, for every $\beta \in \mathcal{B}^N$, the function $x \to F^\beta(x)$ is L-Lipschitz continuous with $L := \max_{\alpha\in\mathcal{A}^N,\beta\in\mathcal{B}^N} \|B(\alpha,\beta)\|$, and F^β is monotone. Then we apply Theorem 5.1 to conclude that problem (5.3) admits a unique solution and Algorithm Ho-3 converges to this solution. $\square$

Remark 5.3. Algorithm Ho-3 is not a Newton-like method. Indeed, by the definitions of β^{k+1} and x^{k+1}, we have

$$G(x^k) = B(\alpha_{x^k}^{\beta^{k+1}},\beta^{k+1})x^k - c(\alpha_{x^k}^{\beta^{k+1}},\beta^{k+1}),$$
$$B(\alpha_{x^{k+1}}^{\beta^{k+1}},\beta^{k+1})x^{k+1} - c(\alpha_{x^{k+1}}^{\beta^{k+1}},\beta^{k+1}) = 0.$$

The first equation involves a minimizer $\alpha_{x^k}^{\beta^{k+1}} \in \mathcal{A}_{x^k}^{\beta^{k+1}}$, while for the second equation we have a minimizer $\alpha_{x^{k+1}}^{\beta^{k+1}} \in \mathcal{A}_{x^{k+1}}^{\beta^{k+1}}$. Hence the two matrices $B(\alpha_{x^k}^{\beta^{k+1}},\beta^{k+1})$ and $B(\alpha_{x^{k+1}}^{\beta^{k+1}},\beta^{k+1})$ in general differ.

Notice that direct Newton's method for the max-min problem leads to

$$(\beta^{k+1},\alpha^{k+1}) := \arg\max_{\beta\in\mathcal{B}^N}\min_{\alpha\in\mathcal{A}^N} \left(B(\alpha,\beta)x^k - c(\alpha,\beta)\right);$$
$$\text{compute } x^{k+1}\in\mathbb{R}^N, \text{ solution of } B(\alpha^{k+1},\beta^{k+1})x - c(\alpha^{k+1},\beta^{k+1}) = 0$$

(equivalently, $x^{k+1} = x^k - B(\alpha^{k+1},\beta^{k+1})^{-1}G(x^k)$ and $\alpha^{k+1} = \alpha_{x^k}^{\beta^{k+1}}$). Since the max-min functional is neither convex nor concave, Newton's method may not be convergent.

Step (i) of Algorithm Ho-3 requires us to solve exactly a subproblem $F^\beta(x)=0$. In practice, this subproblem will be solved approximately (by using, for instance, Howard's algorithm (Algorithm Ho-1)), that is, $\|F^{\beta_k}(x^k)\|$ smaller than a given threshold. A modified algorithm becomes the following.

ALGORITHM Ho-4. Consider a given $\beta^0 \in \mathcal{B}^N$ and a sequence $(\eta_k)_{k\geq 0} \in \mathbb{R}^+$. Iterate for $k \geq 0$:

(i) Find x^k such that $\|F^{\beta^k}(x^k)\| \leq \eta_k$ (resolution at fixed β^k).

(ii) Set $\beta^{k+1} := \text{argmax}_{\beta\in\mathcal{A}} F^\beta(x^k)$.

If $G(x^k) = F^{\beta^{k+1}}(x^k) = 0$, then stop;

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

THEOREM 5.4. *Assume that the assumptions of Theorem* 5.2 *hold. Let* $(\eta_k)_{k\geq 0}$ *be a sequence of* $\mathbb{R}^+$, *with* $\sum_{k\geq 0}\eta_k < \infty$. *Then the sequence of iterates* (x^k) *given by Algorithm Ho*-4 *converges to the unique solution* x^* *of* $G(x^*)=0$. *Furthermore, we have the lower bound estimate*

$$x^k \geq x^* - C\eta_k, \quad \textit{with } C := \max_{\alpha\in\mathcal{A}^N,\,\beta\in\mathcal{B}^N} \|B(\alpha,\beta)^{-1}\|. \tag{5.4}$$

Proof. In the modified Algorithm Ho-4, $F^{\beta^k}(x^k)$ is not necessarily 0 and the sequence x^k is no longer monotone. For a given $x\in\mathbb{R}^N$, let us denote by α_x^β a minimizer of $F^\beta(x)$ (i.e., $F^\beta(x) = B(\alpha_x^\beta, \beta)x - c(\alpha_x^\beta, \beta)$). Let $\delta \in \mathbb{R}^N$, and x, y be in $\mathbb{R}^N$ be such that $F^\beta(x) \leq F^\beta(y) + \delta$. We have

$$B(\alpha_x^\beta, \beta)x - c(\alpha_x^\beta, \beta) = F^\beta(x) \leq F^\beta(y) + \delta \leq B(\alpha_x^\beta, \beta)y - c(\alpha_x^\beta, \beta) + \delta.$$

Taking into account the monotonicity of $B(\alpha_x^\beta, \beta)$, we get $x \leq y + \|B(\alpha_x^\beta, \beta)^{-1}\|\|\delta\|$. Therefore, F^β satisfies

$$\forall x, y \in \mathbb{R}^N \ \forall \delta \in \mathbb{R}^N \ F^\beta(x) \leq F^\beta(y) + \delta \Longrightarrow x \leq y + C\|\delta\|, \tag{5.5}$$

with C given in (5.4). Let us denote $\delta^k := F^{\beta^k}(x^k)$. First, we have $G(x^k) - G(x^*) = G(x^k) \geq F^{\beta^k}(x^k) = \delta^k$, and hence

$$x^k - x^* \geq -C\|\delta^k\| \geq -C\eta_k. \tag{5.6}$$

On the other hand,

$$\begin{aligned} F^{\beta^{k+1}}(x^{k+1}) - \delta^{k+1} = 0 = F^{\beta^k}(x^k) - \delta^k \\ \leq F^{\beta^{k+1}}(x^k) - \delta^k, \end{aligned}$$

and we deduce that

$$x^{k+1} - x^k \leq C(\eta_k + \eta_{k+1}), \tag{5.7}$$

and thus for any $q \geq p + 1$,

$$x^q - x^p \leq \sum_{j=p}^{q-1} C(\eta_j + \eta_{j+1}) \leq 2C \sum_{j \geq p} \eta_j. \tag{5.8}$$

We deduce from (5.6) and (5.8) that (x^k) is bounded. Moreover, from (5.8) we get that $\limsup x^k - \liminf x^k \leq 0$, and hence x^k is convergent to some limit x^*, which satisfies $G(x^*) = 0$ (as in the proof of Theorem 5.1). ☐

5.2. Application to a two-player game. Consider a two-player exit time problem involving an evader and an opponent: First, let us fix a time step $\varepsilon > 0$ and a given convex domain Ω of $\mathbb{R}^2$. At the beginning of the game the evader is at $x \in \Omega$; his goal is to exit Ω as soon as possible, and the opponent wants to delay his exit as long as possible. At each step (ε being one time step), the evader can choose a direction $a \in \mathbb{R}^2$ ($\|a\| = 1$), while the opponent can either accept or reverse evader's choice (control $b = \pm 1$). The evader then moves distance $\sqrt{2\varepsilon}$ in the direction ba, i.e., from x to $x + \sqrt{2\varepsilon}ba$. If we denote by $u(x)$ the time needed for the evader to exit Ω, then, following [19], u is a function from $\mathbb{R}^2 \to \mathbb{R}^+$ which satisfies the min-max problem

$$u(x) = \min_{|a|=1} \max_{b=\pm 1} \left(\varepsilon + u\left(x + \sqrt{2\varepsilon}ab\right)\right), \quad x \in \Omega, \tag{5.9a}$$

$$u(x) = 0 \quad x \in \partial\Omega. \tag{5.9b}$$

Remark 5.5. It is also known that, in the limit $\varepsilon \to 0$, the problem (5.9) is an approximation of the mean curvature propagation problem

$$\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)|\nabla u| + 1 = 0, \quad x \in \Omega, \quad u = 0 \text{ on } \partial\Omega.$$

We refer to Kohn and Serfaty [19] for details and related problems.

Now (5.9) can be replaced by an approximation on a discrete grid: Consider a two-dimensional grid $\mathcal{G} = (X_\ell)$ of $\bar{\Omega}$. We denote by U_ℓ an approximated value of $u(X_\ell)$. We denote by $[U](X)$ an interpolated value at point X of the values $[U_\ell]$ on the grid points, i.e., for $X \in \bar{\Omega}$,

$$[U](X) := \sum_{\ell} \gamma_\ell^X U_\ell,$$

where (γ_ℓ^X) are such that $\gamma_\ell^X \geq 0$, $\sum_\ell \gamma_\ell^X = 1$, and $\sum_\ell \gamma_\ell^X X_\ell = X$. The discretized equation we consider is then

$$\max_{|a|=1} \min_{b=\pm 1} \left(U_\ell - [U]\left(X_\ell + \sqrt{2\varepsilon} ab \right) - \varepsilon \right) = 0 \quad \forall X_\ell \in \Omega, \tag{5.10a}$$

$$U_\ell = 0 \quad \forall X_\ell \in \partial\Omega, \tag{5.10b}$$

and this can be written in the abstract form

$$\max_{|a|=1} \min_{b=\pm 1} \left(B^{a,b} U - c^{a,b} \right) = 0, \tag{5.11}$$

with unknown vector $U = (U_\ell)_{X_\ell \in \Omega}$ and where $B^{a,b}$ is a monotone matrix and $c^{a,b}$ a vector. In order to simplify the maximization operation, instead of (5.11) we consider a finite dimensional optimization problem

$$\max_{a \in \mathcal{A}} \min_{b=\pm 1} \left(B^{a,b} U - c^{a,b} \right) = 0, \tag{5.12}$$

where $\mathcal{A} := \{a_k = (\cos(\theta_k), \sin(\theta_k)),\ k = 1, \dots, N_a,\ \text{with } \theta_k = \frac{2\pi k}{N_a}\}$ and $N_a \geq 1$ is a given integer.

In Figure 5.1, we give the approximated solution of (5.12) obtained by using Algorithm Ho-3. Table 5.1 summarizes the CPU time and the number of iterations. In the first column, we give the number of Howard's iterations k used in Algorithm

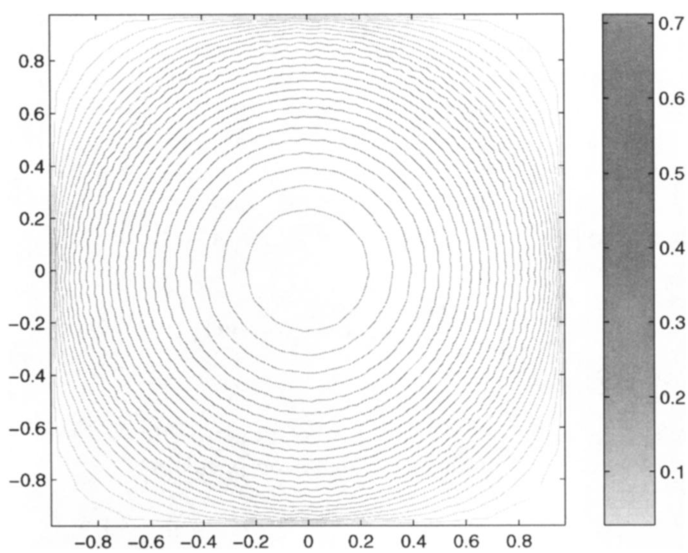

FIG. 5.1. *Contour plot of* U, *with* $N_x = N_y = 80$ *and* $\varepsilon = 0.01$.

Ho-3, while in the second column we give the total number of iterations (the sum over all k of all Howard's iterations for all the subproblems $F^{\beta_k}(x) = 0$ solved by Algorithm Ho-1) and which also corresponds to the number of linear systems to be solved. In this example, the linear systems are solved by using the sparse linear solver of the UMFPACK library [36].[4]

TABLE 5.1
Number of iterations of Howard's algorithm (Algorithm Ho-3) to reach convergence and CPU times (the total number of iterations corresponds to the number of linear systems solved). We have used $\varepsilon = 0.01$ and $N_a = 40$ discrete controls for variable a, and $\Omega = [-1, 1]^2$.

N_x^2	h_x	Iterations (Howard)	Iterations (total)	CPU time (seconds)
20^2	0.105	11	69	2.56
40^2	0.051	11	78	8.01
80^2	0.025	13	107	48.45

5.3. The double-obstacle problem. We consider the following "double-obstacle problem": find $x \in \mathbb{R}^N$ solution of

$$\max\left(\min\left(Ax - b, x - g\right), x - h\right) = 0, \tag{5.13}$$

where A is a given matrix of $\mathbb{R}^{N\times N}$ and b, g, and h are in $\mathbb{R}^N$. 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 (5.13) is then equivalent to finding $x \in \mathbb{R}^N$ solution of $G(x) = 0$.

We first rewrite the function F as follows:

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

where $\mathcal{A} = \{0, 1\}$ and $B(\alpha)$ is defined as in (2.3):

$$B_{ij}(\alpha) := \begin{cases} A_{ij} & \text{if } \alpha_i = 0, \\ \delta_{ij} & \text{if } \alpha_i = 1 \end{cases} \qquad \text{and} \quad c_i(\alpha) := \begin{cases} b_i & \text{if } \alpha_i = 0, \\ g_i & \text{if } \alpha_i = 1, \end{cases} \tag{5.14}$$

with $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. In what follows, we consider that the matrices $B(\alpha)$ satisfy the assumption (H1). Also, we define, for every $\beta \in \mathcal{A}^N$, $F^\beta(x) : \mathbb{R}^N \to \mathbb{R}^N$ by

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

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 (5.13) is equivalent to

$$\text{find } x \in \mathbb{R}^N, \ \max_{\beta\in\mathcal{A}^N} F^\beta(x) = 0. \tag{5.15}$$

To solve this problem, we can consider the policy iteration algorithm (Algorithm Ho-3) with $\mathcal{A} = \mathcal{B} := \{0, 1\}$. More precisely, we apply the algorithm with the following choice:

$$\beta_i^{k+1} := \begin{cases} 0 & \text{if } \ F^\beta(x^k)_i \geq (x^k - h)_i, \\ 1 & \text{otherwise.} \end{cases} \tag{5.16}$$

[4]Using also Scilab (http://www.scilab.org) and the SCISPT interface of Pincon.

For every $k \geq 0$, the equation $F^{\beta^k}(x) = 0$ is an obstacle problem in the form of (2.2). Hence it can be solved with Howard's algorithm (Algorithm Ho-2) in at most N resolutions of linear systems.

We can establish the following convergence result, exactly as in the proof of Theorem 4.1.

THEOREM 5.6. *Assume* (H1). *Then there exists a unique solution of* (5.13), *and Algorithm Ho-*3, *together with the choice* (5.16), *converges in at most* $N+1$ *iterations. It converges in at most* N *iterations if we start with* $\beta_i^0 = 1\ \forall i$.

Remark 5.7. 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 resolutions of linear systems.

5.4. Numerical illustration for the double-obstacle problem. Note that the following example is chosen for illustrative purposes rather than for performance of the methods.

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

$$(5.17)\quad \begin{cases} \max\left(\min\left(-\dfrac{U_{i-1}-2U_i+U_{i+1}}{\Delta s^2} + m(s_i),\ U_i - g(s_i)\right), U_i - h(s_i)\right) = 0, \\ \hfill i = 1,\dots,N, \\ U_0 = u_\ell, \quad U_{N+1} = u_r, \end{cases}$$

with $u_\ell = 1$ and $u_r = 0.8$ (left and right border values, respectively), $\Delta s = \frac{1}{N+1}$ and $s_i = i\Delta s$, $m(s) \equiv 0$, $g(s) := \max(0, 1.2 - ((s-0.6)/0.1)^2)$, and $h(s) := \min(2, 0.3 + ((s-0.2)/0.1)^2)$. Figure 5.2 shows the numerical solution, obtained with $N = 99$.

This problem can easily be reformulated as

$$(5.18)\qquad G(x) := \max(\min(Ax - b, x - g), x - h) = 0,$$

where A is a tridiagonal $N \times N$ matrix (the corresponding matrices $B(\alpha)$ defined by (2.3) satisfy assumption $(H1)$). As in section 5.3, we can write

$$G(x) = \max_{\beta\in\mathcal{B}^N} \min_{\alpha\in\mathcal{A}^N} B(\alpha,\beta)x - c(\alpha,\beta).$$

With Howard's algorithm (*Algorithm* Ho-3), we can solve (5.18) with $k = 14$ global iterations, involving a total number of 88 resolutions of linear systems.

On the other hand, if we use Newton's method as in Remark 5.3, then depending on the initial condition the algorithm may not always converge.

Remark 5.8. In the one-dimensional case ($N = 1$), we consider the problem

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

with $A > 0$ and b, g, and h are given constants with $g < h$. We note that if $x^* = b/A \in (g,h)$ and $\gamma = 1 < A$ and if the starting point x^0 satisfies $x^0 \notin (g,h)$, then Newton's method does not converge (see Figure 5.3). Conversely, we remark that when $\gamma > A$, Newton's method always converges (for any starting iterate $x^0 \in \mathbb{R}$).

Hence, instead of solving $G(x^*) = 0$ and in view of the previous remark, we consider the equivalent problem of solving $G_\gamma(x^*) = 0$, where

$$G_\gamma(x) := \max\left(\min(Ax - b,\ \gamma(x-g)),\ \gamma(x-h)\right),$$

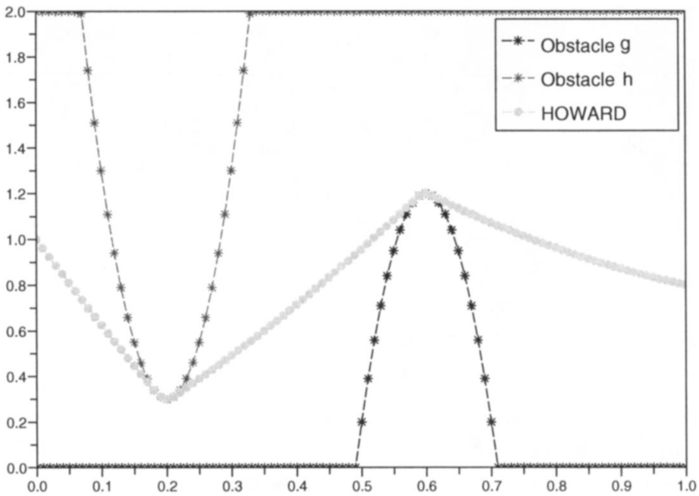

FIG. 5.2. *Solution of the double-obstacle problem, $N = 99$. Values U_j are plotted with respect to s_j. Solution obtained with Howard's algorithm or with Newton's method.*

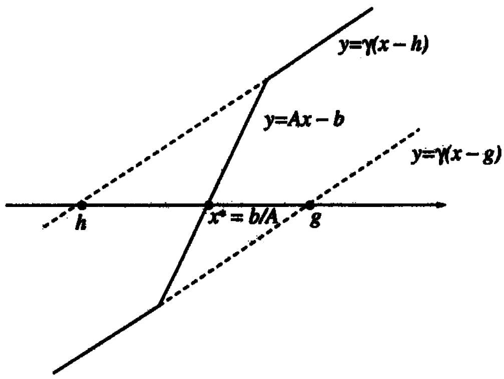

FIG. 5.3. *Divergence of Newton's method in the one-dimensional case.*

with $\gamma > 0$ a given (large) constant. We obtain the exact solution x^* after eight iterations and using $\gamma = 10^4$; see Table 5.2. Numerically, we have observed the convergence of Newton's method as soon as $\gamma > \max_i(A_{ii})$.

Now we consider the primal-dual active set algorithm of Ito and Kunisch [17, section 4] for the double-obstacle problem. In our setting, it amounts to applying Newton's method to find the zero of

$$\widetilde{G}_\gamma(x) := \max\left(\min(Ax - b,\ Ax - b + \bar{\lambda} + \gamma(x - g)),\ Ax - b + \bar{\lambda} + \gamma(x - h)\right)$$

for a given constant $\gamma > 0$ and a given $\bar{\lambda} \in \mathbb{R}^N$. In the numerical tests given in Table 5.2, we have taken $\bar{\lambda} \equiv 0$. We refer to [17] for the meaning and the role of $\bar{\lambda}$ and γ.

Although the primal-dual active set algorithm converges also a in finite number of iterations, it converges to an approximate solution of $\widetilde{G}_\gamma(\cdot) = 0$. For γ large enough, we recover a good approximation of the exact solution. Howard's algorithm needs more iterations here (but this algorithm is more general since it can be applied to

TABLE 5.2
Primal dual active set algorithm [17] *and Newton's method.*

γ	Primal-Dual: iterations	$\|\|G(x^k)\|\|_\infty$	Newton: iterations	$\|\|G(x^k)\|\|_\infty$
10^2	4	$2.4\ 10^{-1}$	divergent	-
10^4	7	$8.7\ 10^{-3}$	8	$1.6\ 10^{-12}$
10^6	9	$1.0\ 10^{-4}$	8	$1.6\ 10^{-12}$
10^{10}	9	$1.0\ 10^{-8}$	8	$1.6\ 10^{-12}$

general max-min problems such as in section 5.2). Finally, Newton's method may diverge if γ is too small (and depending on the starting point) but converges very quickly to the exact solution for γ large enough.

Appendix A. Proof of Lemma 3.2.

Let i be in $\mathbb{I}$. Suppose on the contrary that there exist some $\delta > 0$ and a subsequence $h_n \in \mathbb{R}^N$, with $\|h_n\| \to 0$, such that $d(\alpha_i^{x+h_n}, \mathcal{A}_{x,i}) \geq \delta\ \forall n \geq 0$. Let $K_\delta := \{a \in \mathcal{A},\ d(a, \mathcal{A}_{x,i}) \geq \delta\}$, the function $f : \mathcal{A} \longmapsto \mathbb{R}$ defined by $f(a) := \big[B^a x - c^a\big]_i$, and

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

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

$$f(\alpha_i^{x+h_n}) - f(\alpha_i^x) \geq f(\bar{a}) - f(\alpha_i^x) > 0. \tag{A.1}$$

Let $C := \max_{\alpha \in \mathcal{A}^N} \|B(\alpha)\|$. We have that $\|F(y) - F(z)\| \leq C\|y - z\|$ for every $y, z \in \mathbb{R}^N$. Moreover, $(F(x+h) - F(x))_i = f(\alpha_i^{x+h}) - f(\alpha_i^x) + (B(\alpha^{x+h})h)_i$. Hence $f(\alpha_i^{x+h}) - f(\alpha_i^x) \leq 2C\|h\|$. Taking $h := h_n \to 0$ we obtain a contradiction with (A.1). $\square$

Appendix B. Proof of Theorem 4.2.

(i) The only difficulty is to prove that $x^1 \geq x^0$; otherwise the proof of $x^{k+1} \geq x^k$, for $k \geq 1$, is the same as in Theorem 4.1. First, in the case $\alpha_i^1 = 1$, we have $x_i^1 = g_i$ and $(Ax^0 - b)_i > (x^0 - g)_i$. If $(x^0 - g)_i > 0$, then $(Ax^0 - b)_i > 0$, 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 $(Ax^0 - b)_i \leq (x^0 - g)_i$ and $(Ax^1 - b)_i = 0$. If $(Ax^0 - b)_i > 0$, then $(x^0 - g)_i > 0$, which contradicts the assumption on x^0. Hence $(Ax^0 - b)_i \leq 0$. In particular, $(Ax^0)_i \leq (Ax^1)_i$. 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 $B(\alpha^1)$.

(ii) Now we turn to the proof of the second assertion. For every $k \geq 0$, we set

$$I^k := \{i \in \mathbb{I},\ x_i^k > g_i\} \quad \text{and} \quad m_k := \mathrm{Card}(I^k).$$

Since $(x^k)_k$ is increasing, so is $(m_k)_k$. Moreover, (m_k) is bounded and thus convergent. Let us assume that $m_{k+1} = m_k$ for some $k \geq 0$. In this case, we have $I^{k+1} = I^k$, and for any $i \in I^k$, we have $\alpha_i^k = 0 = \alpha_i^{k+1}$. On the other hand, for $i \notin I^k$, we have $x_i^k = g_i$ and $x_i^{k+1} = g_i$ (here we use the fact that $x^k \geq x^0 \geq g$). We have also by the proof of Theorem 4.1 that α_i^k is nonincreasing. In particular, if $\alpha_i^k = 0$, then $\alpha_i^{k+1} = 0$. Hence the only case when $\alpha_i^k \neq \alpha_i^{k+1}$ is for $i \notin I^k$, $\alpha_i^k = 1$, and $\alpha_i^{k+1} = 0$. However, in this case, we see that $(Ax^{k+1} - b)_i = 0 = x_i^{k+1} - g_i$, and thus we have

$(B(\alpha^k)x^{k+1} - c(\alpha^k))_i = 0$. In particular, $B(\alpha^k)x^{k+1} - c(\alpha^k) = 0$, which means that x^{k+1} is a solution of the same linear system as x^k, and thus $x^{k+1} = x^k$, and the algorithm has converged at iteration k.

Therefore, we have $m_{k+1} \geq m_k + 1$ until convergence is reached ($m^k = m^{k+1}$ with $k = q$ for which $x^q = x$). We obtain $m_q \geq m_0 + q$, and this concludes the proof. □

Acknowledgment. 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 Appl. Math. 30, SIAM, Philadelphia, 2005.

[2] M. Akian, *Méthodes Multigrilles en Contrôle Stochastique*, Ph.D. thesis, University Paris 9, Paris, 1990.

[3] M. Akian, J.L. Menaldi, and A. Sulem, *On an investment-consumption model with transaction costs*, SIAM J. Control Optim., 34 (1996), pp. 329–364.

[4] G. Barles and P.E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptot. Anal., 4 (1991), pp. 271–283.

[5] R. Bellman, *Functional equations in the theory of dynamic programming.* V. *Positivity and quasi-linearity*, Proc. Natl. Acad. Sci. USA, 41 (1955), pp. 743–746.

[6] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[7] M. Bergounioux, K. Ito, and K. Kunisch, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.

[8] O. Bokanowski, B. Bruder, S. Maroso, and H. Zidani, *Numerical approximation of a superreplication problem under gamma constraints*, SIAM J. Numer. Anal., to appear.

[9] F. Bonnans and P. Rouchon, *Commande et Optimisation de Systèmes Dynamiques*, Editions de l'Ecole Polytechnique, Palaiseau, 2005.

[10] J. Bolte, A. Daniilidis, and A. Lewis, *Tame functions are semismooth*, Math. Program. Ser. B, 117 (2009), pp. 5–19.

[11] F. Camilli and M. Falcone, *Approximation of control problems involving ordinary and impulsive controls*, ESAIM Control Optim. Calc. Var., 4 (1999), pp. 159–176.

[12] M.G. Crandall and P.-L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19.

[13] X. Chen, Z. Nashed, and L. Qi, *Smoothing methods and semismooth methods for nondifferentiable operator equations*, SIAM J. Numer. Anal., 38 (2000), pp. 1200–1216.

[14] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002), pp. 865–888.

[15] R.A. Howard, *Dynamic Programming and Markov Processes*, The MIT Press, Cambridge, MA, 1960.

[16] K. Ito and K. Kunisch, *Optimal control of elliptic variational inequalities*, Appl. Math. Optim., 41 (2000), pp. 343–364.

[17] K. Ito and K. Kunisch, *Semi-smooth Newton methods for variationnal inequalities of the first kind*, M2AN Math. Model. Numer. Anal., 37 (2003), pp. 41–62.

[18] K. Ito and K. Kunisch, *Parabolic variational inequalities: The Lagrange multiplier approach*, J. Math. Pures Appl. (9), 85 (2006), pp. 415–449.

[19] R.V. Kohn and S. Serfaty, *A deterministic-control-based approach to motion by curvature*, Comm. Pure Appl. Math., 59 (2006), pp. 344–447.

[20] B. Kummer, *Generalized Newton and NCP methods: Convergence, regularity, actions*, Discuss. Math. Differ. Incl. Control Optim., 20 (2000), pp. 209–244.

[21] B. Kummer, *Newton's method for nondifferentiable functions*, in Advances in Mathematical Optimization, Math. Res. 45, J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Klatte, B. Kummer, K. Lomatzsch, K. Tammer, M. Vlach, and K. Zimmerman, eds., Akademie-Verlag, Berlin, 1988, pp. 114–125.

[22] H.J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer-Verlag, NewYork, Berlin, Heidelberg, 2001.

[23] D. Lamberton and B. Lapeyre, *Introduction au Calcul Stochastique Appliqué à la Finance*, 2nd ed., Ellipses, Paris, 1997.

[24] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.
[25] M. Mnif and A. Sulem, *Optimal risk control and dividend policies under excess of loss reinsurance*, Stochastics, 77 (2005), pp. 455–476.
[26] R. Munos and H. Zidani, *Consistency of a simple multidimensional scheme for HJB equations*, C. R. Acad. Sci. Paris Ser. I, 340 (2005), pp. 499–502.
[27] B. Øksendal, *Stochastic Differential Equations. An Introduction with Applications*, 6th ed., Springer-Verlag, Berlin, 2003.
[28] B. Øksendal and A. Sulem, *Applied Stochastic Control of Jump Diffusions*, Springer-Verlag, Berlin, 2005.
[29] H. Pham, *Optimal stopping, free boundary, and American option in a jump-diffusion model*, Appl. Math. Optim., 35 (1997), pp. 145–164.
[30] H. Pham, *Optimisation et Contrôle Stochastique Appliquès á la Finance*, Collect. Math. Appl., Springer-Verlag, Berlin, 2007.
[31] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18(1993), pp. 227–244.
[32] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Program., 58(1993), pp. 353–367
[33] M.L. Puterman and S.L. Brumelle, *On the convergence of policy iteration in stationary dynamic programming*, Math. Oper. Res., 4 (1979), pp. 60–69.
[34] M.S. Santos, *Accuracy of numerical solutions using the equation residuals*, Econometrica, 68 (2000), pp. 1377–1402.
[35] M.S. Santos and J. Rust, *Convergence properties of policy iteration*, SIAM J. Control Optim., 42 (2004), pp. 2094–2115.
[36] T.A. Davis, *UMFPACK*, http://www.cise.ufl.edu/research/sparse.