Numerical Algorithms for Deterministic Impulse Control Models with Applications

Grass, D.; Chahim, M.

Publication date:
2012

Citation for published version (APA):
Grass, D., & Chahim, M. (2012). *Numerical Algorithms for Deterministic Impulse Control Models with Applications*. (CentER Discussion Paper; Vol. 2012-081). Econometrics.

No. 2012-081

NUMERICAL ALGORITHMS FOR DETERMINISTIC IMPULS CONTROL MODELS WITH APPLICATIONS

By

D. Grass, M. Chahim

19 October, 2012

ISSN 0924-7815

Numerical algorithms for deterministic Impulse Control models with applications

D. Grass[I,b] and M. Chahim[II,a]

[I]*Dept. of Operations Research and Systems Theory, ORCOS, Vienna University of Technology, Austria*
[II]*Dept. of Econometrics & Operations Research, CentER, Tilburg University, The Netherlands*

Abstract

In this paper we describe three different algorithms, from which two (as far as we know) are new in the literature. We take both the size of the jump as the jump times as decision variables. The first (new) algorithm considers an Impulse Control problem as a (multipoint) Boundary Value Problem and uses a continuation technique to solve it. The second (new) approach is the continuation algorithm that requires the canonical system to be solved explicitly. This reduces the infinite dimensional problem to a finite dimensional system of, in general, nonlinear equations, without discretizing the problem. Finally, we present a gradient algorithm, where we reformulate the problem as a finite dimensional problem, which can be solved using some standard optimization techniques. As an application we solve a forest management problem and a dike heightening problem. We numerically compare the efficiency of our methods to other approaches, such as dynamic programming, backward algorithm and value function approach.

Key words: Impulse Control Maximum Principle, Optimal Control, BVP, gradient method, continuation.

JEL-codes: C61, D90, 032, 033

1 Introduction

For many problems in the area of economics and operations research it is realistic to allow for jumps in the state variable. Take, for example, a firm that increases the capital stock by a lumpy investment, or the decrease of the volume of a natural resource after each a drilling. This paper therefore considers optimal control models in which the time moment of these jumps and the size of the jumps are taken as (new) decision variables. Blaquière (1977a; 1977b; 1979; 1985) extends the standard theory on optimal control by deriving a Maximum Principle, the so called Impulse Control Maximum Principle, that gives necessary (and sufficient) optimality conditions for solving such problems. Chahim et al. (2012b) present the necessary optimality

[a]Dieter Grass (`dieter.grass@tuwien.ac.at`) is a Dr. at the institute of Operations Research and Control Systems (ORCOS).

[b]Mohammed Chahim (`m.chahim@tilburguniversity.edu`) is a doctoral candidate at CentER Graduate School. He thanks the Netherlands Organization for Scientific Research (NWO) for financial support.

conditions of the Impulse Control Maximum Principle based on the current value formulation. In Chahim et al. (2012b) also a transformation is designed, which ensures that the application of the Impulse Control Maximum Principle can be applied to problems with a fixed cost. For a review of the literature applying the impulse control maximum principle, we refer to Chahim et al. (2012b).

Like Blaquière (1977a; 1977b; 1979; 1985) and Chahim et al. (2012b), we consider a framework where the number of jumps is not known. This distinguishes our approach from, e.g., Liu et al. (1998) and Wu and Teo (2006) where a gradient method is used assuming the number of jumps is known, and Augustin (2002, pp. 71-81) where the Impulse Control Maximum Principle is used for a fixed number of jumps (see e.g. Rempala (1990)) Other approaches in the literature is the value function approach found in Neuman and Costanza (1990), where a value function is defined for a fixed number of jumps and Erdlenbruch et al. (2011) or Eijgenraam et al. (2011) where dynamic programming is the tool of choice.

In the literature two different algorithms based on the Impulse Control Maximum Principle (Blaquière (1977a; 1977b; 1979; 1985) and Chahim et al. (2012b)) are derived. Luhmer (1986) derived a forward algorithm (starts at time 0) and Kort (1989, pp. 62-70) derived a backward algorithm (starts at final time horizon T). Both algorithms have some drawbacks. To initialize the forward algorithm the initial costate(s) value(s) is the choice variable. A similar drawback holds for the backward algorithm. Here information on the state variable(s) at the end of the planning period is needed, i.e. this (these) value(s) is (are) the choice variables(s).

In this paper we describe three different algorithms, from which two (as far as we know) are new in the literature. We take both the size of the jump and the jump times as decision variables. The first (new) algorithm considers an Impulse Control problem as a (multipoint) Boundary Value Problem and uses a continuation technique to solve it. The second (new) approach is the continuation algorithm that requires the canonical system to be solved explicitly. This reduces the infinite dimensional problem to a finite dimensional system of, in general, nonlinear equations, without discretizing the problem. Finally, we present a gradient algorithm, where we reformulate the problem as a finite dimensional problem, which can be solved using some standard optimization techniques. As an application we solve a forest management problem and a dike heightening problem. We numerically compare the efficiency of our methods to other approaches, such as dynamic programming, backward algorithm and value function approach.

This paper is organized as follows. In Section 2.1 we introduce the type of optimal control problem we consider in this paper. In Section 3 we describe the three algorithm suitable for solving Impulse Control problems. In Section 3.1 we introduce some notation and show that the necessary conditions can be restated as a (multipoint) boundary value problem (BVP). Second, we describe the continuation algorithm in Section 3.2. Third, we describe the gradient algorithm in Section 3.3, which is developed by Hou and Wong (2011). In Section 4 we introduce two applications, one deals with forest management (Section 4.1), and one deals with dike heightening (Section 4.2). The numerical results for both applications are presented in Section 5. We compare our found results with the results found in the literature. Finally, in Section 6 we conclude and give recommendations for future research.

2 An Impulse Control model

In this section we introduce a general Impulse Control model and provide the necessary optimality conditions.

2.1 The Model

Let us denote x as the state variable, u as an ordinary control variable and v^i as the impulse control variable, where x and u are piecewise continuous functions of time[1]. We denote r as the discount rate leading to the discount factor e^{-rt}. The terminal time or horizon date of the system or process is denoted by $T > 0$, T^+ stands for the time moment just after T, and $x(T^+)$ stands for the state value after a possible jump at time T. The profit of the system between jumps is given by $F(x(t), u(t), t)$, whereas $G(x(t), v^i, t)$ is the profit function associated with the i-th jump, and $S(x(T^+))$ is the salvage value, i.e. the total costs or profit associated with the system after time T. Finally, $f(x(t), u(t), t)$ describes the continuous change of the state variable over time between the jump points and $g(x(t), v^i, t)$ is a function that represents the instantaneous (finite) change of the state variable when there is an impulse or jump.

The above results in the following optimal control problem

$$\max_{u(\cdot),N,\tau_i,v^i} \left\{ \int_0^T \mathrm{e}^{-rt}\, F(x(t), u(t), t)\, \mathrm{d}t \right\} + \sum_{i=1}^{N} \mathrm{e}^{-r\tau_i}\, G(x(\tau_i^-), v^i, \tau_i) + \mathrm{e}^{-rT}\, S(x(T^+)), \tag{1a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t), t) \quad \text{for} \quad t \in [0, T] \setminus \{\tau_1, \ldots, \tau_N\}, \tag{1b}$$

$$x(\tau_i^+) - x(\tau_i^-) = g(x(\tau_i^-), v^i, \tau_i) \quad \text{for} \quad i \in \{1, \ldots, N\}, \tag{1c}$$

$$x(0^-) = x_0 \in \mathbb{R}^n, \quad u(t) \in \mathcal{U} \subset \mathbb{R}^m, \quad v^i \in \mathcal{V} \subset\in \mathbb{R}^l, \quad t \in [0, T]. \tag{1d}$$

For $N \in \mathbb{N}$ we assume the jump times to be sorted as

$$\tau_i \in [0, T] \quad \text{with} \quad 0 \le \tau_1 < \ldots < \tau_N \le T, \tag{1e}$$

where

$$x(\tau_i^+) = \lim_{t \downarrow \tau_i} x(t) \quad \text{and} \quad x(\tau_i^-) = \lim_{t \uparrow \tau_i} x(t).$$

We assume that the domains, $\mathcal{U}$ and $\mathcal{V}$ are bounded convex sets. Further we impose that F, f, g and G are continuously differentiable in x on $\mathbb{R}^n$ and v^i on $\mathcal{V}$, $S(x)$ is continuously differentiable in x on $\mathbb{R}^n$, and that g and G are continuous in τ. Finally, when there is no jump, i.e. $v^i = 0$, we assume that

$$g(x(t), 0, t) = 0,$$

for all x and t.

2.2 Necessary Optimality Conditions

We apply the Impulse Control Maximum Principle in current value formulation derived in Chahim et al. (2012b) to (1)[2] the resulting necessary optimality conditions are presented in Theorem 1.

[1] Note that the necessary optimality conditions presented in Theorem 1 also hold for measurable controls. Applications typically have piecewise continuous functions.

[2] Other references deriving the necessary optimality conditions for the Impulse Control problems are Blaquière (1977a; 1977b; 1979; 1985), Seierstad (1981) and Seierstad and Sydsæter (1987)

Before we state Theorem 1, let us define the Hamiltonian $\mathcal{H}$ and the Impulse Hamiltonian $\mathcal{IH}$ as

$$\mathcal{H}(x,u,\lambda,t) := F(x,u,t) + \lambda f(x,u,t), \tag{2a}$$
$$\mathcal{IH}(x,v,\lambda,t) := G(x,v,t) + \lambda g(x,v,t), \tag{2b}$$

and define the following abbreviations

$$\mathcal{H}[s] := \mathcal{H}(x(s),u(s),\lambda(s),s), \tag{2c}$$
$$\mathcal{IH}[s] := \mathcal{IH}(x(s^-),v,\lambda(s^+),s), \tag{2d}$$
$$G[s] := G(x(s^-),v,s), \tag{2e}$$
$$g[s] := g(x(s^-),v,s). \tag{2f}$$

Theorem 1 (Impulse control maximum principle)**.**

Let for $N \in \mathbb{N}$ with $N > 0$ $(x^*(\cdot), u^*(\cdot), N, \tau_1^*, \ldots, \tau_N^*, v^{1*}, \ldots,, v^{N*})$ be an optimal solution of (1). Then there exists a (piecewise absolute continuous) adjoint variable $\lambda(\cdot)$ such that the following conditions hold:

$$u^*(t) \in \operatorname*{argmax}_u \mathcal{H}(x^*(\cdot),u,\lambda(\cdot),t), \quad t \in [0,T], \tag{3a}$$
$$\dot{\lambda}(t) = r\lambda(t) - \frac{\partial}{\partial x}\mathcal{H}(x^*(\cdot),u^*(t),\lambda(\cdot),t), \quad t \in [0,T] \setminus \{\tau_1,\ldots,\tau_N\}. \tag{3b}$$

For every $\tau_i,\ i = 1,\ldots N$, we have

$$\frac{\partial}{\partial v}\mathcal{IH}(x^*(\tau_i^{*-}),v,\lambda(\tau_i^{*+}),\tau_i^*)(v - v^{i*}) \le 0, \tag{3c}$$
$$\lambda(\tau_i^{*+}) - \lambda(\tau_i^{*-}) = -\frac{\partial}{\partial x}\mathcal{IH}(x^*(\tau_i^{*-}),v^{i*},\lambda(\tau_i^{*+}),\tau_i^*), \tag{3d}$$
$$\mathcal{H}[\tau_i^{*+}] - \mathcal{H}[\tau_i^{*-}] + rG[\tau_i^*] - \frac{\partial}{\partial \tau}\mathcal{IH}[\tau_i^{*+}] \begin{cases} > 0 & \tau_i = 0 \\ = 0 & \tau_i \in (0,T). \\ < 0 & \tau_i = T \end{cases} \tag{3e}$$

For $t \in [0,T] \setminus \{\tau_1,\ldots,\tau_N\}$ it holds that

$$\frac{\partial}{\partial v}\mathcal{IH}(x^*(t),0,\lambda(t),t)v \le 0. \tag{3f}$$

The transversality condition is

$$\lambda(T^+) = \frac{\partial}{\partial x}S(x^*(T^+)). \tag{3g}$$

Proof: see Blaquière (1977a; 1985) □

To simplify the presentation and to concentrate on the main concepts of the numerical algorithm, besides the earlier made assumption, we further make the following assumptions.

Assumption 1. *For every time horizon $T \ge 0$ there exists a unique optimal solution of (1), with a finite number of jumps (which in general depends on T).*

This assumption is needed for the boundary value problem approach and the continuation algorithm. If this assumption does not hold, both algorithms will not generate a solution since the number of jumps is not finite. This assumption is not required for the gradient algorithm, since the number of jumps are fixed.

Assumption 2. *Let for $T > 0$ the jump times be $(\tau_i)_{i=1}^N$ with $0 < \tau_1 < \ldots < \tau_N < T$, and $\bar{x}(T) := (x(\tau_1^-), x(\tau_1^+), v_1, \ldots, x(\tau_N^-), x(\tau_N^+), v_N)$ be the vector of left and right limits of the states together with the optimal impulse control values for the given time horizon T. Then in a neighborhood of T the solution vector $\bar{x}(T)$ is continuous.*

We need this assumption again for both the boundary value problem approach and the continuation algorithm. For both algorithms T is a continuation variable. During the continuation process T is increased and the conditions for possible jumps are monitored.

Assumption 3. *The model does not include a continuous control $u(\cdot)$, i.e., $u: \mathbb{R} \to \emptyset$, and $v^i \in \mathbb{R},\ i \geq 0$.*

For simplicity we state this assumption. Then the boundary value problem approach is still a suitable method to solve the problem. The gradient method and the continuation algorithm depend on whether the system is explicitly solvable or not.

Assumption 4. *Condition (3c) implies*

$$\frac{\partial}{\partial v}\mathcal{IH}(x^*(\tau_i^{*-}), v^{i*}, \lambda(\tau_i^{*+}), \tau_i^*) = 0 \tag{4}$$

and with $\frac{\partial^2}{\partial v^2}\mathcal{IH}(x^*(\tau_i^{*-}), v^{i*}, \lambda(\tau_i^{*+}), \tau_i^*) < 0$ *this yields*

$$v^i = v(x(\tau_i^-), \lambda(\tau_i^+), \tau_i). \tag{5}$$

In general condition (3c) does not imply that the optimal impulse control value can be found as the arg max of the Impulse Hamiltonian. For simplicity we restrict ourself to such function in this paper.

3 Numerical Algorithms

In this section we describe three different algorithms to solve Impulse Control problems. We state a (multipoint) boundary value problem for Impulse Control problems in Section 3.1 which is (as far as we know) new in the literature, describe the gradient method approach developed by Hou and Wong (2011) in Section 3.3, and finally we describe a second new approach that we call the continuation algorithm in Section 3.2.

3.1 (Multipoint) Boundary Value Approach

In this section we describe a (multipoint) boundary value problem (BVP), that is useful to solve Impulse Control problems. The idea behind the boundary value approach is that between two jumps the system of differential equations (canonical system) combined with the boundary conditions (begin and end conditions) is solved. After each found jump the (multipoint) BVP is updated to find the next jump.

To formulate the (multipoint) BVP we introduce the following notation for the canonical system dynamics:

$$\dot{x}(t) = h_1(x(t), \lambda(t), t), \tag{6a}$$

$$\dot{\lambda}(t) = h_2(x(t), \lambda(t), t). \tag{6b}$$

For the conditions at a jumping time τ it holds that:

$$j^x(x(\tau^+), x(\tau^-), \lambda(\tau^+), \tau) := x(\tau^+) - x(\tau^-) - g[\tau], \tag{6c}$$

$$j^\lambda(x(\tau^-), \lambda(\tau^+), \lambda(\tau^-), \tau) := \lambda(\tau^+) - \lambda(\tau^-) + \frac{\partial}{\partial x}\mathcal{IH}[\tau], \tag{6d}$$

$$\begin{aligned} j^\tau(x(\tau^-), x(\tau^+), \lambda(\tau^+), \lambda(\tau^-), \tau) := \mathcal{H}[\tau^+] - \mathcal{H}[\tau^-] + \\ rG[\tau] - \frac{\partial}{\partial \tau}\mathcal{IH}[\tau]. \end{aligned} \tag{6e}$$

Now let $(x^*(\cdot), u^*(\cdot), N, \tau_1^*, \dots, \tau_N^*, v^{1*}, \dots, v^{N*})$ be an optimal solution of (1) with $0 < \tau_1^* < \dots < \tau_N^* < T$. Then the necessary conditions yield the following (multipoint) BVP:

$$\dot{x}_i(t) = h_1(x_i(t), \lambda_i(t), t), \quad t \in [\tau_{i-1}, \tau_i],\ i = 1, \dots, N+1, \tag{7a}$$

$$\dot{\lambda}_i(t) = h_2(x_i(t), \lambda_i(t), t), \quad t \in [\tau_{i-1}, \tau_i],\ i = 1, \dots, N+1, \tag{7b}$$

$$j^x(x_i(\tau_i^+), x_i(\tau_i^-), \lambda_i(\tau_i^+), \tau_i) = 0, \quad i = 1, \dots, N, \tag{7c}$$

$$j^\lambda(x_i(\tau_i^-), \lambda_i(\tau_i^+), \lambda_i(\tau_i^-), \tau_i) = 0, \quad i = 1, \dots, N, \tag{7d}$$

$$j^{\tau_i}(x_i(\tau_i^-), x_i(\tau_i^+), \lambda_i(\tau_i^+), \lambda_i(\tau_i^-), \tau_i) = 0, \quad i = 1, \dots, N, \tag{7e}$$

$$\mathcal{S}(x_{N+1}(T), \lambda_{N+1}(T)) = 0, \tag{7f}$$

$$x_1(0) - x_0 = 0, \tag{7g}$$

where (7f) denotes the transversality condition (3f).

For notational simplicity we define $\tau_0 := 0$ and $\tau_{N+1} := T$. After defining $t(s) := \tau_i - i\Delta\tau_i + \Delta\tau_i s$, with $\Delta\tau_i := \tau_i - \tau_{i-1}$, we rewrite (7) into

$$\dot{x}_i(s) = \Delta\tau_i h_1(x_i(s), \lambda_i(s), t(s)), \quad s \in [i-1, i],\ i = 1, \dots, N+1 \tag{8a}$$

$$\dot{\lambda}_i(s) = \Delta\tau_i h_2(x_i(s), \lambda_i(s), t(s)), \quad s \in [i-1, i],\ i = 1, \dots, N+1 \tag{8b}$$

$$j^x(x_i(i^+), x_i(i^-), \lambda_i(i^+), \tau_i) = 0, \quad i = 1, \dots, N \tag{8c}$$

$$j^\lambda(x_i(i^-), \lambda_i(i^+), \lambda_i(i^-), \tau_i) = 0, \quad i = 1, \dots, N \tag{8d}$$

$$j^i(x_i(i^-), x_i(i^+), \lambda_i(i^+), \lambda_i(i^-), \tau_i) = 0, \quad i = 1, \dots, N \tag{8e}$$

$$\mathcal{S}(x_{N+1}(N+1), \lambda_{N+1}(N+1)) = 0, \tag{8f}$$

$$x_1(0) - x_0 = 0. \tag{8g}$$

The jump times $\tau_i,\ i = 1 \dots, N$, appear as unknown variables.

To handle the case $\tau_N = T$ we introduce the (unknown) variables

$$\begin{aligned} \mathrm{xT} &:= x_{N+1}(T^+), \\ \mathrm{lT} &:= \lambda_{N+1}, (T^+), \end{aligned}$$

together with the additional boundary conditions

$$j^x(\mathrm{xT}, x_{N+1}(N+1), \mathrm{lT}, T) = 0, \tag{9a}$$

$$j^\lambda(x_{N+1}(N+1), \mathrm{lT}, \lambda_{N+1}(N+1), T) = 0, \tag{9b}$$

and replace (8f) by

$$\mathcal{S}(\mathrm{xT}, \mathrm{lT}) = 0. \tag{9c}$$

The case $\tau_1 = 0$ can be treated in an analogous way. We therefore set

$$\begin{aligned} \mathrm{x0} &:= x_1(0^+), \\ \mathrm{l0} &:= \lambda_1(0^+), \end{aligned}$$

together with the additional boundary conditions

$$j^x(\mathrm{x0}, x_0, \mathrm{l0}, 0) = 0, \tag{10a}$$
$$j^\lambda(x_0, \mathrm{l0}, \lambda_1(0), 0) = 0, \tag{10b}$$

and replace (8g) by

$$x_1(0) - \mathrm{x0} = 0. \tag{10c}$$

During the continuation process it may be of interest to determine the exact value of end time T where the solution jumps at the end time and additionally the condition (8e) is satisfied. In general this characterizes the crossing from a jump at the boundary to an interior jump. For that case the time horizon T is considered as a free variable and the condition

$$j^{N+1}(x_{N+1}(N+1), \mathrm{xT}, \lambda_{N+1}(N+1), \mathrm{lT}, T) = 0 \tag{11}$$

is appended to (9).

Initializing the BVP

To find the solution of a specific problem of type (1) we can apply a continuation strategy with respect to the time horizon T. Therefore, as a first step we have to determine an initial (optimal) solution.

Due to Assumption 1, the initial condition together with the transversality condition yield the necessary equations for $T = 0$. This solution can be used as a starting point for paths, which for a “small” time horizon do not exhibit a jumping point.

3.2 Continuation Algorithm

Let us consider the initial value problem (IVP) (8a) and (8b) on the time interval $[i-1, i]$ with

$$\dot{y}(s) = \Delta\tau_i h_1(y(s), \mu(s), t(s)), \quad s \in [i-1, i], \tag{12a}$$
$$\dot{\mu}(s) = \Delta\tau_i h_2(y(s), \mu(s), t(s)), \quad s \in [i-1, i]. \tag{12b}$$

With initial conditions

$$y(i-1) = x(\tau_i),\ \ \mu(i-1) = \lambda(\tau_i), \tag{12c}$$

the solution can formally be written as

$$\begin{aligned} y(i) - y(i-1) &= \Delta\tau_i \int_{i-1}^{i} h_1(y(s), \mu(s), t(s))\,\mathrm{d}s, \\ \mu(i) - \mu(i-1) &= \Delta\tau_i \int_{i-1}^{i} h_2(y(s), \mu(s), t(s))\,\mathrm{d}s, \end{aligned}$$

or even more general as an implicit equation

$$F(y(i-1), \mu(i-1), y(i), \mu(i), \tau_{i-1}, \tau_i) = 0.$$

To simplify notation, we introduce the following notation:

$$y_{2i} := \begin{pmatrix} x(\tau_i^-) \\ \lambda(\tau_i^-) \end{pmatrix} \quad y_{2i+1} := \begin{pmatrix} x(\tau_i^+) \\ \lambda(\tau_i^+) \end{pmatrix}, \quad 0 \le i \le N.$$

Then the system (8) can be stated as

$$\Omega_0(y_0, y_1, \tau_0) = 0 \in \mathbb{R}^{3n}, \tag{13a}$$

$$\Omega_{N+1}(y_{2N}, y_{2N+1}, \tau_{N+1}) = 0 \in \mathbb{R}^{3n}, \tag{13b}$$

$$\Omega_i = \Upsilon(y_{2i}, y_{2i+1}, \tau_i) = 0 \in \mathbb{R}^{2n+1}, \quad 1 \le i \le N, \tag{13c}$$

$$\Gamma_i = F(y_{2i+1}, y_{2(i+1)}, \tau_i, \tau_{i+1}) = 0 \in \mathbb{R}^{2n}, \;\; 0 \le i \le N, \tag{13d}$$

where (13a) denotes the initial condition, (13b) the transversality condition, (13c) the connecting condition for interior jumping points, and (13d) the solution of the IVP. Thus in total we have $8n + N(4n+1)$ equations ((13a) generates $3n$ equations, (13b) also generates $3n$ equations, (13c) generates $N(2n+1)$ equations, and finally (13d) generates $(N+1)2n$ equations) and the same number of unknowns $(y_0, \dots, y_{2(N+1)+1}, \tau_1, \dots, \tau_N)$ ($y_0, \dots, y_{2(N+1)+1}$ are $2n(2(N+1)+2)$ variables and $\tau_1, \dots, \tau_N$ are N variables, gives a total of $8n + N(4n+1)$ variables).

$$\Omega = [\Omega_0 \quad \Omega_1 \dots \Omega_N]' \in \mathbb{R}^{8n+N(2n+1)}, \tag{14a}$$

$$\Gamma = [\Gamma_0 \quad \Gamma_1 \dots \Gamma_{N+1}]' \in \mathbb{R}^{2nN}. \tag{14b}$$

If the IVP (12) can be solved explicitly, the formulation (14) has the advantage of reducing the infinite dimensional problem to a finite dimensional system of, in general, nonlinear equations, without discretizing the problem.

3.3 Gradient Algorithm

If the dynamics (1b) and the integral part of the objective function (1a) are simple enough to solve them explicitly, then the problem can be restated (without numerical discretization) as a finite dimensional problem. This can then be solved by some standard optimization algorithm, e.g., the numerical optimizer `fmincon` under MATLAB.

Problem (1) can be written as

$$\begin{aligned} &\max_{N,\tau_i,v^i} \sum_{i=0}^{N} \Gamma(x(\tau_i^+), x(\tau_{i+1}^-), t_i, t_{i+1}) + \\ &\sum_{i=1}^{N} \mathrm{e}^{-r\tau_i}\, G(x(\tau_i^-), v^i, \tau_i) + \mathrm{e}^{-rT}\, S(x(T^+)), \quad i = 0, \dots, N, \end{aligned} \tag{15a}$$

$$\text{s.t.} \quad x(t_{i+1}^-) = \Phi(x(t_i^+), t_i, t_{i+1}) \quad \text{for} \quad i = 0, \dots, N, \tag{15b}$$

$$x(\tau_i^+) - x(\tau_i^-) = g(x(\tau_i^-), v^i, \tau_i) \quad \text{for} \quad i = 1, \dots, N, \tag{15c}$$

$$x(0^-) = x_0 \in \mathbb{R}^n, \tag{15d}$$

with

$$\tau_k = t_k, \quad k \in \{0, 1, \dots, N, N+1\}, \quad t_{N+1} = T, \tag{15e}$$

$$\Gamma(x(t_i^+), x(t_{i+1}^-), t_i, t_{i+1}) = \int_{t_i}^{t_{i+1}} \mathrm{e}^{-rt}\, F(x(t), t)\, \mathrm{d}t, \tag{15f}$$

$$\Phi(x(t_i^+), t_i, t_{i+1}) = x(t_i^+) + \int_{t_i}^{t_{i+1}} f(x(t), t)\, \mathrm{d}t. \tag{15g}$$

Setting

$$y = (x(t_0^-), x(t_0^+), \dots, x(T^-), x(T^+), v^1, \dots, v^N, \tau_1, \dots, \tau_N)', \tag{16}$$

problem (15) becomes a finite dimensional maximization problem. To keep the notation simple, in a first step we subsequently assume that the jumps only occur within the interior of the interval $[t_0,\ T]$. Therefore $\tau_k = t_k,\ k = 1,\dots,N$ and $y \in \mathbb{R}^{4N+4}$ (i.e. y consists of $N+2$ left and $N+2$ right limits, N jumps, and N jump times). In that case the doubling (left and right limit) of the initial and end state is superfluous but allows an immediate generalization in case that a jump also occurs at the beginning or/and the end.

Next we derive the necessary optimality conditions, which, of course, reproduce the necessary optimality conditions from the Impulse Control Maximum Principle. First we start with the derivatives (gradients) of the equality constraints (15b)-(15d). In the new coordinates y_i these constraints become

$$
\begin{aligned}
&c_1 = y_1 - x_0 = 0,\\
&c_{2+k} = y_{2k+1} - y_{2k} - \Phi(y_{2k}, y_{2(N+2)+N+k}, y_{2(N+2)+N+k+1}) = 0,\ k = 0,\dots,N,\\
&c_{2+N+1+k} = y_{2(k+1)} - y_{2k+1} = 0,\ k = 0, N+1,\\
&c_{2+N+1+k} = y_{2(k+1)} - y_{2k+1} - g(y_{2k+1}, y_{2(N+2)+k}, y_{2(N+2)+N+k}) = 0,\ k = 1,\dots,N.
\end{aligned}
$$

Therefore the derivatives are calculated as

$$
\begin{aligned}
&\frac{\partial c_1}{\partial y_1} = 1,\\
&\frac{\partial c_{2+k}}{\partial y_{2k+1}} = 1,\\
&\frac{\partial c_{2+k}}{\partial y_{2k}} = -1 - \partial^{(1)}\Phi,\\
&\frac{\partial c_{2+k}}{\partial y_{2(N+2)+N+k}} = -\partial^{(2)}\Phi,\\
&\frac{\partial c_{2+k}}{\partial y_{2(N+2)+N+k+1}} = -\partial^{(3)}\Phi,\\
&\frac{\partial c_{2+N+1+k}}{\partial y_{2(k+1)}} = 1, \quad k = 0,\dots,N+1,\\
&\frac{\partial c_{2+N+1+k}}{\partial y_{2k+1}} = -1, \quad k = 0,\ N+1,\\
&\frac{\partial c_{2+N+1+k}}{\partial y_{2k+1}} = -1 - \partial^{(1)}g, \quad k = 1,\dots,N,\\
&\frac{\partial c_{2+N+1+k}}{\partial y_{2(N+2)+k}} = -\partial^{(2)}g, \quad k = 1,\dots,N,\\
&\frac{\partial c_{2+N+1+k}}{\partial y_{2(N+2)+N+k}} = -\partial^{(3)}g, \quad k = 1,\dots,N.
\end{aligned}
$$

Rewriting the objective function (15a) in the coordinates y we find

$$
\begin{aligned}
V \;=\;& \Gamma(y_1, y_2, y_{2(N+2)+N+1}, y_{2(N+2)+N+2})\\
&+ \sum_{i=1}^{N} \Gamma(x(t_i^+), x(t_{i+1}^-), t_i, t_{i+1})\\
&+ \sum_{i=1}^{N} \mathrm{e}^{-r\tau_i}\, G(x(\tau_i^-), v^i, \tau_i) + \mathrm{e}^{-rT}\, S(x(T^+)),
\end{aligned}
$$

and the derivatives are given as

$$\frac{\partial V}{\partial y_1} = 1.$$

For a thorough discussion and motivation we refer to Hou and Wong (2011).

In order to find the optimal solution using the gradient algorithm we need some information about the structure of the problem, i.e. have some knowledge about the optimal number of jumps. Neuman and Costanza (1990) use the value function approach and assume that for each initial state x, the value function V is well behaved, in the sense that there is an index k such that V_k (where V_k denotes the value function having k-jumps) is greater than the rest of the V_i's, the V_i's are nondecreasing for $i \leq k$ and monotonically decreasing for $i \geq k$. The main reason for this assumption is that this guarantees that only a finite number of steps is necessary to achieve the optimum.

To overcome this problem we use the solution provided by the continuation algorithm to initialize the gradient method approach. From numerical experiments we know that the continuation algorithm has provided the same (optimal) solution for impulse control problems solved using the backward algorithm, dynamic programming, or the value function approach. We have no proof that the algorithm converges or finds the optimal solution for all Impulse Control problems.

4 Two applications

4.1 A forest management model

To exemplify the numerical techniques we use a model described in Neuman and Costanza (1990) where the optimal solution for forest management is derived using impulse control. It consists, at time t, of one state $w(t) \in \mathbb{R}_+$ denoting the size of the forest and one impulse control $z^i \in \mathbb{R}_+$ denoting the size of the cut (of the forest). The dynamics of the forest is described by a logistic term $g(y(t))$. Forest growth is then presented by

$$\dot{w}(t) = g(w(t) := w(t)(a - bw(t)), \qquad t \geq 0,$$

with a and b positive constants and $w(t) \in \mathbb{R}_+$. At time zero the size of the forest is equal to some initial value, i.e.

$$w(0) = x \geq 0.$$

When management is imposed on forest evolution, the forest is cut at times $\tau_i \in \mathbb{R}_+$ ($i = 1, \ldots N$ with N the number of cuts) such that the size of the forest changes by:

$$w(\tau_i^+) - w(\tau_i^-) = z^i \quad \text{for} \quad i \in \{1, \ldots, N\}.$$

The total benefit generated by the dynamic system is given by

$$q(x) + \int_0^T f(w(s), s)e^{-rt}dt + \sum_{i=1}^{N} k(w(\tau_i), \tau_i, z^i)e^{-r\tau_i} + p(w(T^-)e^{-rT},$$

where $q(x)$ is the initial cost function, f is the profit function of the system per unit time, and k is the cost of the impulse z^i applied to the state $w(\tau_i)$ at time τ_i.

The impulse cost function is given by

$$k(w,\tau,z) = D + K(w,z) = D - g_0 z + g_1 z^2 \quad \text{with } D < 0,$$

where D can be considered as a fixed cost for cutting the forest and $K(w,z)$ being the partial profit generated by cutting the forest, g_0 and g_1 are some positive constants. If $z = 0$ we assume that $k(w,\tau,0) = 0$. The initial cost function is given by

$$q(x) = -q_0(x - x_0),$$

where q_0 is a positive constants and x_0 is some bound imposed on the states, due to either ecological or practical constraints. The profit of the system is given by a constant f_0, i.e.

$$f(w(t),t) = f_0,$$

with f_0 some positive constant. Finally, the salvage value is defined as

$$p(w(T^-)) = g_0(w(T^-) - x_0) - g_1(w(T^-) - x_0)^2.$$

Summing up, the optimal control problem can be written as

$$\begin{aligned}
&\max_{N,\tau_i,z^i} \Bigg\{ -q_0(x - x_0) + \int_0^T \mathrm{e}^{-rt} f_0 \,\mathrm{d}t \\
&\qquad + \sum_{i=1}^{N} \mathrm{e}^{-r\tau_i}(D - g_0 z^i) + \mathrm{e}^{-rT}(g_0(w(T^-) - x_0) - g_1(w(T^-) - x_0)^2) \Bigg\}, && \text{(17a)}\\
\text{s.t.}\quad &\dot{w}(t) = w(t)(a - bw(t)) \quad \text{for} \quad t \in [0,T] \setminus \{\tau_1, \dots, \tau_N\}, && \text{(17b)}\\
&w(\tau_i^+) - w(\tau_i^-) = z^i \le 0 \quad \text{for} \quad i \in \{1, \dots, N\}, && \text{(17c)}\\
&w(0^-) = x \ge 0. && \text{(17d)}\\
&w \in \mathbb{R}_+, \quad z^i \in (-\infty, 0] \quad \text{and} \quad \tau_i \in [0,T], && \text{(17e)}
\end{aligned}$$

where r denotes the discount rate. For the analysis of this model the Impulse Control Maximum Principle is used, where the details are presented in Appendix A.1.

4.2 Dike heightening problem

This section describes a problem taken from Chahim et al. (2012a) where the optimal timing of the heightening of a dike is studied. The cost-benefit-economic decision problem contains two types of cost, namely investment cost and cost due to damage (caused by failure of protection by the dikes). It consist, at time t, of one state $H(t) \in \mathbb{R}_+$ denoting the height of the dike relative to the initial situation, i.e. $H(0) = 0$ (cm) and one impulse control variable v^i denoting the i-th dike heightening of the dike. It is assumed that between two heightenings the dike height does not change, i.e. the dynamics of the dike are presented by

$$\dot{H}(t) = 0.$$

The dike increases at times $\tau_i \in \mathbb{R}_+$ ($i = 1, \dots N$, with N the number of heightenings) such that the height of the dike is increased by

$$H(\tau_i^+) - H(\tau_i^-) = v^i \quad \text{for} \quad i \in \{1, \dots, N\}.$$

The objective consist of two parts. The first part is the total (discounted) expected damage cost, which is given by

$$\int_0^T S(t)e^{-rt}dt + \frac{S(T)e^{-rT}}{r},$$

where $S(t)$ denotes the expected damage at time t, i.e. $S(t) = P(t)V(t)$, where $P(t)$ stands for the flood probability and $V(t)$ the damage of a flood (million €) at time t. The flood probability $P(t)$ (1/year) in year t is defined as

$$P(t) = P_0 e^{\alpha\eta t} e^{-\alpha H(t)}, \tag{18}$$

where α (1/cm) stands for the parameter in the exponential distribution regarding the flood probability, η (cm/year) is the parameter that indicates the increase of the water level per year, and P_0 denotes the flood probability at $t = 0$. The damage of a flood V(t) (million €) is given by

$$V(t) = V_0 e^{\gamma t} e^{\zeta H(t)}, \tag{19}$$

in which γ (per year) is the parameter for economic growth, and ζ (1/cm) stands for the damage increase per cm dike height. V_0 (million €) denotes the loss by flooding at time $t = 0$. The second part of the objective is the total (discounted) investment cost

$$\sum_{i=1}^{N} I(v^i, H(\tau_i^-))e^{-r\tau_i},$$

where $H(\tau^-)$ denotes the height of the dike (in cm) just before the dike update at time τ (left-limit of $H(t)$ at $t = \tau$). The investment cost is given by

$$I(v^i, H(\tau^-)) = \begin{cases} a_0(H(\tau^-) + v^i)^2 + b_0 v^i + c_0 & \text{for } v^i \neq 0 \\ 0 & \text{for } v^i = 0, \end{cases}$$

for suitably chosen constants a_0, b_0 and c_0. Summing up, the Impulse Control model can be written as

$$\begin{aligned}
& \min_{v^i,\tau_i,N} \Big(\int_0^T S(t)e^{-rt}dt + \sum_{i=1}^{N} I(v^i, H(\tau_i^-))e^{-r\tau_i} + e^{-rT}\frac{S(T)}{r}, && \text{(20a)} \\
\text{s.t.}\quad & \dot{H}(t) = 0, \quad \text{for} \quad t \in [0,T] \setminus \{\tau_1, \ldots, \tau_N\}, && \text{(20b)} \\
& H(\tau_i^+) - H(\tau_i^-) = v^i > 0, \quad \text{for} \quad i \in \{1, \ldots, N\}, && \text{(20c)} \\
& H(0^-) = H_0 = 0, && \text{(20d)} \\
& H \in \mathbb{R}_+, \quad v^i \in [0, \infty) \quad \text{and} \quad \tau_i \in [0,T]. && \text{(20e)}
\end{aligned}$$

For the analysis of this model the impulse control maximum principle is used, where the details are carried out in Appendix A.2. For an extensive description of the model we refer to Chahim et al. (2012a).

5 Numerical results

In this section we present results for two different applications using the continuation algorithm and make a comparison with results derived using other approaches.

5.1 The forest model

In this section we present the results for the optimal forest management problem described in the previous section. The parameter value presented in Table 1 are taken from Neuman and Costanza (1990).

r	a	b	D	f_0	g_0	g_1	q_0	x_0	y_0	T
0.05	0.2059	0.00344	-190	-15	24.5	0	40	5	34.4	8

Table 1: Parameter values for the optimal forest management model

τ	z_i	$w(\tau^-)$	$w(\tau^+)$
0	0	34.4	34.4
1	-24.2	37.35	13.08
8	0	32.43	32.43
Discounted revenue		-441.1751	

Table 2: Result of value function approach found in Neuman and Costanza (1990)

τ	z_i	$w(\tau^-)$	$w(\tau^+)$
0	0	34.400	34.400
0.8216	-23.5757	36.8383	13.2626
8	0	33.290	33.290
Discounted revenue		-438.2973	

Table 3: Result of the continuation algorithm

The results we derive using the continuation algorithm are presented in Table 3. The results of Table 3 are similar to the results found in Neuman and Costanza (1990) presented in Table 2. The continuation algorithm (same holds for BVP algorithm) has two advantages over the value function approach described in Neuman and Costanza (1990). First, we do not have to discretize the time horizon. This results in a better objective value and hence a better solution to the original problem. In Figure 1 we plot the size of the forest as a function of time. Initially, the size of the forest increases, then at a some time instance the forest is cut. Hence, the size of the forest jumps downward and then grows again. Second, we did not have to solve the problems for different number of cuts to find the optimal solution to our forest management problem.

5.2 The dike heightening model

In this section we will present the optimal solution for a dike. The parameter value presented in Table 4 are taken from Eijgenraam et al. (2011). In Table 5 the solution for three different approaches are presented. In the second column the results for the continuation algorithm are given, the third column presents the results found by the backward algorithm used in Chahim et al. (2012b), and finally in the fourth column the results for dynamic programming (DP) are given taken from Eijgenraam et al. (2011).

Unlike, dynamic programming, both the continuation algorithm and the backward algorithm do

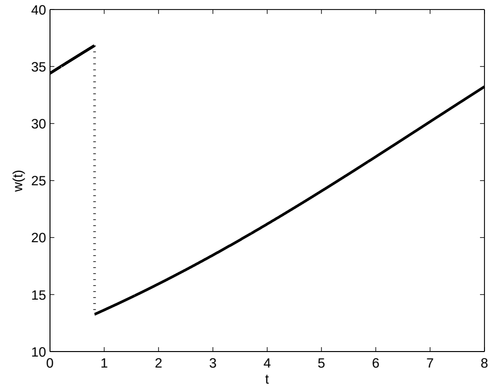

Figure 1: Dynamics of the forest over time

a_0	b_0	c_0	V_0	r	P_0
0.0014	0.6258	16.6939	1564.9	0.04	1/2270
H_0	α	η	γ	ζ	T
0	0.33027	0.32	0.02	0.003774	300

Table 4: Parameter values for dike 10

not need to discretize time. However, for the initialization of the backward algorithm, we need the discretization of the state at the end of the time horizon (final stage), i.e $H(T)$ and dynamic programming requires the discretization of time and of the heights (states) for each stage. The continuation algorithm does not need any input on the state variable $H(T)$. Even though the solutions for the backward algorithm and the continuation algorithm are similar, the continuation algorithm (same holds for the BVP approach) finds the optimal solution without running the algorithm for different end heights $H(T)$. In Chahim et al. (2012a) the authors discretize the state variable as is required for the dynamic programming approach in Eijgenraam et al. (2011) and take that $H(T)$ that minimizes the 20a.

Approach[a]	BA	DP	CA
$(\tau_i : u_i)$	272.8 : 52.18	274 : 51.84	272.7 :52.21
	217.0 : 56.43	219 : 55.68	217.0 :56.45
	160.1 : 56.90	162 : 57.60	160.0:56.90
	103.0 : 56.95	104 : 57.60	103.0 :56.96
	45.9 : 56.96	46 : 57.60	45.8 :56.96
$H(T)$	279.41	280.32	279.48
Total cost	40.03	40.04	40.03

[a] Backward algorithm (BA), dynamic programming (DP), and continuation algorithm (CA)

Table 5: Results for dike 10.

6 Conclusions and Recommendations

We described three different numerical methods to solve Impulse Control problems. The first (new) algorithm considers an Impulse Control problem as a (multipoint) Boundary Value Problem and uses a continuation technique to solve it. The second (new) approach is the continuation algorithm that requires the canonical system to be solved explicitly. This reduces the infinite dimensional problem to a finite dimensional system of, in general, nonlinear equations, without discretizing the problem. The third algorithm we present is a gradient algorithm, where we reformulate the problem as a finite dimensional problem, which can be solved using some standard optimization techniques. We used the continuation algorithm to solve the optimal forest management problem (same results found for the boundary value problem approach) and the dike heightening problem. Although numerical results found by the continuation algorithm (same holds for the boundary value problem approach) are at least as good as the results found in the literature, a formal proof for the boundary value problem approach and the continuation algorithm finding the optimal solution is subject for future research.

Appendix A Necessary Optimality Conditions for the applications

A.1 The forest management model

Let us define the current value Hamiltonian

$$\mathcal{H}(w(t), \lambda(t), t) := f_0 + \lambda(t)w(t)(a - wy(t)), \tag{21}$$

and the current value Impulse Hamiltonian

$$\mathcal{IH}(z^i, \lambda(t), t) := D - g_0 z_i + g_1 z_i^2 + \lambda(t) z^i. \tag{22}$$

We obtain the adjoint equation

$$\dot{\lambda}(t) = (r - a + 2bw(t))\lambda, \tag{23}$$

with the transversality condition

$$\lambda(T) = g_0 - 2g_1 w(T^-). \tag{24}$$

The jump conditions are

$$-g_0 + g_1 z_i + \lambda(\tau_i^+) = 0, \quad \text{for } i = 1, \ldots, N \tag{25}$$

$$\lambda(\tau_i^+) - \lambda(\tau_i^-) = 0, \quad \text{for } i = 1, \ldots, N \tag{26}$$

from which we can conclude that the costate $\lambda(t)$ is continuous at every jump point. The condition for determining the optimal switching time τ_i is

$$\begin{aligned} &\lambda(\tau_i^+)w(\tau_i^+)(a - bw(\tau_i^+)) - \lambda(\tau_i^-)w(\tau_i^-)(a - bw(\tau_i^-)) \\ &\quad + rD - rg_0 z_i + rg_1 z_i^2 \begin{cases} > 0 & \text{for } \tau_i^* = 0 \\ = 0 & \text{for } \tau_i^* \in (0, T) \\ < 0 & \text{for } \tau_i^* = T. \end{cases} \end{aligned} \tag{27}$$

A.2 The dike heightening model

Let us define the current value Hamiltonian

$$\mathcal{H}(t, H(t)) = -S_0 e^{\beta t} e^{-\theta H(t)}, \tag{28}$$

and the current value Impulse Hamiltonian

$$\mathcal{IH}(t, H(\tau^-), v^i, \lambda(t)) = -I(v^i, H(\tau_i^-)) + \lambda(t)v^i = -A_0(H(\tau^-) + v^i)^2 \\ -b_0 v^i - c_0 + \lambda(t)v^i, \tag{29}$$

and obtain the adjoint equation

$$\dot{\lambda}(t) = r\lambda(t) - \theta S_0 e^{\beta t} e^{-\theta H(t)}, \tag{30}$$

with the transversality condition

$$\lambda(T) = \frac{\theta S_0 e^{\beta T} e^{-\theta H(T)}}{r}. \tag{31}$$

The jump conditions are

$$-I_u\left(u_i, H\left(\tau_i^-\right)\right) + \lambda\left(\tau_i^+\right) = 0 \quad \text{for } i = 1, \ldots, N, \tag{32}$$

$$\lambda(\tau_i^+) - \lambda(\tau_i^-) == I_H\left(u_i, H(\tau_i^-)\right) \quad \text{for } i = 1, \ldots, N \tag{33}$$

The condition for determining the optimal switching time τ_i is

$$S_0 e^{\beta\tau_i}\left(e^{-\theta H(\tau_i^-)} - e^{-\theta H(\tau_i^+)}\right) - rI\left(u_i, H(\tau_i^-)\right) \begin{cases} > 0 & \text{for } \tau_i^* = 0 \\ = 0 & \text{for } \tau_i^* \in (0, T) \\ < 0 & \text{for } \tau_i^* = T. \end{cases} \tag{34}$$

Appendix B Implementation in MATLAB

For the subsequents sections we assume that a solution of (1) and time horizon T has already been detected given by $(x^*(\cdot), v_i^*, \tau_i),\ i = 1, \ldots, N$ with $0 < \tau_1 < \tau_2 < \ldots < \tau_N < T$. In the first section we consider the case where a solution of the canonical system between two adjacent jumps can analytically be found. Therefore the problem can be reduced to a finite number of nonlinear equations, see Sect. **??**.

B.1 Continuation Algorithm

For the actual implementation in MATLAB a vector x is introduced

$$x = (y(\tau_1^-), y(\tau_1^+), \ldots, y(\tau_N^-), y(\tau_N^+), \tau_1, \ldots, \tau_N)' \tag{35a}$$

with

$$y(t) := (x^*(t), \lambda(t)). \tag{35b}$$

This vector consists of the left and right side limits of the states and costates at the jumping times and the (interior) jumping times appended at the end. To continue the solution along a parameter value, the initial states or time horizon `MATCONT` is used. Therefore the main `MATCONT` file, where the system is defined has to be provided.

```
function out = iocmodelDiscrete4matcont
%
% Standard ode file for MATCONT

out{1} = @init;
out{2} = @fun_eval;
```

```
out{10}= @interiorjumpfunc;
out{11}= @reachtimehorizon;
out{12}= @jumpingtimesvstimehorizon;
out{13}= @negativetime;
%
% -------------------------------------------------------------------------
function out = fun_eval(t,geny,x0,par,T)
global GIV
aid=GIV.aid;
arcnum=GIV.arcnum;
jid=GIV.jid;
y=geny(GIV.gDVC);
tp=[GIV.IT geny(GIV.JTC).' T];

initres=[];
transres=[];
connecres=[];
dynres=[];
interiorjumpres=[];
for ii=1:arcnum+1
    yLR=y(:,(2*ii-1):2*ii);
    if ii==1
        initres=GIV.IC(tp(ii),yLR,[par,T],aid(1),x0);
    elseif ii==arcnum+1
        transres=GIV.TC(tp(ii),yLR,[par,T],aid(end));
    end
    connecres=[connecres; ...
        GIV.JC(tp(ii),yLR,[par,T],jid(ii))];
    if arcnum>1 && ii>=2 && ii<=arcnum
        interiorjumpres=[interiorjumpres; ...
            GIV.IJC(tp(ii),yLR,[par,T],aid(ii-1),jid(ii))];
    end
    if ii<=arcnum
        yI=y(:,2*ii:(2*ii+1));
        dynres=[dynres; ...
            GIV.CS(tp(ii:ii+1),yI,[par,T],aid(ii))];
    end
end
out=[initres;transres;connecres;dynres;interiorjumpres];

%-------------------------------------------------------------------------
function out=interiorjumpfunc(t,geny,x0,par,T)
global GIV
aid=GIV.aid;
arcnum=GIV.arcnum;
jid=GIV.jid;
y=geny(GIV.gDVC);
yLR=y(:,(2*(arcnum+1)-1):2*(arcnum+1));
```

```
if jid(end)
    out=GIV.IJC(T,yLR,[par,T],aid(end),jid(end));
else
    out=1;
end

%---------------------------------------------------------------------------
function out=reachtimehorizon(t,geny,x0,par,T)
global GIV

out=GIV.TH-T;

%---------------------------------------------------------------------------
function out=jumpingtimesvstimehorizon(t,geny,x0,par,T)
global GIV
tp=[geny(GIV.JTC)];

if isempty(tp)
    out=1;
else
    out=min(T-tp);
end

function out=negativetime(t,geny,x0,par,T)
global GIV

%---------------------------------------------------------------------------
out=min([geny(GIV.JTC);T]);
```

Abbreviations

```
GIV=GlobalImpulseVariable
genDynVarCoordinates=gDVC
InitialTime=IT
JumpTimeCoordinates=JTC
TimeHorizon=TH
InteriorJumpCondition=IJC
CanonicalSystem=CS
TransversalityCondition=TC
```

The function fun_eval file defines the ascribing equations. These equations are stated in model specific functions and the function names are defined in the global variable GIV. The fields of the global variable GIV

arcnum the number of arcs $y(t),\ t \in [\tau_i, \tau_{i+1}],\ i = 0, \ldots, N$ between two adjacent jumping times.

jumparg (jid) an integer vector storing an identifier for each jump. The first and last entry denotes if a jump at the initial or end time occurs. If no jump occurs it is set to zero, otherwise to some integer larger than zero.

InitialTime (IT) stores the initial time t_0.

TimeHorizon (TH) stores the time horizon of the problem T.

CanonicalSystem (CS) function where the canonical system is described.

InteriorJumpCondition (IJC) function for the interior jumping condition (7e).

TransversalityCondition (TC) function for the transversality condition (7f).

genDynVarCoordinates (gDVC) the matrix of coordinates for the left and right side limits of the states and costates of vector x.

JumpTimeCoordinates (JTC) the coordinates of vector x storing the jumping times.

Further variables used in the listing

geny variable denoting x of (35a).

y matrix, where the column consist of $y(\tau_i^\pm),\ i = 1, \ldots, N$, as defined in (35b).

yLR the left and right side limits $y(\tau^\pm)$ at a specific jumping time τ.

yI the two column matrix consisting of the right side limit $y(\tau_i^+)$ and the left side limit of the next jumping time $y(\tau_{i+1}^+)$.

tp a vector consisting of the initial time, jumping times and time horizon.

x0 is a vector of the initial states $x(0)$.

par is a vector of the parameter values of the model.

T is the actual time horizon, which need not be equal to the time horizon of the problem stored in GIV.TH.

initres residuum of the initial condition.

transres residuum of the transversality condition.

connecres residuum of the connection between two adjacent arcs.

dynres residuum derived from the equations of the canonical system.

interiorjumpres residuum derived from the interior jumping conditions.

The user functions used within the `MATCONT` syntax

interiorjumpfunc returns the value of the interior jumping condition at jumping times. This value is monitored during the continuation process. If it changes sign the necessary jumping condition for an interior jump is satisfied and an interior jump may occur.

reachtimehorizon if the continuation is done with respect to the time horizon this value is monitored to check if the final time horizon is reached.

B.2. Gradient Algorithm

To solve problem (15) numerically the MATLAB function `fmincon` can be used. Therefore a file describing the objective function and its derivative has to be provided together with a file describing the constraints and the corresponding derivatives. The syntax (as we need it) of the function is

```
x = fmincon(fun,x0,A,b,[],[],lb,ub,nonlcon,opt),
```

where `fun` and `nonlcon` denote the files for the objective function and (nonlinear) constraints, respectively. To apply the gradient algorithm lower and upper bounds of the vector y have to be provided `ub` and `vb`. These bounds should be chosen in a way that the interesting state and control space is covered. If during the calculations the bounds are hit one can increase the bounds to stay in the interior. furthermore we can assure that the jumping times are ordered and do not exceed the time horizon. These are linear inequalities

$$\tau_1 - \tau_2 \leq 0, \ldots, \tau_{N-1} - \tau_N \leq 0,$$

which can be provided by a matrix inequality of the form $Ay \leq 0$. The vector `x0` is some approximated solution of the problem. For the gradient algorithm the options have at least to consist of

```
opt=optimset('GradObj','on','GradConstr','on');
```

The m-file for the constraints has to return a vector for (nonlinear) equality and inequality constraints and the corresponding derivatives. If the problem does not consist of inequality constraints empty vectors have to be returned.

References

Augustin, D. (2002). *Sensitivitätsanalyse und Echtzeit-Steuerung optimaler Steuerprozesse mit Zustandbeschränkungen sowie für Multiprozess- und Impulsprobleme.* PhD thesis, Westfälische Wilhelms-Universität, Münster.

Blaquière, A. (1977a). Differential games with piece-wise continuous trajectories. In Hagedorn, P., Knobloch, H., and Olsder, G., editors, *Diffential Games and Applications*, pages 34–69. Springer-Verlag, Berlin.

Blaquière, A. (1977b). Necessary and sufficient conditions for optimal strategies in impulsive control and application. In Aoki, M. and Morzzolla, A., editors, *New trends in Dynamic System Theory and Economics*, pages 183–213. Academic Press, New York.

Blaquière, A. (1979). Necessary and sufficient conditions for optimal strategies in impulsive control. In Lui, P. and Roxin, E., editors, *Differential Games and Control Theory III, Part A*, pages 1–28. Marcel Dekker, New York.

Blaquière, A. (1985). Impulsive optimal control with finite or infinite time horizon. *Journal of Optimization Theory and Applications*, 46(4):431–439.

Chahim, M., Brekelmans, R. C. M., den Hertog, D., and Kort, P. M. (2012a). An impulse control approach for dike height optimization. *To appear in Optimization Methods and Software.*

Chahim, M., Hartl, R., and Kort, P. (2012b). A tutorial on the deterministic impulse control maximum principle: Necessary and sufficient optimality conditions. *European Journal of Operational Research*, 219(1):18–26.

Eijgenraam, C. J. J., Brekelmans, R. C. M., den Hertog, D., and Roos, C. (2011). Safe dike heights at minimal costs: the homogenous case. Working Paper, Tilburg University, Tilburg.

Erdlenbruch, K., Jean-Marie, A., Moreaux, M., and Tidball, M. (2011). Optimality of impulse harvesting policies. *Economic Theory*, pages 1–31.

Hou, S. and Wong, K. (2011). Optimal impulsive control problem with application to human immunodeficiency virus treatment. *Journal of Optimization Theory and Applications*, 151(2):385–401.

Kort, P. (1989). *Optimal Dynamic Investment Policies of a Value Maximizing Firm*. Springer-Verlag.

Liu, Y., Teo, K., Jennings, L., and Wang, S. (1998). On a class of optimal control problems with state jumps. *Journal of Optimization Theory and Applications*, 98(1):65–82.

Luhmer, A. (1986). A continuous time, deterministic, nonstationary model of economic ordering. *European Journal of Operations Research*, 24(1):123–135.

Neuman, C. and Costanza, V. (1990). Deterministic impulse control in native forest ecosystems management. *Journal of Optimization Theory and Applications*, 66(2):173–196.

Rempala, R. (1990). Lecture notes in control and information sciences. In Sebastian, H. and Tammer, K., editors, *System Modelling and Optimization*, pages 387–393. Springer Berlin / Heidelberg, Berlin.

Seierstad, A. (1981). Necessary conditions and sufficient conditions for optimal control with jumps in the state variables. Memorandum from instutute of economics, University of Oslo, Oslo.

Seierstad, A. and Sydsæter, K. (1987). *Optimal Control Theory with Economic Applications*. Elsevier Science Pub. Co., Amsterdam.

Wu, C. and Teo, K. (2006). Global impulsive optimal control computation. *Journal of Industrial and Management Optimization*, 2(4):435–561.