

A simulation-and-regression approach for stochastic dynamic programs with endogenous state variables*

Michel Denault[†] Jean-Guy Simonato[‡] Lars Stentoft[§]

May 2011

Abstract

We investigate the optimum control of a stochastic system, in the presence of both exogenous (control-independent) stochastic state variables and endogenous (control-dependent) state variables. Our solution approach relies on simulations and regressions with respect to the state variables, but also grafts the endogenous state variable into the simulation paths. That is, unlike most other simulation approaches found in the literature, no discretization of the endogenous variable is required. The approach is meant to handle several stochastic variables, offers a high level of flexibility in their modeling, and should be at its best in non time-homogenous cases, when the optimal policy structure changes with time. We provide numerical results for a dam-based hydropower application, where the exogenous variable is the stochastic spot price of power, and the endogenous variable is the water level in the reservoir.

Keywords: stochastic control, approximate dynamic programming, simulation and regression, least-squares Monte Carlo, hydropower management.

1 Introduction

We propose a new solution approach for stochastic dynamic control problems involving both exogenous and endogenous state variables, i.e. control-independent and control-dependent variables. The approach relies on simulations and regressions (also known as LSMC, for Least Squares and Monte Carlo) to approximate the value function and the underlying conditional expectations.

*The authors are thankful to Michèle Breton, Matt Davison, Pierre L'Écuyer and Ilya Ryshov for their helpful comments, and to Mathieu Rousseau and Sofiane Tafat for their programming help. They gratefully acknowledge funding from HEC Montréal (Denault, Simonato) and NSERC (Denault).

[†]HEC Montréal (Management Sciences) and GERAD. Corresponding author.

[‡]HEC Montréal (Finance)

[§]HEC Montréal (Finance)

Stochastic control problems are found in a wide variety of engineering and management applications, from portfolio optimization to avionics and from gas storage to supply chain management. The main features of the type of problems we consider, are explained in terms of the control (the decisions) and the two types of state variables:

- the evolution of the endogenous variable through time depends on the decisions;
- there is great value in the making the right decisions at the right time;
- the decisions now have an impact on the availability of decisions (options) later;
- the decisions are made with incomplete information about the future, given the stochastic, exogenous state variables.

Recently, gas storage has been an important development vector for stochastic dynamic programs with endogenous and exogenous variables (most authors explicitly recognize the applicability of their methods beyond gas storage). The endogenous variable is the level of gas in storage and the exogenous variable is the gas price. We cite for now Chen and Forsythe ([CF07] and [CF10]), and Thomson, Davison and Rasmussen [TDR09], who solve the control problems through partial differential equations techniques. Such methods are typically efficient but less flexible than simulation-based techniques.

The use of simulations and regressions to solve stochastic DP problems goes back at least to Keane and Wolpin [KW94]. The approach became very popular in financial engineering after the papers of Longstaff and Schwartz [LS01] and Tsitsiklis and Van Roy [TVR01], who set option pricing as an optimal stopping time problem.

Generalizing from financial option pricing, the method of simulations and regressions was extended to gas storage by Boogert and de Jong [BdJ08] and Carmona and Ludkovsky [CL10]. In opposition to our approach, Boogert and de Jong discretize the gas level, and run a separate regression for each gas level. Carmona and Ludkovsky introduce a “quasi-simulation” of the endogenous variable and regress on both the exogenous and the endogenous variables; this is also the starting point of our approach. Our algorithm differs from that of [CL10] on three main counts. First, we build forward-optimal endogenous variable paths by relying on existing policies, while the authors do a guessing step but provide little detail on guiding the guess. Second, we propose a method to account for the impact of upper and lower bounds of the endogenous variable on the regressions. Finally, we suggest a rather natural approach to solve the problem of *clustering*, cf. [CL10, p. 366].

Stochastic dynamic programming methods based on simulations and regressions fall in the broad area of Approximate Dynamic Programming, that is, dynamic programming methods that rely on approximations of the value function or the policies. Two main references for ADP are the books of Bertsekas [Be07] and Powell [Po11]. An approximate dynamic programming method without either parametric function fitting nor simulation is proposed by Lai, Margot and Secomandi [LMS10] for the gas storage problem; see also the related paper by Secomandi [S10]. Nascimento and Powell (2009) [NP09] bring another approach, completely in the ADP line and using simulations and parametric function approximations

but still quite different from the papers mentioned above. Indeed, the authors do not work on a full set of simulation paths, but rather rely on iterative improvements of the value function.

Although the solution method we propose applies in general to stochastic control problem with exogenous variables and (bounded) endogenous variables, we refer throughout the paper to the management of a dam-based hydropower system to support the intuition. Our motivation to depart from gas storage is that multi-scale (short and long term) storage and supplementary stochastic variables such as water inflows are natural features of the hydro problem, to be explored in subsequent work.

The approach that we propose can apply to a broad number of problems. In production and manufacturing, consider supply chain management, or oil and mine exploitation. Moving away from the storage of physical assets, one could take capital as the endogenous variable, and tackle portfolio management optimization, or underground exploration rights acquisition.

In the next section, we set the problem up. We describe our approach in section 3, and briefly discuss convergence results from the literature in section 4. Numerical evidence is presented in section 5; the paper concludes with section 6. Appendix A provides the details of our Markov-chain benchmark approach.

2 Problem Description

In this section, we describe the problem’s main characteristics, indicate its underlying hypotheses and set the notation up.

We consider a stochastic control problem where a state variable varies (deterministically) with time according to the control choices that are made; we call such a variable “endogenous”. The payoff during each period of time is determined by the control choice and a stochastic, exogenous state variable. Although the set up is completely general, we believe it helps the intuition to refer to specific variables in describing the method. We will thus consider throughout the paper the case of a dam-based hydropower producer: the endogenous variable is the amount of energy (as water) behind the dam, the exogenous variable is the spot price of power, and the control is the amount of power to produce and sell at each moment. Improvements to this bare-bones case are discussed later.

We shall frame our problem within a finite horizon, with a discrete number of time periods during which neither the exogenous variable nor the control decision can change. The control space is also finitely discretized.

The components of our model can be defined as follows:

Time as a moment is represented by $t \in [0, T]$, though t is also used to represent the *period* from (time point) t to $t + 1$, a duration we simply call “one time period”. Context should relieve any possible confusion between time point and time period.

The exogenous state variable S_t , which can be thought of as a spot power price during period t . It is revealed at the beginning of each period.

The endogenous state variable L_t , which can be thought of as the amount of power (contained as water) in the reservoir. Simple operational constraints are given by bounds $L_{\min} \leq L_t \leq L_{\max}$: control decision that would bring L_t beyond those bounds are not allowed.

The above are the state variables of our dynamic programming formulation. The control variable is u , defined immediately; with it, we can also define the payoff associated to control decisions.

Control variable u_t directly influences the endogenous variable level, and partly determines the payoff. As mentioned, not all controls may be available, depending on the endogenous variable level. We denote by $\mathcal{U}(S, L, t)$ the set of admissible regimes when starting with state values (S, L) and for period t . The control can be thought of as the amount of power produced and sold. In the simplest case, we consider only three regimes and $u_t \in \{+1, 0, -1\}$. Depending on the application, u_t may or may not be sign-constrained; we come back later to the meaning of a negative u_t in our dam context.

Payoff $\pi_t(u; S, L)$ is the cashflow obtained when running regime u during period t , starting with state variables values (S, L) . (Throughout the paper, we leave aside time discounting, without loss of generality). A typical payoff function would be

$$\pi_t(u_t; S_t, L_t) = qu_t S_t \phi(L_t)$$

where function ϕ can be used to account for operational constraints on the endogenous variable, and q is used to scale the unitless decision value into an amount of energy.

and to bring some flexibility to the decision variable through the fixed scalar q .

We can now set an objective for the control problem, which is to find the *value function*

$$V_t(S, L) \equiv \sup_{u \in \mathcal{U}(S, L, t)} \mathbb{E} \left\{ \sum_{s=t}^{T-1} \pi_s(u; S_s, L_s) + W(S_T, L_T) \mid (S_t, L_t) = (S, L) \right\} \quad (1)$$

where $W(S, L)$ is the terminal value of the system with the given values of the state variables. The deterministic state variable L_t abides the state equation

$$L_{t+1} = h(u_t; L_t); \quad (2)$$

note that given time- t information, the level of L_{t+1} is known. The stochastic state variable follows a Markov transition probability distribution $p_S(S_{t+1}|S_t)$.

The underlying optionality of the problem is as follows. The level of the endogenous variable influences which control decisions, and thus which payoffs, are available. But the control decision at time t influences the endogenous variable level for all future time periods, making some future payoffs unavailable. For example, water not used now may be used later, but at a power price unknown at the present time.

To account for this optionality, we use the principle of optimality of dynamic programming to rewrite the value function (1) in a recursive manner:

$$V_t(S_t, L_t) = \sup_{u \in \mathcal{U}(S_t, L_t, t)} \left\{ \pi_t(u; S_t, L_t) + \mathbb{E}_t \left\{ V_{t+1}(S_{t+1}, h(u; L_t)) \right\} \right\} \quad (3)$$

where the subscript on \mathbb{E}_t indicates that the expectation is taken with respect to the information at time t on random variable S_{t+1} .

3 Solution Approach

In this section, we describe our solution approach to solve the dynamic programming recursion equation (3).

3.1 A simulation and regression approach

We need to mark the difference between the *exogenous* state variable S and the *endogenous* state variable L . A standard dynamic programming approach is to discretize all state variables; it is well-known to suffer from the curse of dimensionality. Another approach, inspired by papers such as that of Longstaff and Schwartz [LS01], is to use a combination of Monte Carlo simulations and least-squares regressions (“LSMC”) on the exogenous state variable, to handle both the conditional expectation and the value function approximation problems. A standard way to adapt the LSMC approach to an endogenous state variable L_t would be to discretize it, and simulate the exogenous variable: see Boogert and de Jong [BdJ08]. While numerically feasible, the efficiency of this approach suffers from the discretization process. Note that the state variable L_t cannot be simulated like the exogenous state variable, as it depends on the control. A numerically appealing approach is to try to determine appropriate values of L_t for each simulated scenario, backwards from the time horizon end T , see Carmona and Ludkovski [CL10] who call “quasi-simulation” this treatment of the endogenous variable. This idea allows one to regress on both state variables and avoid any discretization of state variables. Our main contributions with respect to [CL10] are three-fold. We propose a specific approach to “quasi-simulation” path building; we directly account for the impact of upper and lower bounds on the storage level; and we propose a “backwash” approach to alleviate the storage levels randomization steps which break apart the optimality of decisions for all paths at once, when applied (see the end of section 4.1 of [CL10]).

The recursive solution first proceeds as follows. Assume that K scenarios (paths) for the exogenous variable have been generated: $(S_0^{(k)}, \dots, S_T^{(k)})$, for $k = 1, \dots, K$. Let the value function at $t + 1$

$$V_{t+1} \left(S_{t+1}^{(k)}, L_{t+1}^{(k)} \right)$$

be known for each scenario $k = 1, \dots, K$; the origin of $L_{t+1}^{(k)}$ is discussed below.

Define a time-reversed state equation

$$\overleftarrow{h}(u_t; L_{t+1}) = L_t$$

and define a time- t , control-dependent level $L_t^{(k)}(u_t)$,

$$L_t^{(k)}(u_t) = \overleftarrow{h}\left(u_t; L_{t+1}^{(k)}\right) \quad (4)$$

for each possible control $u_t \in \{+1, 0, -1\}$ and scenario $k = 1, \dots, K$. These are the possible time- t levels, given the time- $(t+1)$ levels, for each scenario.

Then regress the following values, K in each of the three cases,

$$\pi_t\left(+1; S_t^{(k)}, L_t^{(k)}(+1)\right) + V_{t+1}\left(S_{t+1}^{(k)}, L_{t+1}^{(k)}\right) \quad \text{on} \quad \left(S_t^{(k)}, L_t^{(k)}(+1)\right) \quad (5)$$

$$\pi_t\left(0; S_t^{(k)}, L_t^{(k)}(0)\right) + V_{t+1}\left(S_{t+1}^{(k)}, L_{t+1}^{(k)}\right) \quad \text{on} \quad \left(S_t^{(k)}, L_t^{(k)}(0)\right) \quad (6)$$

$$\pi_t\left(-1; S_t^{(k)}, L_t^{(k)}(-1)\right) + V_{t+1}\left(S_{t+1}^{(k)}, L_{t+1}^{(k)}\right) \quad \text{on} \quad \left(S_t^{(k)}, L_t^{(k)}(-1)\right) \quad (7)$$

Note that “regressing on pairs $\left(S_t^{(k)}, L_t^{(k)}(u_t)\right)$ ” is taken to mean that we regress on a basis formed with the pairs. Figure 1 illustrates the regression procedure, displaying the endogenous variable levels at play, and the values that serve as input to the regression.

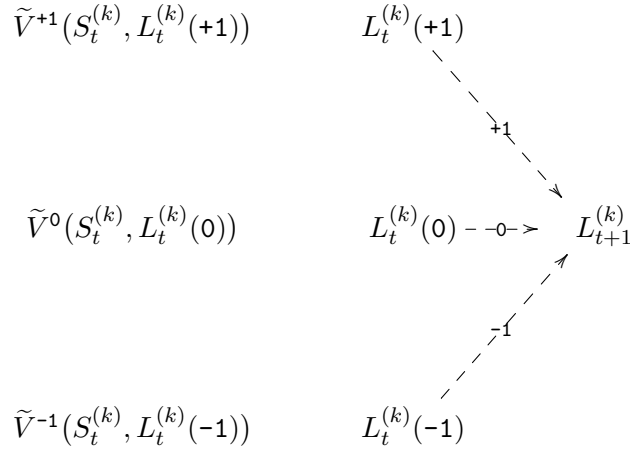


Figure 1: **Regression procedure.** On the right, a graphical view of possible endogenous variable levels $L_t^{(k)}$, given the time- $t+1$ level $L_{t+1}^{(k)}$ (for a specific path k). On the left, the values that enter the regression procedure.

We thus obtain three regression surfaces, denoted $\tilde{V}^{+1}(S, L)$, $\tilde{V}^0(S, L)$, $\tilde{V}^{-1}(S, L)$, corresponding to each control decision; each regression surface approximates the value function at time t , when taking the corresponding decision.

Figure 2 illustrates three regressions surfaces obtained for the third period of the four-period example of section 5. The highest surface at any price-water level coordinates provides the (approximate) optimal decision, for those coordinates.

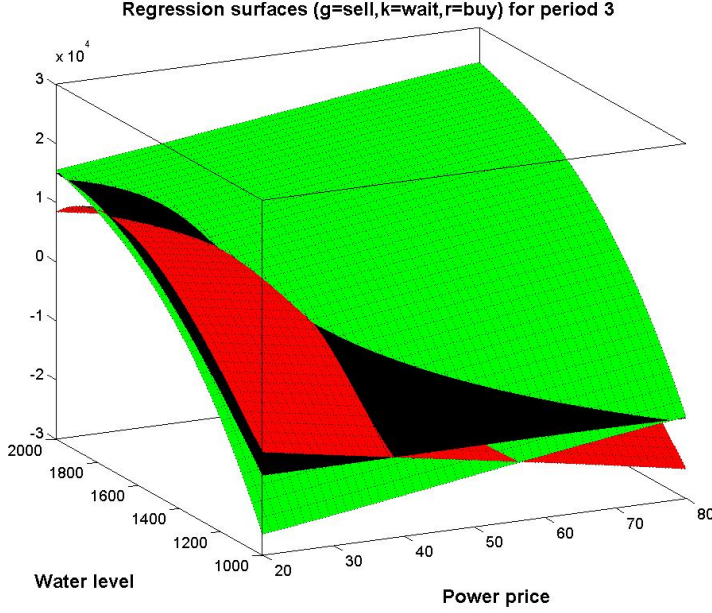


Figure 2: Regression surfaces corresponding to each of three decisions, superimposed.

3.2 Building optimal storage level paths

To avoid the discretization of the storage level (the endogenous state variable), we would like to associate a path of storage levels to each exogenous state variable path.

At first sight, one may be tempted to use the three regression surfaces to associate a time- t level $L_t^{(k)}$ with each scenario: project each of the three points $(S_t^{(k)}, L_t^{(k)}(u_t))$, $u_t \in \{+1, 0, -1\}$ on its corresponding regression surface, and let the highest value determine the time- t “antecedent” level $L_t^{(k)}$:

$$L_t^{(k)} = L_t^{(k)}(\hat{u}_t)$$

where

$$\hat{u}_t = \arg \max_{u_t \in \{+1, 0, -1\}} \tilde{V}^{u_t}(S_t^{(k)}, L_t^{(k)}(u_t))$$

This approach is flawed as it compares values with different initial conditions; somehow, it tracks *backward-optimality*, i.e. the best decision going from $t + 1$ to t . What we need are *forward-optimal* decisions and paths. We therefore suggest the following approach to build forward-optimal paths.

Consider first a specific scenario k and the time- t , control-dependent level $L_t^{(k)}(+1)$ associated to decision $u_t = +1$. Compare between themselves the values associated with each possible time- t decision, *always starting from* $L_t^{(k)}(+1)$ and identify the decision that reaps the largest expected value,—simply read on the regression surfaces—

$$u_t^* = \arg \max_{u_t \in \{+1, 0, -1\}} \tilde{V}^{u_t}(S_t^{(k)}, L_t^{(k)}(+1)) \quad (8)$$

Note that by construction, only one decision maps $L_t^{(k)}(+1)$ to the known scenario- k , time- $(t+1)$ level $L_{t+1}^{(k)}$; that is, decision $+1$. The winning decision u_t^* may or may not be $(+1)$; if it is, we have already identified one forward-optimal decision that leads to $L_{t+1}^{(k)}$. See Figure 3 for an illustration.

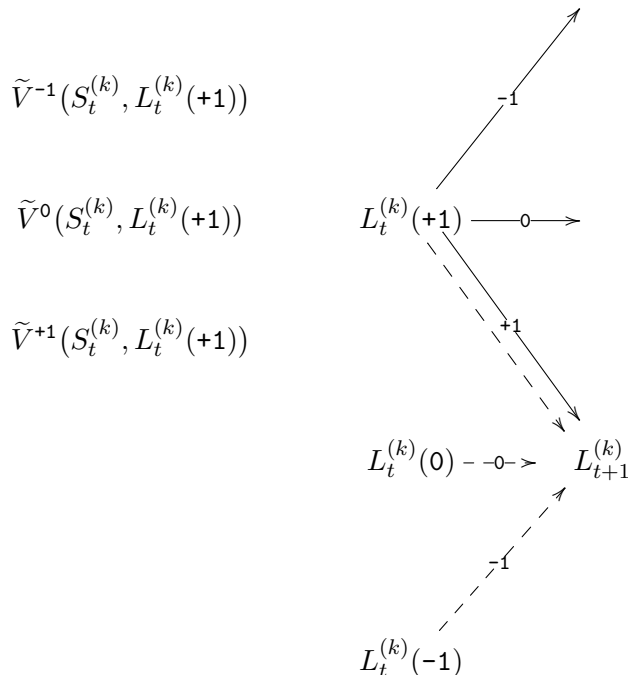


Figure 3: Endogenous variable path-building: for the case of level $L_t^{(k)}(+1)$, the three values $\tilde{V}^{-1}(S_t^{(k)}, L_t^{(k)}(+1))$, $\tilde{V}^0(S_t^{(k)}, L_t^{(k)}(+1))$, and $\tilde{V}^{+1}(S_t^{(k)}, L_t^{(k)}(+1))$, are looked up on the regression surfaces and compared between themselves. Only one decision $(+1)$ leads to $L_{t+1}^{(k)}$.

Repeat the procedure for the other two possible time- t levels $L_t^{(k)}(0)$ and $L_t^{(k)}(-1)$ obtained from the time-reverse state equation (4). We find two more u_t^* , which again, may or may not be respectively (0) and (-1) , i.e. the decisions that lead back to $L_{t+1}^{(k)}$.

If for at least one of the three $L_t^{(k)}(u_t)$ the optimal time- t decision brings this time- t level to the time- $(t+1)$ level $L_{t+1}^{(k)}$, then for scenario k , we have identified at once an optimal decision for time period t , and a time- t level for the endogenous variable. There is no guarantee that this will occur, or that it will occur exactly once, so provisions must be made for such cases, see the implementation details below. Note that, in practice, we have experienced very few cases where no optimal decision was available.

This procedure is then repeated for each scenario k , thereby associating to each simulation path k of the *exogenous* variable(s), a time- t level for the *endogenous* variable, building one more step of a path that is forward-optimal by construction. It is important to note that no extra regressions need to be performed to decide on optimal time- t levels associated to each

path.

The algorithm described so far in this section and section 3.1 is used to approximate the optimal policy that underlies the value function in (3). The main difficulty in ensuring that the approximation is convergent with the simulation effort, is that we have very little control over the distribution of the endogenous variable levels, which could end up anywhere as we back up from the horizon end to time 0. This is the topic of the next section.

Note that we need to be able to start the procedure at the horizon T , i.e. to have a temporal boundary condition giving the time- T value, $V(S_T, L_T)$. Various such conditions can be considered, which do not impact our description of the approach, though of course they change the numerical results. See the implementation details.

3.3 Dealing with boundary conditions and clustering

The main advantage of incorporating the endogenous variable into the simulation paths, is that it avoids the discretization of that variable. This comes at a cost.

First, we need to somehow enforce the spatial boundary conditions of the endogenous variable (a discretization approach enforces such conditions more naturally). Here, the difficulty is that we cannot keep the endogenous variable within its bounds $L_{\min} \leq L_t \leq L_{\max}$ by simply constraining which decisions, and thus which “antecedent” levels, are allowed during the backward pass. While this would ensure that no simulation path ever goes beyond the bounds, it would also mean that the boundaries’ influence is never taken into account in the regression procedure. For example, the cost of entering a full-regime production period with almost no water behind the dam, is a piece of information that would never be accounted for by the corresponding regression surface.

Second, we do not control the distribution of values of the endogenous variable that are considered, except at the final time T . The danger here is the occurrence of *clustering*, a situation where the levels of the endogenous variable become lumped together around one or a few spots. With clustering, the regression surfaces of section 3.1 are built from oddly distributed values of the endogenous variable, and their quality suffers.

Fortunately, it appears to be possible to somehow play each problem against the other. The basic idea is the following.

To solve the first problem, we need to let the paths of levels L_t cross the boundaries, so that the cost of decisions that lead outside the boundaries can be integrated in the regressions. Of course, some payoff penalty has to be incurred when leaving $[L_{\min}, L_{\max}]$. Akin to the penalty methods of constrained optimization, the penalty is always increasing with the violation, and can be linear, polynomial, exponential, etc. This can solve the problem of boundary enforcing. However, paths of the endogenous variable tend to leave the feasible zone $[L_{\min}, L_{\max}]$ for good, once they step out. This is a natural effect of the penalty; remember these paths are being built backwards in time. Indeed, forward-optimality means that valuable decisions are often ones that move the water level out of the penalty zone; following those valuable decisions during a backward pass tends to lead to the penalty zone. Which means that there is a *leakage* effect: before the backward recursion has reached time 0,

there are fewer endogenous variable values being sampled within the feasible zone, perhaps none at all. Our proposed solution to the leakage problem is to randomly assign a new endogenous variable level to any path that wanders too far beyond the feasible zone. What is “too far” must be treated with some care; see the implementation details. The value attributed to the simulation path for that point of time and the reassigned level, must be read on the regression surface; there is no obvious other good choice.

A key point here is that the leakage problem, which we created as a by-product of the solution to our first problem (regression information around the boundaries), not only can be solved in practice, but that reassigning those out-of-bounds endogenous variable values at random within the feasible zone goes a long way to solve the second problem, clustering. The price to pay is optimality. In the best of cases, we would build endogenous variable paths that cover the whole control period, and that correspond to optimal decisions at each time period. However, with each random reassignment, the path is broken in two pieces that are not optimal at their junction. Our tests tend to indicate that the impact of the reassignments is quite reasonable.

3.4 Implementation details

We discuss in this section our experience with some details of the approach.

3.4.1 Bounds on the endogenous variable and penalties

Recall that we assumed static bounds $L_{\min} \leq L_t \leq L_{\max}$ for the value of the endogenous variable. Let us discuss only L_{\max} ; the idea is the same for L_{\min} . We implement a supplementary bound L_{\max}^+ beyond L_{\max} , such that $L_{\max} \leq L_{\max}^+$ (equivalently, $L_{\min} \geq L_{\min}^-$). Then the following rules are used:

L_{\max} A penalty is applied when computing the value of $V_t(S_t^{(k)}, L_t^k)$ for any level L_t^k above L_{\max} .

L_{\max}^+ If the selected path- (k) , time- t level L_t^k is above L_{\max}^+ , then this level is re-assigned randomly to a level in $[L_{\min}, L_{\max}]$.

We are left to decide on the value of L_{\max}^+ and the intensity of the penalty. We choose to place L_{\max}^+ at the highest level that is reachable in one time step, starting from L_{\max} . This means that the levels above L_{\max}^+ , i.e. those that are reassigned, can only be reached in two time steps.

We choose a penalty function that is linear in the level above L_{\max} , specifically

$$\max\{L_t - L_{\max}, 0\};$$

the value is then impacted through the payoff function, which could typically be of the form be

$$\begin{aligned} \pi_t^{\text{penal}}(u_t; S_t, L_t) &= qu_t S_t - \max\{L_t - L_{\max}, 0\} qu_t S_t \\ &= \min\{q, q - L_t + L_{\max}\} u_t S_t \end{aligned} \tag{9}$$

meaning that going beyond the limits brings no value, but bears no “punitive” penalty.

Other penalty functions can of course be used, for example linear with a multiplicative factor, or exponential. We found no advantage in doing so; in particular, an exponential penalty is quite delicate to handle. These preferences are admittedly the result of numerical experience.

3.4.2 Functional bases

Our experience with various bases (stepwise, piecewise linear, or more general polynomials) for the exogenous variable S_t , the endogenous variable L_t and their product, steered us towards simple choices that worked just as well and were computationally undemanding. In the numerical tests below, we use monomials of S_t , L_t , and of the product of S_t and L_t . A simple scaling procedure helps avoid numerical instabilities.

Note that in his extensive study of various bases used with the simulation and regression algorithm proposed by Longstaff and Schwartz, Stentoft [St04a] concluded in the same general direction as our numerical experience.

3.4.3 Other cases related to the antecedent endogenous variable level

In section 3.2, it became clear that while exploring potential antecedent endogenous variable levels $L_t^{(k)}(u_t)$ (recall the time-reverse state equation (4)), it can be that three, two, one or none of these levels optimally lead to $L_{t+1}^{(k)}$. If two or three choices are available, we simply pick one at random. Notice however that this flexibility could be used to push the path in a preferred direction, for example to help avoid clusters. On the other hand, if no optimal choice exists, we can either cancel simulation k or pick a suboptimal antecedent level. Given that this problem has happened very rarely in practice so far, we expect little impact from either approach. In the numerical tests below, we pick the suboptimal antecedent level that is closest to being optimal.

3.4.4 Temporal boundary conditions

The time- T levels of the endogenous variable, for each scenario k , are generated randomly within the feasible area, i.e. between L_{\min} and L_{\max} .

The *value* associated to each scenario k at time T must be dependent on the level $L_T^{(k)}$, which should be straightforward. The dependence on $S_T^{(k)}$ is less obvious. We use a boundary condition

$$W\left(S_T^{(k)}, L_T^{(k)}\right) = \max_{u \in \mathcal{U}\left(S_T^{(k)}, L_T^{(k)}, T\right)} \left\{ \pi_T\left(u; S_T^{(k)}, L_T^{(k)}\right) \right\} \quad (10)$$

that in fact corresponds to “selling everything now, at the current price” (operational constraints may well make this unrealistic). An average price over some past period, or an expected price for the future, could be used. All this of course depends largely on the ap-

plication, and while it influences the numerical results, it is of lesser interest to explain our approach.

3.5 Summary of the algorithm

It is useful at this point to sum up the various steps of the algorithm. For the sake of simplicity, we keep with $\mathcal{U}(S, L, t) = \{+1, 0, -1\}$.

1. **Initialization:**

- (a) Choose a set of basis functions for the state variables, S_t and L_t ;
- (b) Randomly generate K paths for the exogenous variable S_t , ($t = 0, 1, \dots, T$);
- (c) Randomly generate K time- T levels of the endogenous variable L_T , within the range $[L_{\min}, L_{\max}]$, for example with a uniform distribution.
- (d) Compute time- T values according to a boundary condition, for example (10).

2. **Backward recursion:** for all times from $t = T - 1$ to $t = 0$:

- (a) Compute the regression surfaces $\tilde{V}^u(S, L)$, $u \in \{+1, 0, -1\}$ according to a *penalized* version of (5)–(7), meaning that if $L_t^{(k)}(u)$ is outside $[L_{\min}, L_{\max}]$, a penalty is applied to the regressand (using the penalized payoff (9) for example). Store the regression parameters.
- (b) For each of the $3K$ candidate levels $L_t^{(k)}$, compute the most valuable decision, using (8).
- (c) Associate a level $L_t^{(k)}$ for each path k , according to the results in the above step, that is, ensuring that the move from $L_t^{(k)}$ to $L_{t+1}^{(k)}$ is an identified “most valuable decision”. (If there are no such decisions, choose the decision that is the least suboptimal. If there are multiple “most valuable” decisions, choose a decision at random among them.) If $L_t^{(k)}$ is outside of $[L_{\min}^-, L_{\max}^+]$, randomly reassign it to $[L_{\min}, L_{\max}]$ (the backwash technique)
- (d) Compute the K values $V_t(S_t^{(k)}, L_t^{(k)})$ as a sum of payoffs until time T along path (k). In the case of paths whose level has just been reassigned (see step above), use instead the value on the regression surface.

3. **Out-of-sample tests:** if desired, run out-of-sample tests, retaining solely the regression parameters from the above steps.

4 Convergence results from the literature

Convergence analysis for LSMC algorithms is involved, to say the least. Most existing results are concerned with the analysis of American option pricing algorithms and stopping time

problems; see Clément, Lamberton, Porter [CLP02] and Egloff [E05]. One must in fact distinguish between two types of LSMC algorithms. In the first, one uses the values read on the regression surfaces during the backward pass; in the other type, one uses the realized values along the paths (the regressions are used to make decisions, but not to provide value approximations). Typically, the first type appears to be easier to analyse, while the second works better in practice; this holds for American option pricing as well as for more general problems like the one we consider. Carmona and Ludkovski [CL08] provide very detailed analysis work for a method of the first type, which they nickname after Tsitsiklis and Van Roy (see [TVR01]). They also concede that the analysis of methods of the second type, —nicknamed after Longstaff and Schwartz by the authors—, is considerably more difficult, and in fact intractable in the current state of science.

The algorithm that we use is fundamentally of the second type (“Longstaff and Schwartz”), or in fact a hybrid, because of the backwash procedure. Beyond the sources of error identified in [CL08] (time and policy discretizations, projection on bases, and Monte Carlo sampling), we must add the error due to the occasional breaking of optimal paths caused by the backwash procedure. For lack of theoretical proofs of convergence, we provide in the next section numerical results that lend support to the good behaviour of our approach.

5 Numerical experiments

We present here our numerical experiments designed to illustrate the performance of the approach we propose. The first experiment is a simple, intuitive, four half-day periods case; the second experiment models 16 weeks worth of half-day long periods (224 periods), and is characterized by important daily, weekly and monthly seasonality.

Both models share the same dam-based, hydro-plant background; providing a real application makes the discussion more intuitive. Specifically, we consider a power system containing between $L_{\min} = 1000$ and $L_{\max} = 2000$ units of energy; also, a counterparty (or market) can buy ($u_t = +1$) or sell ($u_t = -1$) 180 units of energy per period, always at the current spot price S_t . The spot price models and seasonalities are discussed below.

Note that we run the experiments as if it were possible to transform purchased energy into water behind the dam with 100% efficiency. This is of course impossible, but in fact, as long as there is a sufficient domestic market demand to fulfil, the efficiency assumption is easy to motivate in the sense that the purchased energy would simply be used to satisfy the local demand.

For both experiments, we compare our approach with a benchmark, dynamic programming approach which is described in appendix A. The appendix covers the geometric brownian motion case of section 5.2, but the method can be easily adapted to the simpler process in section 5.1.

All codes are programmed in Matlab. For the benchmark program, the heaviest loops are passed to an embedded C code.

5.1 A simple, four-period experiment

We first test our approach on a stylized four-period (five time points) case. Spot prices follow a uniform distribution, \$60 wide and centered around, respectively, \$50, \$30, \$50, \$50, and \$30, at times 1, 2, 3, 4, and 5. There is no dependence of prices through time. The initial dam level is 1500 units of energy, and it is assumed that the residual value at the end of the fourth period is the reservoir’s content times the time-5 spot price.

Note that the optimal policy is clear if the price path follows the means of the distributions: sell, buy, sell, sell, which leaves the reservoir rather empty at the end of the fourth period.

Table 1 provides statistics on average realized value over 20 runs with $n_{\text{oos}} = 100\,000$ out-of-sample price paths simulations, for both our approach and the benchmark approach. Running times to build the policies (excluding out-of-sample tests) are a few seconds at the longest. To avoid combining policy and out-of-sample errors, we control for the out-of-sample paths, i.e. the same paths are provided to the 20 policies.

Sim. and reg.	Learning paths	Mean	St.Dev.
	1 000	11 602 \$	75 \$
	10 000	11 707 \$	18 \$
	100 000	11 710 \$	5 \$
Benchmark		11 927 \$	

Table 1: **Values for out-of-sample tests with the four-period experiment.** “Learning paths” is the number of simulation paths used to build the policies. “Mean” is an average of the values at time 0, over 20 runs of simulations, each defining its own policy, and using the same 100 000 out-of-sample paths for all 20 runs. In all cases, the basis functions were monomials up to cubic in the price, up to cubic in the water levels, and linear in the cross term (price times water level). “St.Dev.” is the sample standard deviation of the 20 simulation runs. “Benchmark” is a price obtained with a discretization of prices and water levels $m = n = 1001$.

Policies can be compared visually, though of course the final test of a policy is the ability to return a high value. Figure 4 shows the policies obtained with the benchmark approach and the simulations and regressions approach.

The results indicate on one hand that high numbers of simulations are not necessary. However, increasing even further the numbers of learning simulations or bases do not seem to lead to much better policies, in that the average value reaches a plateau before converging to the benchmark value. As a consequence, the *circa* two-percent suboptimality could in large part be due to the backwash procedure. Finally, notice how the reduction in standard deviation scales approximately with the increase in the number of learning paths.

5.2 Results for a long term, high seasonality experiment

Our second experiment consists of 224 half-day periods, or 16 weeks at two periods per day. Spot prices follow a geometric brownian motion with almost no drift, $\mu = 0.0001$, but very high volatility $\sigma = 0.8$. Seasonal factors are applied as follows:

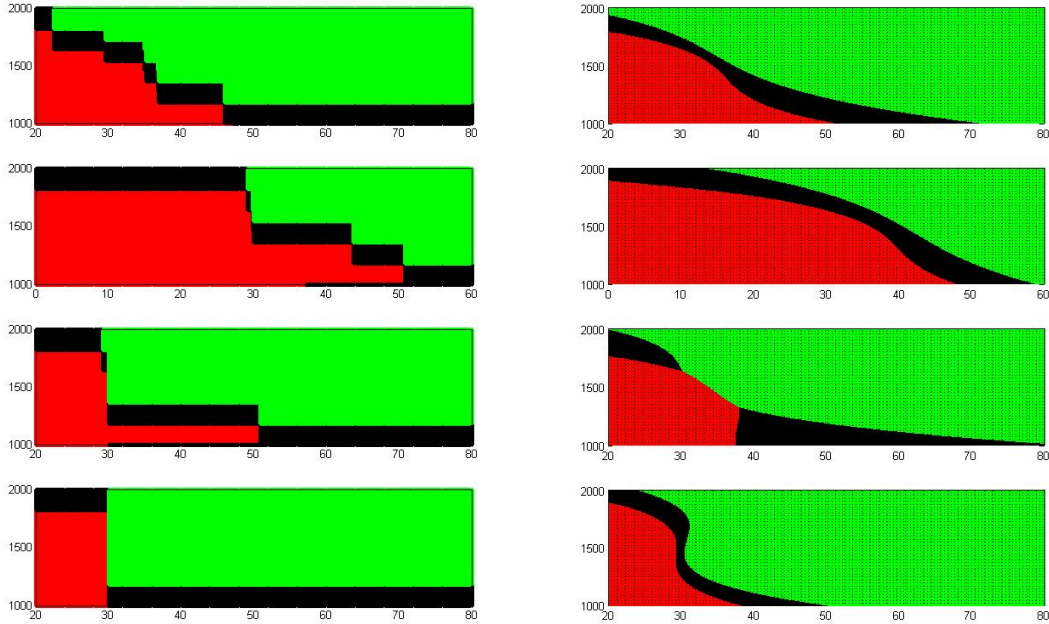


Figure 4: **View of policies.** The policies derived from the benchmark approach (left) and the simulations and regressions approach (right) are displayed for time periods one (top) to four (bottom) of our four-period example. Power prices are on the x-axis, water levels on the y-axis. The south-west areas are “buy” zones, the north-east are “sell” zones, the black areas correspond to “do nothing”. The benchmark policies are obtained with a discretization $m = n = 1001$. The simulation and regression policies are obtained with 100 000 simulation paths. The basis functions were monomials up to cubic in the price, up to cubic in the water levels, and linear in the cross term.

Daily seasonality: For weekdays, every peak, half-day period is followed by an off-peak half-day period where the prices are lowered by a deterministic factor of $\exp(-0.5)$.

Weekly seasonality: Weekend days (four periods) have their prices lowered by a deterministic factor of $\exp(-0.5)$. There are no peak vs off-peak hour effects during the weekend.

Monthly seasonality: A high-price month (four weeks in fact) is followed by a low-price month, whose prices are lowered by a factor of $\exp(-0.5)$.

For example, the price for an off-peak half-day during the week and in a low-price month, is the price simulated by the brownian motion times $\exp(-0.5)^2$. The same holds for the price for any half-day during the weekend and in a low-price month.

The system is assumed to start with a peak period, the first of five weekdays, and the first day of a high-priced month.

Table 2 provides statistics on average realized value for out-of-sample price paths simulations, again for both the benchmark and the simulation and regression approaches. The

running time to build the policies (excluding out-of-sample tests) is about three and a half minutes (for one of the 20 runs, and with 75 000 learning simulations). Again, we control for the out-of-sample paths to single the policies impact.

Sim. and reg.	Learning paths	Mean	St.Dev.
	1 000	240 512 \$	4 096 \$
	5 000	242 714 \$	475 \$
	25 000	242 765 \$	270 \$
	75 000	242 900 \$	128 \$
Benchmark		247 576 \$	

Table 2: **Values for out-of-sample tests with the 224-period experiment.** “Learning paths” is the number of simulation paths used to build the policies. “Mean” is an average of the values at time 0, over 20 runs of simulations, each defining its own policy, and using the same 100 000 out-of-sample paths for all 20 runs. In all cases, the basis functions were monomials up to quadratic in the price, up to quadratic in the water levels, and up to quadratic in the cross terms. “St.Dev.” is the sample standard deviation of the 20 simulation runs. “Benchmark” is a price obtained with a discretization of prices and water levels $m = n = 1001$.

Here again, the analysis of the numerical results indicates that very high numbers of simulations are not necessary. However, more effort to build policies (going beyond 75 000 simulations, using more bases) does not improve the average value much more. Here again, and despite the much higher number of time periods, the benchmark value is missed by about two percent, just like in the four-period, low seasonality case. Finally, notice again how the standard deviation decrease scales approximately with the number of simulations increase.

6 Conclusion

We present an approximate dynamic programming algorithm based on simulations and regressions (LSMC). As suggested by [CL10], both exogenous and endogenous state variable are tacked onto (quasi-)simulation paths, so that neither state variable is discretized. We innovate first by specifying an approach to build “*forward-optimal*” paths of the endogenous state variable. We also suggest a *backwash* procedure that deals at the same time with two difficulties: providing valuable regression information around the bounds of the endogenous state variable, and avoiding the clustering effect that the “quasi-simulation” can lead to. Numerical tests indicate that the approach is quite robust to the number of basis functions, runs well with reasonable numbers of simulations, and yields policies that are suboptimal by only a slight margin, when compared to a Markov-chain, finely discretized D.P. benchmark.

The main interest in simulation and regression methods is their scalability in the number of variables. Although we provide for the moment evidence with only one exogenous variable and one endogenous variable, the method should scale easily with the number of exogenous

variables: for example, power demand and water inflows could be added quite naturally to our hydropower model.

Increasing slightly the number of endogenous variables would be reasonably more demanding, in that the number of regressions would increase as a power of the discretization of the control decision. Increasing the number of decision variables would have a similar effect, the effect being an increase in the number of regressions as a function of their discretization. For continuous decisions variables, a three-point discretization (as we tested) may appear too limiting, as a higher discrete count would offer better control. Note however that there are non-trivial cases where “bang-bang” policies (sell all or buy all) are optimal; see [BdJ08] and [S10]. In such cases, there is no need to refine the decision possibilities.

Further work includes the extensions mentioned above: several exogenous and endogenous state variables, and multi-variate decisions. At that point, introducing appropriate variance-reduction techniques will become necessary.

One big challenge that lies ahead is the introduction of risk constraints in our currently pure profit model. Risk management may also need to bear on the values of the variables themselves: think of the water level above and below a dam, for example.

A second important topic of research is the introduction of decisions that kick in only after a number of periods. Term contracts for hydropower is a prime example. A simplistic approach would be to set power aside (as water) until the sale, which could be months away. This could severely limit the flexibility of the system, and its value. Clearly, introducing such forward-bearing decisions in a backward-moving algorithm may prove to be a challenge.

A The benchmark dynamic programming method

We describe here the discretization approach used to compute benchmark values for our examples. The approach relies on plain backward recursion D.P. and Markov chains. We present the case of the geometric brownian motion; simpler cases can be readily adapted. We assume here that the exogenous state variable is described by

$$S_t = f(t) \times \tilde{S}_t$$

where $f(t)$ is a deterministic seasonal component given by:

$$f(t) = e^{\beta_1 d_{1,t} + \beta_2 d_{2,t}}$$

and

$$d_{1,t} = \begin{cases} 1 & \text{if date } t \text{ is a night, or a weekend} \\ 0 & \text{otherwise} \end{cases}$$

$$d_{2,t} = \begin{cases} 1 & \text{if date } t \text{ is a month of low consumption} \\ 0 & \text{otherwise} \end{cases}$$

where β_1 and β_2 are fixed parameters. The seasonally adjusted portion of the state variable evolves as:

$$\tilde{S}_{t+1} = \tilde{S}_t e^{(\alpha - \frac{1}{2}\sigma^2)\eta + \sigma\sqrt{\eta} \varepsilon_{t+1}}$$

where η is the length, in years, between t and $t + 1$, α is an annual drift parameter, σ is an annual standard deviations parameter, and ε_{t+1} is a Normal(0, 1) noise. Because we want an homogenous chain through time to simplify the computations, we define the adjusted log of the state variable as

$$\ln \tilde{S}_t = \ln S_t - \beta_1 d_{1,t} - \beta_2 d_{2,t}$$

whose distribution can be used to compute the expectations appearing in the dynamic programming recursions. The adjusted variable is a Gaussian with mean $\ln \tilde{S}_t + (\alpha - \frac{1}{2}\sigma^2)\eta$ and variance $\sigma^2\eta$. The distribution function of this variable, required later for the computations of the transition probability matrix elements, is then given by

$$\Phi_{\ln \tilde{S}}(a | \ln \tilde{S}_t) = N\left(\frac{a - (\alpha - \frac{1}{2}\sigma^2)\eta - \ln \tilde{S}_t}{\sigma\sqrt{\eta}}\right) \quad (11)$$

where $N(\cdot)$ is the distribution function of a standard normal random variable.

A.1 Computing the value function

For the purpose of computing the recursive equation of the value function, we use an homogenous discrete-time Markov chain. As shown in Duan and Simonato [DS01], one can construct this chain in such a way that, as the number of states goes to infinity, it converges

to the target stochastic process over the time points $0, 1, 2, \dots$. In the context of the recursive formulation of a dynamic programming problem, the value function computed with this chain will converge to the theoretical function values.

The idea behind the construction of an m states approximating Markov chain is as follows. The real line, the support of the log of the adjusted exogenous state variable distribution, is first partitioned into m distinct cells. A numerical value is then assigned to each cell, yielding the m possible values. Given one of these values, the probability of reaching another value is computed as the probability to land in a given cell η years from now. For the process assumed above, these probabilities can be conveniently computed with the distribution function. The endogenous variable is also discretized and allowed to take n different values.

More formally, denote the discretized exogenous and endogenous variables to be $\mathbf{s} = [s_1, s_2, \dots, s_m]'$ and $\mathbf{l} = [l_1, l_2, \dots, l_n]'$. In these vectors, the first entry is the smallest element while the last entry is the largest. For the endogenous variable, the minimum and maximum values are dictated by physical restrictions. These values are denoted by L_{\min} and L_{\max} which implies that $l_1 = l_{\min}$ while $l_n = l_{\max}$. The probability matrix describing the transition of the exogenous state variable from one state to the other over a period of η year is

$$\mathbf{P} = \begin{bmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{bmatrix}.$$

Here, m is an odd integer and $s_{(m+1)/2} = \ln(\tilde{S}_0)$ is the current seasonally adjusted level. For the endogenous variable, we have $l_{istart} = L_0$ where $istart$ is the index value of the discretized endogenous variable value which is the closest to the current value. In such a context, for an adjusted state level s_i , an endogenous level l_j , and a maturity of T periods, the value function can be computed in τ time steps and with the following recursive system:

$$V_{i,j}(s_i, l_j, t) = \max_{u_{i,j}} \left[\pi_{i,j}(u_{i,j}; s_i, l_j) + e^{-r\eta} \sum_{k=1}^m p_{i,k} \times V_{k,\omega(u_{i,j})}(s_k, l_{\omega(u_{i,j})}, t+1) \right]$$

with

$$V_{i,j}(s_i, l_j, T) = l_j \times e^{s_i + \beta_1 d_{1,t} + \beta_2 d_{2,t}}$$

and

$$u_{i,j} \in \begin{cases} \{+1, 0, -1\} & \text{for } j = j_{low} \text{ to } j_{high} \\ \{0, -1\} & \text{for } j = 1 \text{ to } j_{low} - 1 \\ \{0, +1\} & \text{for } j = j_{high} + 1 \text{ to } m \end{cases}$$

and where

- j_{low} is the lowest index value of the discretized endogenous variable levels for which the change in level can be meet; j_{high} is the highest index value of the discretized water levels allowing to meet the upward change in the level. These index values can be found with:

$$j_{low} = \min\{j \text{ such that } l_j - q_- > l_{\min}\}$$

$$j_{high} = \max\{j \text{ such that } l_j + q_+ < l_{\max}\}$$

where q_- is the change in level associated with decision $u = -1$ and q_+ the change in level associated with decision $u = +1$. It is assumed here that changes in the endogenous variable are always at these levels.

- $\pi_{i,j}(u_{i,j}; s_i, l_j) = q(u_{i,j}) \times \exp(s_i + \beta_1 d_{1,t} + \beta_2 d_{2,t})$ with

$$q(u_{i,j}) = \begin{cases} q_- \times u_{i,j} & \text{if } u_{i,j} = +1 \\ q_+ \times u_{i,j} & \text{if } u_{i,j} = -1 \\ 0 & \text{if } u_{i,j} = 0 \end{cases}$$

- $l_{\omega(u_{i,j})}$ is the endogenous variable level computed with

$$l_{\omega(u_{i,j})} = RoundToNearest(l_j - q(u_{i,j}))$$

where *RoundToNearest* is a function rounding the level to the nearest discretized value and $\omega(u_{i,j})$ is the index value corresponding to the identified level.

A.2 Computing the Markov chain level and probabilities

To construct the vector \mathbf{s} , an overall interval covering the set of representative log levels over a period of $T \times \eta$ years is defined. This interval is written as $[\ln \tilde{S}_0 - B_s, \ln \tilde{S}_0 + B_s]$. The quantity B_s is computed using the conditional standard deviation of the power price return over the horizon of the problem. Formally,

$$B_s = \delta(m) \times \sigma \times \sqrt{T \times \eta}$$

where $\sigma \times \sqrt{T \times \eta}$ is the standard deviation of the adjusted level return over a period of $T \times \eta$ years. The scaling factor $\delta(m)$ is an increasing function of m , satisfying some mild partition conditions stating that $\delta(m)$ should grow at a rate smaller than m . In this study we opt for $\delta(m) = \ln(m)$. We then divide the above interval equally into $m - 1$ parts to obtain m discrete values

$$s_i = \ln \tilde{S}_0 + \frac{2i - m - 1}{m - 1} B_s$$

with $i \in \{1, \dots, m\}$ for the log of the adjusted levels. Note that $s_1 = \ln \tilde{S}_0 - B_s$ and $s_m = \ln \tilde{S}_0 + B_s$ are the minimum and maximum values. In order to have $\ln \tilde{S}_0$ among the state levels, m needs to be an odd integer which obtains $s_{\frac{m+1}{2}} = \ln \tilde{S}_0$.

To compute the elements of matrix \mathbf{P} , we define m cells that are constructed as $C_i = [c_i, c_{i+1})$ for $i = \{1, \dots, m\}$ and where $c_1 = -\infty$, $c_i = \frac{s_i + s_{i-1}}{2}$ for $i = 2$ to m and $c_{m+1} = +\infty$. Using the above cells and states, the transition probabilities i.e. the probability of landing in cell C_j given a current price s_i are computed with

$$p_{ij} = \Phi_{\ln \tilde{S}}(c_{j+1} | s_i) - \Phi_{\ln \tilde{S}}(c_j | s_i)$$

where $\Phi_{\ln \tilde{S}}(\cdot)$ is the distribution function of the logarithm of the adjusted exogenous variable level given in equation (11).

For the endogenous variable, the discrete quantities are computed with

$$l_i = l_{\min} + (i - 1)\Delta_l$$

where

$$\Delta_l = \frac{(l_{\max} - l_{\min})}{n - 1}$$

for $i \in \{1, \dots, n\}$.

References

- [Be07] BERTSEKAS, D. (2007), *Dynamic Programming and Optimal Control*, Athena Scientific.
- [BdJ08] BOOGERT, A., AND C. DE JONG (2008), "Gas Storage Valuation Using a Monte Carlo Method", *The Journal of Derivatives*, 15, 81–98.
- [CL08] CARMONA, R., AND M. LUDKOVSKI (2008), "Pricing asset scheduling flexibility using optimal switching", *Applied Mathematical Finance*, Special Issue on Commodities, 15(5-6), 405-447.
- [CL10] CARMONA, R., AND M. LUDKOVSKI (2010), "Valuation of energy storage: an optimal switching approach", *Quantitative Finance*, 10(2), 359–374.
- [CF07] CHEN, Z., AND P.A. FORSYTH (2007), "A semi-lagrangian approach for natural gas storage valuation and optimal operation", *SIAM Journal on Scientific Computing*, 30(1), 339–368.
- [CF10] CHEN, Z., AND P.A. FORSYTH (2010), "Implications of a regime-switching model on natural gas storage valuation and optimal operation", *Quantitative Finance*, 10(2), 159–176.
- [CLP02] CLÉMENT, E., LAMBERTON, D. AND P. PROTTER (2002), "An analysis of a least squares regression algorithm for American option pricing", *Finance and Stochastics* 6, 449-471.
- [DS01] DUAN, J.C. AND J.-G. SIMONATO (2001), "American option pricing under GARCH by a Markov chain approximation", *Journal of Economic Dynamics and Control*, 25, 1689–1718.
- [E05] EGLOFF, D. (2005), "Monte Carlo algorithms for optimal stopping and statistical learning", *Annals of Applied Probability* 15(2), 1396-1432.
- [KW94] KEANE, M.P., AND K.I. WOLPIN (1994), "The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte Carlo evidence.", *The Review of Economics and Statistics*, 76(4), 648–672.
- [LMS10] LAI, G., MARGOT, F. AND N. SECOMANDI (2010), "An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation", *Operations Research* 58(3), 564–582.
- [LS01] LONGSTAFF, F.A., AND E.S. SCHWARTZ (2001), "Valuing American options by simulations: a simple least squares approach", *Review of Financial Studies* 14, 113-148.

- [NP09] NASCIMENTO, J. AND W. POWELL (2009), “An optimal approximate dynamic programming algorithm for the energy dispatch problem with grid-level storage”, *Working paper*, Department of Operations Research and Financial Engineering, Princeton University.
- [Po11] POWELL, W. (2011), *Approximate Dynamic Programming, Solving the Curses of Dimensionality*, Second edition, John Wiley and Sons.
- [S10] SECOMANDI, N. (2010), ”Optimal commodity trading with a capacitated storage asset”, *Management Science* 56(3), 449–467.
- [St04a] STENTOFT, L. (2004a), ”Assessing the Least Squares Monte Carlo approach to American option valuation”, *Review of Derivatives Research* 7(3), 129-168.
- [TDR09] THOMPSON, M., DAVISON, M, AND H. RASMUSSEN (2009), ”Natural gas storage valuation and optimization: a real options application”, *Naval Research Logistics*, 56, 226–238.
- [TB02] TSENG, C.L., AND G. BARZ (2002), ”Short-term generation asset valuation: a real options approach”, *Operations Research* 50(2), 297-310.
- [TVR01] TSITSIKLIS, J.N., AND B. VAN ROY (2001), ”Regression methods for pricing complex American-style options”, *IEEE Transactions on Neural Networks* 12(4), 694-703.