Bloomberg

Overview of American Monte Carlo Pricing in DLIB

January 31, 2021

Quantitative Analytics
Bloomberg L.P.

Abstract

This document provides an overview of the Bloomberg implementation of American Option Pricing

Keywords. Monte Carlo, American Option, Bermudan Option, Least Squares Regression, Longstaff-Schwartz, AMC Freezing.

DISCLAIMER

Contents

1 Overview

1.1 American Options

An American call option on an underlying asset S with expiry T and strike K, which is the prototype for all American options, is a contract giving the holder the right, but not the obligation, to purchase the asset at any time $t < T$ by paying the option writer K. The time at which the holder exercises his option is called the *exercise time* τ_e, and the *payoff* for an American call option will be $(S(\tau_e) - K)^+$, whereas the American put option will be $(K - S(\tau_e))^+$.

$\tau_e : \Omega \to \overline{\mathbb{R}}_{\geq 0}$

More generally, any contract with embedded optionality implies that the option holder has either the right to *exercise into* the underlying contract (a call option), or the issuer has the right to *cancel out of* the underlying contract (a callable deal). In each case there is an accompanying schedule of *exercise dates* (or *callable schedule*) on which the right to exercise or cancel may be exercised. The European option is characterized by a single exercise date at expiration, whereas the American option is characterized by the option to exercise or cancel at any time before expiration[1]. The Bermudan option is midway between the European and American, in that one may exercise at any one of a specified set of dates between settlement and maturity.

See Andersen & Piterbarg for details

Callable deals are conceptually no different from call options, in that one may construct a "synthetic" callable deal by selling the underlying and simultaneously purchasing a call option. Ratchet floaters, snowballs, ratchet range accruals, Nikkei-linked down-in notes, and all deals with knock-out features, are examples of Bermudan callable path-dependent deals which can be priced as a combination of a non-call base deal and an early exercise-into deal.

A basic premise for valuing an American option might be that *if one knows in advance* the time τ_e at which he will optimally exercise the option, then pricing need be no more difficult than that of a European option. In particular, if Π and C are the prices of the American and European calls respectively,

$$\Pi(T; S_0, K) \quad = \quad C(\tau_e; S_0, K) = \mathbb{E}^{\mathbb{Q}}\left[\frac{(S(\tau_e) - K)^+}{B(\tau_e)}\right], \tag{1.1}$$

where $\mathbb{Q}$ is the risk-neutral measure associated with the market account $B(t)$. Of course, if things were as simple as (1.1), then one could generate many Monte Carlo paths and compute the European option price as the value of the American option. What this simplistic analysis overlooks is the concept of *path dependence*, which is to say that each path ω will have its own path-dependent time $\tau_e(\omega)$ at which to optimally exercise. Rejecting the idea of a universal, path-independent time τ_e, we are led to the idea of an *exercise strategy* τ which, based on information known at time t, dictates whether or not one is better off exercising or holding. Given a strategy τ, we see that the τ_e should be understood as a random variable giving the *first time to exercise* along a given path using strategy τ. We consequently obtain the following American option pricing formula[2]:

MC PROBLEM

$$\boxed{\Pi(T; S_0, K) \quad = \quad \max_{\tau} \; \mathbb{E}^{\mathbb{Q}}\left[\frac{(S(\tau_e) - K)^+}{B(\tau_e)}\right],} \tag{1.2}$$

[1] If the option has an initial "lock out" period (call protection), then exercise is allowed only after a specified date.

[2] We use the concise notation $\mathbb{E}_t^{\mathbb{Q}}[Y] := \mathbb{E}\left[Y(t^+) \,\middle|\, S(t)\right]$ for the conditional expectation of the random variable $Y(t^+)$ conditioned on $S(t)$. In our case, $Y(t^+)$ is the value of the option having *not exercised* through time t, but using all path information $S_\omega(t) = S(0 \leq s \leq t)$.

where the random variable τ_e takes values interval $(0, T]$. In the case of an American Put option, an example of a strategy would be to exercise whenever the option value $\mathbb{E}_t^{\mathbb{Q}}\left[\frac{(S(t)-K)^+}{B(t)}\right]$ is less than or equal to its intrinsic value, and so τ_e would be the first time at which the option has no premium[3]. Two tasks therefore are arise - the calculation of computing the 1 *conditional* expectations appearing in $\mathbb{E}_t^{\mathbb{Q}}[\cdot]$, and 2 implementing an optimal strategy.

1.2 Bermudan Options and Path Dependent Pricing

A Bermudan option is similar to an American option, except that the allowed exercise times are discretely specified by the set of times $\{t_0, t_1, \ldots, t_N\}$, as opposed to the *continuum* of possible exercise times $[t_0, T]$ permitted in the American option's definition. Numerically, continuous parameters are always modeled using a discrete set of points, and by increasing the density of these points one obtains increasingly accurate approximations. Specifically, one replaces the interval $t \in [t_0, T]$ with the finite set $t_j \in \{t_0, t_1, \ldots, t_N\}$, and one expects that by letting $|\,t_{k+1} - t_k| \to 0$ as $N \to \infty$, one can simulate the continuous situation to any desired accuracy. In other words, American options can be viewed as a limiting case of Bermudans, and are numerically modeled as Bermudans. Consequently, in what follows we will consider Bermudan options, and denote their set of exercise dates by $\{t_1, t_2, \ldots, t_N\}$.

The expression in (1.2) can be rewritten for Bermudan options as follows:

Bermudan

$\tau_e : \Omega \longrightarrow \{t_1, \ldots, t_N\}$

$$\Pi(T; S_0, K) \quad = \quad \max_{\tau} \; \mathbb{E}^{\mathbb{Q}}\left[\frac{(S(t_j) - K)^+}{B(t_j)}\right], \tag{1.3}$$

where now τ_e takes values in the finite set $\{t_1, t_2, \ldots, t_N\}$.

Recall that the expression (1.3) was derived by focusing on the path-dependent exercise time $\tau_e(\omega)$. There is an alternative perspective which, while giving an equivalent formulation, lends itself more directly to a backwards algorithm commonly encountered in Dynamic Programming problems.

In this alternate approach, one exploits the fact that an American option contains within itself many future-starting American options. First, it is easy to determine the value of the option at a given state $S(t_k)$ *if we exercise*, namely

$$\text{Exercise} \xrightarrow{\text{value}} \quad \frac{(S(t_k) - K)^+}{B(t_k)}.$$

The other possibility is that we *don't exercise*, in which case the time t_k value of the option will be the discounted value of the option held until (at least) time t_{k+1}, averaged over all path continuations $(t_k, S(t_k)) \to (t_{k+1}, S(t_{k+1}))$:

$\Pi_{t_{k+1}}$, price in next exercise date

$$\text{Don't exercise} \xrightarrow{\text{value}} \quad \mathbb{E}_{t_k}^{\mathbb{Q}}\left[\frac{\Pi(t_{k+1}, t_N; S_{t_{k+1}}, K)}{B(t_{k+1})}\right]. \tag{1.4}$$

[3] If the option always has some premium, such as an American Call option on a stock which pays no dividends, then it is never optimal to early exercise, and we effectively have an equivalent European option.

This quantity (1.4) is appropriately referred to as the *hold value*, *continuation value*, or *conditional expectation of continuation*. An optimal strategy is to take the maximum of the two possibilities:

See p.31, 32 of Andersen and Piterbarg for more details.

$$\boxed{\frac{\Pi(t_k, t_N; S_{t_k}, K)}{B(t_k)} \quad = \quad \max\left[\frac{(S(t_k)-K)^+}{B(t_k)}, \mathbb{E}^{\mathbb{Q}}_{t_k}\left[\frac{\Pi(t_{k+1}, t_N; S_{t_{k+1}}, K)}{B(t_{k+1})}\right]\right]}, \tag{1.5}$$

thereby giving a recursive definition of the Bermudan call price $\Pi(t_k, T; S_{t_k}, K)$. Algorithmically, we need only start the induction from the pathwise determined exercise values at the final time t_N:

(terminal condition)
$$\frac{\Pi(t_N, t_N; S_{t_N}, K)}{B(t_N)} = \frac{(S(t_N)-K)^+}{B(t_N)}.$$

In §2.1 we describe the least squares regression method for approximating conditional expectations in Monte Carlo simulations. In §2 we further describe the method of Longstaff-Schwartz which combines least squares regression approximation of continuation values with Dynamic Programming to price Bermudan options. Finally, in §3 we describe DLIB's support for Monte Carlo pricing of American, Bermudan, and Callable options. A standard reference for these topics is [Gla].

2 Least Squares Monte Carlo

2.1 Conditional Expectation and Least Squares Regression

As described in §1.2, computing a pathwise optimal exercise time $\tau_e(\omega)$ requires the calculation of the conditional expectation of continuation. Unfortunately, continuation values cannot be computed exactly. Nonetheless, in the context of Monte Carlo simulation, there is an effective method of approximating these values.

Conditional Expectation using Embedded Monte Carlo

(Andersen & Piterbarg, Glasserman for more details)

Markovian property

One approach to computing the conditional expectation (1.4), namely $\mathbb{E}^{\mathbb{Q}}_{t_k}[\cdot]$, is to generate at each time t_k using $S(t_k)$ as an initial condition, many Monte Carlo paths and to then compute the expected value directly. This technique is referred to as *embedded* Monte Carlo, as it generates Monte Carlo paths within an existing Monte Carlo simulation. Of course, on account of the recursive nature (1.5), this procedure needs to be replicated N times, as each $\Pi(t_{k+1}, t_N; S_{t_{k+1}}, K)$ will require averaging over many $\Pi(t_{k+2}, t_N; S_{t_{k+2}}, K)$. Specifically, if we consistently use M paths, then at t_0 we generate M paths (each of length N), and at t_1 an additional M^2 paths (each of length $N-1$, and so on until t_{N-1} where we introduce an additional M^N paths (each of length one). Even if $M = 1000$ and $N = 10$, we have a prohibitive number ($\sim 10^{30}$) of paths to generate! Clearly, embedded Monte Carlo is not a feasible method for computing conditional expectations. On the other hand, if we bear in mind that conditional expectations at any given time t_k are themselves random variables, which is to say *functions* of the possible states[4], there presents an opportunity to introduce ideas from Functional Analysis. This is the approach of Least Squares Monte Carlo described in the following section. ↳ see Fx below.

[4]More precisely, all states conditioned on the state of the world at t_k, which includes $S_\omega(t_j), t_j < t_k$ as well.

Conditional Expectation using Least Squares Regression

Spaces of functions on an interval are mathematically infinite dimensional, but it is possible to *approximate* these idealized (and numerically inaccessible) spaces with finite-dimensional subspaces. The most common examples are truncating the first d terms of a function's Taylor expansion to produce an approximating polynomial, or truncating the first d terms of a periodic function's Fourier expansion to produce an approximating trigonometric polynomial.

Conceptually, infinite-dimensional function spaces have a metric structure analogous to Euclidean space[5] and the process of approximating a function can be viewed as projecting it into a finite dimensional subspace. Choosing a basis of functions for the approximating subspace, such as $\{1, x, x^2, \ldots, x^d\}$ in the first example where the finite-dimensional subspace is the collection of degree d polynomials, translates the problem into one of finding coefficients that optimally represent the function in terms of the d basis functions. Lest this appear an unduly complicated description of truncating a function's Taylor series, it gives a uniform context to the general problem of approximating a function in terms of a small collection of basis functions. For example, in some applications one uses the Laguerre[6] polynomials:

Taylor's basis might be problematic for computing purposes (see Atkinson, p. 207)

$$\{L_0(x) = 1, L_1(x) = 1 - x, L_2(x) = 1 - 2x + 0.5x^2, \ldots L_d(x)\}.$$

The technique introduced by Longstaff and Schwartz [LS] applies the above idea to the "conditional expectation of continuation" function defined by (1.4). If we specify the basis functions as $\{\psi_1(x), \ldots, \psi_d(x)\}$, then we seek coefficients $\{\lambda_1, \ldots, \lambda_d\}$ such that

$$\text{Search } \lambda_i \ / \quad \mathbb{E}^{\mathbb{Q}}_{t_k}\left[\frac{\Pi(t_{k+1}, t_N; \cancel{x}, K)}{B(t_{k+1})}\right] \approx \sum_{j=1}^{d} \lambda_j \psi_j(x). \tag{2.1}$$

$$\simeq B(t_k) \cdot \mathbb{E}^{\mathbb{Q}}_{t_k}\left[\underbrace{\frac{\Pi(t_{k+1}, t_N, S(t_{k+1}), K)}{B(t_{k+1})}}_{\text{Hold value, see appendix}} \,\middle|\, S(t_k) = x\right] = H_k(x)$$

$\exists F_{t_k}$ by factorization lemma (B. Driver)

In the context of a Monte Carlo simulation employing M paths $\{\omega_i\}_{i=1}^M$, with generated states $x_i = S_{\omega_i}(t_k)$, then we seek λ_j which solve the "Least Squares Regression" problem of minimizing

$$\{\lambda_j^{MC}\}_{j=1}^{d} = \underset{\lambda}{\arg\min} \sum_{\substack{i=1 \\ (x_i = S(\omega_i, t_k))}}^{M} \left[D_{i,k} \cdot \Pi_{t_{k+1}}\left(S_{\omega_i}(t_{k+1})\right) - \sum_{j=1}^{d} \lambda_j \psi_j(x_i)\right]^2. \qquad D_{i,k} \overset{d}{:=} \frac{B_{\omega_i}(t_k)}{B_{\omega_i}(t_{k+1})} \qquad [1]$$

See my notes there for justification.

In §A.2 we give an example showing how the least squares regression is performed in practice.

2.2 Least Squares Regression in Several Variables

In the previous section we discussed the conditional expectation as a function of a single independent variable, namely the state $S_\omega(t_k)$. One example of a "multi-dimensional" deal is that of an equity basket in which the underlying is actually a vector of prices. Moreover, depending on the

$\| F - F' \|_{L^2(\Omega)} = \mathbb{E}\left[|F - F'|^2\right]$

[5]A purely probabilistic interpretation of length is to use the quadratic expectation of the random variable.

[6]Laguerre polynomials are an example of an *orthogonal* basis, which have the important property that lower-order coefficients need not be recomputed when projecting into higher dimensional approximating subspaces.

KEY REMARK: we cannot generate (sub)paths for each conditional expectation we want, i.e., is not feasible. Thus, we follow a different approach:

$$(1) \quad H_k(x_i) \simeq \frac{1}{M'} \sum_{j=1}^{M'} \frac{B_{\tilde\omega_j}(t_k)}{B_{\tilde\omega_j}(t_{k+1})} \Pi_{t_{k+1}}\left(S_{\tilde\omega_j}(t_{k+1})\right) \ / \ \tilde\omega_j \text{ paths where } S_{\tilde\omega_j}(t_k) = S_{\omega_i}(t_k) = x_i \quad \forall i' \in \{1, \ldots, M'\}$$

(that is, new paths starting at that point for the underlying)

$$H_k(x) \simeq \sum \beta_j \psi_j(x) \ / \ [1], \text{ so no need to simulate.}$$

$$\neq \frac{1}{M} \sum_{j=1}^{M} D_{i,j} \, \Pi_{t_{k+1}}\left(S_{\omega_j}(t_{k+1})\right) \quad \text{for } \omega_j \ / \ S_{\omega_j}(t_k) = x_j \neq x_i.$$

↳ changes for each path

complexity of the Bermudan deal being priced, the pricing of the option in a given state may have additional dependencies hidden in its dependence on the path ω. One example is a payoff which depends on some combination of the underlying's current value and the value of a previously paid coupon. Another example is an Asian option whose valuation depends on a moving window of fixed duration over which the underlying asset price is averaged. In this case, the valuation depends on the stochastic start time of the optimal window, and therefore averages over different windows can be introduced as additional path-dependent variables.

When calculating exercise values and performing regression, one is actually *obliged* to account for path-dependent information used to compute the cash-flows of the deal, such as coupon history for rachet floaters/snowballs, or event triggers in Nikkei down-and-in notes and knock-out deals, and therefore the user must take proper account of the deal specifics when identifying the regression variables.

several variables
not just the strike

Mathematically, we may write the price $\Pi(t_k, t_N; \mathbf{x}, \mathbf{k})$ where $\mathbf{x} = [x_1, \ldots, x_r] \in \mathbb{R}^r$ is a vector of path-dependent quantities (including, of course, $x_1 :\overset{d}{=} S(t_k)$) and $\mathbf{k}$ is a vector of deal-specific constants generalizing the strike price of the call or put. This multi-variable generalization is easily handled by considering function spaces associated with functions of several independent arguments $f(\mathbf{x}) = f(x_1, \ldots, x_r)$. These spaces also contain finite-dimensional subspaces which provide a corresponding collection of basis functions $\{\Psi_1(\mathbf{x}), \ldots, \Psi_d(\mathbf{x})\}$. Given a basis $\{\psi_1(\mathbf{x}), \ldots, \psi_d(\mathbf{x})\}$ for the single-variable case discussed earlier, it is typical to use a basis of functions for the multi-variable case constructed from products of single-variable basis functions. For example, in the case of polynomials where $r = 2$ and $d = 3$, we would construct the following two-variable monomials:

$$\begin{aligned}
\Psi_1(x,y) &= 1, \qquad \Psi_2(x,y) = x, \qquad \Psi_3(x,y) = y, \\
\Psi_4(x,y) &= x^2, \qquad \Psi_5(x,y) = xy, \qquad \Psi_6(x,y) = y^2, \\
\Psi_7(x,y) &= x^3, \qquad \Psi_8(x,y) = x^2y, \qquad \Psi_9(x,y) = xy^2, \qquad \Psi_{10}(x,y) = y^3.
\end{aligned}$$

More generally, if $\psi_i(x), i = 1, \ldots, d$ are basis functions for the one-variable case, then the product functions

$$\Psi_{\widehat{i_1 \ldots i_r}}(\mathbf{x}) = \psi_{i_1}(x_1) \cdot \cdots \cdot \psi_{i_r}(x_r),$$

where the multi-index $\widehat{i_1 \ldots i_r}$ satisfies $i_1 + \ldots i_r \leq d$, can be used as a basis of functions for the multi-variable case. (see Stone - Weierstrass's Th. in Folland).

2.3 Dynamic Programming and the Longstaff-Schwartz Algorithm

In this section we illustrate the Dynamic Programming method which, combined with the Least Squares regression method, is the algorithm of Longstaff-Schwartz for pricing Bermudan options. Detailed descriptions of the Dynamic Programming algorithm (§A.1), the Least Squares algorithm (§ A.2), and their consolidation into the Longstaff-Schwartz algorithm (§ A.3) for early-exercise option pricing are provided in the appendices.

We consider a Bermudan put with strike $K = 9.0$ on an asset whose present value is 8.0, in the presence of zero interest rates (so no discounting necessary), and for which five path evolutions are simulated. The asset values at the three exercise dates $\{t_1, t_2, t_3\}$ are tabulated in Table 2.1.

$r = 0 \Rightarrow B = 1$

* See (8.52) Glasserman, LS algorithm sets $V(t_i) = \begin{cases} V^{ex}(t_i) & \text{if } \hat{V}^{hold}(t_i) \le V^{ex}(t_i) \to \text{EXERCISE.} \\ V(t_{i+1}) & \text{if } \hat{V}^{hold}(t_i) > V^{ex}(t_i) \text{, NOT } \hat{V}^{hold}(t_i) \text{, see (8.51) there.} \end{cases}$

↑ estimated using Least Squares ↳ HOLD

path	t_0	t_1	t_2	t_3
1	8.0	11.0	15.0	18.0
2	8.0	10.0	11.0	6.0
3	8.0	9.0	6.0	6.0
4	8.0	6.0	5.0	8.0
5	8.0	4.0	3.0	2.0

Table 2.1: Stock Values $S(t_k)$

$K = 9$ → $\phi(x) = (K - x)_+$ (put option)

path	t_0	t_1	t_2	t_3
1	1.0	0.0	0.0	0.0
2	1.0	0.0	0.0	3.0
3	1.0	0.0	3.0	3.0
4	1.0	3.0	4.0	1.0
5	1.0	5.0	6.0	7.0

Table 2.2: Exercise Values $\text{Exer}(t_k)$ $= (K - S(t_k))_+$

(at t_2)

path	$S(t_2)$	Hold Value $f(S(t_2))$ f^{t_2}	Exer(t_2)	Decision	OptionValue(t_3)	OptionValue(t_2) *
1	15.0	0.25	0.0	Hold	0.0	0.0
2	11.0	1.71	0.0	Hold	3.0	3.0
3	6.0	3.53	3.0	Hold	3.0 (see *)	3.0
4	5.0	3.89	4.0	Exercise	NA	4.0
5	3.0	4.62	6.0	Exercise	NA	6.0

$\rightarrow \frac{(K - S_\omega(t_3))_+}{B(t_3) = 1}$ (paths 1–3) $\rightarrow (K - S_\omega(t_2))_+$ (paths 4–5)

Table 2.3: Continuation Values using Regression Coefficients $[a_0, a_1] = [5.71, -0.36]$, and how they are used in the Calculation of Option Values at t_2.

by def, $\mathbb{E}_{t_2}^{Q,\text{approx}}\left[\frac{\Pi(t_3, t_3, S(t_3), K)}{B(t_3) = 1} \,\middle|\, S(t_2) = x\right] \underset{=}{B(t_2)}^{1} = \mathbb{E}_{t_2}^{Q,\text{approx}}\left((K - S(t_3))_+ \mid S(t_2) = x\right)$

Hold value ↑

Formulating the regression problem $f(x) = a_0^{t_2} + a_1^{t_2} x$ to compute regression coefficients at t_2 from

$f(S_{t_2}) = (K - S_{t_3})_+ \Rightarrow$ $f(15.0) = 0.0, \quad f(6.0) = 3.0, \quad f(11.0) = 3.0, \quad f(5.0) = 1.0, \quad f(3.0) = \cancel{6}.0$ 7

leads to following least squares matrix problem: ↗ fixed coefficient

$$f(x) = a_0 + a_1 x \Rightarrow [a_0, a_1] \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 15.0 & 11.0 & 6.0 & 5.0 & 3.0 \end{bmatrix} = \begin{bmatrix} 0.0 & 3.0 & 3.0 & 1.0 & \cancel{6}.0 \end{bmatrix}; \quad \to a_0^{t_2}, a_1^{t_2}$$

[5.71666667 -0.36458333] ⟶ Table 2.3.

whose solution is $[a_0, a_1] = [5.71, -0.36]$. Consequently, we construct a table of continuation values at t_2, their comparison with exercise values at t_2, and the new valuation of the option at t_2, possibly propagating backwards the option value at t_3 in the event of a hold decision.

The algorithm now repeats itself. The regression problem $f^{t_1}(x) = a_0^{t_1} + a_1^{t_1} x$ to compute the regression coefficients at t_1 from

$f^{t_1}(x) = \mathbb{E}_{t_1}^{Q,\text{app}}(\ldots \mid S_{t_1} = x) \Rightarrow$ $f(11.0) = 0.0, \quad f(10.0) = 3.0, \quad f(9.0) = 3.0, \quad f(6.0) = 4.0, \quad f(\cancel{3.0}) = 6.0$ 4.0

leads to following least squares matrix problem (see $S(t_1)$ and OptionValue(t_2)):

$$f^{t_1}(x) = a_0^{t_1} + a_1^{t_2} x \qquad [a_0, a_1] \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 11.0 & \cancel{9.0} & 9.0 & 6.0 & \cancel{5.0} \end{bmatrix} = \begin{bmatrix} 0.0 & 3.0 & 3.0 & 4.0 & 6.0 \end{bmatrix}$$

10.0 4.0

[7.74117647 -0.61764706] The other numbers is the result with the wrong data, but results in the same hold data.

whose solution is $[a_0, a_1] = [\cancel{9.86, -0}.83]$. Consequently, we construct a table of continuation values at t_1, their comparison with exercise values at t_1, and the new valuation of the option at t_1, possibly propagating back the option value at t_2 in the event of a hold decision. ⇒ Table 2.4

Finally, we compute the value of the Bermudan put option as the expected value of the OptionValue(t_1) over the five paths giving $(0 + 3 + 3 + 4 + 6)/5 = 3.2$. Note that the path-dependent exercise time $\tau_e(\omega)$ takes the values $\{\infty, t_3, t_3, t_2, t_2\}$.

↓ value is always 0.

Previous table

see prev. page

path	$S(t_1)$	Hold Value $f(S(t_1))$	Exer(t_1)	Decision	Option Value(t_2)	Option Value(t_1)
1	11.0	0.70	0.0	Hold	0.0	0.0
2	10.0	2.37	0.0	Hold	3.0	3.0
3	9.0	2.37	0.0	Hold	3.0	3.0
4	6.0	4.86	3.0	Hold	4.0	4.0
5	4.0	5.70	5.0	Hold	6.0	6.0

Table 2.4: Continuation Values using Regression Coefficients $[a_0, a_1] = [9.86, -0.83]$, and how they are used in the Calculation of Option Values at t_1.

Π_0^{Eur}

Note that the simulated price of the European put would be $(0 + 3 + 3 + 1 + 7)/5 = 2.8$. Although one expects the Bermudan put to always be at least as valuable as its European version, agreement with theory can be expected only in the limit as the number of paths and the polynomial degree approach infinity. (See Glasserman, Longstaff-Schwartz gives a lower bound, negative bias, so if large enough, this inequality is not satisfied).

3 American, Bermudan, and Callable Options in DLIB

For deals with embedded optionality, in particular American, Bermudan, and callable deals, DLIB provides an "Option" menu with the following selections: `None, European, American`.

3.1 Americans as Bermudans

The first observation we make concerning the pricing of American options in DLIB is that they are internally treated as Bermudan options. In other words, the concept of a continuous exercise schedule is replaced with a discrete exercise schedule, albeit of sufficiently high frequency, and then priced as a Bermudan deal.

3.2 Regression Variables

The second observation is the requirement of specifying regression variables and regression functions. Strictly speaking, this is not a requirement of the deal specification, but rather an implementation requirement related to the Least Squares Regression Monte Carlo pricing algorithm. In BLAN, for example, a callable script will not be validated if regression information is not provided. Additionally, it is critical that the specified regression variables be either *independent*, or at worst only mildly correlated, such as the underlying stocks in an equity basket.

As an example, while specifying both the Libor rate and the Swap rate of long tenor as regression variables would introduce only mild dependence and be acceptable, specifying two functions of Libor alone (such as Libor-3M plus two different spreads) would constitute essentially a single regression variable, the second one being redundant. A less extreme case, but no more desirable, is that of choosing Libor-6M and Libor-3M, where the variables are very strongly correlated. Although strong correlation between regression variables does not present any numerical difficulty, these examples illustrate poor choices of regression variables since they will not optimally capture the relevant state variables influencing exercise decisions. The subject of properly choosing underlyings for regression variables is a subtle one, and in §4.2 we gives some results comparing the pricing accuracy associated with different sets of choices.

Remark 3.1. In certain BLAN scripts as well as non-BLAN Templates such as the `"Structured Product"` template, where the script is neither modifiable or transparent, the selection of regression variables for four-factor LMM callable deals uses the following internal logic:

- Base regression variables associated with deal underlyings[7]:
  1. Always include the underlying Libor index for the deal currency
  2. If a fixed coupon is derived from a foreign currency then use the default ticker[8] for the currency
  3. If a floating coupon is derived from Libor/CMS underlyings then use the tickers for those underlying observables[9]
- Supplemental regression variables are dynamically associated with each call date T_k, for which we define the *Time to Maturity* $\tau_k := T_{mat} - T_k$:
  1. SWAP-N_0 where $N_0 = F(\tau_k)$
  2. SWAP-N_5 where $N_5 = F(\tau_k - 5)$
  3. SWAP-N_8 where $N_8 = F(\tau_k - 8)$

  and where the tenor returned by $F(\cdot)$ is determined by the currency as:

  $$F_{USD}(\tau) \quad := \quad \begin{cases} \lceil\tau\rceil \text{ with ceiling granularity of [1y],} \\ \lfloor\tau\rfloor \text{ with flooring granularity of [1y, 2y,}\ldots\text{, 20y, 25y, 30y, 40y, 50y].} \end{cases}$$

  and

  $$F_{EUR}(\tau) \quad := \quad \begin{cases} \lfloor\tau\rfloor \text{ with flooring granularity of [1y, 2y,}\ldots\text{, 12y],} \\ \lfloor\tau\rceil \text{ with rounding granularity of [13y,}\ldots\text{, 45y].} \end{cases}$$

- After any redundancies are removed, no more (though possibly less, depending on τ_k) than the first four regression variables obtained from the above hierarchical logic will be applied.

DLIB currently supports only one type of regression function, namely polynomials. The degree of the polynomial regression function is specified to be quadratic. Additionally, there are two primitive regression variables that are supported: the underlying state (`"Underlying"`), and a past coupon (`"PastCoupon"`).

In addition to the prudent selection of regression variables, the user should be aware of the implications of a prudent choice for the number of paths. In particular, in order to reduce bias by improving the independence between the regression function and the path data to which it is applied, DLIB uses two independent Monte Carlo simulations. The first Monte Carlo simulation (the *pilot run*) is used to determine the regression function by fitting its coefficients to the states associated with an initial generation of paths. The second Monte Carlo simulation (the *pricing*

[7]Note that for a RACL product with a *fixed coupon* which depends indirectly on an underlying being *in-range*, such underlying is included as if it were covered under the rule for a *float coupon*.

[8]For example, if a trade priced in USD depends on a EUR underlying, then EUR-6M would be included.

[9]In the case of a spread such as [CMS30Y - CMS2Y], the difference will be used as the regression variable.

run) then applies those regression functions to the simulated paths[10] from which the continuation values are derived. Needless to say, the user should be aware of the slower performance of American Monte Carlo, in particular with regard to its dependence on the number of simulation paths, when compared with its European analogue.

3.3 AMC Freezing and Greeks

It is important for certain controls in the Monte Carlo simulation to be held fixed in order to achieve a consistency across multiple pricings. The most well-known of these controls is the *seed* for the random-number generation. For example, if the pilot run used in a Longstaff-Schwartz algorithm for determining regression coefficients, or the pilot run used in a control-variate approach to variance reduction, were to use a different seed from the baseline pricing simulation, then results would be unstable. The present section addresses the question of keeping the *exercise decisions* in the base-price simulation of an American option to be consistent with scenario simulations in which market data is shifted for purposes of computing risk-factor sensitivities.

The AMC Freezing methodology refers to the reuse, during a subsequent scenario pricing in the specific context of a Greek[11] calculation, of the exercise decisions associated with an American Monte Carlo base-price simulation (see [Pit]). This approach is mathematically justified by the fact that a small perturbation in model parameters has a second-order impact on the optimal exercise boundary, and hence freezing this exercise boundary is appropriate for sufficiently small[12] bump-sizes. An important consequence of this technique is the resulting stability of Greek calculations, as found for example in the Greek calculations for KRR and Bucketed Vega. Without the AMC Freezing, inaccuracies in the Greek calculations can be observed in disparities between the sum of bucketed Greeks and their corresponding *parallel* Greek (*e.g.* DV01 compared against the sum of the KRR's), as well as irregular behavior with respect to decreasing bump-size.

In terms of implementation, the AMC Freezing technique requires that scenario requests to the pricer for purposes of Greek calculations be identified as such, so that the exercise decisions associated with the base-price can be returned and passed along to subsequent simulations for the scenario repricing and corresponding Greek calculation.

In §4.3 we present some results showing the effectiveness of the AMC Freezing technique in terms of regularity with respect to bump-size, as well as spillage with respect to bucketed vega and KRR.

3.4 BLAN Deals

A detailed example involving a Callable Bermudan Swap contract, which uses the `cancellable` function invoked from the `Callable` BLAN module, is carefully described in the BLAN documentation [Blo1].

[10] The pricing simulation will consume the random numbers generated by continuing the random number sequence from the pilot simulation.

[11] The applicability of AMC freezing will vary with bump-size, and is not applicable to scenario pricing generally.

[12] What constitutes an adequate bump size can depend on many factors. Bump values of 0.01bps for Vega and 0.1bp for KRR are recommended, although 0.1bps for Vega and 1bp for KRR are generally adequate.

3.5 Template Deals

There are many template deals provided with DLIB. The regression variables and their corresponding regression functions for some specific template deals are given in the Table 3.1. Recall that we use r to denote the number of variables, and d for the total polynomial degree. For example, $(r, d) = (2, 2)$ denotes the six two-variable monomials $\{1, x, y, x^2, xy, y^2\}$, used as a basis for the space of regression functions.

Template Deal	Regression Variables	(r, d)
Snowball, Snowbear	`(Underlying IR Rate, PastCoupon)`	$(2, 2)$
Equity Linked Note Callable Knock-Out	`(Underlying IR Rate, Underlying Equity/Index)`	$(2, 2)$
FX Linked Note Callable Knock-Out	`(Underlying IR Rate, Underlying FX Rate)`	$(2, 2)$
PRDC	`(Domestic IR Rate, Foreign IR Rate, FX Rate)`	$(3, 2)$

Table 3.1: Template Deal Regression Variables and Regression Functions.

4 Validation of American Monte Carlo

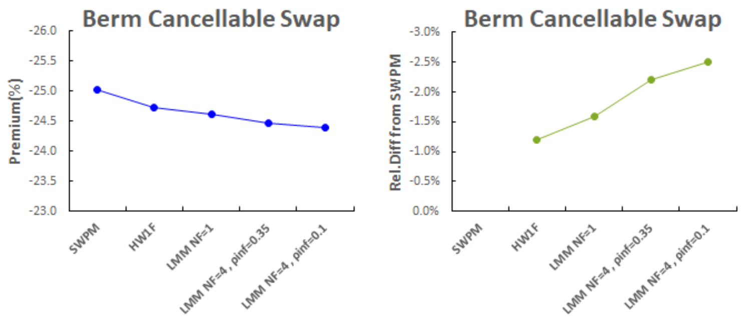
In this section, we present some results which illustrate the accuracy of DLIB's least-squares regression methodology. In particular, comparisons with the model-specific calibration schemes for HW1F and LMM (Libor Market Model) are made against SWPM pricing which uses an independent method not based on simulation.

4.1 Quality of Polynomial Regression

In validating DLIB's methodology of polynomial regression, tests have been conducted for some typical IR derivatives, namely Bermudan Cancellable Swaps and Bermudan Swaptions with 20 year maturity and quarterly call schedule. In these tests, pricings in DLIB have been compared with their counterparts in SWPM. It should be noted that SWPM *does not* use least-squares regression for Bermudan option pricing, but instead applies the direct integration method described in [Blo2]. In this sense, SWPM's Bermudan pricing can be regarded as a more reliable methodology than the Longstaff-Schwartz method used in DLIB, and for this reason it has been chosen as a good benchmark for comparison. Furthermore, since SWPM uses the LGM model which is mathematically equivalent to the Hull-White one-factor model, the AMC testing for DLIB using the Hull-White model can be directly compared with SWPM to assess the regression quality.

In addition to this comparison, the correlation effect of the LMM model is also tested to confirm the *factor dependence* reported by [AA]. The Bermudan prices of LMM converge to SWPM or Hull-White values when the perfect correlation is enforced by the one factor, and decrease in magnitude for increased decorrelation, which is consistent with the conclusions of [AA].

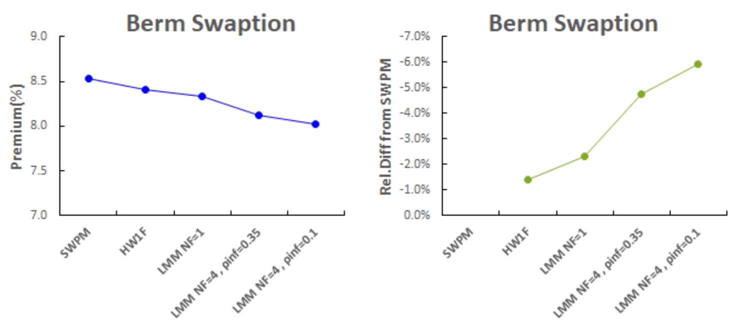
As shown in Figure 4.1 for a Bermudan Cancellable Swap, SWPM matches the Hull-White price within 1% (relative error), and matches one-factor LMM within 1.5% (relative error). It is clearly observed that the enhanced decorrelation of LMM achieved by increasing the number of factors, or decreasing the ρ_∞ parameter, decreases the Bermudan price by about 1% in absolute magnitude.



(a) Premiums using four DLIB pricing models and SWPM.

(b) Relative errors of four DLIB pricing models versus SWPM.

Figure 4.1: Comparison of SWPM pricing and DLIB pricing (HW1 and LMM with different correlation structures) for a Bermudan Cancellable Swap with 20 year maturity and quarterly call schedule.



(a) Premiums using four DLIB pricing models and SWPM

(b) Relative errors of four DLIB pricing models versus SWPM.

Figure 4.2: Comparison of SWPM pricing and DLIB pricing (HW1 and LMM with different correlation structures) for Bermudan Swaption with 20 year maturity and quarterly call schedule.

Additional tests with Bermudan Swaptions reveal a trend consistent with the previous case of the Bermudan Cancellable Swap but show an enhanced sensitivity to the correlation as shown in Figure 4.2. More specifically, the decorrelation decreases the price by 4% in absolute magnitude, which can be regarded as more than *moderate factor dependence*.

4.2 Proper choice and number of regression variables

AMC pricing of a 5Yx5Y European[13] ATM Swaption in LMM was investigated by using as regression variables specific subsets[14] of the four underlyings: Libor3M, CMS1Y, CMS2Y, CMS5Y. Note that it is inappropriate to specify more than the number of factors associated with the model, for

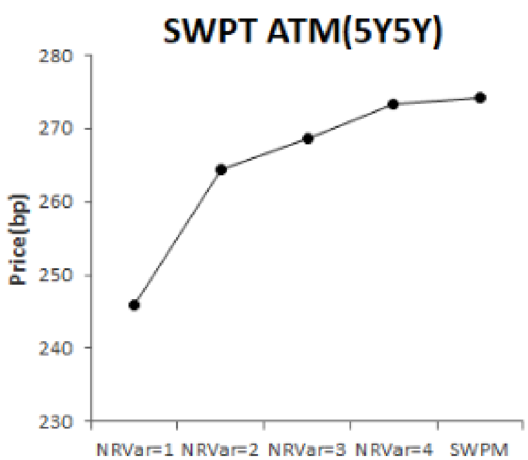
[13] Note that when DLIB prices a European option it employs the AMC methodology, effectively pricing it as a Bermudan with a a single exercise date.

[14] In §A.4 we illustrate how this full set of four regression variables can be specified in BLAN.

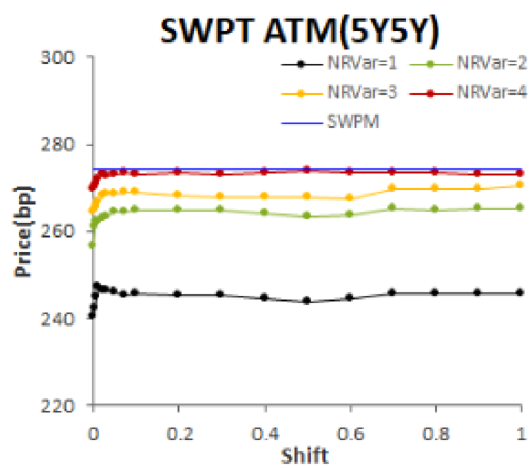
example one for HW1, two for HW2, or four for LMM with 4-factor. The resulting prices are compared with the market price (labeled as SWPM) as shown in Figure 4.3a. The trend clearly shows that using the full set of four regression variables is highly recommended in order to reproduce the market quote.

Also shown below in Figure 4.3b are similar results to Figure 4.3a, but reproduced over a range of values for a constant SLMM shift parameter. These plots demonstrate the consistent advantage to using the full set of four regression variables, independent of how market skew might be implied by the calibration of the SLMM shift parameters, for accurately reproducing the market quote.

Although not specific to AMC pricing, it is worth emphasizing that an improper choice of calibration instruments can also negatively impact the accuracy of pricing, and therefore should be given equal consideration when structuring a DLIB deal.



(a) AMC pricing becomes more accurate, approaching the market price, with sequential inclusion of recommended regression variables.



(b) Consistent improvement shown in Figure 4.3a is reproduced for full range of SLMM shift values.

Figure 4.3: Improved pricing, closely approximating the market (SWPM) value, is achieved when using the recommended underlyings [Libor3M, CMS1Y, CMS2Y, CMS5y] as regression variables.

4.3 Quality of Greeks with AMC Freezing

In this section we present comparisons between the Freezing and non-Freezing methods when computing parallel and bucketed Greeks for a range of bump-sizes.

Callable Swap

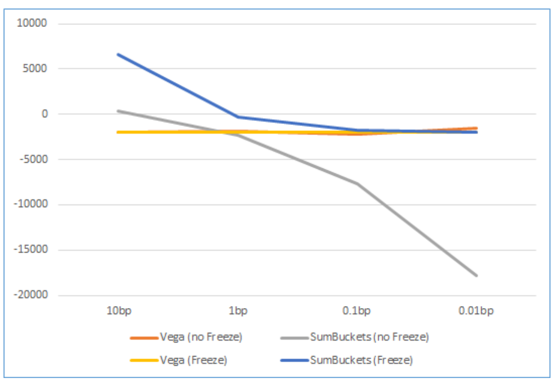
The present deal is a “callable 5 year swap with 1 year non-call period” (5NC1) with a fixed leg coupon of 1.645% and priced on 09/10/19 using 20,000 paths. It has been implemented as a BLAN script, using the regression variables “US0003M, USSWAP1, USSWAP2, USSWAP5” as in §A.4.

The call schedule will consist of 16 quarterly call dates:

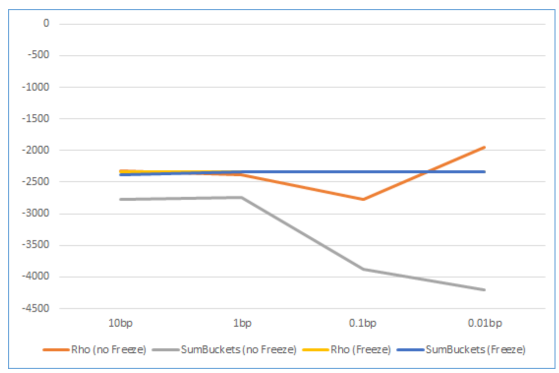
$$\{08/27/20, 11/27/20, 02/25/21, \ldots, 11/27/23, 02/26/24/, 05/28/24\}.$$

The first sensitivity we consider is volatility[15], and therefore we record the (parallel) vega and also the sum of the bucketed vegas. Without AMC Freezing, one finds an undesirable irregular disparity between parallel vega and the sum of the bucketed vega (the difference being the *vega-spillage*), and moreover neither is stable with respect to decreasing bump-size. On the other hand, with AMC Freezing enabled, one observes in Figure 4.4a the vanishing vega-spillage with decreasing bump-size, as well as the stability of the parallel vega.

The second sensitivity we consider is rates, and therefore we record the (parallel) rho and also the sum of the KRR's. Again, without AMC Freezing, one finds an undesirable irregular disparity between parallel rho and the sum of the KRR's (the difference being the *rho-spillage*), and moreover neither is stable with respect to decreasing bump-size. On the other hand, with AMC Freezing enabled, one observes in Figure 4.4b the vanishing rho-spillage with decreasing bump-size, as well as the stability of the parallel rho.



(a) Comparison of parallel and bucketed Vegas with the Freezing technique enabled and disabled, with decreasing bump-size.



(b) Comparison of parallel and bucketed Rhos with the Freezing technique enabled and disabled, with decreasing bump-size.

Figure 4.4: Vega and Rho stability with respect to bump size. Comparison between with-and-without Freezing methodology, and also between parallel shift and the sum of the bucketed shifts. When the Freezing technique is applied, one observes vanishing spillage as bump-size decreases (blue vs. yellow), as well as an exceedingly stable parallel Greek (yellow plot) for all bump sizes.

Remark 4.1. Regarding the presumption that the sum of the bucketed risks should agree with the parallel risk, while true for a linear function of the calibration instruments, is only approximately true for nonlinear functions. In other words, nonlinear effects with respect to larger bump sizes, and therefore the bump-size at which spillage becomes negligible (*e.g.* at 0.1bp for Figure 4.4a), will depend on the specific nature of the deal. This comment equally applies when AMC Freezing is enabled, where the AMC Freezing assumption will have an additional influence on the spillage with large bump size. Modifying the calibration instruments in the above example so as to omit six specific swaptions will remove some of the nonlinear dependencies, as shown in Figure 4.5.

[15]Note that all references to volatility, and volatility bump-size, are in units of bps and refer to Normal volatility.

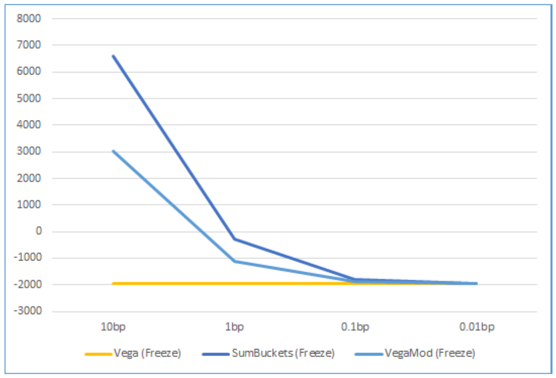


Figure 4.5: Vega stability with respect to bump size. Comparison between parallel shift (orange) and the sum of the bucketed vegas for two sets of calibration instruments. The same deal as shown in Figure 4.4a, but calibrated to six fewer swaption instruments, is shown in light blue.

4.4 Summary and Guidance

As several user options can have a critical impact on the accuracy of the American Monte Carlo simulation, we present a table summarizing the guidance discussed in previous sections.

Setting	Recommended Values	Guidance	Reference
Bump Sizes (bps)	$0.01 \leq$ Vega-bump ≤ 0.1 (Normal Vol) $0.1 \leq$ KRR-bump ≤ 1.0	too small requires more paths[16] too large defeats AMC Freezing[17]	§3.3
Number RegVars	one per model factor HW1F (1), HW2F (2), LMM (4)	use Templates to assure correct number	§4.2
Choice RegVars	underlyings (Libor3M),... calibration instruments (USSWAP2),...	use Templates to avoid dependencies	§4.2

Table 4.1: Guidance on AMC user inputs.

[16]In general, a computed sensitivity falling within standard-error may be indistinguishable from simulation noise, and will require more paths to be numerically meaningful.

[17]In general, re-calibration with large bump sizes is prone to optimizer instabilities. Indeed, AMC Freezing makes it possible to use the small bump sizes necessary for calibration stability.

A Appendices (pseudocodes)

A.1 Algorithm: Dynamic Programming

(Inputs)

Require:
Paths for regression: $\{\omega_j : j = 1 \ldots, N\}$, `no_sims`
Option exercise schedule: $\{t_k : k = \alpha, \ldots, \beta\}$ `exercise_dates`
Regression basis functions $\{\Psi_h(\mathbf{x}) : h = 0, 1, 2, \ldots, r\}$ `regression_functions`
Ensure:
Regression Coefficients: $\lambda_{h,k}$ for basis function Ψ_h at time t_k

1: $\lambda_{h,k} \leftarrow 0$ for all k and h. {regression coefficients}
2: $k \leftarrow \beta$ {start from the last possible exercise time}
3: compute numeraire $\mathcal{N}^j(t_k)$ for all paths ω_j ($k = \beta$)
4: compute exercise value $V^j_{ex}(t_k)$ for all paths ω_j $\quad V^j_{ex}(t_k) :\overset{d}{=} \underset{\text{payoff}}{U}(t_k, \underline{x}^j(t_k))$, p. 30 Andersen & Piterbarg.
5: $V^j_{ct}(t_k) \leftarrow V^j_{ex}(t_k)$ for all j
6: $k \leftarrow k - 1$
7: **while** $(k \geq \alpha)$ **do**
8: compute numeraire $\mathcal{N}^j(t_k)$ for all j
9: compute exercise value $V^j_{ex}(t_k)$ for all j (as above) $\quad \underset{\mathcal{N}(t_k, X(t_k))}{\mathcal{N}(t_k)} \mathbb{E}^N\left(\frac{\overset{V(t_{k+1}, X(t_{k+1}))}{V(t_{k+1})}}{\mathcal{N}(t_{k+1})} \Big| X(t_k) = x\right) =: \underset{\text{hold value}}{H_k(x)} = \lambda_k \cdot \Psi(x) + \underset{\text{error}}{\varepsilon_k}$
10: compute regression variables $\{\mathbf{x}^j(t_k)\}$
11: compute regression coefficients $\lambda_{h,k}$ $\{\underbrace{\frac{\mathcal{N}^j(t_k)}{\mathcal{N}^j(t_{k+1})} V^j_{ct}(t_{k+1})}_{=: \tilde{H}^j(t_k)} \underset{\text{(below)}}{\overset{\text{LstSq}}{=}} \sum_h \lambda_{h,k} \Psi_h(\mathbf{x}^j(t_k))\}$ (below)
12: **for all** j **do**
13: **if** $V^j_{ex}(t_k) \geq \underbrace{\sum_h \lambda_{h,k} \Psi_h(\mathbf{x}^j(t_k))}_{= \tilde{H}^j(t_k)}$ **then**
14: $V^j_{ct}(t_k) \leftarrow V^j_{ex}(t_k)$ {exercise is optimal}
15: **else**
16: $V^j_{ct}(t_k) \leftarrow \sum_h \lambda_{h,k} \Psi_h(\mathbf{x}^j(t_k))$ {hold is optimal} (in Longstaff - Schwartz this is different, see below)
17: **end if**
18: **end for**
19: $k \leftarrow k - 1$ {working backward to the next step: t_k becomes t_{k-1}}
20: **end while** (step 7)
21: **return** $\lambda_{h,k}$ for all h and k.

A.2 Algorithm: Least Squares Regression

Require:
Regression variables: $\mathbf{x}_j(t_k)$ for each path ω_j and fixed time t_k
Regression basis functions $\{\Psi_h(\mathbf{x}) : h = 0, 1, 2, \ldots, r\}$
Numeraires: $\mathcal{N}^j(t_k), \mathcal{N}^j(t_{k+1})$ for each path ω_j and fixed times t_k and t_{k+1}
Continuation values: $V^j_{ct}(t_{k+1})$ for each path ω_j and fixed time t_k
Ensure:
Regression coefficients $\lambda_{h,k}$, for all h and fixed k.

1: $\mathbf{y} = \{y_j \leftarrow \frac{\mathcal{N}^j(t_k)}{\mathcal{N}^j(t_{k+1})} V^j_{ct}(t_{k+1})\} \to \tilde{H}_j$
2: $\boldsymbol{A} = \{a_{j,h} \leftarrow \Psi_h(\mathbf{x}^j(t_k))\}$
3: find $\mathbf{z} = \{z_k\}$ such that $\underset{\text{regression on the number of paths}}{\sum_j} (y_j - \sum_h a_{j,h} z_h))^2 \equiv \|\mathbf{y} - \boldsymbol{A}\mathbf{z}\|^2$ minimized
4: $(\mathbf{y} - \boldsymbol{A}\mathbf{z}) \perp Im(\boldsymbol{A}) \implies (\mathbf{y} - \boldsymbol{A}\mathbf{z}) \in Ker(\boldsymbol{A}') \implies \boldsymbol{A}'(\mathbf{y} - \boldsymbol{A}\mathbf{z}) = \mathbf{0}$ (standard OLS, see Hayashi)
5: $\mathbf{z} = (\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}'\mathbf{y}$
6: **return** $\lambda_{h,k} = z_h$ for all h and fixed k.

Let's omit the t_k dependence for simplicity:

$$a_{jh} = \Psi_h(x^j) \to (A' \cdot A)_{rs} = a_{kr} \cdot a_{ks} = \underset{\text{summation over simulations}}{\sum_k} \Psi_r(x^k) \cdot \Psi_s(x^k) = N \cdot \underset{\text{A\&P, p. 162}}{\bar{\Psi}_{rs}}$$

$$(A' \cdot y)_r = a_{kr} \cdot y_k = \sum_k \Psi_r(x^k) \cdot \tilde{H}_k = N \cdot \underset{\text{A\&P, p. 162}}{\bar{\Omega}_{rs}} \Rightarrow \hat{\lambda}_k = \underset{= 1}{\frac{N}{N}} \cdot \bar{\Psi}_k^{-1} \cdot \bar{\Omega}_k \Rightarrow$$ Same result as in (3.115) A & P (and Glasserman) $\square$

A.3 Algorithm: Longstaff-Schwartz

Require:

Paths for regression: $\{\omega_j : j = 1 \ldots, N\}$,

Option exercise schedule: $\{t_k : k = \alpha, \ldots, \beta\}$

Regression basis functions $\{\Psi_h(\mathbf{x}) : h = 0, 1, 2, \ldots, r\}$

Regression Coefficients: $\lambda_{h,k}$ for basis function Ψ_h at time t_k (from dynamic programming algorithm)

Ensure:

Option price V at time 0

for $j = 1$ to N **do**

  $V^j \leftarrow 0$ {path value at current time, overwritten at each time t_k evaluation}

  for $k = \alpha$ to β **do**

    compute exercise value $V^j_{ex}(t_k)$ $= \mathcal{U}(t_k, x^j(t_k))$

    compute regression variables $\mathbf{x}^j(t_k)$ {evaluated on path ω_j evaluated at t_k}

    $\tilde{H}^j(t_k) =$ $V^j_{ct}(t_k) \leftarrow \sum_h \lambda_{h,k}\Psi_h(\mathbf{x}^j(t_k))$ {use regression results to estimate continuation value}

    if $V^j_{ex}(t_k) \geq V^j_{ct}(t_k)$ **then** (if this never happens, $V^j = 0$)

      (see below) $V^j \leftarrow \frac{1}{\mathcal{N}^j(t_k)} V^j_{ex}(t_k)$

      break {exercised, done with the path (hit the "Snell Envelope")}

    end if

  end for{over all exercise times t_k}

end for{over all paths ω_j}

$V \leftarrow \mathcal{N}(0)\frac{1}{N}\sum_{j=1}^N V^j$ {option value}

return V

$\hookrightarrow V = \frac{1}{N}\sum_{j=1}^{N} \frac{\mathcal{N}(0)}{\mathcal{N}^j(t_k)} V^j_{ex}(t_k)$

A.4 BLAN: Regression Variables

```
let Libor = market("US0003M Index") in
let Swap1 = market("USSWAP1 Index") in
let Swap2 = market("USSWAP2 Index") in
let Swap5 = market("USSWAP5 Index") in
let regression_vars(call_date, payment_date) = [Libor, Swap1, Swap2, Swap5] in

cancellable_of_schedule(..., regression_vars, ...)
```

References

[AA] L. Andersen and J. Andreasen. Factor Dependence of Bermudan Swaption Price: Fact or Fiction? (March 2000). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=209988.

[Blo1] Bloomberg. The little BLANer: An Interest Rate Contract Tutorial. DOCS #2071571⟨GO⟩.

[Blo2] Bloomberg. Pricing Structured Notes in YASN and SWPM. DOCS #2084167⟨GO⟩.

[Gla] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2010.

[Li] M. Li. Computing CVA with American Monte Carlo. Bloomberg Internal Publication, `TEAM 500316774`⟨GO⟩, July 2015.

[LS] F. Longstaff and E. Schwartz. Valuing American Options by Simulation: A Simple Least-Squares Approach. *The Review of Financial Studies* **14**(2001), 113–147.

[Pit] V. Piterbarg. Computing Deltas of Callable Libor Exotics in Forward Libor Models. *Journal of Computational Finance* **7**(2003). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=396180.