AAD and Least Squares Monte Carlo: Fast Bermudan-style Options and XVA Greeks

LUCA CAPRIOTTI * † YUPENG JIANG † ANDREA MACRINA† ‡

September 23, 2016

Abstract

We show how Adjoint Algorithmic Differentiation (AAD) can be used to calculate price sensitivities in regression-based Monte Carlo methods reliably and orders of magnitude faster than with standard finite-differences approaches. By discussing in detail examples of practical relevance, we demonstrate how accounting for the contributions associated with the regression functions is crucial to obtain accurate estimates of the Greeks for Bermudan-style options and XVA applications.

1 Introduction

The efficient calculation of the risk factor sensitivities of financial derivatives, also known as the "Greeks", is an essential component of modern risk management practices. Indeed, the aftermath of the recent financial crisis has seen remarkable changes in the market practice whereby financial institutions need to quantify (and risk-manage) counterparty, funding and capital risk exposures, collectively known as XVA, in large portfolios, see e.g. Crépey et al. (2014).

The traditional approach for the calculation of the Greeks is the so-called *bump and reval* or *bumping* technique. This comes with a significant computational cost as it generally requires repeating the calculation of the P&L of a portfolio under hundreds of market scenarios in order to form finite-difference estimators. As a result, in many cases, even after deploying vast amounts of computer power, these calculations cannot be completed in a practical amount of time.

Conversely, *Adjoint Algorithmic Differentiation* (AAD), a numerical technique recently introduced to financial engineering (see e.g., Capriotti (2011), Capriotti et al. (2011), Capriotti and Giles (2010, 2012), Henrard (2011)), has proven to be effective for speeding up the calculation of risk factor sensitivities, both for Monte Carlo (MC) and deterministic numerical methods, see Capriotti and Lee (2014) and Capriotti et al. (2015).

The main ideas underlying AAD can be illustrated by considering a computer implemented function of the form

$$Y = \mathtt{FUNCTION}(X) \tag{1}$$

mapping a vector $X \in \mathbb{R}^n$ to a vector $Y \in \mathbb{R}^m$ through a sequence of intermediate steps

$$X \to \ldots \to U \to V \to \ldots \to Y. \tag{2}$$

Here, the real-valued vectors U and V represent variables used in the calculation. Each step may be a distinct high-level function or even a specific instruction.

*Quantitative Strategies, Global Markets, Credit Suisse Group
†Department of Mathematics, University College London
‡Department of Actuarial Science, University of Cape Town

Electronic copy available at: http://ssrn.com/abstract=2842631

AAD, sometimes also simply known as adjoint mode of Algorithmic Differentiation (AD), results from "propagating" the derivatives of the final output with respect to all the intermediate variables – the so called *adjoints* – until the derivatives with respect to the independent variables are formed. Using the standard AD notation, the adjoint of any intermediate variable V_k is defined by

$$\bar{V}_k = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial V_k}, \tag{3}$$

where $\bar{Y}$ is a vector in $\mathbb{R}^m$. For each of the intermediate variables U_i, by applying the chain rule, we get

$$\bar{U}_i = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial U_i} = \sum_{j=1}^{m} \bar{Y}_j \sum_k \frac{\partial Y_j}{\partial V_k} \frac{\partial V_k}{\partial U_i},$$

which corresponds to the adjoint mode equation for the intermediate step represented by the function $V = V(U)$. We thus have a function of the form $\bar{U} = \bar{V}(U, \bar{V})$ where

$$\bar{U}_i = \sum_k \bar{V}_k \frac{\partial V_k}{\partial U_i}.$$

Starting from the adjoint of the outputs $\bar{Y}$, we may apply this rule to each step in the calculation, working from the right to the left,

$$\bar{X} \leftarrow \ldots \leftarrow \bar{U} \leftarrow \bar{V} \leftarrow \ldots \leftarrow \bar{Y} \tag{4}$$

until we obtain $\bar{X}$, namely, the linear combination of the rows of the Jacobian of the function $X \to Y$:

$$\bar{X}_i = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial X_i} \tag{5}$$

for $i = 1, \ldots, n$.

In the adjoint mode, the cost does not increase with the number of inputs, but it is linear in the number of (linear combinations of the) rows of the Jacobian that need to be evaluated independently. If the full Jacobian is required, one needs to repeat the adjoint calculation m times, setting the vector $\bar{Y}$ equal to each of the elements of the canonical basis in $\mathbb{R}^m$.

One particularly important theoretical result is that given a computer program performing some high-level function (1), the execution time of its adjoint counterpart

$$\bar{X} = \texttt{FUNCTION\_b}(X, \bar{Y}) \tag{6}$$

(with suffix `_b` for "backward" or "bar") that computes the linear combination (5), is bounded by three to four times the cost of execution of the original one. That is,

$$\frac{\mathrm{Cost}[\texttt{FUNCTION\_b}]}{\mathrm{Cost}[\texttt{FUNCTION}]} \leq \omega_A \tag{7}$$

where $\omega_A \in [3, 4]$, see Griewank (2000).

In this paper we present the application of AAD to regression-based MC approaches (also known as Least Squares MC) such as those that are widely used for Bermudan-style options, seeCarriere (1996), Longstaff and Schwartz (2001), Tsitsiklis and Van Roy (2001), or for XVA applications, see Cesari et al. (2009). We will develop the AAD implementation of the well-known least squares algorithm for the computation of conditional expectations, and we will investigate numerically the impact on the Greeks arising from the sensitivities of the regression functions, a component that is generally ignored for Bermudan-style options by invoking arguments of quasi-optimality of the exercise boundary.

Electronic copy available at: http://ssrn.com/abstract=2842631

The paper is organised as it follows: In the next section, the regression-based MC algorithm for both Bermudan-style options and XVA is presented. In Section 3 we discusses the AAD algorithms for the regression-based MC method. Here we also includes material on how to handle the discontinuities within path-wise differentiation. We give two numerical examples, the *best of two stocks Bermudan-style call* and its corresponding XVA in Section 4. Here we show how smoothening out discontinuities associated with suboptimal exercise boundaries improves the accuracy of the Greeks of Bermudan-style options, and why the contribution to the sensitivities arising from the regression boundaries is essential for an accurate computation of XVA sensitivities. The efficiency and accuracy of AAD is also compared with bump and reval approaches in the same section.

2 Valuation of Bermudan-style options and XVA by regression-based Monte Carlo

2.1 Bermudan-style options

While European-style options can be exercised only at final maturity, Bermudan-style options can be exercised on multiple dates up to trade expiry. We denote by $T_1, \ldots, T_M$ the exercise dates of the option and define $\mathcal{D}(t) = \{T_m \geq t\}$. We denote by $\eta(t)$ the smallest integer such that $T_{\eta(t)+1} > t$. An *exercise policy* is represented mathematically by a *stopping time* taking values in $\mathcal{D}(t)$. We denote by $\mathcal{T}(t)$ the set of stopping times taking values in $\mathcal{D}(t)$.

A rational investor will exercise the option that he holds in such a way as to maximise its economic value. As a result, the value of a Bermudan-style option is the supremum of the option value over all possible exercise policies. With the notation introduced above, the value of a Bermudan-style option at time t can therefore be expressed by

$$\frac{V(t)}{N(t)} = \sup_{\tau \in \mathcal{T}(t)} \mathbb{E}\left[\frac{E(\tau)}{N(\tau)}\right] , \tag{8}$$

where $E(t)$ is the exercise value of the option, and $N(t)$ is the chosen *numéraire*[1], see Andersen and Piterbarg (2010). In this equation, $V(t)$ is to be interpreted as the early-exercise value of the option conditional on exercise not having taken place *strictly before* time t.

A useful concept is the *hold value* of the Bermudan-style option. We denote by $H(t)$ the value of the Bermudan-style option when the exercise dates are restricted to $\mathcal{D}(T_{\eta(t)+1})$, that is

$$\frac{H(t)}{N(t)} = \mathbb{E}_t\left[\frac{V\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)}\right] , \tag{9}$$

where we have assumed, for simplicity of exposition, no cashflow between t and $T_{\eta(t)+1}$.

The option holder, following an optimal exercise policy, will exercise his option if the exercise value is larger than the hold value, i.e.,

$$V\left(T_{\eta(t)}\right) = \max\left(E\left(T_{\eta(t)}\right), H\left(T_{\eta(t)}\right)\right) . \tag{10}$$

This, when combined with Equation (9), leads to the so-called *dynamic programming* formulation:

$$\frac{H(t)}{N(t)} = \mathbb{E}_t\left[\max\left(\frac{E\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)}, \frac{H\left(T_{\eta(t)+1}\right)}{N\left(T_{\eta(t)+1}\right)}\right)\right] , \tag{11}$$

[1] In the following, for simplicity of notation, we will set $N(0) = 1$.

for $T_\eta \leq t < T_{\eta+1}$, and $\eta = 1, \ldots, M-1$. Starting from the terminal condition $H(T_M) \equiv 0$, Equation (11) defines a backward iteration in time for $H(t)$. By definition, this is also equal to $V(t)$ if t is not an exercise date, i.e. if $T_{\eta(t)} < t < T_{\eta(t)+1}$. Conversely, if t is an exercise date, $t = T_{\eta(t)}$, then $V(T_{\eta(t)}) = \max(E(T_{\eta(t)}), H(T_{\eta(t)}))$.

The dynamic programming formulation above implies that the stopping time, defining the optional exercise date as seen at time t, is given by

$$\tau^\star = \inf[T_m \geq t : E(T_m) \geq H(T_m)] \ . \tag{12}$$

The optimal exercise strategy defined by Equation (12) requires the computation of the hold value $H(t)$, $m = \eta(t)+1, \ldots, M-1$. In a setting in which the underlying risk factor process $\{X(t)\}_{0 \leq t \leq T}$ is a generic k-dimensional Markov process, the hold value $H(t)$ is a function of the state vector at time t. That is,

$$H_t(x) := \mathbb{E}\left[\frac{N\left(X(t)\right)}{N\left(X(T_{m+1})\right)} V\left(X\left(T_{m+1}\right)\right) \Big| X(t) = x \right] \ . \tag{13}$$

When the dimension of the Markov process k is small enough, the conditional expectation value in Equation (13) can be computed in a straightforward way by discretising the risk-factor process and performing standard backward induction on a tree or a grid, or by discretising an associated PDE. Here we refer to, e.g. Wilmott et al. (1995). However, the complexity of grid-based calculations is exponential in the dimension of the Markov process and numerical implementations become infeasible when $k \geq 4$. As we will review in Section 2.3, regression-based MC techniques provide an effective way of computing conditional expectation values of the form (13).

2.2 XVA

We next consider the computation of the Credit Valuation Adjustment (CVA) and the Debt Valuation Adjustment (DVA) as the main measures of a dealer's counterparty credit risk, see e.g. Crépey et al. (2014). For a given portfolio of trades with the same investor or institution, the CVA (*resp.* DVA) aims to capture the expected loss (*resp.* gain) associated with the counterparty (*resp.* dealer) defaulting in a situation in which the position, netted for any collateral posted, has a positive mark-to-market for the dealer (*resp.* counterparty).

This can be evaluated at time $T_0 = 0$ by

$$XVA = -\mathbb{E}\left[\mathbb{I}(\tau_c \leq T) \frac{L_c}{N(\tau_c)} \left(V(\tau_c)\right)^+ + \mathbb{I}(\tau_d \leq T) \frac{L_d}{N(\tau_d)} \left(V(\tau_d)\right)^- \right] \ , \tag{14}$$

where τ_c (*resp.* τ_d) is the default time of the counterparty (*resp. the dealer*), $V(t)$ is the net present value of the portfolio or *netting set* at time t from the dealer's point of view (the so-called conditional future exposure), L_c (*resp.* L_d) is the loss given default of the counterparty (*resp.* the dealer), and $\mathbb{I}(\tau_c \leq T)$ (*resp.* $\mathbb{I}(\tau_d \leq T)$) is the indicator that the counterparty's (*resp.* dealer's) default happens before the longest deal maturity T in the portfolio. Here, for simplicity of notation, we consider the unilateral CVA and DVA, the generalization to the bilateral formulation, see e.g. Crépey et al. (2014), being straightforward.

Equation (14) is typically computed on a discrete time grid of "horizon dates" $0 = T_0 < T_1 < \ldots < T_M$. For instance, we may have

$$XVA \simeq -\sum_{m=1}^{M} \mathbb{E}\left[L_c(\mathrm{SP}_c(T_{m-1}) - \mathrm{SP}_c(T_m)) \frac{(V(T_m))^+}{N(T_m)} + L_d(\mathrm{SP}_d(T_{m-1}) - \mathrm{SP}_d(T_m)) \frac{(V(T_m))^-}{N(T_m)} \right] \ , \tag{15}$$

where $\mathrm{SP}_c(t)$ (*resp.* $\mathrm{SP}_d(t)$) is the survival probability of the counterparty (*resp.* the dealer) up to time t, e.g., conditional on a realization of the default intensity (or hazard rate) process in a Cox framework, see Lando (1998). Here we assume that the default times τ_c and τ_d are conditionally independent of the portfolio values V_{τ_c} and V_{τ_d}, respectively. In general, the right hand side of Equation (15) depends on several correlated random market factors, including interest rate, recovery amounts, and all the market factors the net conditional future exposure of the portfolio, $V(t)$, depends on. As such, its calculation typically requires a MC simulation.

In the k-dimensional Markov setting introduced above the conditional future exposure $V(T_m)$ is a function $V(X(T_m))$ of the state vector at time T_m. However, only for vanilla securities and simple models for the evolution of the risk factors, such conditional future exposures can be expressed in closed form, see e.g. Cesari et al. (2009). In order to illustrate how the conditional future exposure can be computed by means of regression-based MC, we consider a specific example in which the underlying portfolio contains a basket of Bermudan-style options.

2.3 Conditional expectation values and Bermudan-style options by regression

Regression methods are based on the observation, see e.g., Friedman et al. (2001), that given a real-valued random input vector $X \in \mathbb{R}^d$ and Y a real valued random output, the conditional expectation $\mathbb{E}[Y|X]$ is the function of X that best approximates in the least squares sense the output Y. That is,

$$\mathbb{E}[Y|X] = \mathrm{argmin}_c \mathbb{E}\left[(Y-c)^2\right] . \tag{16}$$

In particular, assuming that the conditional expectation is a linear function of some unknown vector of parameters β,

$$\mathbb{E}[Y|X] = \beta^T X, \tag{17}$$

Equation (16) reduces to the well-known linear regression conditions, giving for the optimal vector of parameters

$$\beta = \mathbb{E}\left[XX^T\right]^{-1}\mathbb{E}[XY] . \tag{18}$$

In the context of the valuation of Bermudan-style options, the hold value (13) on an exercise date T_m is fassumed to be of the form

$$\hat{H}_m(x) = \beta_m^T \psi(x) \tag{19}$$

where $\psi(x) = (\psi_1(x), \ldots, \psi_d(x))^T$ is a vector of d basis functions and $\beta_m = (\beta_{1m}, \ldots, \beta_{dm})^T$ is the vector of coefficients to be determined by regressing $N(X(T_m))/N(X(T_{m+1}))V(X(T_{m+1}))$ versus $\psi(X(T_{m+1}))$. This gives

$$\beta_m = \Psi_m^{-1}\Omega_m , \tag{20}$$

where we define the $d \times d$ matrix

$$\Psi_m = \mathbb{E}\Big[\psi(X(T_m))\psi^T(X(T_m))\Big] \tag{21}$$

and the $d \times 1$ vector

$$\Omega_m = \mathbb{E}\left[\frac{N(X(T_m))V(X(T_{m+1}))}{N(X(T_{m+1}))}\psi(X(T_m))\right] . \tag{22}$$

These equations provide a straightforward recipe to compute the regression coefficients β_m by substituting Ψ_m and Ω_m with their sample average over N_{MC} MC replications. These are:

(R1) Simulate N_{MC} independent MC paths $X_m^{(n)}$ of $X(T_m)$ by the recursion

$$X_{m+1}^{(n)} = F\left(T_m, X_m^{(n)}, \theta\right) \tag{23}$$

for $m = 0, \ldots, M-1$, and $n = 1, \ldots, N_{\mathrm{MC}}$. Here F is a function based on the chosen models for the risk factors, and θ is a vector of model parameters.

(R2) For $n = 1, \ldots, N_{\mathrm{MC}}$, compute the terminal payoff of the contract by setting

$$V_M^{(n)} = E_M^{(n)} \,, \tag{24}$$

where $E_M := E(X_M^n)$ is the final exercise value of the option.

(R3) Apply the following backward induction steps for $m = M-1, \ldots, 1$:

(a) Compute the MC sample average of Ψ_m and ${\Omega_m}^2$ by

$$\Psi_m = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} (\psi_m^{(n)})^T \,, \tag{25}$$

$$\Omega_m = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} \frac{N_m^{(n)} V_{m+1}^{(n)}}{N_{m+1}^{(n)}} \,, \tag{26}$$

where $\psi_m^{(n)} := \psi\left(X_m^{(n)}\right)$ and $N_m^{(n)} := N(X_m^{(n)})$.

(b) Compute the regression coefficients β_m by matrix inversion and multiplication:

$$\beta_m = \Psi_m^{-1} \Omega_m. \tag{27}$$

(c) For the estimate of the hold value $H_m^{(n)} := H_m\left(X_m^{(n)}\right)$, set

$$H_m^{(n)} = \beta_m^T \psi_m^{(n)} \,, \tag{28}$$

for $n = 1, \ldots, N_{\mathrm{MC}}$.

(d) For the estimate of the Bermudan-style option value at time T_m, set

$$V_m^{(n)} = \max\left(E_m^{(n)}, H_m^{(n)}\right) \,, \tag{29}$$

where $E_m^{(n)} := E(X_m^{(n)})$ is the exercise value at time T_m, for $n = 1, \ldots, N_{\mathrm{MC}}$.

(R4) Compute the MC estimate of the Bermudan-style option at time T_0 by

$$V_0 = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} \frac{V_1^{(n)}}{N_1^{(n)}} \,. \tag{30}$$

This approach was introduced by Tsitsiklis and Van Roy (2001) and they showed that the estimator V_0 converges for $n \to \infty$ to the true value $V(0)$ provided that the representation (19) holds exactly.

A modification of this algorithm was proposed by Longstaff and Schwartz (2001) and it entails replacing Equation (29) in Step R3 (d) with

$$V_m^{(n)} = \begin{cases} E_m^{(n)} & \text{if } \; E_m^{(n)} > H_m^{(n)} \,, \\ N_m^{(n)} V_{m+1}^{(n)} / N_{m+1}^{(n)} & \text{otherwise,} \end{cases} \tag{31}$$

which, in the examples considered, was shown to lead to more accurate results. In the following, however, for simplicity of exposition, we will consider the estimator in Eq. (29).

[2] Here and in the following, to keep the notation simple, we do not introduce different symbols for expectations and the respective sample averages.

2.4 Lower bound algorithm for Bermudan-style options

The hold value obtained by regression as described in the previous section defines an exercise policy whereby on each exercise date T_m the option is exercised if

$$E(X(T_m)) > \beta_m^T \psi(X(T_m)) \ . \tag{32}$$

Such policy, being an approximation of the solution of the the dynamic programming equation (11), will in general correspond to a suboptimal stopping time. As a result, when used in a second, independent, MC simulation, the exercise policy obtained by regression, will result in a lower-bound estimator for the Bermudan-style option value. The corresponding algorithm can be schematically described as it follows.

For each MC replication indexed by $n = 1, \ldots, N_{\text{MC}}$ perform steps (L1) to (L4) below:

(L1) Simulate the path $X_m^{(n)}$ of the risk factor vector $X(T_m)$ as in (R1).

(L2) For $m = 1, 2, ..., M-1$, compute the approximate hold value of the option at time T_m using the associated regression vector β_m, and regression functions ψ, by

$$H_m^{(n)} = \beta_m^T \psi_m^{(n)} \tag{33}$$

with the hold value at expiry T_M set to zero.

(L3) Compute the path-wise estimator for the discounted cash-flows of the option

$$P^{(n)} = \sum_{m=1}^{M} \left[\mathbf{1}^{(n)}(t_1, t_m) \mathbf{1} \left(E_m^{(n)} > H_m^{(n)} \right) \frac{E_m^{(n)}}{N_m^{(n)}} \right] \ , \tag{34}$$

where

$$\mathbf{1}^{(n)}(t_1, t_m) = \left(\prod_{i=1}^{m-1} \mathbf{1} \left(H_i^{(n)} > E_i^{(n)} \right) \right) \tag{35}$$

and the convention $\mathbf{1}^{(n)}(t_1, t_1) = 1$.

(L4) Compute the MC estimate of the Bermudan-style option at time $T_0 = 0$ by

$$V_0 = \frac{1}{N_{\text{MC}}} \sum_{n=1}^{N_{\text{MC}}} P^{(n)} \ . \tag{36}$$

2.5 XVA by regression

As described in Section 2.2, the calculation of the XVA in Equation (15) requires the conditional future exposure $V(t)$ on a set of dates determined by a discretisation time grid $T_1 \ldots T_M$. The regression algorithm described in the previous section can be easily adapted to compute such quantity. Indeed, the conditional value of each of the options contained in the netting set can be obtained using the same least squares procedure. Once the regression algorithm is completed, we can use the regression functions to compute the hold value of each option in the portfolio on the discretisation time grid by $H_m = \beta_m^T \psi_m$. If the discretisation time T_m is not an exercise opportunity for the option under consideration, then this is also its conditional future exposure. Conversely, the conditional future exposure is obtained by comparing the hold value to the exercise value as in Equations (29) and (31). These observations translate in the following algorithm.

For each MC replication indexed by $n = 1, \ldots, N_{\text{MC}}$ perform steps (X1) to (X3) below:

(X1) Simulate the path $X_m^{(n)}$ of the risk factor vector by the recursion:

$$X_{m+1}^{(n)} = F\left(T_m, X_m^{(n)}, \theta\right) , \tag{37}$$

for $m = 0, \ldots, M-1$. Simulate the path of the counterparty's and the dealer's default intensity, $\lambda_m^{d,c} = \lambda^{d,c}(T_m)$ by the recursions

$$\lambda_{m+1}^{\mathrm{c},(n)} = G_c\left(T_m, \lambda_m^{\mathrm{c},(n)}, \theta\right) , \tag{38}$$

$$\lambda_{m+1}^{\mathrm{d},(n)} = G_d\left(T_m, \lambda_m^{\mathrm{d},(n)}, \theta\right) , \tag{39}$$

for $m = 0, \ldots, M-1$, where G_c (*resp.* G_d) is the function describing the dynamics of the counterparty's (*resp.* the dealer's) hazard rate.

(X2) Compute the (discretised) path-wise survival probabilities for the counterparty and the dealer by

$$\mathrm{SP}_m^{\mathrm{c},(n)} = \exp\left[-\sum_{j=0}^{m-1} \lambda_j^{\mathrm{c},(n)} \left(T_{j+1} - T_j\right)\right] , \tag{40}$$

$$\mathrm{SP}_m^{\mathrm{d},(n)} = \exp\left[-\sum_{j=0}^{m-1} \lambda_j^{\mathrm{d},(n)} \left(T_{j+1} - T_j\right)\right] , \tag{41}$$

for $m = 1, 2, ..., M$.

(X3) For $m = 1, 2, ..., M-1$, approximate the hold value of the p-th option in the portfolio at time T_m using the associated regression vector $\beta_{p,m}$ and regression functions ψ_p by

$$H_{p,m}^{(n)} = \beta_{p,m}^T \psi_{p,m} , \tag{42}$$

with $p = 1, \ldots, P$. The hold value at the expiry date T_M is set to zero. The conditional expectation value of the portfolio is given by $V_m^{(n)} = \sum_{p=1}^{P} V_{p,m}^{(n)}$, where

$$V_{p,m}^{(n)} = \begin{cases} \max\left\{H_{p,m}^{(n)}, E_{p,m}^{(n)}\right\} , & \text{if } T_m \text{ is an exercise date for the } p\text{-th option} \\ H_{p,m}^{(i)} , & \text{otherwise,} \end{cases} \tag{43}$$

for $m = 1, 2, ..., M$, where $E_{p,m}^{(n)}$ is the exercise value of the p-th option at time T_m on the n-th path.

(X4) Compute the path-wise XVA by

$$XVA^{(n)} = -\sum_{m=1}^{M} \left[L_c \left(\mathrm{SP}_{m-1}^{\mathrm{c},(n)} - \mathrm{SP}_m^{\mathrm{c},(n)}\right) \frac{\left(V_m^{(n)}\right)^+}{N_m^{(n)}} + L_d \left(\mathrm{SP}_{m-1}^{\mathrm{d},(n)} - \mathrm{SP}_m^{\mathrm{d},(n)}\right) \frac{\left(V_m^{(n)}\right)^-}{N_m^{(n)}} \right] . \tag{44}$$

(X5) Form the MC estimator

$$XVA = \frac{1}{N_{\mathrm{MC}}} \sum_{n=1}^{N_{\mathrm{MC}}} XVA^{(n)} . \tag{45}$$

An alternative approach was recently suggested in Joshi and Kwon (2016) which we will not consider here, leaving it as the subject of further research.

3 The AAD algorithm for regression-based Monte Carlo

3.1 Function regularisations

In the context of MC methods, Capriotti and Giles (2012) show that AAD allows to calculate the sensitivities by differentiating the relevant estimator on a path by path basis. As a path-wise method, the MC estimators must satisfy specific regularity conditions, see Glasserman (2004). For instance, all the functions appearing in each step leading to the computation of the payout estimator must be Lipschitz continuous. This requirement is usually cited in the literature as a shortcoming of path-wise schemes. Indeed, it potentially limits their practical utility as the majority of the payout functions commonly used for structured derivatives contains discontinuities, e.g., in the form of digital features, random variables counting discrete events, or barriers.

Fortunately the Lipschitz continuity requirement turns out to be more of a theoretical than a practical limitation. Indeed, a practical way of addressing non-Lipschitz estimators is to smoothen out the singularities they contain. Such a procedure introduces a finite bias in the estimates of the sensitivities. However, as we illustrate below, such bias can be reduced to levels that are considered acceptable for risk management purposes and is comparable with the bias industry practitioners face when computing sensitivities by finite differences.

In this Section, we describe a general way to address the most common singularities arising in practice. This can be achieved by observing that in most cases the singularities in the payout functions, although not necessarily implemented as such, can be expressed in terms of Heaviside functions. For instance, the payoff function of a digital option is,

$$P(X(T)) = \mathbf{1}(X(T) > K) \; , \tag{46}$$

while the payoff of a knock-out, path-dependent option with barrier monitored at the time $T_1, \ldots, T_M$ is of the form

$$P(X(T_1), \ldots, X(T_M)) = \prod_{m=1}^{M} \mathbf{1}(X(T_m) > B_m) \tag{47}$$

where B_m is a k-dimensional vector representing the barrier level for the corresponding state variable vector at time T_m. The singularities in such payoff functions can be regularised by replacing the indicator function with one of its smoothened counterparts. A very common choice, for instance, is to approximate the step function with a "call spread" payoff functions,

$$\mathbf{1}\left(x > K\right) \approx \mathcal{H}_\delta^{\text{cs}}\left(x - K\right) = \left(\min\left(\frac{x - (K - \delta)}{2\delta}, 1\right)\right)^+ , \tag{48}$$

where $\delta \ll K$. This is a standard choice for digital options, because it has a useful interpretation in terms of the hedging portfolio of a long and a short position in two calls with strike price $K - \delta$ and $K + \delta$. Alternatively, one can approximate the indicator function with a cumulative normal density function with zero mean and standard deviation δ, that is,

$$\mathbf{1}\left(x > K\right) \approx \mathcal{H}_\delta^{\text{cn}}\left(x - K\right) = \int_{-\infty}^{x-K} \frac{\exp\left(-u^2/2\delta^2\right)}{\delta\sqrt{2\pi}} \mathrm{d}u. \tag{49}$$

Both regularisations give rise to functions that are Lipschitz continuous with respect to x and can be differentiated in a straightforward manner. In particular, the adjoint regularized Heaviside functions read

$$\bar{\mathcal{H}}_\delta(x - K, \bar{x}) = \bar{x}\,\frac{\partial}{\partial x}\mathcal{H}_\delta(x - K) \; . \tag{50}$$

For the Call spread regularization (48) we then have

$$\bar{\mathcal{H}}_\delta^{\text{cs}}(x - K, \bar{x}) = \begin{cases} \bar{x}/2\delta & \text{if } K - \delta \le x \le K + \delta \\ 0 & \text{otherwise} \end{cases} \tag{51}$$

and, for the cumulative normal regularization (49),

$$\bar{\mathcal{H}}^{\text{cn}}_{\delta}(x - K, \bar{x}) = \bar{x}\, \phi_{\delta}(x - K) \tag{52}$$

where $\phi_{\delta}(x)$ is the Normal density function with standard deviation δ. When $\delta \to 0$, both (48) and (49) give the correct derivative of the Heaviside function in the distributional sense, i.e., the Dirac delta function. However, while approaching this limit, the derivative of such regularised Heaviside functions is zero or vanishingly small, apart from a very small portion of the sample space where instead it is very large. This leads to exceedingly large variances in the MC sampling of the estimators expressed in terms of such functions, signaling the breakdown of the Lipschitz continuity condition. Hence, the choice of the smoothening parameter δ is necessarily a tradeoff between the bias, vanishing for $\delta \to 0$, and the statistical errors of the MC sampling, diverging in the same limit.

In general, the payoff estimator for Bermudan-style options (34) is not differentiable with respect to the pathway value of the approximate exercise boundary $H_m^{(n)}$, and it requires the regularization described above. A common approximation among practitioners, see e.g., Leclerc et al. (2009), is to assume that that exercise boundary implied by the rule (32) is close to optimality so that the value of the contract is approximatively continuous across the exercise boundary and no regularization is required. As shown in Appendix A, under this assumption, no contribution to the sensitivities is associated with the perturbations of the exercise boundary, and one can therefore keep the regression coefficients fixed while calculating the sensitivities. As discussed in Section 4, depending on the accuracy of the basis functions in representing the exercise boundary, this may or may not be an accurate approximation.

3.2 AAD for the lower bound algorithm for Bermudan-style options

The AAD implementation of the lower bound algorithm for Bermudan-style options described in Section 2.4, producing the MC estimators for the sensitivities of the estimator (36) with respect to a set of model parameters θ_k, $k = 1, \ldots, N_\theta$, given by

$$\bar{\theta}_k = \frac{\partial V(\theta)}{\partial \theta_k}\ , \tag{53}$$

consists of the adjoint of steps (L1) to (L4) executed backwards for each MC replication $n = 1, \ldots, N_{\text{MC}}$. These are:

($\overline{\text{L4}}$) Set the adjoint of the option value $\bar{V} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$ and set

$$\bar{P}^{(n)} = \bar{V}_0 \frac{1}{N_{\text{MC}}}\ . \tag{54}$$

($\overline{\text{L3}}$) Assuming the indicator functions in the estimator in (34) have been regularised as discussed in Section 3.1, compute

$$\begin{aligned}
\bar{H}_m^{(n)} &= \bar{\mathcal{H}}_\delta\left(H_m^{(n)} - E_m^{(n)}, \bar{P}^{(n)}\right)\left(P_m^{(n)} - Q_m^{(n)}\right),\\
\bar{E}_m^{(n)} &= \bar{P}^{(n)} R_m^{(n)} + \bar{\mathcal{H}}_\delta\left(E_m^{(n)} - H_m^{(n)}, \bar{P}^{(n)}\right)\left(P_m^{(n)} - Q_m^{(n)}\right),\\
\bar{N}_m^{(n)} &= -\bar{P}^{(n)} \frac{Q_m^{(n)} \mathcal{H}_\delta\left(E_m^{(n)} - H_m^{(n)}\right)}{N_m^{(n)}}
\end{aligned} \tag{55}$$

for $m = M, \ldots, 1$ and where

$$
\begin{aligned}
P_m^{(n)} &= \sum_{k=m+1}^{M} \frac{E_k^{(n)}}{N_k^{(n)}} \mathbf{1}\left(E_k^{(n)} > H_k^{(n)}\right) \prod_{j=2, j\neq m}^{k-1} \mathbf{1}\left(H_j^{(n)} > E_j^{(n)}\right) , \\
Q_m^{(n)} &= \frac{E_m^{(n)}}{N_m^{(n)}} \mathbf{1}^{(n)}(t_1, t_m) , \\
R_m^{(n)} &= \frac{1}{N_m^{(n)}} \mathbf{1}\left(E_m^{(n)} > H_m^{(n)}\right) \mathbf{1}^{(n)}(t_1, t_m) .
\end{aligned}
$$

Here we also adopt the conventions $\prod_{j=2,j\neq m}^{1} \ldots = 1$, $\sum_{M+1}^{M} \ldots = 0$. Initialise the adjoints of the risk factors so that

$$
\bar{X}_m^{(n)} = \bar{E}_m^{(n)} \frac{\partial E_m^{(n)}}{\partial X_m^{(n)}} + \bar{N}_m^{(n)} \frac{\partial N_m^{(n)}}{\partial X_m^{(n)}} . \tag{56}
$$

($\overline{\text{L2}}$) For $m = M, \ldots, 1$, initialise the adjoint of the regression coefficients β_m to give

$$
\bar{\beta}_m = \sum_{n=1}^{N_{\mathrm{MC}}} \psi_m^{(n)} \bar{H}_m^{(n)} , \tag{57}
$$

as well as the adjoints of the basis functions $\psi_m^{(n)}$,

$$
\bar{\psi}_m^{(n)} = \beta_m \bar{H}_m^{(n)} , \tag{58}
$$

and update the adjoints of the state vector

$$
\bar{X}_m^{(n)} \mathrel{+}= (\bar{\psi}_m^{(n)})^T \frac{\partial \psi_m^{(n)}}{\partial X_m^{(n)}} , \tag{59}
$$

where we use the standard notation $+=$ for the standard addition assignment operator.

($\overline{\text{L1}}$) For $m = M, \ldots, 0$ compute the adjoint of the risk factor evolution such that

$$
\bar{X}_m^{(n)} \mathrel{+}= \bar{X}_{m+1}^{(n)} \frac{\partial F}{\partial X_m^{(n)}} \left(T_m, X_m^{(n)}; \theta\right) , \quad \bar{\theta} \mathrel{+}= \bar{X}_{m+1}^{(n)} \frac{\partial F}{\partial \theta} \left(T_m, X_m^{(n)}, \theta\right) , \tag{60}
$$

where the gradients are computed by applying the rules of adjoint differentiation following the instructions that implement the function F. Finally, the adjoint $\bar{X}_0$ is used to populate the component of $\bar{\theta}$ corresponding to the adjoint of the model parameter X_0.

3.3 AAD for XVA by regression

The AAD implementation of the algorithm for the calculation of XVA described in Section 2.5, producing the MC estimators for the sensitivities of the estimator (45) with respect to a set of model parameters θ_k, $k = 1, \ldots, N_\theta$, given by

$$
\bar{\theta}_k = \frac{\partial XVA(\theta)}{\partial \theta_k} , \tag{61}
$$

consists of the adjoint of steps (X1)-(X5) executed backwards for each MC replication $n = 1, \ldots, N_{\mathrm{MC}}$:

($\overline{\text{X5}}$) Set the adjoint of the XVA value, $\overline{XVA} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$ and

$$\overline{XVA}^{(n)} = \overline{XVA}\frac{1}{N_{\text{MC}}} \; . \tag{62}$$

($\overline{\text{X4}}$) For $m = M, \ldots, 1$ compute:

$$\bar{V}_m^{(n)} = -\frac{\overline{XVA}^{(n)}}{N_m^{(n)}}\left[L_c\left(\text{SP}_{m-1}^{c,(n)} - \text{SP}_m^{c,(n)}\right)\mathbf{1}(V_m^{(n)} > 0) + L_d\left(\text{SP}_m^{d,(n)} - \text{SP}_m^{d,(n)}\right)\mathbf{1}(V_m^{(n)} < 0)\right], \tag{63}$$

$$\bar{N}_m^{(n)} = \frac{\overline{XVA}^{(n)}}{\left(N_m^{(n)}\right)^2}\left[L_c\left(\text{SP}_{m-1}^{c,(n)} - \text{SP}_m^{c,(i)}\right)\left(V_m^{(n)}\right)^+ + L_d\left(\text{SP}_{m-1}^{d,(n)} - \text{SP}_m^{d,(n)}\right)\left(V_m^{(n)}\right)^-\right], \tag{64}$$

$$\overline{\text{SP}}_m^{c,(n)} = \overline{XVA}^{(n)}\left[\frac{V_m^{(n)}}{N_m^{(n)}}(1-\delta_{m,0}) - \frac{V_{m+1}^{(n)}}{N_{m+1}^{(n)}}\right]\mathbf{1}(V_m^{(n)} > 0) \; , \tag{65}$$

$$\overline{\text{SP}}_m^{d,(n)} = \overline{XVA}^{(n)}\left[\frac{V_m^{(n)}}{N_m^{(n)}}(1-\delta_{m,0}) - \frac{V_{m+1}^{(n)}}{N_{m+1}^{(n)}}\right]\mathbf{1}(V_m^{(n)} < 0) \; , \tag{66}$$

where $\delta_{m,n}$ is the Kronecker symbol, and $V_{M+1}^{(n)}/N_{M+1}^{(n)} = 0$.

($\overline{\text{X3}}$) For $m = M, \ldots, 1$, set $\bar{V}_{p,m}^{(n)} = \bar{V}_m^{(n)}$ and compute the adjoint of Equation (43) by

$$\bar{H}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)}\mathbf{1}\left(H_{p,m}^{(n)} > E_{p,m}^{(n)}\right) \; , \quad \bar{E}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)}\mathbf{1}\left(H_{p,m}^{(n)} < E_{p,m}^{(n)}\right) \; , \tag{67}$$

if T_m is an exercise date for the p-th option, and

$$\bar{H}_{p,m}^{(n)} = \bar{V}_{p,m}^{(n)} \; , \quad \bar{E}_{p,m}^{(n)} = 0 \; , \tag{68}$$

otherwise. Initialise the adjoints of the risk factors,

$$\bar{X}_m^{(n)} = \bar{E}_{p,m}^{(n)}\frac{\partial E_{p,m}^{(n)}}{\partial X_m^{(n)}} + \bar{N}_m^{(n)}\frac{\partial N_m^{(n)}}{\partial X_m^{(n)}} \; . \tag{69}$$

Initialise the adjoint of the regression coefficients $\beta_{p,m}$, the adjoints of the basis functions $\psi_{p,m}$, and update the adjoints of the state vector as in Equations (57) to (59).

($\overline{\text{X2}}$) Update the adjoint of the simulated hazard rates according to

$$\bar{\lambda}_m^{c,(n)} \mathrel{+}= -\left(T_{m+1} - T_m\right)\sum_{j=m+1}^{M}\overline{\text{SP}}_m^{c,(n)}\text{SP}_m^{c,(n)} \; , \quad \bar{\lambda}_m^{d,(n)} \mathrel{+}= -\left(T_{m+1} - T_m\right)\sum_{j=m+1}^{M}\overline{\text{SP}}_m^{d,(n)}\text{SP}_m^{d,(n)} \; . \tag{70}$$

($\overline{\text{X1}}$) For $m = M, \ldots, 0$ compute the adjoint of the hazard rate evolution

$$\bar{\lambda}_m^{c,(n)} \mathrel{+}= \bar{\lambda}_{m+1}^{c,(n)}\frac{\partial G_c}{\partial \lambda_m^{c,(n)}}\left(T_m, \lambda_m^{c,(n)}; \theta\right) \; , \quad \bar{\lambda}_m^{d,(n)} \mathrel{+}= \bar{\lambda}_{m+1}^{d,(n)}\frac{\partial G_d}{\partial \lambda_m^{d,(n)}}\left(T_m, \lambda_m^{d,(n)}; \theta\right) \; , \tag{71}$$

$$\bar{\theta} \mathrel{+}= \bar{\lambda}_{m+1}^{c,(n)}\frac{\partial G_c}{\partial \theta}\left(T_m, \lambda_m^{c,(n)}, \theta\right) \; , \quad \bar{\theta} \mathrel{+}= \bar{\lambda}_{m+1}^{d,(n)}\frac{\partial G_d}{\partial \theta}\left(T_m, \lambda_m^{d,(n)}, \theta\right) \; , \tag{72}$$

where the gradients can be computed through the AAD implementation of the function G and of the risk factor evolution as in step ($\overline{\text{L1}}$).

3.4 AAD for the regression algorithm

In the AAD implementations presented in Section 3.2 and Section 3.3 the adjoint of the regression coefficients in Equation (57) do not contribute to the calculation of sensitivities. As mentioned in Section 3.1, and illustrated with numerical examples in Section 4, this can be justified as a reasonable approximation in the case of Bermudan-style options when the exercise boundary approximated by regression is close to the optimal one. However, such arguments are generally approximate for Bermudan-style options, and cannot be invoked at all when regression is utilised for XVA. In this case, the contributions associated with the sensitivities of the regression coefficients must be taken into account in order to obtain accurate estimates of the model parameters sensitivities.

In this section, we discuss how these contributions can be computed by the AAD implementation of the least squares MC algorithm described in Section 2.3.

($\overline{\text{R4}}$) Skip this step if the regression algorithm is used in conjunction with a second independent simulation for Bermudan-style options or in the context of XVA. Initialise the adjoint of the option value $\bar{V} = 1$, the adjoint of the model parameters $\bar{\theta} = 0$, the adjoint of the regression coefficients $\bar{\beta}_m = 0$, $m = 1, \ldots, M-1$ and set

$$\bar{V}_1^{(n)} = \frac{\bar{V}}{N_{\text{MC}}} \frac{1}{N_1^{(n)}} , \qquad \bar{N}_1^{(n)} = -\frac{\bar{V}}{N_{\text{MC}}} \frac{V_1^{(n)}}{\left(N_1^{(n)}\right)^2} \tag{73}$$

for $n = 1, \ldots, N_{\text{MC}}$.

($\overline{\text{R3}}$) For $m = 1$ to $M-1$, we have:

($\bar{\text{d}}$) For $n = 1, \ldots, N_{\text{MC}}$ compute

$$\bar{E}_m^{(n)} = \bar{V}_m^{(n)} \mathbf{1}\Big(E_m^{(n)} > H_m^{(n)}\Big), \qquad \bar{H}_m^{(n)} = \bar{V}_m^{(n)} \mathbf{1}\Big(E_m^{(n)} < H_m^{(n)}\Big), \tag{74}$$

and initialise the adjoint of the risk factor path value $X_m^{(n)}$ by

$$\bar{X}_m^{(n)} = \bar{E}_m^{(n)} \frac{\partial E_m^{(n)}}{\partial X_m^{(n)}} . \tag{75}$$

($\bar{\text{c}}$) Set

$$\bar{\beta}_m += \sum_{n=1}^{N_{\text{MC}}} \psi_m^{(n)} \bar{H}_m^{(n)}, \tag{76}$$

and initialise the adjoint of the basis functions ψ as in Equation (58)[3].

($\bar{\text{b}}$) Compute the adjoint of the two intermediate variables Ω_m and Ψ_m in Equation (27) using the results in Giles (2008) by

$$\bar{\Omega}_m = \Psi_m^{-T} \bar{\beta}_m , \qquad \bar{\Psi}_m = -\bar{\Omega}_m \beta_m^T . \tag{77}$$

($\bar{\text{a}}$) For $n = 1, \ldots, N_{\text{MC}}$ compute the adjoint of Equation (26) by

$$\begin{aligned} \bar{\psi} &+= \frac{1}{N_{\text{MC}}} \mathcal{N}_m^{(n)} V_{m+1}^{(n)} \bar{\Omega}_m , \\ \bar{V}_{m+1}^{(n)} &= \frac{1}{N_{\text{MC}}} \mathcal{N}_m^{(n)} (\psi_m^{(n)})^T \bar{\Omega}_m , \\ \bar{\mathcal{N}}_m^{(n)} &= \frac{1}{N_{\text{MC}}} V_{m+1}^{(n)} (\psi_m^{(n)})^T \bar{\Omega}_m , \end{aligned} \tag{78}$$

[3]Note that β_m will contain the sensitivities of the second simulation.

where $\mathcal{N}_m^{(n)} = N_m^{(n)} / N_{m+1}^{(n)}$, and compute the adjoint of Equation (25) by

$$\bar{\psi}_m^{(n)} += \frac{1}{2N_{\text{MC}}} \left(\bar{\Psi}_m^{(n)} + \left(\bar{\Psi}_m^{(n)} \right)^T \right) \psi_m^{(n)}. \tag{79}$$

Then update the adjoint of the risk factor vectors by

$$\bar{X}_m^{(n)} += \frac{\bar{\mathcal{N}}_m^{(n)}}{N_{m+1}^{(n)}} \frac{\partial N_m^{(n)}}{\partial X_m^{(n)}} + \bar{\psi}^T \frac{\partial \bar{\psi}}{\partial X_m^{(n)}}, \qquad \bar{X}_{m+1}^{(n)} += -\frac{\bar{\mathcal{N}}_m^{(n)} \mathcal{N}_m^{(n)}}{N_{m+1}^{(n)}} \frac{\partial N_{m+1}^{(n)}}{\partial X_{T_{m+1}}^{(n)}}. \tag{80}$$

($\overline{\text{R2}}$) Compute the adjoint of the risk factor vector at expiry by

$$\bar{X}_M^{(n)} = \bar{V}_M^{(n)} \frac{\partial E_M^{(n)}}{\partial X_M^{(n)}}, \tag{81}$$

for $n = 1, \ldots, N_{\text{MC}}$.

($\overline{\text{R1}}$) For $m = M, \ldots, 0$, compute the adjoint of the risk factor evolution as in step ($\overline{\text{L1}}$).

4 Numerical results

4.1 Best of two assets Bermudan-style option

As a first example, we consider a Bermudan-style option on the maximum of two assets under a standard lognormal model for the asset price processes $\{X_1(t)\}$ and $\{X_2(t)\}$. The payoff function at exercise time t is given by

$$(\max \{X_1(t), X_2(t)\} - K)^+, \tag{82}$$

where K is the strike price. A similar example is also studied in Glasserman (2004), Broadie and Glasserman (1997) and Broadie et al. (2004). Andersen and Broadie (2004) obtain nearly exact estimates for this example using twelve basis functions, including the value of a European-style option. We assume the option can be exercised every three months up to 3 years. We assume the underlying assets are independent geometric Brownian motion processes with the same initial values $X(0)$, volatilities σ, and dividend rates d. In particular, we choose $X_1(0) = X_2(0) = 1$, $\sigma_1 = \sigma_2 = 0.2$ and $d_1 = d_2 = 0.1$ and the risk-free rate $r = 0.05$. We compute the Bermudan-style option prices, the *Deltas* (i.e., the sensitivity with respect to the spot values, $X_i(0)$) and the *Vegas* (i.e., the sensitivity with respect to the volatilities σ_i) for $K = 0.9$, 1.0 and 1.1 with AAD and bumping. We also compare these with their "exact" value, which is obtained by a partial differential equation approach. The numerical results are found in Table 1 and in Figure 1. Throughout this section, the MC uncertainties are computed using the binning technique of Capriotti and Giles (2010).

K = 0.9	Price	*Delta*	*Vega*
AAD	0.2012(2)	0.415(3)	0.456(2)
PDE	0.20107	0.41423	0.45740
K = 1.0			
AAD	0.1396(1)	0.337(2)	0.486(2)
PDE	0.13959	0.33588	0.48440
K = 1.1			
AAD	0.0941(2)	0.258(1)	0.463(2)
PDE	0.09431	0.25635	0.46253

Table 1: *Prices, Deltas and Vegas for the Bermudan-style option in (82) with three different strike values. The smoothening parameter in the call spread regularization (48) is* $\delta = 0.005$*. The number of MC paths is 400,000. The MC uncertainty (in parenthesis) is computed using the binning technique of Capriotti and Giles (2010) with 20 bins for each set of simulations.*

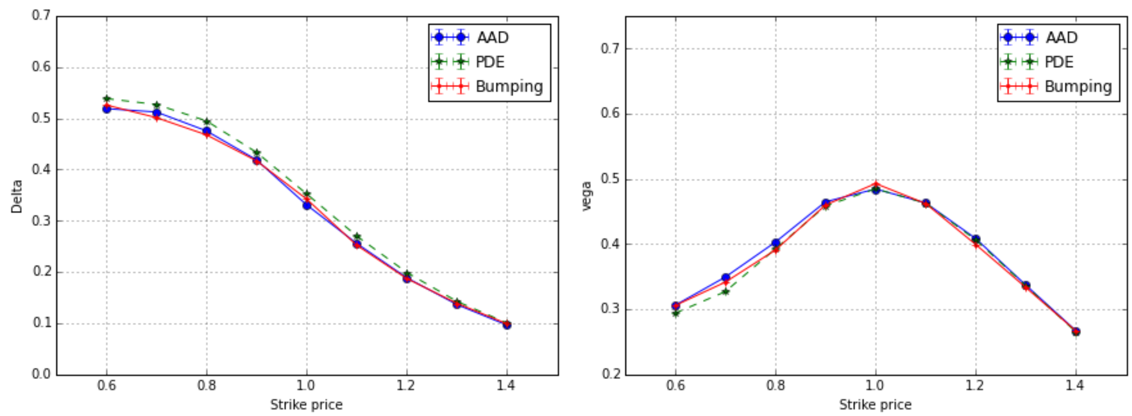

Figure 1: *Deltas (left panel) and Vegas (right panel) for the Bermudan-style option in (82) as a function of strike. The smoothening parameter in the call spread regularization (48) is* $\delta = 0.005$*. The number of MC paths is 400,000. The MC uncertainty (in parenthesis) is computed using the binning technique with 20 bins for each set of simulations.*

In the table and figure above, the smoothening parameter for the calculation of the Greeks, discussed in Section 3.1, $\delta = 0.005$, was chosen as a reasonable compromise between variance and bias of the estimator. This is illustrated in Figure 2, showing how for $\delta = 0.005$ the bias introduced by the finite δ is of the same order of magnitude of the statistical uncertainty for the chosen computational budget, and is negligible for any practical application.

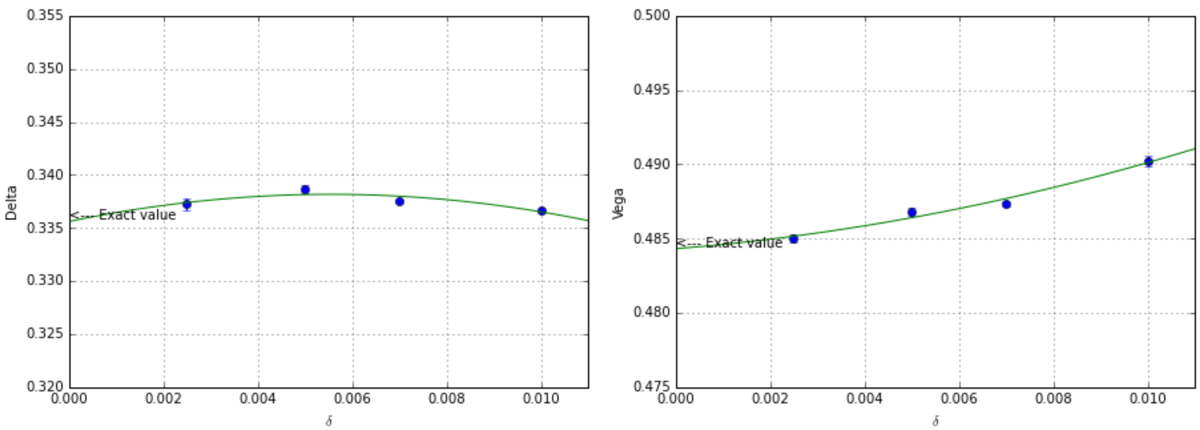

Figure 2: *AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option in (82) for* $K = 1$ *vs the smoothening parameter* δ *for the call spread regularization (48). The number of simulated paths is 3,000,000 for* $\delta = 0.01$ *and is increased as* δ *is decreased in order to keep statistical uncertainties roughly constant. The values in the graphs are fitted based on a quadratic polynomial function (green lines).*

As discussed in Section 3.1, neglecting to smoothen out the exercise boundary, although common in the financial practice, introduces a bias in the computation of sensitivities because the exercise boundary is in general not optimal. This is illustrated in Figure 3, in which we compare the *Delta*, with and without smoothening, for different choices of the basis functions. Here for *Delta*, the smoothened estimator turns out to be more accurate, especially when the exercise boundary obtained by regression is a less accurate approximation of the real one. However, it is difficult to establish a priori whether the unsmoothened estimator provides a smaller or a larger bias than the smoothened one. This is because the bias introduced by the lack of smoothening may be offset by the bias arising from the sub-optimality of the exercise boundary. This is illustrated in the right panel of Figure 3 showing that for Vega the smoothened and unsmoothened estimators have a comparable accuracy. In any case, a consideration to keep in mind is that smoothening the exercise boundary is generally required to obtain stable second-order risk values.

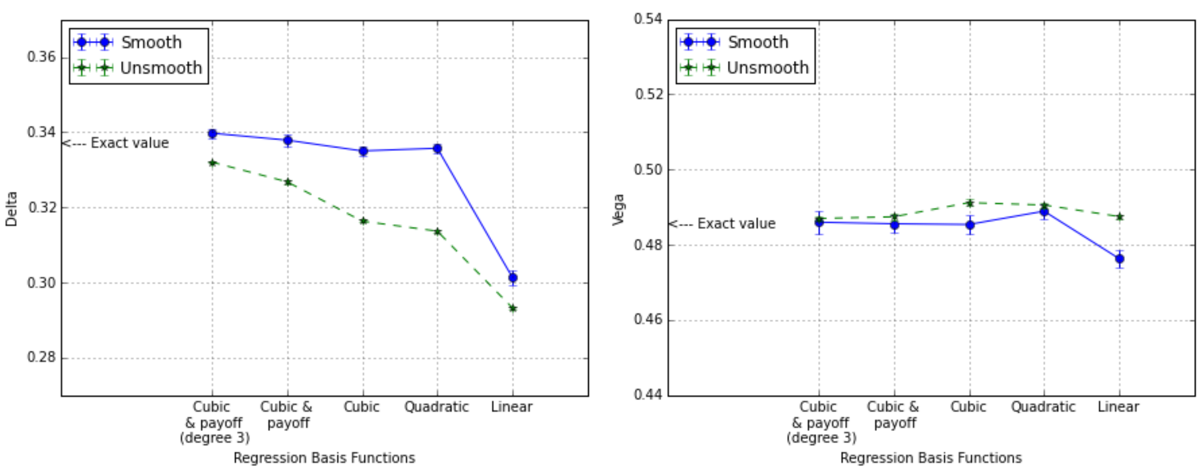

Figure 3: *AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option in (82) for* $K = 1$ *as obtained with the unsmoothed and the smoothened estimators with the call-spread approach (48) for five choices of the regression basis functions. The number of simulated paths is 1,000,000 with 20 bins and the smoothening component* $\delta = 0.005$.

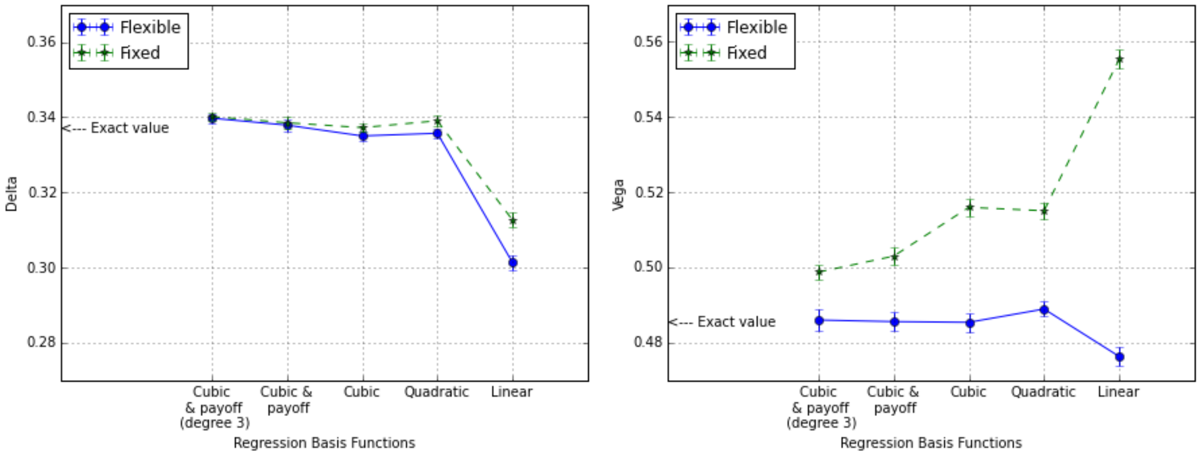

Figure 4: *AAD Delta (left panel) and Vega (right panel) of the Bermudan-style option in (82) for $K = 1$ as obtained by keeping the exercise boundary fixed (Fixed) and accounting instead for its contributions to the sensitivities (Flexible). The number of simulated paths is 1,000,000 with 20 bins and the call-spread smoothening component $\delta = 0.005$.*

The quasi-optimality of the exercise boundary is also generally invoked among industry practitioners as a justification for neglecting the contributions to the sensitivities arising from the exercise boundary. Clearly, the quality of this approximation is dependent on the accuracy of the regression functions in reproducing the actual exercise boundary. This is illustrated in Figure 4 where we plot *Delta* and *Vega* of the Bermudan-style option (82) for different choices of the regression basis functions. Here we compare the results obtained by the AAD algorithm, as described in Section 3.4, in the case where a) the exercise boundary is kept fixed, and b) when accounting instead for its contributions to the sensitivities. As expected, the difference between the two approaches vanishes as the regression functions become more accurate. However, as it is shown for the *Vega*, it can lead to a significant bias if a simple (e.g., linear or quadratic) representation of the exercise boundary is adopted.

4.2 XVA sensitivities

As another example, we compute the sensitivities of XVA (14) for the same option defined in Equation (82). Here, for simplicity, we assume that the hazard rate and the volatility are piecewise constant. The XVA sensitivities with respect to some of the model parameters, namely the term structure of hazard rates of the counterparty and of volatilities of the underlying, obtained with the AAD algorithm described in Section 2.5 are compared with the ones obtained by standard bumping in Tables 2 and 3. As expected, the AAD sensitivities are in excellent agreement with those obtained by bumping with any discrepancies attributable to the bias of the finite-difference approach completely masked in this example by the MC uncertainties.

	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5
AAD	0.00298(1)	0.00297(2)	0.00294(1)	0.00283(1)	0.00271(1)	0.00255(1)
Bumping	0.00298(1)	0.00297(2)	0.00294(1)	0.00283(1)	0.00271(1)	0.00255(1)
	σ_6	σ_7	σ_8	σ_9	σ_{10}	σ_{11}
AAD	0.00236(1)	0.00218(1)	0.00201(1)	0.00183(1)	0.00165(1)	0.001450(8)
Bumping	0.00236(1)	0.00218(1)	0.00201(1)	0.00183(1)	0.00165(1)	0.001450(8)

Table 2: *The XVA sensitivities with respect to the piecewise volatility. Both sets of the results are computed with 1,000,000 MC paths and 25 bins.*

	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
AAD	0.03337(1)	0.03336(2)	0.03329(2)	0.03306(3)	0.03269(3)	0.03218(3)
Bumping	0.03337(1)	0.03336(2)	0.03329(1)	0.03306(3)	0.03269(3)	0.03218(3)
	λ_6	λ_7	λ_8	λ_9	λ_{10}	λ_{11}
AAD	0.03151(3)	0.03074(4)	0.02989(4)	0.02893(4)	0.02784(4)	0.02669(4)
Bumping	0.03151(3)	0.03074(3)	0.02989(4)	0.02893(4)	0.02784(4)	0.02669(4)

Table 3: *The XVA sensitivities with respect to the piecewise hazard rates. Both sets of the results are computed with 1,000,000 MC paths and 25 bins.*

Similar to the case of Bermudan-style option Greeks, keeping the regression boundary fixed while computing the sensitivities, may introduce a bias. However, for XVA, this issue is much more severe than in the case of the Bermudan-style option Greeks because no quasi-optimality argument can be invoked. As shown in Figure 5, the XVA sensitivities against the volatilities obtained by bumping and AAD with flexible boundaries are almost identical. Instead, the results of AAD with fixed boundaries are remarkably different and, if used for risk management, this would lead to significant mis-hedging. On the other hand, as expected, the XVA sensitivities with respect to the hazard rates are not affected by the contribution arising from the regression functions. This is because the hazard rates do not enter the computation of the portfolio value conditional on default, and hence do not appear in the regression boundaries.

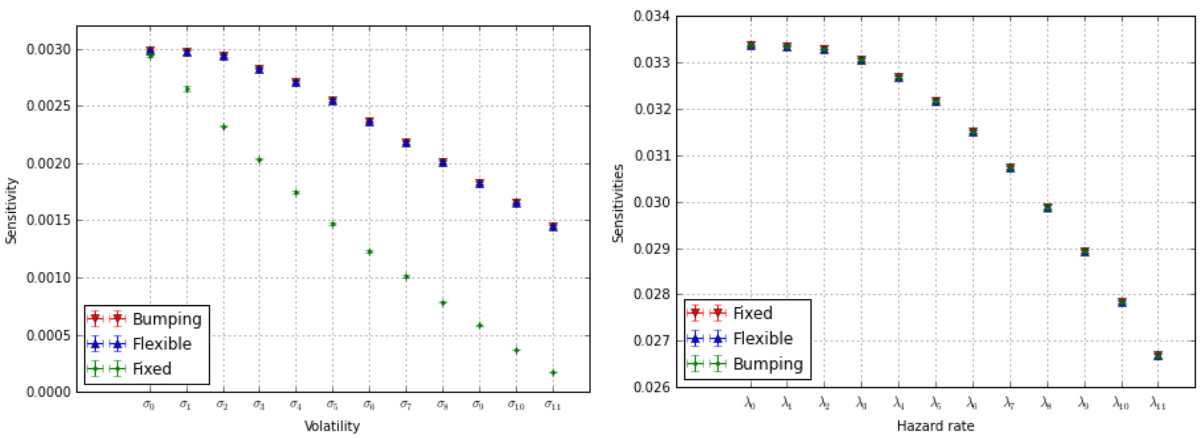

Figure 5: *CVA sensitivities with respect to the piecewise volatility (left panel) and hazard rate (right panel), computed by AAD with flexible boundaries, fixed boundaries and bumping. The number of MC paths is 1,000,000 and the number of bins is 25.*

Finally, the remarkable computational efficiency of the AAD approach is illustrated in Figure 6. Here we plot the cost of computing the XVA sensitivities with respect to the term structure of the counterparty hazard rate and the underlying volatility, relative to the cost of performing a single valuation. The calculation of the sensitivities by means of AAD can be performed in about three times the cost of computing the XVA value, that is, well within the theoretical bound (7). In contrast, the cost of bumping, for one-sided finite-difference estimators, is in general $(1 + N)$ times the cost of as single valuation, where N is the number of model parameters, i.e., in this case over 20 times the cost of computing the value of the XVA.

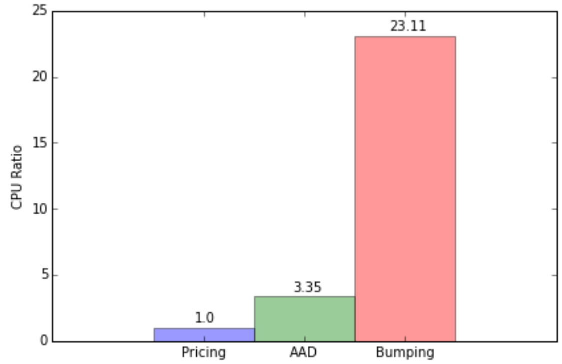

Figure 6: *Ratio of the CPU time required for the calculation of the CVA, and its sensitivities, and the CPU time spent for the computation of the CVA alone.*

5 Conclusions

We have shown how Adjoint Algorithmic Differentiation (AAD) can be used to implement efficiently the computation of sensitivities in regression-based MC methods. By discussing in detail examples of practical relevance, we have demonstrated how accounting for the sensitivities contributions associated with the regression functions is crucial to obtain accurate estimates of the Greeks in XVA applications and for Bermudan-style options, especially when the exercise boundary is not particularly accurate. We also show how smoothening out the discontinuities associated with suboptimal exercise boundaries can lead in some situations to more accurate estimates of the sensitivities.

From a computational stand point, similarly to what previously found in other MC and PDE settings, c.f. Capriotti and Giles (2012), Capriotti et al. (2015), the proposed method allows the computation of the complete first-order risk at a cost which is at most four times the cost of calculating the value of the portfolio itself. This typically results in orders of magnitude of savings in computational time with respect to standard finite-difference approaches, thus making AAD an ever so indispensable technique in the toolbox of modern financial and insurance engineering.

Acknowledgments

It is a pleasure to thank Jacky Lee, for numerous useful discussions, and for a careful reading of the manuscript. The opinions and views expressed in this paper are uniquely those of the authors, and do not necessarily represent those of Credit Suisse Group.

References

Andersen, L. and Broadie, M. (2004). Primal-dual simulation algorithm for pricing multidimensional Americans options. *Management Science*, 50(9):1222–1234.

Andersen, L. and Piterbarg, V. (2010). *Interest Rate Modeling, Volume I: Foundations and Vanilla Models*. Atlantic Financial Press.

Broadie, M. and Glasserman, P. (1997). Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8):1323–1352.

Broadie, M. N., Glasserman, P., et al. (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7:35–72.

Capriotti, L. (2011). Fast greeks by algorithmic differentiation. *Journal of Computational Finance*, 3:3–35.

Capriotti, L. and Giles, M. (2010). Fast correlation risk by adjoint algorithmic differentiation. *Risk*, 23:79–85.

Capriotti, L. and Giles, M. (2012). Algorithmic differentiation: adjoint greeks made easy. *Risk*, 25:92–98.

Capriotti, L., Jiang, Y., and Macrina, A. (2015). Real-time risk management: An AAD-PDE approach. *International Journal of Financial Engineering*, 02(04).

Capriotti, L. and Lee, S. J. (2014). Adjoint credit risk management. *Risk*, 27:92–98.

Capriotti, L., Peacock, M., and Lee, J. (2011). Real time counterparty credit risk management in Monte Carlo. *Risk*, 24:86–90.

Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19(1):19–30.

Cesari, G., Aquilina, J., Charpillon, N., Filipovic, Z., Lee, G., and Manda, I. (2009). *Modelling, pricing, and hedging counterparty credit exposure: a technical guide*. Springer Science & Business Media.

Crépey, S., Bielecki, T. R., and Brigo, D. (2014). *Counterparty Risk and Funding–A Tale of Two Puzzles*. Chapman and Hall/CRC Financial Mathematics Series.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Giles, M. B. (2008). Collected matrix derivative results for forward and reverse mode algorithmic differentiation. In *Advances in Automatic Differentiation*, pages 35–44.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer, New York.

Griewank, A. (2000). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Frontiers in Applied Mathematics, Philadelphia.

Henrard, M. (2011). Adjoint algorithmic differentiation: Calibration and implicit function theorem. *OpenGamma Quantitative Research*, (1).

Joshi, M. and Kwon, O. K. (2016). Least squares monte carlo credit value adjustment with small and unidirectional bias. *preprint available on SSRN*.

Lando, D. (1998). On Cox processes and credit risky securities. *Review of Derivatives research*, 2(2-3):99–120.

Leclerc, M., Liang, Q., and Schneider, I. (2009). Fast Monte Carlo Bermudan greeks. *Risk*, 22:84–88.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14(1):113–147.

Tsitsiklis, J. N. and Van Roy, B. (2001). Regression methods for pricing complex American-style options. *Neural Networks, IEEE Transactions on*, 12(4):694–703.

Wilmott, P., Howison, S., and Dewynne, J. (1995). *The Mathematics of Financial Derivatives*. Cambridge University Press, Cambridge.

A American-style options sensitivities with optimal regression boundaries

This result can be obtained by writing formally the derivative of the payoff estimator (34),

$$\partial_\theta P(X) = \sum_{m=1}^{M} \partial_\theta \left[\mathbf{1}(t_1, t_m) \times \mathbf{1}(H_m < E_m) \right] \frac{E_m}{N_m} + \sum_{m=1}^{M} \mathbf{1}(t_1, t_m) \times \mathbf{1}(H_m < E_m)\, \partial_\theta \frac{E_m}{N_m} \,, \tag{83}$$

so that the pathwise derivative estimate of the sensitivities is expressed by

$$\bar{\theta}(t) = \mathbb{E}_0 \left[\sum_{m=1}^{M} \mathbf{1}(t_1, t_m) \times \mathbf{1}(H_m < E_m)\, \partial_\theta \frac{E_m}{N_m} \right] + \Delta\bar{\theta}$$

where $\Delta\bar{\theta}$ is the correction associated with the sensitivities of the exercise boundary. This can in turn be rewritten as

$$\begin{aligned} \Delta\bar{\theta} \;&=\; \mathbb{E}_t \left[\partial_\theta \left[\mathbf{1}(H_1 < E_1) \right] \frac{E_1}{B_1} + \partial_\theta \left[\mathbf{1}(H_1 > E_1) \right] \times \sum_{m=2}^{M} \mathbf{1}(t_2, t_m) \times \mathbf{1}(H_m < E_m) \frac{E_m}{N_m} \right] \\ &+\; \mathbb{E}_t \left[\mathbf{1}(H_1 > E_1) \sum_{m=2}^{M} \partial_\theta \left[\mathbf{1}(t_2, t_m) \times \mathbf{1}(H_m < E_m) \right] \frac{E_m}{N_m} \right]. \end{aligned} \tag{84}$$

By conditioning on t_1, and defining the approximate hold value at time t_i as[4]

$$\frac{\tilde{H}_i}{B_i} = \mathbb{E}_i \left[\sum_{m=i}^{M} \mathbf{1}(t_i, t_m) \times \mathbf{1}(H_m < E_m) \frac{E_m}{N_m} \right] \tag{85}$$

one obtains

$$\begin{aligned} &\Delta\bar{\theta} = \\ &\mathbb{E}_t \left[\delta_1 (\partial_\theta E_1 - \partial_\theta H_1) \left(\frac{H_1}{N_1} - \frac{\tilde{H}_1}{N_1} \right) + \mathbf{1}(H_1 > E_1) \mathbb{E}_1 \left[\sum_{m=2}^{M} \partial_\theta \left[\mathbf{1}(t_2, t_m) \times \mathbf{1}(H_m < E_m) \right] \frac{E_m}{N_m} \right] \right] \,, \end{aligned} \tag{86}$$

where δ_i is Dirac's delta of the form $\delta(E_i - H_i)$. Finally, by iterating the procedure, one gets

$$\Delta\bar{\theta} = \mathbb{E}_t \left[\sum_{m=1}^{M} \mathbf{1}(t_1, t_m) \delta_m (\partial_\theta E_m - \partial_\theta H_m) \left(\frac{H_m}{N_m} - \frac{\tilde{H}_m}{N_m} \right) \right] \,, \tag{87}$$

where we have used $H_M \equiv 0$. As a result, when the exercise boundary is optimal, i.e. $H_m = \tilde{H}_m$, the correction due to the discontinuity of the exercise boundary $\Delta\bar{\theta}$ is zero.

[4]Note that this is in general different from H_i because the latter is determined by the regression in the first simulation.