Risk publications

Risk

RISK MANAGEMENT • DERIVATIVES • REGULATION

Risk.net September 2018

Cutting edge
Valuation adjustments

Efficient Simm-MVA calculations for callable exotics

Efficient Simm-MVA calculations for callable exotics

Margin valuation adjustment for callable trades subject to the standard initial margin model requires sensitivities of future trade values to quotes. Fortunately, sensitivities of future trade values to model parameters can be combined with future parameter-to-quote Jacobians to achieve this, and these sensitivities can be computed efficiently via differentiation through least-squares Monte Carlo. Here, Alexandre Antonov, Serguei Issakov and Andrew McClelland facilitate this through algorithmic differentiation, where the propagation rule is similar to that employed for pathwise exposure sensitivities, as required for credit valuation adjustment sensitivities

We begin this article by introducing the standard initial margin model (Simm) and margin valuation adjustment (MVA), demonstrating the need for future sensitivities and sketching our proposed technique for producing them efficiently. The Simm determines how much initial margin (IM) must be posted for non-cleared, bilaterally margined trades. This article focuses on callable trades such as Bermudans, which are non-cleared but have high volumes, and as such will contribute significantly to IM requirements.[1] The Simm inputs are (first-order) trade sensitivities to quantities such as market quotes, eg, swap rates or implied volatilities. These are aggregated to approximate a 99% value-at-risk over a 10-day margin period of risk. The main calculations involved are outlined here, leaving specific details to the International Swaps and Derivatives Association (2017).

Let $\partial_{Q_m} V$ denote the sensitivity of a trade's value with respect to the mth quote, Q_m, and let there be N_Q quotes. In practice, Simm-IM requires sensitivities for each trade with a counterparty; however, one can work with a single trade without loss of generality, as sensitivities are summed across trades at the first step of aggregation. The Simm-IM for a standalone Bermudan is:

$$\text{IM} = \text{IM}_{\text{Delta}} + \text{IM}_{\text{Vega}} + \text{IM}_{\text{Curvature}}$$

where IM_{Delta} requires deltas over swap rates, and where IM_{Vega} and $\text{IM}_{\text{Curvature}}$ require vegas over implied volatilities; curvatures are approximated via the gamma-vega relationship for vanillas. The IMs have similar structures, so it suffices to focus on one – say, IM_{Delta} – that simplifies to:

$$\text{IM}_{\text{Delta}} = \sqrt{\sum_{k,l} (\partial_{S_k} V \partial_{S_l} V)(\rho_{kl} \text{RW}_k \text{RW}_l)} \tag{1}$$

Here, $\partial_{S_k} V$ is the delta over the kth swap rate, RW_k are risk weights and ρ_{kl} are correlations; twelve swap rates are considered (2W to 30Y tenors). The formula has been simplified by (a) ignoring concentration factors and (b) using single-curve pricing, purely to ease exposition.[2] Similar sensitivities are required for IM_{Vega} and $\text{IM}_{\text{Curvature}}$, although these are over the at-the-money (ATM) implied volatility surface. Specifically, these require $\partial_\nu V$, the sensitivity over a flat shift of the surface holding skew constant. Computing IM thus distills to computing sensitivities over current quotes, and it is useful to write $\text{IM} = \text{IM}(\partial_Q V)$, where $\partial_Q V$ is the gradient against the full quote vector, Q. It is also useful to use $\text{IM}(t) = \text{IM}(\partial_{Q(t)} V(t))$ to emphasise that Simm-IM computed on future dates will require sensitivities to quotes prevailing on future dates.

MVA is the expected lifetime cost of funding IM postings (see Green & Kenyon (2015) for motivation). In short, IM is funded by the dealer, whereas VM is funded by hedging gains and trade cashflows. It is computed as:

$$\begin{aligned}\text{MVA} &= \mathbb{E}_{t_0}\biggl[\int_{t_0}^{T} \mathrm{e}^{-R(t)} \text{IM}(t)(r_F(t) - r(t))\,\mathrm{d}t\biggr] \\ &\approx \frac{1}{N_P} \sum_{p=1}^{N_P} \sum_{i=1}^{N_T} \mathrm{e}^{-R_p(t_i)} \text{IM}_p(t_i)(r_{Fp}(t_i) - r_p(t_i))\Delta t_i \end{aligned} \tag{2}$$

where $r(t)$ is the overnight rate, $r_F(t)$ is the dealer's funding rate and $R(t)$ is an unspecified integrated discounting rate, which can incorporate things such as hazard rates. For bilaterally margined trades, $\text{IM}(t) = \text{IM}(\partial_{Q(t)} V(t))$ will generally be highly non-linear and MVA will not be available in closed form. The Monte Carlo approximation (2) uses N_P paths discretised at N_T 'observation dates', t_i. On these dates, future Simm-IM figures are required for all paths, and thus so are the constituent future sensitivities:

$$\partial_{Q_{p,i}} V_{p,i} \equiv \partial_{Q_p(t_i)} V_p(t_i)$$

Callables do not allow closed-form values or sensitivities, and using nested calls to the pricing function for future sensitivities is burdensome. Indeed, such trades require numerically expensive methods such as partial differential equations, lattices or regression methods, eg, least-squares Monte Carlo (LSMC). Computing the full set of derivatives via one-sided finite differences requires $N_Q \times N_P \times N_T$ pricing function calls, though the N_Q factor can be removed by applying adjoint algorithmic differentiation (AAD) to the pricing function (see, for example, Giles & Glasserman 2006). Critically, if the pricing function uses Monte Carlo with N_P paths, producing future sensitivities becomes $\mathcal{O}(N_P^2)$.

To understand the proposal of this article, consider generating raw exposures or future values. This requires all future values $V_{p,i}$, which costs $N_P \times N_T$ pricing function calls if done by 'brute force'. Again, if pricing is an $\mathcal{O}(N_P)$ operation, computing future values is $\mathcal{O}(N_P^2)$, just as

[1] *Here, 'callable' refers to any trade allowing one counterparty to call, cancel or exercise at times prior to maturity.*

[2] *Concentration factors depend only on sensitivities, and multi-curve pricing simply introduces additional parameters over which to differentiate.*

for future sensitivities. However, future values can be produced by LSMC using a single call to a regression-based pricing function, which reduces he expense to $\mathcal{O}(N_P)$. As will be seen, future sensitivities over model parameters, θ, are available via a special flavor of algorithmic differentiation (AD) propagation applied to a single LSMC run, again at a cost of $\mathcal{O}(N_P)$. The sensitivities over parameters are transformed into sensitivities over quotes via Jacobians.

One may refer to Antonov *et al* (2016a) for a general treatment of AD-on-LSMC for credit valuation adjustment (CVA) sensitivities, and it is useful to note that this method requires differentiation through regression coefficients with respect to parameters, and not simply the differentiation of a fitted regression function with respect to states or regression variables. To preview this, assume the time-t_i regression uses N_B basis functions of state processes $X_{p,i}$, $\phi(X_{p,i}) \equiv [\phi_n(X_{p,i})]$, and has fitted coefficients $\alpha \equiv [\alpha_n]$. The coefficients clearly embed θ dependence:

$$V_{p,i} = V(t_i, X_{p,i}, \theta) = \phi(X_{p,i})\alpha(\theta)$$
$$\implies \partial_\theta V_{p,i} = \partial_\theta V(t_i, X_{p,i}, \theta) = \phi(X_{p,i})\partial_\theta \alpha(\theta) \quad (3)$$

requiring full differentiation through LSMC.[3] The propagation used here produces sensitivities at t_i with respect to parameters governing model behaviour after t_i, ignoring dependence on parameters prior to t_i; this mimics the calculation of parameter sensitivities at t_0 in practice. The parameters θ referenced here include yield and volatility curve knots, which are commonly differentiated over in standard AD, and thus the parameters governing behaviour after t_i are taken here to be the knots located after t_i. To make this clear, let $\vartheta_i \subseteq \theta$ be the set of parameter knots located after t_i. For example, if $t_i = t_0 + 5\text{Y}$, ϑ_i consists of elements of θ associated with knots located > 5Y from the present, and if $t_i = t_0$, $\vartheta_0 = \theta$. Standard AD for CVA sensitivities produces full pathwise sensitivities $\partial_\theta V_i = \partial_{\vartheta_0} V_i$, while the AD propagation used here produces $\partial_{\vartheta_i} V_i$. Once available, parameter sensitivities are transformed into future quote sensitivities via future Jacobians linking parameters to quotes prevailing on trajectories, $\partial_{\vartheta_i} Q_i$, which involve vanillas and can be produced rapidly.

To our knowledge, this is the first method available to produce future Simm sensitivities for callables, eg, where LSMC is necessary. Others use regression functions to expedite or approximate future VAR calculations and can be very useful for projecting IM for cleared trades (see, for example, Andersen *et al* 2017; Green & Kenyon 2015). This is done, however, without the use of sensitivities, which are central to accurate Simm-IM projections. For example, Green & Kenyon (2015) simulate risk factors $L_{p,i}$ using a high-dimensional model and compute values on these trajectories $V_{p,i}$ using a low-dimensional pricing model, independently for each path and time step. A regression linking the two, $V_{p,i} \approx \phi(L_{p,i})\beta$, is then fitted, which allows for rapid revaluation of the trade (or portfolio) under perturbed risk factors, as required for clearing-house VAR-based IM. One possibility for adapting this approach to the current setting is to use Simm quotes as risk factors, ie, $L_i = Q_i$, and to differentiate the fitted regression function. Unfortunately, building the $V_{p,i}$ by brute force for callables is computationally expensive. Moreover, fitting regressions to N_Q-dimensional correlated inputs with accuracy is difficult enough, and requiring their derivatives to also be accurate makes matters even worse. This stands in contrast to the method proposed here, which differentiates via typical LSMC regressions with respect to embedded parameters. In Andersen *et al* (2017), standard errors of fitted local regressions for $V_{p,i}$ are multiplied by tail probabilities to approximate the VAR. Extending this approach to computing sensitivities over state variables is possible but intractable, as 12 delta sensitivities will require a 12-factor model. One thus encounters the dimensionality issues cited when trying to extend the work of Green & Kenyon (2015).

Future quote sensitivities via future parameter sensitivities

This section reviews how current quote sensitivities can be computed from current parameter sensitivities and Jacobians, before discussing the analogous calculation on future scenarios as required by brute force. This helps to build intuition regarding how future parameter sensitivities will be used and sets up for the case where they are produced efficiently via AD-on-LSMC. To begin, consider a risk-neutral diffusion model of N_F interest rate factors $X(t)$:

$$\mathrm{d}X(t) = \mu_X(t, X(t), \theta(t))\,\mathrm{d}t + \sigma_X(t, X(t), \theta(t))\,\mathrm{d}W(t) \quad (4)$$

The parameters active at t, $\theta(t)$, are time dependent and interpolated from the parameter vector θ; this is comprised of a yield curve and volatility parameters fitted to current calibration quotes such as swap rates and implied volatilities, denoted here by C_0. The current value is $V_0 = V(t_0, X_0, \theta)$, with all parameters θ $(= \vartheta_0)$ being relevant.[4] It is straightforward to compute current sensitivities to parameters, $\partial_\theta V_0$, possibly using AD. These can be transformed into quote sensitivities via a Jacobian, $J_0 = \partial_\theta Q_0$, as is standard practice.

Consider first the sensitivities against the 12 swap rates required for Simm, and set $N_Q = 12$ momentarily. In an idealised setting, where θ consists only of 12 yield curve knots at the Simm tenors, one could invert J_0 and compute the requisite 12 quote sensitivities via:

$$\partial_{Q_0} V_0 = \partial_\theta V_0 J_0^{-1} \quad (5)$$

If the model is calibrated using $N_\theta > N_Q = 12$ yield curve knots (eg, given the natural choice of bootstrapping instruments), J_0 is not square and inversion is not possible. Discussing this is crucial, as non-square Jacobians will be standard when computing future quote sensitivities that $N_{\vartheta_i} > N_Q$, because one must ensure there are sufficient knots to compute N_Q sensitivities for all t_i. This will not happen if the model is built with sparse yield curve knots at offset tenors of, say, $[\ldots, 20\text{Y}, 25\text{Y}]$, which is representative of typical curve construction. When t_i approaches the 20Y offset, the only knots left to differentiate over are those at offsets of 20Y and 25Y. The approach adopted here is to build curves with a sufficient number of knots in θ so as to ensure the knots in ϑ_i allow for N_Q bucketed sensitivities to be computed for each t_i. An example of this is given later in the paper.

Mathematically, bucketing is equivalent to introducing a vector of 12 auxiliary parameters, ϵ, where each introduces an additive shift to all knots

[3] *AD-on-LSMC easily extends to θ-dependent basis functions.*

[4] *The fact that parameters following trade maturity may not be relevant is ignored with no loss of generality.*

located in the corresponding bucketed interval; sensitivities against ϵ produce a vector of bucketed sensitivities. Let $\theta^{\epsilon} = \theta + \epsilon$ represent knot values after the (piecewise-constant) shifts have been applied, eg:

$$\theta^{\epsilon} = [\theta_1 + \epsilon_1, \theta_2 + \epsilon_2, \theta_3 + \epsilon_2, \dots]$$

where the knots θ_2 and θ_3 reside within the second bucket. Given this, the gradient $\partial_{\epsilon} V_0 = \partial_{\theta} V_0 \partial_{\epsilon} \theta^{\epsilon}$ has 12 elements, and the Jacobian $J_0^{\epsilon} = \partial_{\epsilon} Q_0 = \partial_{\theta} Q_0 \partial_{\epsilon} \theta^{\epsilon}$ is a 12-×-12 square matrix. The transformation in (5) can then be performed with $\partial_{\epsilon} V_0$ and J_0^{ϵ} in place of the raw non-bucketed quantities. The same rationale can be applied to current vega sensitivities, which completes the analysis for current Simm-IM calculations.

Consider now computing future quote sensitivities at t_i on path p. The best approximation of reality is that a new set of calibration instruments would be produced, $C_{p,i} = C(t_i; X_{p,i}, \theta)$, from which a new model would be calibrated with parameters $\theta_{C_{p,i}}$. Critically, $\theta_{C_{p,i}} \neq \vartheta_i$ in general, most obviously because the knot locations will shift, as they are typically chosen at fixed offsets (tenors) to t_i. After recalibration, parameter sensitivities $\partial_{\theta_{C_{p,i}}} V_{p,i}$ would be computed and transformed into quote sensitivities $\partial_{Q_{p,i}} V_{p,i}$ with use of $\partial_{\theta_{C_{p,i}}} Q_{p,i}$. Clearly, recalibrating the model is expensive, and so is the calculation of parameter sensitivities. Performing this for all $N_T \times N_P$ simulation nodes lies at the heart of the MVA problem.

The key insights of this article are twofold. The first is, after the appropriate use of bucketing, the parameter sensitivities $\partial_{\theta_{C_{p,i}}} V_{p,i}$ are well approximated by parameter sensitivities produced off the original model, $\partial_{\vartheta_i} V_{p,i}$. The second is these sensitivities are well approximated using the output of AD-on-LSMC, where propagation carries a cost of only $\mathcal{O}(N_P)$. This propagation is a modified version of that used for CVA sensitivities. Using a simple definition of CVA, eg, ignoring default probabilities, discounting and collateral, and using a single observation date, it is clear how the two propagation rules are related:[5]

$$\begin{aligned} \text{CVA} &= \mathbb{E}_{t_0}[(V_i)^+] \\ \implies \partial_{\theta}\text{CVA} &= \partial_{\theta}\mathbb{E}_{t_0}[(V_i)^+] = \mathbb{E}_{t_0}[1_{(V_i>0)} \partial_{\theta} V(t_i, X(t_i, \theta), \theta)] \end{aligned}$$

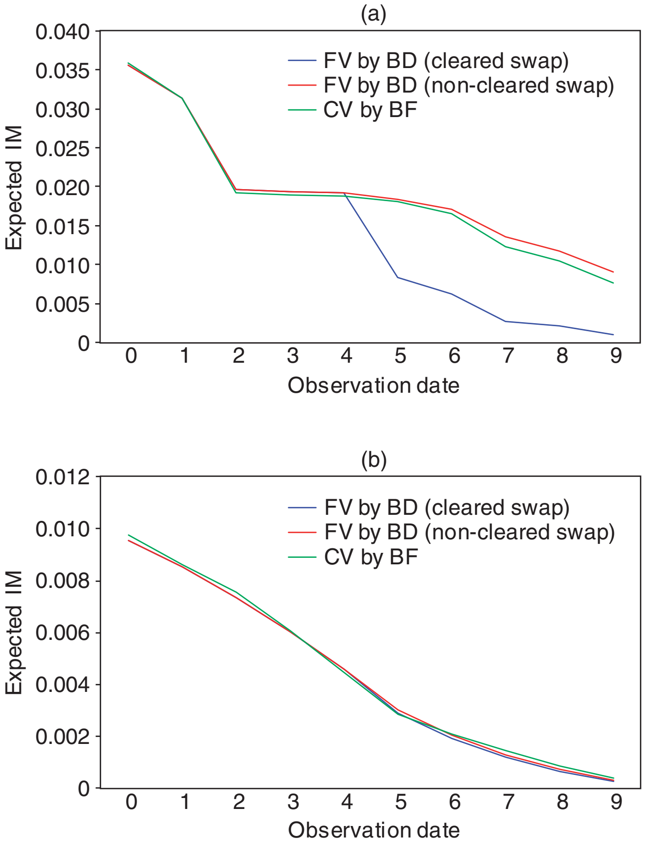
The sensitivities $\partial_{\theta} V(t_i, X(t_i, \theta), \theta)$ include the impact of θ upon the trajectory $X(t_i, \theta)$. MVA requires future sensitivities along a given trajectory, and thus one must omit this dependence during propagation. In practice, this means prematurely terminating propagation of sensitivities through $\partial_{X_i} V_i$, which basically means ignoring parameters influencing behaviour prior to t_i. Aside from producing $\partial_{\vartheta_i} V_{p,i}$ at a cost of $\mathcal{O}(N_P)$, using AD represents enormous synergies for institutions that have already implemented an AD framework.

With $\partial_{\vartheta_i} V_{pi}$ available, future Jacobians $J_{p,i} = \partial_{\vartheta_i} Q_{p,i}$ are produced analytically, and both are bucketed and combined to produce future quote sensitivities:

$$\begin{aligned} \partial_{Q_{p,i}} V_{p,i} &= \partial_{\varepsilon_i} V_{p,i} (J_{p,i}^{\varepsilon_i})^{-1} \\ &= (\partial_{\vartheta_i} V_{p,i} \partial_{\varepsilon_i} \vartheta_i^{\varepsilon_i})(\partial_{\vartheta_i} Q_{p,i} \partial_{\varepsilon_i} \vartheta_i^{\varepsilon_i})^{-1} \end{aligned}$$

In the above, the ε_i are analogous to the ϵ at t_0 and correspond to buckets chosen at t_i to align with the offset tenors of the quotes. These buckets essentially slide across knot locations as t_i progresses.

1 Expected IM profiles



(a) Expected delta Simm-IM. (b) Expected vega Simm-IM

Naturally, the $V_{p,i}$ from LSMC are approximations, and thus so are the $\partial_{\vartheta_i} V_{p,i}$ from AD-on-LSMC. Their accuracy is discussed in the next section after presenting differentiation through an LSMC algorithm, and in particular through regression coefficients. A comparison with brute force results is also offered in the following section (see figure 1), supporting the claim that the approximations are sufficiently accurate for practical use.

Future parameter sensitivities for callables via AD-on-LSMC

This section outlines enough of the AD-on-LSMC propagation for $\partial_{\vartheta_i} V_{p,i}$ to make it accessible. The next subsection discusses how LSMC produces future values for callables; this is followed by a demonstration of how parameter sensitivities of intermediate quantities are propagated. The final subsection discusses the accuracy of these sensitivities.

■ **LSMC for exposures.** Pricing callables relies on ‘backward’ (backward-in-time) inductive procedures. The main quantity of interest is the continuation value (CV), $V(t, X(t), \theta)$. It is a hold value, representing the value of subsequent payments given an exercise (or call) strategy, assuming exercise has not yet occurred.[6]

[5] *For now, the possibility that the trade has exercised into an underlying, been called, etc, is ignored for simplicity.*

[6] *Formally, the exercise strategy for a callable should reflect the funding of future IM requirements, ie, future MVA (see Green & Kenyon 2017). This is ignored here given the existing complexity of the problem at hand.*

A Bermudan gives the buyer the right to exercise into an underlying swap on exercise dates $\{T_j\}_{j=1}^{M}$. For simplicity, assume payment and exercise dates coincide, and suppose the underlying swap pays cashflows $c_j \equiv c(T_j)$ at each T_j. It is assumed cashflows are known functions of X_j and θ, as is the case for fix-in-arrears swaps, but this is purely to ease exposition; the algorithm can readily handle complicated path-dependent cashflows. Let $V(t)$ and $S(t)$ denote the CV of the Bermudan and the value of the underlying swap, respectively. Pricing a Bermudan, or any callable, via LSMC involves two repeated steps on instrument dates $\{T_j\}_{j=1}^{M}$ (see Antonov *et al* (2015) for more detail).

- **Transformations.** For example, take the maximum of two values, or append cashflows to value:

$$V_j^- = \max(V_j, S_j) \quad \text{and} \quad S_j^- = S_j + c_j \tag{6}$$

where $V_j^- = V(T_j^-)$ and $S_j^- = S(T_j^-)$ denote valuations just prior to exercise time T_j.

- **Projection.** Calculate earlier values as discounted conditional expectations over subsequent values:

$$S_j = N_j \mathbb{E}_{T_j}\left[\frac{S_{j+1}^-}{N_{j+1}}\right] \quad \text{and} \quad V_j = N_j \mathbb{E}_{T_j}\left[\frac{V_{j+1}^-}{N_{j+1}}\right] \tag{7}$$

where N_j is the model numeraire. LSMC computes these conditional expectations by regression when closed-form representations are not available, eg, for V_j. To be clear, projection produces the entire vector $V_j = [V_{p,j}]$ from a single regression, and it is assumed for simplicity that the output of the single-pass, regression-based procedure in (6) and (7) yields an unbiased CV estimator, though the AD propagations to be discussed readily extended to two-pass approaches. Clearly, these regressions can also be used to project to observation dates t_i as well as exercise dates T_j, thus making the $V_{p,i}$ available.

There is one additional complication to be addressed before we discuss AD-on-LSMC. For CVA, the notion of a future value (FV) is necessary. The difference between CVs and FVs is CVs do not account for exercise behaviour prior to t_i, while FVs do. FVs are denoted using lowercase, v_i, while uppercase is reserved for CVs, V_i. A Bermudan FV is equal to the Bermudan CV if exercise has not occurred before t_i, and it is equal to the underlying swap value if it has:

$$v_{p,i} = V_{p,i}(1 - \mathbb{1}_{p,i}) + S_{p,i}\mathbb{1}_{p,i} \tag{8}$$

where $\mathbb{1}_{p,i}$ are pathwise exercise indicators.

This is complicated further by the fact that, upon being exercised, the underlying swap is often cleared, at which point $v_{p,i} = S_{p,i}$ no longer contributes to counterparty exposures and $\partial_{Q_{p,i}} V_{p,i}$ no longer contributes to Simm-IM. For this reason, we introduce:

$$\hat{v}_{p,i} = V_{p,i}(1 - \mathbb{1}_{p,i})$$
$$\partial_{Q_{p,i}} \hat{v}_{p,i} = \partial_{Q_{p,i}} V_{p,i}(1 - \mathbb{1}_{p,i})$$

for exposure and sensitivity contributions in the cleared-underlying case. Crucially, for MVA the $\mathbb{1}_i$ are applied to sensitivities and thus do not impact propagation.

- **AD-on-LSMC propagation.** AD is more efficient than finite differences for CVA sensitivities (see Giles & Glasserman 2006). It applies the chain rule, where derivatives of intermediate operations are propagated to produce sensitivities of outputs to inputs. For CVA, a gradient is produced, ∂_θCVA. For MVA, however, one requires all $\partial_{\vartheta_i} V_{p,i}$ as an input; MVA itself is not a sensitivity. As will be seen, the sensitivities $\partial_{\vartheta_i} V_{p,i}$ require a propagation similar to $\partial_\theta V_{p,i}$, essentially the θ sensitivities of the exposures matrix. Importantly for AD, the number of outputs, $N_T \times N_P$, is much larger than the number of inputs, N_θ, and thus tangent AD (TAD) is more efficient than AAD. One benefit of TAD is sensitivities can be propagated in real time as LSMC runs, meaning intermediate sensitivities need not be written to memory (ie, using 'tape'). An algorithm applying TAD to LSMC for $\partial_\theta V_{p,i}$ is presented in Antonov *et al* (2016a). It is named backward differentiation (BD), as LSMC projections are recursively applied backward through calendar dates; as TAD propagates alongside LSMC, sensitivities also propagate backward. A key strength of BD is it differentiates by hand the matrix operations involved in projection by regression (7), allowing us to use matrix calculus identities and caching to gain efficiencies. A mild modification of BD is required for $\partial_{\vartheta_i} V_{p,i}$, as will be seen momentarily.

The hardest aspect of AD-on-LSMC is propagating through projection. Consider the values $V_{i+1} = [V_{p,i+1}]$, where each $V_{p,i+1}$ depends on $X_{p,i+1}$ and θ. The latter dependence can be reduced to ϑ_{i+1}, as only parameters relevant after t_{i+1} impact values at t_{i+1}. Proceeding by induction, assume one has both $\partial_{\vartheta_{i+1}} V_{i+1}$ and $\partial_{X_{i+1}} V_{i+1}$. The task is to track sensitivities through the projection of V_{i+1} to V_i and propagate them for $\partial_{\vartheta_i} V_i$ and $\partial_{X_i} V_i$. To this end, let $R(\cdot)$ denote the output of the regression operation. Ignoring the numeraire for simplicity,[7] one has $V_i = R_i \equiv R(V_{i+1}, X_i)$, such that:

$$\partial_{\vartheta_i} V_i = \partial_{V_{i+1}} R(V_{i+1}, X_i)(\partial_{\vartheta_i} V_{i+1} + \partial_{X_{i+1}} V_{i+1} \partial_{\vartheta_i} X_{i+1})$$
$$\partial_{X_i} V_i = \partial_{V_{i+1}} R(V_{i+1}, X_i)\partial_{X_{i+1}} V_{i+1} \partial_{X_i} X_{i+1} + \partial_{X_i} R(V_{i+1}, X_i) \tag{9}$$

Dealing first with non-regression components, it is clearly the case that $\partial_{\vartheta_i} V_{i+1} = [0, \partial_{\vartheta_{i+1}} V_{i+1}]$. For $\partial_{\vartheta_i} X_{i+1}$ and $\partial_{X_i} X_{i+1}$, assume for simplicity that X_{i+1} is produced from X_i via the sampling scheme $X_{p,i+1} = X(t_{i+1}, X_{p,i}, \vartheta_i, u_{p,i})$, where $u_{p,i}$ is a random deviate; $\partial_{\vartheta_i} X_{i+1}$ and $\partial_{X_i} X_{i+1}$ are available by simply differentiating this element-wise.

The quantities $\partial_{V_{i+1}} R_i$ and $\partial_{X_i} R_i$ involve regression differentiation. As per (3), fitting the t_i regression requires N_B basis functions $\phi_i \equiv [\phi_{n,i}(\cdot)]$. It is assumed $\phi_{n,i}(\cdot)$ are θ independent, eg, polynomials or splines in X_i, but with minimal effort one can extend to θ-dependent bases including transforms of swaps, European swaptions, etc. One then selects parameters $\alpha_i \equiv [\alpha_{i,n}]$ so as to minimise:

$$\chi^2 = \mathbb{E}_{t_i}[(V_{i+1} - \alpha_i \phi_i(X_i))^2] \simeq \frac{1}{N_P}\sum_{p=1}^{N_P}(V_{p,i+1} - \alpha_i \phi_i(X_{p,i}))^2 \tag{10}$$

Minimising χ^2 over α_i yields:[8]

$$\alpha_i = (\phi_i(X_i)^{\mathrm{T}} \phi_i(X_i))^{-1} \phi_i(X_i)^{\mathrm{T}} V_{i+1}$$

[7] *The numeraire will simply introduce the quantities* $\partial_{\vartheta_i}(N_i/N_{i+1})$ *and* $\partial_{X_i}(N_i/N_{i+1})$.

[8] *This analysis readily extends to weighted least squares, using shrinkage, as well as solving normal equations directly instead of by inversion.*

making it clear $\alpha_i = \alpha_i(\vartheta_i)$ given its dependence on V_{i+1}, which depends on ϑ_i through ϑ_{i+1} and X_{i+1}. From this, the fitted set of V_i is:

$$\begin{aligned} V_i &= R(V_{i+1}, X_i) = \phi_i(X_i)\alpha_i \\ &= \underbrace{\phi_i(X_i)(\phi_i(X_i)^{\mathrm{T}}\phi_i(X_i))^{-1}\phi_i(X_i)^{\mathrm{T}}}_{\equiv M(\phi_i(X_i))} V_{i+1} \\ &= M(\phi_i(X_i))V_{i+1} \end{aligned} \tag{11}$$

Given this, differentiating the regression output over states and parameters reduces to evaluating:

$$\partial_{V_{i+1}} R_i = M(\phi_i(X_i)) \quad \text{and} \quad \partial_{X_i} R_i = \partial_{X_i} M(\phi_i(X_i))V_{i+1} \tag{12}$$

Given (12), it is clear (θ, X_{i+1}) dependencies embedded in V_{i+1} are propagated into those for V_i via multiplication by $M(\phi_i(X_i))$. Capturing the X_i dependence introduced via the regression requires differentiation of $M(\phi_i(X_i))$ with respect to X_i. This can be simplified by noting that, for large N_P, elements of $\partial_{X_i}\alpha_i$ tend to zero. In other words, for fixed basis functions, changing X_i and re-running LSMC to t_i will produce the same coefficients asymptotically, implying:

$$\partial_{X_i} M(\phi_i(X_i)) = \partial_{X_i}\phi(X_i)\alpha_i$$

Finally, 'full' BD for $\partial_\theta V_{p,i}$ used towards CVA sensitivities will propagate $\partial_{X_{p,i}} V_{p,i}$ over the θ dependence in $X_{p,i}$; 'modified' BD for $\partial_{\vartheta_i} V_{p,i}$ used towards MVA ignores this channel.

■ **Accuracy of AD-on-LSMC sensitivities.** As already noted, the values $V_{p,i}$ output by LSMC are an approximation, and thus so are the sensitivities $\partial_{\vartheta_i} V_{p,i}$ output by AD-on-LSMC. There are established results regarding LSMC convergence. If $V_{p,i}(\theta; N_P, N_B)$ is the LSMC approximation using N_P paths and N_B basis functions, it can be shown that:

$$\lim_{N_P, N_B \to \infty} V_{p,i}(\theta; N_P, N_B) = V(t_i, X_{p,i}, \theta)$$

under appropriate technical conditions, where the convergence is in probability (see, for example, Wang & Caflisch (2010) and the references therein). This article proposes using $\partial_{\vartheta_i} V_{p,i}(\theta; N_P, N_B)$, though it does not seek to formally establish the conditions under which:

$$\begin{aligned} \lim_{N_P, N_B \to \infty} \partial_{\vartheta_i} V_{p,i}(\theta; N_P, N_B) &= \partial_{\vartheta_i} \lim_{N_P, N_B \to \infty} V_{p,i}(\theta; N_P, N_B) \\ &= \partial_{\vartheta_i} V(t_i, X_{p,i}, \theta) \end{aligned}$$

given the article's practical scope. Such asymptotics would in any case be no guarantee the output $\partial_{\vartheta_i} V_{p,i}(\theta; N_P, N_P)$ are accurate for typical combinations of controls (N_P, N_B) and basis functions used in practice. Indeed, significant validation is required to ensure $V_{p,i}(\theta; N_P, N_B)$ itself approximates exposures well. This is important, as the quality of the two approximations must be intimately related.

To gain intuition, consider computing $\partial_\theta V(t_i, X_i, \theta; N_P, N_P)$ by finite difference. One starts with the current (vector) X_i and runs LSMC twice: once for θ and once for a perturbed θ'. Thus, the θ sensitivity literally compares one LSMC result with another at a neighbouring parameter value. Now assume the adopted LSMC controls produce accurate approximations for $V(t_i, X_i, \theta; N_P, N_P)$, which are free of excessive oscillation and have well-behaved asymptotes, and assume also this has been validated for a range of θ values. Under such conditions, differentiating LSMC over parameters should be expected to produce sensitivity approximations of a quality sufficient for MVA calculations. This is supported by the numerical results presented in figure 1.

A practical suggestion for improving the quality of sensitivities is to recognise that basis functions must be capable of spanning the sensitivities, and to choose or extend them accordingly. In cases where a larger set of basis functions is necessary, more paths are required to ensure the quality of the fitted function. However, the cost of an incremental increase in N_B and N_P pales in comparison to that of moving to an $\mathcal{O}(N_P^2)$ brute force method. As a final note, an alternative to differentiating through LSMC is to recursively compute:

$$\partial_{\vartheta_i} V(t_i) = \mathbb{E}_{t_i}[\partial_{\vartheta_i} V(t_{i+1}, \vartheta_{i+1}, X_{i+1}(\vartheta_i))]$$

ignoring the numeraire for simplicity. However, N_θ regressions are significantly more expensive than differentiating one regression over N_θ inputs, owing to efficiencies in propagating matrix operations (11). Moreover, they do not leverage existing AD machinery.

A Bermudan swaption numerical example

The modified BD algorithm is applied to an 11Y-semi-versus-semi ATM Bermudan receiver with a 2.01% swap rate (the yield curve is 2% flat). The exercise frequency is 1Y, starting at 5Y, and the observation frequency is 1Y. We use a Hull-White one-factor model with 5% mean-reversion and flat left-interpolated volatility calibrated to an ATM diagonal of 20% swaption volatilities. The yield curve knots are spaced at 3M intervals, the volatility knots at 1Y intervals, and sensitivities over these are bucketed according to Simm quote-tenor intervals, as per our previous discussion. These are transformed into deltas over swap rates with tenors 3M, 6M, 1Y, 2Y, 3Y, 5Y, 10Y and 12Y, and into vegas over swaption volatilities ending at 11Y and starting at 1Y, 2Y, ..., 10Y to coincide with exercise dates. Sensitivities over swap rates at 2W and 1M are effectively loaded onto the 3M swap rate to allow for a coarser interpolation. Discretising at a 2W frequency would allow enough knots for a full set of Simm sensitivities; however, it has minimal impact on final IM figures, as these tenors attract similar risk weights and high correlations under Simm.

The following simulations are performed here: (a) modified BD for FVs of a Bermudan exercising into a cleared swap; (b) modified BD for FVs of a Bermudan exercising into a non-cleared swap; and (c) brute force for the CVs of a Bermudan (ignoring prior exercises). For each path and observation date, future $\mathrm{IM}_{\mathrm{Delta}}$ and $\mathrm{IM}_{\mathrm{Vega}}$ are computed by applying the relevant risk weights and volatilities.[9] This exercise concisely demonstrates several important points. First and foremost, regarding quality, if BD produces a good approximation of brute force, the IM profiles for all three simulations should agree prior to the first exercise date (5Y); afterwards, they should diverge given differences in the Bermudan's nature for cleared swap-versus-non-cleared swap-versus-ignoring exercise behaviour altogether.

Figure 1 presents expected future IM profiles $\mathbb{E}_{t_0}[\mathrm{IM}(t_i)]$ for each simulation: part (a) is $\mathrm{IM}_{\mathrm{Delta}}$ and part (b) is $\mathrm{IM}_{\mathrm{Vega}}$. The accompanying table A presents the associated CPU times incurred on a standard laptop.[10]

[9] $\mathrm{IM}_{\mathrm{Curvature}}$ *is disregarded for compactness. Note that it depends on vegas and* $\mathrm{IM}_{\mathrm{Vega}}$ *offers a gauge of their quality.*

[10] *Full numerical results are available in Antonov et al (2016b).*

As seen, IM for the three simulations agree (to the fourth decimal) prior o the first exercise date for both $\mathrm{IM}_{\mathrm{Delta}}$ and $\mathrm{IM}_{\mathrm{Vega}}$, confirming accuracy. $\mathrm{M}_{\mathrm{Delta}}$ for the cleared and non-cleared cases predictably diverge after the irst exercise date, while their vegas coincide. The term $\mathbb{1}_i \partial_{Q_i} S_i$ drops ›ut in the cleared case, significantly affecting delta IM but not vega IM ıs it vanishes in both cases, irrespective of clearing. This highlights the ıeed to handle exercise behaviour correctly. Finally, the IM profiles for he FV in the non-cleared case closely mimic those of the CV after the irst exercise, owing to similar behaviours of in-the-money Bermudans and ;waps post-exercise. Naturally, the magnitude of the disagreement grows vith time.

Convergence studies have determined BD converges for simulations ·equiring 25–40 seconds of CPU time. For BF, this increases to 3.0–3.5 ıours, which is simply too long for practical use, even where MVA calculations are required only infrequently. As a rule of thumb, it was found hat BD offered an acceleration factor of 300–400 over BF, supporting the :laim that BD is an efficient and practicable method for Simm-MVA.

Interestingly, the timing results reveal calculating Jacobians dominates he actual calculation of sensitivities, and this step is amenable to further optimisations. The sensitivities over parameters and Jacobian inversion cost very little by comparison. Critically, this suggests that the BD running time will grow only mildly with increases in the number of trades in a counterparty portfolio, whereas for BF it is clearly proportional to the number of trades given the cost of parameter sensitivities per trade. ■

A. Computation times

	BD (secs)			BF (hours)
No. of paths	Parameter sensitivities	Jacobian calculation	Jacobian inversion	Total
2000	0.2	20	1.8	2.5
4000	0.41	41	2.7	15

BF denotes brute force

Alexandre Antonov is a director at Standard Chartered in London. Andrew McClelland is a director in the quantitative research team at Numerix in New York, and Serguei Issakov is a San Francisco-based senior vice-president in the quantitative research group at Numerix.
Email: alexandre.antonov@sc.com, isakov@numerix.com,
clelland@numerix.com

REFERENCES

Andersen L, M Pykhtin and A Sokol, 2017
Rethinking the margin period of risk
Journal of Credit Risk 13(1), pages 1–45

Antonov A, S Issakov and S Mechkov, 2015
Backward induction for future values
Risk January, pages 92–97

Antonov A, S Issakov, M Konikov, A McClelland and S Mechkov, 2016a
PV/XVA Greeks for callable exotics by algorithmic differentiation
Working Paper, SSRN

Antonov A, S Issakov and A McClelland, 2016b
Efficient SIMM-MVA calculations for callable exotics
Working Paper, September, SSRN

Capriotti L, Y Jiang and A Macrina, 2016
AAD and least squares Monte Carlo: fast Bermudan-style options and XVA Greeks
Working Paper, September, SSRN

Giles M and P Glasserman, 2006
Smoking adjoints: fast Monte Carlo Greeks
Risk January, pages 92–96

Green A and C Kenyon, 2015
MVA by replication and regression
Risk April, pages 82–87

Green A and C Kenyon, 2017
XVA at the exercise boundary
Risk February, pages 128–133

Huge B and A Savine, 2017
LSM reloaded: differentiate xVA on your iPad Mini
Working Paper, June, SSRN

ISDA, 2017
ISDA SIMM methodology
Version R1.3, March, International Swaps and Derivatives Association

Wang Y and R Caflisch, 2010
Pricing and hedging American-style options: a simple simulation-based approach
Journal of Computational Finance 13(4), pages 85–125

GUIDELINES FOR THE SUBMISSION OF TECHNICAL ARTICLES

Risk.net welcomes the submission of technical articles on topics relevant to our readership. Core areas include market and credit risk measurement and management, the pricing and hedging of derivatives and/or structured securities, the theoretical modelling of markets and portfolios, and the modelling of energy and commodity markets. This list is not exhaustive.

The most important publication criteria are originality, exclusivity and relevance. Given *Risk.net* technical articles are shorter than those in dedicated academic journals, clarity of exposition is another yardstick for publication. Once received by the quant finance editor and his team, submissions are logged and checked against these criteria. Articles that fail to meet the criteria are rejected at this stage.

Articles are then sent to one or more anonymous referees for peer review. Our referees are drawn from the research groups, risk management departments and trading desks of major financial institutions, in addition to academia. Many have already published articles in *Risk.net*. Authors should allow four to eight weeks for the refereeing process. Depending on the feedback from referees, the author may attempt to revise the manuscript. Based on this process, the quant finance editor makes a decision to reject or accept the submitted article. His decision is final.

Submissions should be sent to the technical team (technical@infopro-digital.com).

PDF is the preferred format. We will need the LaTeX code, including the BBL file, and charts in EPS or XLS format. Word files are also accepted.

The maximum recommended length for articles is 4,500 words, with some allowance for charts and formulas. We reserve the right to cut accepted articles to satisfy production considerations.