Adjoint Credit Risk Management

Luca Capriotti and Jacky Lee

Quantitative Strategies, Investment Banking Division, Credit Suisse Group,
Eleven Madison Avenue, New York City, NY 10010-3086, United States of America

(Dated: July 30, 2013)

Adjoint Algorithmic Differentiation is one of the principal innovations in risk management of the recent times. In this paper we show how this technique can be used to compute real time risk for credit products.

Keywords: Adjoint Algorithmic Differentiation, Credit Risk, Derivatives Pricing

Introduction

The aftermath of the recent financial crisis has seen a dramatic shift in the credit derivatives markets, with a conspicuous reduction of demand for complex, capital intensive products, like bespoke collateralized debt obligations (CDO), and a renewed focus on simpler and more liquid derivatives, like credit default indices and swaptions.

In this background, dealers are quickly adapting to a business model geared towards high-volume, lower-margin products for which managing efficiently the trading inventory is of paramount importance. As a result, the ability to produce risk in real time is rapidly becoming one of the keys to running a successful trading operation.

A recently introduced technology for real time risk is *Adjoint Algorithmic Differentiation* (AAD) (Capriotti, 2011; Capriotti and Giles, 2010, 2012). This powerful technique allows the fast computation of risk without the necessity of repeating the valuation of the portfolio multiple times as in traditional *bump and reval* (or bumping) approaches. In contrast to computational solutions based on parallel architectures like GPUs and FPGAs, AAD does not require investments in new infrastructure or additional computational resources. Rather, AAD is a straightforward mathematical technique that can be easily implemented and integrated in existing analytics software.

The remarkable efficacy of AAD was recently demonstrated in a variety of risk management problems in the context of highly time consuming Monte Carlo valuations, including counterparty credit management (Capriotti *et al.*, 2011). In this paper, we demonstrate how this technique can be extremely effective also for simpler credit products, typically valued by means of faster semi analytical techniques. We will show how AAD provides orders of magnitude savings in computational time and makes the computation of risk in real time - with no additional infrastructure investment - a concrete possibility.

Pricing of Credit Derivatives

The key concept for the valuation of credit derivatives, in the context of the models generally used in practice, is the *hazard rate*, λ_u, representing the probability of default of the reference entity between times u and $u+du$, conditional on survival up to time u. By modelling the default event of a reference entity i as the first arrival time of a Poisson process with deterministic intensity λ^i_u, the survival probability, $Q_i(t,T)$, is given by

$$Q(t,T;\lambda^i) = \exp\left[-\int_t^T du\ \lambda^i_u\right] . \tag{1}$$

In the hazard rate framework, the price of a credit derivative can be expressed mathematically as

$$V(\theta) = V(\boldsymbol{\lambda}(\theta), \theta) , \tag{2}$$

where, $\boldsymbol{\lambda} = (\lambda^1, \ldots, \lambda^N)$ are the hazard rate functions for N credit entities referenced in a given contract. Here we have indicated generically with $\theta = (\theta_1, \ldots, \theta_{N_\theta})$ the vector of model parameters, *e.g.*, credit spreads, recovery rates, volatilities, correlation and the market prices of the interest rate instruments used for the calibration of the discount curve.

In general, the valuation of a credit derivative can be separated in a

Calibration Step :

$$\theta \to \boldsymbol{\lambda}(\theta)$$

for the construction of the hazard rate curve given liquidly traded CDS prices, a term structure of recoveries and a given discount curve, and a

Pricing Step :

$$\theta \to V(\boldsymbol{\lambda}(\theta), \theta)$$

mapping the hazard rate curves and the other parameters to which the pricing model is explicitly dependent on, to the price of the credit derivative. The pricing step is obviously specific to the particular credit derivative under valuation. Instead, the calibration step is the same for any derivative priced within the hazard rate framework. For the purpose of the discussion below it is useful to recall the main steps involved in the calibration of a hazard rate curve.

Calibration step

The hazard rate function λ_u in Eq. (1) is commonly parameterized as piece-wise constant with M knot points at time $(t_1, \ldots, t_M)$, $\lambda = (\lambda_1, \ldots, \lambda_M)$, such that

$$\lambda_u = \lambda_{n-1} = \frac{1}{t_n - t_{n-1}} \ln \left(\frac{Q(t, t_{n-1}; \lambda)}{Q(t, t_n; \lambda)} \right)$$

for $t_{n-1} \leq u < t_n$ and t_0 equal to the valuation date. In the calibration step, the hazard rate knot points are calibrated from the price, or equivalently the credit (par) spreads $(s_1, \ldots, s_M)$, of a set of liquidly traded CDS with maturities $T_1, \ldots, T_M$ *e.g.*, using the standard bootstrap algorithm (O'Kane, 2011).

Such calibration can be expressed mathematically as solving a system of M equations

$$G_j(\lambda, \theta) = 0 \ , \tag{3}$$

$j = 1, \ldots M$, with

$$G_j(\lambda, \theta) = s_j - \frac{\mathcal{L}(t, T_j; \lambda, \theta)}{\mathcal{A}(t, T_j; \lambda, \theta)} \ , \tag{4}$$

where $\mathcal{L}(t, T_j; \lambda, \theta)$ and $\mathcal{A}(t, T_j; \lambda, \theta)$ are, respectively, the expected loss and *credit risky* annuity for a T_j maturity CDS contract starting at time t[1]. These are defined as

$$\mathcal{L}(t, T; \lambda, \theta) = \int_t^T du \ Z(t, u; \theta) \, (1 - R_u) \left(-\frac{dQ(t, u; \lambda)}{du} \right), \tag{5}$$

and (e.g., for continuously paid coupons)

$$\mathcal{A}(t, T; \lambda, \theta) = \int_t^T du \ Z(t, u; \theta) Q(t, u; \lambda) \ . \tag{6}$$

Here, $Z(t, u; \theta)$ is the discount factor from time t to time u, $Q(t, u; \lambda)$ (*resp.* $-dQ(t, u; \lambda)/du$) is the probability that the reference entity survives up to (*resp.* defaults in an infinitesimal interval around) time u, and R_u is the expected percentage recovery upon default at time u. The latter is generally expressed as a piecewise constant function with the same discretization of the hazard rate function, say $R = (R_1, \ldots, R_M)$.

[1] Note that although the credit spreads s_j are contained in the model parameter vector θ the risky annuity and the expected loss do not depend explicitly on them.

The calibration equations (3) and (4) are based on the definition of par spread s_i as break-even coupons c making the value of CDS

$$V_{\mathrm{CDS}}(t, T; \theta) = \mathcal{L}(t, T; \lambda, \theta) - c \, \mathcal{A}(t, T; \lambda, \theta), \tag{7}$$

worth zero[2]. Since both the expected loss and the risky annuity at time T_i depend on hazard rate points λ_j with $j \leq i$, the calibration equations can be solved iteratively starting from $i = 1$, by keeping fixed the hazard rate knot points λ_j with $j < i$, and solving for λ_i.

Through the calibration process, the system of M equations (3) defines *implicitly* the function $\lambda = \lambda(\theta)$, linking the hazard rate to the credit spreads, the term structure of expected recovery and the discount factors. These are in turn a function of the market instruments that are used for the calibration of the discount curve.

Challenges in the Calculation of Credit Risk

The computation of the sensitivities of the price of the credit derivative (2) with respect to the model parameters θ can be performed by means of the chain rule

$$\frac{\partial V}{\partial \theta_k} = \frac{\partial V}{\partial \theta_k} + \sum_{j=1}^{M} \frac{\partial V}{\partial \lambda_j} \frac{\partial \lambda_j}{\partial \theta_k} \ , \tag{8}$$

where the first term captures the explicit dependence on the model parameters θ through the pricing step, and the second term captures the implicit dependence via the calibration step.

The computation of the calibration component of the prices sensitivities with standard bump and reval approaches is particularly onerous because it involves repeating the calibration step for each perturbation. Especially for portfolio of simple credit derivatives, like CDS, this can easily represent the bulk of the computational burden. In addition, finite-size perturbations of credit spreads, recovery or interest rates often correspond to inputs that do no admit an arbitrage-free representation in terms of a non-negative hazard rate curve, thus making the robust and stable computation of sensitivities challenging.

[2] Note that since the standardization of CDS contracts in 2008, liquidly traded CDS are characterized by a standard coupon and are generally quoted in terms of upfronts or quote spreads. Both mark types can be mapped to a dollar value of a CDS contract by means of a market standard parameterization (ISDA, 2013), and hazard rates can be equivalently bootstrapped from these marks using Eq. (7). Credit (par) spreads remain nonetheless commonly used in the market practice as risk factors for credit derivatives. The analysis of this paper can be easily formulated in terms of quote spreads or upfronts.

Adjoint Calculation of Risk

Both the computational costs and stability of the calculation of credit risk can be effectively addressed by means of the AAD implementation of the chain rule (8). In particular, the adjoint of the algorithm consisting of the *Calibration Step* and *Pricing Step*, described above reads

$\overline{Pricing\ Step}$:

$$\bar{\theta}_k = \bar{V}\frac{\partial V}{\partial \theta_k} \qquad \bar{\lambda}_j = \bar{V}\frac{\partial V}{\partial \lambda_j} \ , \tag{9}$$

$\overline{Calibration\ Step}$:

$$\bar{\theta}_k = \bar{\theta}_k + \sum_{j=1}^{M} \bar{\lambda}_j \frac{\partial \lambda_j}{\partial \theta_k} \ . \tag{10}$$

Here we have used the standard 'bar' notation to indicate adjoint variables and adjoint functions. In particular, we recall that, given a function

$$Y = Y(X) \ , \tag{11}$$

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

$$X \ \rightarrow \ \ldots \ \rightarrow \ U \ \rightarrow \ W \ \rightarrow \ \ldots \ \rightarrow \ Y \ ,$$

where the real vectors U and W represent intermediate variables used in the calculation, its adjoint counterpart reads

$$\bar{X} = \overline{Y}(X, \bar{Y}) \ , \tag{12}$$

where the adjoint of the output $\bar{Y}$ is an arbitrary vector in $\mathbb{R}^m$ and the adjoint of the input $\bar{X}$, is given by

$$\bar{X}_i = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial X_i}, \tag{13}$$

with $i = 1, \ldots, n$. Note that the vector $\bar{Y}$ allows one to select a specific linear combination of the rows of the Jacobian $\partial Y_j / \partial X_i$ with respect to which derivatives are computed. In the case of scalar function, $\bar{Y}$ is a scalar that can be set to one[3].

The adjoint function (12) can be implemented by reversing the order of the computations in the original function as

$$\bar{X} \ \leftarrow \ \ldots \ \leftarrow \ \bar{U} \ \leftarrow \ \bar{V} \ \leftarrow \ \ldots \ \leftarrow \ \bar{Y} \ ,$$

where the adjoint of any intermediate variable U_k is defined as

$$\bar{U}_k = \sum_{j=1}^{m} \bar{Y}_j \frac{\partial Y_j}{\partial U_k}.$$

The key theoretical result is that, given a computer program performing some high-level function (11), the execution time of its adjoint counterpart (12) calculating the linear combination (13) is bounded by approximatively 4 times the cost of execution of the original one (Capriotti and Giles, 2012).

Given the definitions above, it is immediate to verify that for $\bar{V} = 1$, each $\bar{\theta}_k$ computed by means of the adjoint of the pricing and calibration steps, Eqs. (9) and (10), gives the price sensitivity in Eq. (8).

Although in the following we will give explicit examples of the adjoint of the pricing step for portfolios of CDS and credit default index swaptions, here we focus our discussion on the adjoint of the calibration step in Eq. (10) which is a time consuming and numerically challenging step common to all pricing applications within the hazard rate framework.

Implicit Function Theorem

The adjoint of the calibration step $\theta \rightarrow \lambda(\theta)$ can be produced following the general rules of AAD. The associated computational cost can be generally expected to be of the order of the cost of performing the bootstrap algorithm a few times (but approximately less than 4 according to the general result of AAD quoted above). This in itself is generally a very significant improvement with respect to bump and reval approaches, involving repeating the bootstrap algorithm as many times as sensitivities required. However, following the suggestions of (Christianson, 1998; Henrard, 2011), a much better performance can be obtained by exploiting the so-called *implicit function theorem*, as described below.

By differentiating with respect to θ the calibration identity (3) we get

$$\frac{\partial G_i}{\partial \theta_k} + \sum_{j=1}^{M} \frac{\partial G_i}{\partial \lambda_j}\frac{\partial \lambda_j}{\partial \theta_k} = 0 \ ,$$

for $i = 1, \ldots, M$, and $k = 1, \ldots, N_\theta$, or equivalently

$$\frac{\partial \lambda_i}{\partial \theta_k} = -\left[\left(\frac{\partial G}{\partial \lambda}\right)^{-1} \frac{\partial G}{\partial \theta}\right]_{ik} \ . \tag{14}$$

This relation allows the computation of the sensitivities of $\lambda(\theta)$, locally defined in an implicit fashion by Eqs. (3) and (4), in terms of the sensitivities of the function (4).

In the specific case, when $\theta_k \neq s_j$ for $j = 1, \ldots, M$, *i.e.* when considering sensitivities with respect to market risk factors other than the credit spreads, Eq. (14) can

[3] It is worth noting that, as customary in elementary calculus, the same symbol Y is used for both the output of the function and the function itself. Similarly, the same symbol $\bar{Y}$ is used for both the adjoint function and the adjoint input, as customary in the adjoint literature.

be expressed in turn as

$$\frac{\partial \lambda_i}{\partial \theta_k} = - \left[\left(\frac{\partial s(\lambda, \theta)}{\partial \lambda} \right)^{-1} \frac{\partial s(\lambda, \theta)}{\partial \theta} \right]_{ik} . \tag{15}$$

Here we have used that θ_k is not a credit spread so that $\partial G / \partial \theta_k = -\partial s(\lambda, \theta)/\partial \theta_k$, where the par spread *functions*

$$s(\lambda, \theta) = (s_1(\lambda, \theta), \ldots, s_M(\lambda, \theta)) ,$$

$$s_j(\lambda, \theta) = \frac{\mathcal{L}(t, T_j; \lambda, \theta)}{\mathcal{A}(t, T_j; \lambda, \theta)} , \tag{16}$$

are defined by Eqs. (3) and (4).

In the case of credit spread sensitivities, $\theta_k = s_k$, Eq. (14) simplifies as follows

$$\begin{aligned} \frac{\partial \lambda_i}{\partial s_k} &= \sum_{j=1}^{M} \left[\frac{\partial s(\lambda, \theta)}{\partial \lambda} \right]^{-1}_{ij} \frac{\partial G_j}{\partial s_k} \\ &= \sum_{j=1}^{M} \left[\frac{\partial s(\lambda, \theta)}{\partial \lambda} \right]^{-1}_{ij} \frac{\partial s_j}{\partial s_k} = \left[\frac{\partial s(\lambda, \theta)}{\partial \lambda} \right]^{-1}_{ik} , \end{aligned} \tag{17}$$

where we have used that the par spread functions do not explicitly depend on the credit spreads s_k.

Equations (15) and (17) express the implicit function theorem in the context of hazard rate calibration. These allow the computation of the sensitivities $\partial \lambda_i / \partial \theta_k$ by *i)* evaluating the sensitivities of the par spread functions with respect to the model parameters, $\partial s_j(\lambda, \theta)/\partial \theta_k$, and the hazard rates, $\partial s_k(\lambda, \theta)/\partial \lambda_i$ and *ii)* solving a linear system, *e.g.*, by Gaussian elimination. This method is significantly more stable and efficient than the naïve approach of calculating the derivatives of the implicit functions $\theta \to \lambda(\theta)$ by differentiating directly the calibration step either by bump and reval or by applying AAD to the calibration step. This is because $s(\lambda, \theta)$ in Eq. (16) are *explicit* functions of the hazard rate and the model parameters that are easy to compute and differentiate.

Combining the implicit function theorem with adjoint methods results in extremely efficient risk computations, as we will demonstrate below.

Adjoint of the Calibration Step

All the sensitivities necessary to compute Eqs. (15) and (17) can be obtained through the adjoint of the function

$$s_j = s_j(\lambda, \theta)$$

defined by Eq. (16), namely, using the definitions (11) and (12),

$$(\bar{\lambda}, \bar{\theta}) = \bar{s}_j(\lambda, \theta, \bar{s}_j) ,$$

where the scalar $\bar{s}_j$ is the adjoint of the j-th par spread with $j = 1, \ldots, M$. By applying the rules of AAD, this can be implemented as

$$\begin{aligned} \overline{\mathcal{A}}_j &= -\bar{s}_j \frac{\mathcal{L}(t, T_j; \lambda, \theta)}{\mathcal{A}(t, T_j; \lambda, \theta)^2} \\ \overline{\mathcal{L}}_j &= \bar{s}_j \frac{1}{\mathcal{A}(t, T_j; \lambda, \theta)} \\ (\bar{\lambda}, \bar{\theta}) &\mathrel{+}= \overline{\mathcal{A}}(t, T_j; \lambda, \theta, \overline{\mathcal{A}}_j) , \\ (\bar{\lambda}, \bar{\theta}) &\mathrel{+}= \overline{\mathcal{L}}(t, T_j; \lambda, \theta, \overline{\mathcal{L}}_j) , \end{aligned}$$

where $\overline{\mathcal{A}}(t, T_j; \lambda, \theta, \overline{\mathcal{A}}_j)$ and $\overline{\mathcal{L}}(t, T_j; \lambda, \theta, \overline{\mathcal{L}}_j)$ are the adjoints of $\mathcal{A}(t, T_j; \lambda, \theta)$ and $\mathcal{L}(t, T_j; \lambda, \theta)$, respectively. Here we have used the standard AAD notation for the increment operator += (Capriotti and Giles, 2012).

Combining AAD and the implicit function theorem results therefore in the following algorithm for the adjoint of the calibration routine, $\bar{\theta} = \bar{\lambda}(\theta, \bar{\lambda})$:

1. Execute $(\bar{\lambda}, \bar{\theta}) = \bar{s}_j(\lambda, \theta, \bar{s}_j)$ with $\bar{s}_j = 1$ for $j = 1, \ldots, M$. This gives the derivatives:

   $$\bar{\lambda}_{ij} = \frac{\partial s_j}{\partial \lambda_i} \qquad \bar{\theta}_{kj} = \frac{\partial s_j}{\partial \theta_k} ,$$

   for $i = 1, \ldots, M$, and $k = 1, \ldots, N_\theta$.

2. Find the matrix $\partial \lambda / \partial \theta$ by solving the linear system

   $$\frac{\partial s}{\partial \lambda} \frac{\partial \lambda}{\partial \theta} = - \frac{\partial s}{\partial \theta} .$$

3. Return:

   $$\bar{\theta}_k = \sum_{i=1}^{M} \bar{\lambda}_i \frac{\partial \lambda_i}{\partial \theta_k} ,$$

   for $k = 1, \ldots, M$.

The adjoint of the calibration algorithm described above is extremely efficient. Indeed, as illustrated in Fig. 1, the sensitivities of the hazard rate with respect to the credit spreads, and interest rates instruments can be computed in $\sim 25\%$ *less* time than performing a single bootstrap.

Applications

Credit Default Swaps

As a first example we consider the calculation of price sensitivities for a (portfolio of) CDS. In this case, the adjoint of the pricing step simply reads, from Eq. (7),

$$\begin{aligned} \overline{\mathcal{L}} &= \bar{V} \\ \overline{\mathcal{A}} &= -\bar{V} c \\ (\bar{\lambda}, \bar{\theta}) &= \overline{\mathcal{A}}(t, T; \lambda, \theta, \overline{\mathcal{A}}) , \\ (\bar{\lambda}, \bar{\theta}) &\mathrel{+}= \overline{\mathcal{L}}(t, T; \lambda, \theta, \overline{\mathcal{L}}) , \end{aligned}$$

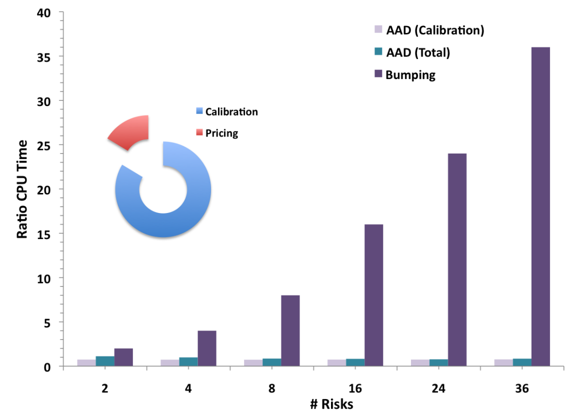

FIG. 1 Cost of computing the sensitivities with respect to the credit spreads and interest rates instruments - relative to the cost of a single valuation - as a function of the number of sensitivities.

where the risky annuity and expected loss (and their adjoint counterparts) are those of the CDS in the portfolio. In this case, as illustrated in Fig. 1, the cost of the pricing step is a small portion ($\sim 10\%$) of the overall cost of computing the sensitivities which is instead dominated by the cost of the calibration step. As a result, all the sensitivities can be obtained by means of AAD for $\sim 15\%$ less than the cost of performing a single valuation. In typical applications, where computing sensitivities with respect to 18 spread tenors and interest rate instruments is commonplace, this results in a reduction of the computational cost by a factor of 50 or more.

Credit Default Index Swaptions

As a second example, also of significant practical relevance, we consider credit default index swaptions. The value of these instruments at time t is given by

$$V_t = Z(t, T_E; \theta)\times \\ \mathbb{E}_t\Big(\max\Big(\zeta\Big[V_{\rm iCDS}(T_S, T_E) + L(T_E) - P_E\Big], 0\Big)\Big) \quad (18)$$

where $\zeta = 1$ for a payer and $\zeta = -1$ for a receiver option, $V_{\rm iCDS}(T_E, T_M)$ is the value at time T_E of the underlying credit default index swap (long protection) with standard coupon rate and maturity T_M, P_E is the exercise fee, and $L(T_E)$ is the value at time T_E of the loss given default associated to the names that have defaulted before expiry,

$$L(T_E) = \sum_{i=1}^{N} I(\tau^i < T_E) N^i (1 - R^i_\tau),$$

where N is the number of names in the index, I is the indicator function, and N^i, τ^i and R^i_u are the notional, default time and recovery function of the i-th name in the portfolio[4].

According to the *de facto* market standard model (Pedersen, 2003) the value at time T_E of the random quantity given by the sum of the loss amount, $L(T_E)$, and the value of the credit default index swap, $V_{\rm iCDS}(T_E, T_M)$, is modelled in terms of a single state variable, the default adjusted forward spread s_{T_E}, as

$$V_{\rm iCDS}(T_E, T_M) + L(T_E) = \\ N_{tot}\mathcal{A}_{\rm isda}\left(s_{T_E}, T_E, T_M\right)\left(s_{T_E} - c\right), \quad (19)$$

where c is the fixed rate in the underlying credit default index swap and $N_{tot} = \sum_{i=1}^{N} N^i$ is the total notional of the index. Here $\mathcal{A}_{\rm isda}(s, t, T)$ is the standardized risky annuity of Eq. (6) calculated assuming a flat term structure of the credit spread s, according to the standard ISDA conventions (ISDA, 2013). In the simplest setting, the default adjusted forward spread is assumed lognormally distributed,

$$s_{T_E} = F_{T_E} \\ \times \exp\Big[-\frac{1}{2}\sigma^2_{T_E}(T_E - t) + \sigma_{T_E}\sqrt{T_E - t}\,\tilde{Z}\Big], \quad (20)$$

where σ_{T_E} is the volatility of the default adjusted forward spread, $\tilde{Z}$ is a standard normal random variable and the forward F_{T_E}, can be determined by taking the expectation of both sides of Eq. (19) giving,

$$G_F(F_{T_E}, \boldsymbol{\lambda}, \theta) \equiv \\ V^{adj}_{\rm iCDS}(T_E, T_M; \boldsymbol{\lambda}, \theta) - V^{isda}_{\rm iCDS}(T_E, T_M; F_{T_E}, \theta) = 0\,. \quad (21)$$

The first term in the equation above,

$$V^{adj}_{\rm iCDS}(T_E, T_M; \boldsymbol{\lambda}, \theta) = \\ \mathbb{E}_t\Big[V_{\rm iCDS}(T_E, T_M) + L(T_E)\Big],$$

can be computed according to the standard hazard rate model using the time t default and recovery curves of the index constituents:

$$V^{adj}_{\rm iCDS}(T_E, T_M; \boldsymbol{\lambda}, \theta) = \tilde{\mathcal{L}}(t, T_E; \boldsymbol{\lambda}, \theta) + Z(t, T_E; \theta)\times \\ \sum_{i=1}^{N} N^i\left(\mathcal{L}(T_E, T_M; \lambda^i, \theta) - c\,\mathcal{A}(T_E, T_M; \lambda^i, \theta)\right), \quad (22)$$

with

$$\tilde{\mathcal{L}}(t, T_E; \boldsymbol{\lambda}, \theta) = Z(t, T_E; \theta)\sum_{i=1}^{N} N^i \tilde{\mathcal{L}}^i(t, T_E; \lambda^i, \theta)$$

where $\tilde{\mathcal{L}}^i(t, T; \lambda^i, \theta)$ is defined by setting in Eq. (5) $Z(t, u; \theta) \to 1$ to reflect that the loss amounts occurred

[4] Here for simplicity of exposition we assume that no names in the index have defaulted at valuation time.

before option expiry are settled at T_E. The second term can be computed instead by numerical integration over the distribution of s_{T_E}, Eq. (20),

$$V^{isda}_{\mathrm{iCDS}}(T_E, T_M; F_{T_E}, \theta) = \mathbb{E}_t\Big[N_{tot}\mathcal{A}_{\mathrm{isda}}\left(s_{T_E}, T_E, T_M\right)\left(s_{T_E} - c\right)\Big] \ . \quad (23)$$

The calibration equation (21) defines implicitly the loss adjusted forward spread, F_{T_E}, as a function of its volatility σ_{T_E}, the hazard rates and expected recoveries of the index constituents, and the risk parameters of the discount curve, in short

$$F_{T_E} = F_{T_E}(\boldsymbol{\lambda}; \theta) \ . \quad (24)$$

For a given set of input parameters θ and the calibrated hazard rates for the index constituents $\boldsymbol{\lambda}$, the pricing algorithm consists of the following steps:

Step 1 Calibrate the forward by solving the calibration equation (21). This involves computing Eq. (22) using the hazard rate model and Eq. (23) by numerical integration for each trial value of F_{T_E}.

Step 2 Compute the option value (18) using Eq. (19), *e.g.*, using Gaussian quadrature

$$V_t = Z(t, T_E) \sum_{k=1}^{L} w_k \phi(x_k; F_{T_E}, \theta) P_k \ , \quad (25)$$

where $\phi(x_k; F_{T_E}, \theta)$ is the probability density function of s_{T_E},

$$P_k = \Big(\zeta\Big[N\mathcal{A}_{\mathrm{isda}}\left(x_k, T_E, T_M\right)\left(x_k - c\right) - P_E\Big]\Big)^+ \ , \quad (26)$$

L is the number of quadrature points, and w_k the quadrature weights.

The adjoint of the implicit forward function (24),

$$(\bar{\boldsymbol{\lambda}}, \bar{\theta}) = \overline{F}_{T_E}(\boldsymbol{\lambda}, \theta, \bar{F}_{T_E}) \ , \quad (27)$$

can be computed by means of the implicit function theorem, similarly to what we described for the adjoint of the hazard rate calibration. More explicitly, one first computes the adjoint of the calibration function (21)

$$(\bar{F}_{T_E}, \bar{\boldsymbol{\lambda}}, \bar{\theta}) = \overline{G}_F(F_{T_E}, \boldsymbol{\lambda}, \theta, \bar{G}_F)$$

with

$$\overline{G}_F = \overline{V}^{adj}_{\mathrm{iCDS}}(T_E, T_M; \boldsymbol{\lambda}, \theta, \bar{G}_F) - \overline{V}^{isda}_{\mathrm{iCDS}}(T_E, T_M; F_{T_E}, \theta, \bar{G}_F) \ . \quad (28)$$

Here

$$(\bar{\boldsymbol{\lambda}}, \bar{\theta}) = \overline{V}^{adj}_{\mathrm{Idx}}(T_E, T_M; \boldsymbol{\lambda}, \theta, \bar{V}^{adj}_{\mathrm{iCDS}}) \ ,$$

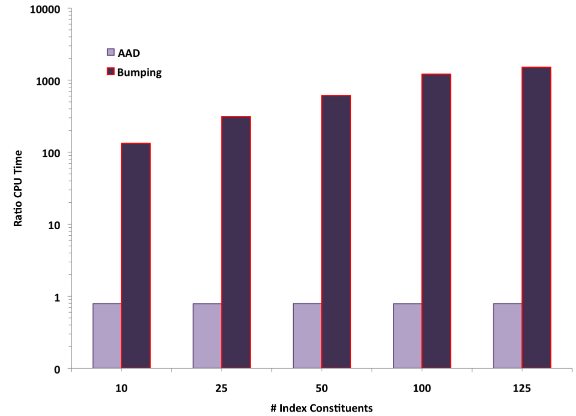

FIG. 2 Cost of computing the sensitivities with respect to the volatility, the constituents' credit spreads and interest rate instruments - relative to the cost of performing a single valuation - as a function of the number of index constituents.

and

$$(\bar{F}_{T_E}, \bar{\theta}) = \overline{V}^{isda}_{\mathrm{iCDS}}(T_E, T_M; F_{T_E}, \theta, \bar{V}^{isda}_{\mathrm{iCDS}}) \ ,$$

are the adjoints of Eqs. (22) and (23), respectively. For $\bar{G}_F = 1$ Eq. (28) gives $\bar{F}_{T_E} = \partial G_F/\partial F_{T_E}$, $\bar{\lambda}^i_j = \partial G_F/\partial \lambda^i_j$, and $\bar{\theta}_k = \partial G_F/\partial \theta_k$, for $i = 1, \ldots, N$, $j = 1, \ldots, M$, $k = 1, \ldots, N_\theta$. Applying the implicit function theorem to the function G_F one finally obtains the outputs of the function in Eq. (27):

$$\bar{\lambda}^i_j = \bar{F}_{T_E} \frac{\partial F_{T_E}}{\partial \lambda^i_j} = -\left(\frac{\partial G_F}{\partial F_{T_E}}\right)^{-1} \frac{\partial G_F}{\partial \lambda^i_j} \ ,$$

$$\bar{\theta}_k = \bar{F}_{T_E} \frac{\partial F_{T_E}}{\partial \theta_k} = -\left(\frac{\partial G_F}{\partial F_{T_E}}\right)^{-1} \frac{\partial G_F}{\partial \theta_k} \ .$$

The adjoint of the pricing algorithm consists therefore of the following steps:

Step $\bar{2}$ Set:

$$\bar{Z} = \bar{V} \ \frac{V_t}{Z(t, T_E; \theta)} \ ,$$

and

$$\bar{\theta} = \bar{Z}(t, T_E; \theta, \bar{Z}) \ ,$$

where $\bar{Z}(t, T; \theta, \bar{Z})$ is the adjoint of the discount function. Then compute the adjoint of the Gaussian quadrature Eqs. (25) and (26), namely set $\bar{F}_{T_E} = 0$, and

$$\bar{\phi}_k = \bar{V} Z(t, T_E; \theta) w_k P_k \ ,$$

$$(\bar{F}_{T_E}, \bar{\theta}) \mathrel{+}= \bar{\phi}(x_k; F_{T_E}, \theta, \bar{\phi}_k) \ ,$$

for $k = 1, \ldots, L$, where $\bar{\phi}(x_i; F_{T_E}, \theta, \bar{\phi}_i)$ is the adjoint of the probability density function. Note that due to the linearity of the adjoint function with respect to the adjoint input, these instructions can be re-expressed in terms of a numerical integration of the form

$$(\bar{F}_{T_E}, \bar{\theta}) = Z(t, T_E; \theta) \sum_{k=1}^{L} w_k \bar{\phi}(x_k; F_{T_E}, \theta, \bar{V}) P_k \ ,$$

i.e., the adjoint of a Gaussian quadrature can be expressed in terms of the quadrature of the adjoint of the integrand.

Step $\bar{1}$ Set $\bar{\boldsymbol{\lambda}} = 0$ and execute the adjoint of the implicit forward function (24),

$$(\bar{\boldsymbol{\lambda}}, \bar{\theta}) \mathrel{+}= \overline{F}_{T_E}(\boldsymbol{\lambda}, \theta, \bar{F}_{T_E}) \ ,$$

computed as described above. Note that the adjoint function in Eq. (23) can also be expressed in terms of a Gaussian quadrature.

Steps $\bar{2}$ and $\bar{1}$ provide the outputs of the adjoint of the pricing step in Eq. (9). Performing the adjoint of the calibration step (10) as previously described generates the full set of sensitivities.

The remarkable computational efficiency achievable for swaptions is illustrated in Fig. 2. Here we plot the cost of computing the sensitivities with respect to the volatility, the constituents' credit spreads and interest rate instruments - relative to the cost of performing a single valuation - for different numbers of index constituents, ranging from 10 (*e.g.*, for iTraxx SOVX Asia Pacific) to 125 (*e.g.*, for iTraxx Europe or CDX.NA.IG). Combining AAD with the implicit function theorem allows the computation of interest rate and (constituents) credit spread risk in 20% less than the cost of computing the option value, resulting in up to 3 orders of magnitude savings (note the logarithmic scale) in computational time.

Conclusions

In conclusion, we have shown how by combining adjoint ideas with the implicit function theorem one can avoid the necessity of repeating multiple times the calibration of the hazard rate curves which, especially for flow products, often represent the bottle neck in the computation of spread and interest rate risk for credit products. This typically results in orders of magnitudes savings in computational time with respect to the standard bump and reval method. In addition, since AAD produces analytical derivatives rather than finite differences approximations the calculation is much more robust numerically than bumping, which is instead often affected by the problem that arbitrary perturbations of credit spreads, recovery rates or of the discount curve may lead to an arbitrageable hazard rate curve.Their computation simply involves computing the sensitivities of the par spread functions (16) with respect to the hazard rate and the model parameters and solving the a linear system, *e.g.*, by Gaussian elimination.

The adjoint of the calibration step can be naturally combined with the adjoint of the pricing step. This allows one to compute the risk of portfolios of credit products faster than computing the portfolio value alone, thus making possible risk management in real time without onerous investments in calculation infrastructure.

Acknowledgments

The authors are grateful to Duncan Noltingk for a careful reading of the manuscript and useful discussions and suggestions. 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

Capriotti, L., 2011, Journal of Computational Finance **3**, 3.
Capriotti, L., and M. Giles, 2010, Risk **23**, 79.
Capriotti, L., and M. Giles, 2012, Risk **25**, 92.
Capriotti, L., M. Peacock, and J. Lee, 2011, Risk **24**, 86.
Christianson, B., 1998, Optimization Methods and Software **9**(4), 307.
Henrard, M. P. A., 2011, OpenGamma Quantitative Research **1**, 1.
ISDA, 2013, Isda cds standard model, URL `http://www.cdsmodel.com/cdsmodel`.
O'Kane, D., 2011, *Modelling Single-name and Multi-name Credit Derivatives*, The Wiley Finance Series (Wiley).
Pedersen, C. M., 2003, Lehman Brothers Quantitative Credit Research , 1.