EFFICIENT GREEK ESTIMATION IN GENERIC MARKET MODELS

MARK JOSHI AND CHAO YANG

Abstract. We first develop an efficient algorithm to compute Deltas of interest rate derivatives for a number of standard market models. The computational complexity of the algorithms is shown to be proportional to the number of rates times the number of factors per step. We then show how to extend the method to efficiently compute Vegas in those market models.

1. Introduction

Market models have become a standard tool for pricing exotic interest rate derivatives. The most popular of these is the LIBOR market model (LMM) also known as BGM, Brace et al. (1997). In the LMM the underlying variables are market-observable LIBORs with discrete tenors. By construction, the model can reproduce the forward yield curve perfectly and is able to justify the use of Black's formula for European interest rate derivatives (IRDs). Thus calibration of the LMM to the market prices of caplets is automatic. Jamshidian (1997) introduced another class of market models, namely, the (co-terminal) swap-rate market model (SMM) where the underlying variables are overlapping swap-rates with discrete tenors. Consequently, drift computations and deriving bond ratios in the SMM are computationally more demanding than the LMM.

The greater simplicity of the LMM has resulted in its receiving greater attention in both practice and academia than swap-rate market models. For a recent overview of research in the LMM, see Brigo and Mercurio (2006) or Brace (2008). One particular strand of research is to investigate the computational complexity of the model: how many operations are required to evolve the model across one step and how many to compute Greeks? Joshi (2003) presented an algorithm that efficently computes the drifts in the LMM with complexity proportional to the number of rates (n) times the number of factors driving the LMM (F) per step. Since the other computational operations for each step are straight-forward, this resulted in $O(nF)$ operations per step. Giles and Glasserman (2006) developed an adjoint method that computes Deltas and Vegas (sensitivities with respect to the underlying rates and volatility parameters, respectively) with order $O(nF)$ per step.

Date: September 9, 2009.

1991 *Mathematics Subject Classification.* 91B24, 60G40, 60G44. JEL Classification: G13.

Key words and phrases. adjoint method, Delta, Vega, computational order, market model, Monte Carlo simulation.

Less attention has been given to such questions for the swap-rate market model. In fact, there are now many swap-rate market models since one is not restricted to using co-terminal rates. Galluccio et al. (2007) used graph theory to identify three main classes of generic market models, namely the co-terminal, co-initial and co-sliding models, and they introduced a novel calibration technique to allow simultaneous calibration to caplets and swaption prices. This followed on from Galluccio and Hunter (2004) where the co-initial SMM (ciSMM) was investigated and an algorithm to compute the drifts with order $O(n^2)$ per step was derived. Pietersz and van Regenmortel (2005) studied constant maturity SMM (cmSMM) and other generic market models. They developed generic algorithms to evaluate drifts for various market models. The order to compute drifts in the co-terminal SMM (ctSMM) is $O(nF)$, and the order to compute drifts approximately in the cmSMM is $O(nF)$. In addition, their method to derive bond ratios has order $O(n^3)$.

The fact that the rates overlap in the swap-rate market models mean that the analysis is considerably more complicated than for the LMM. However, progress has been made. Joshi and Liesch (2007) introduced efficient algorithms for implementing generic market models with order $O(nF)$ per step for a wide class of market models. Since the computational order of all the operations for each step has been shown to be $O(nF)$, it should be possible to apply the adjoint method as in Giles and Glasserman (2006) to estimate Deltas and Vegas with this order of computations per step. In the case of the co-terminal swap-rate market model, Joshi and Yang (2009) established that this was true for Deltas.

In this paper, we develop efficient algorithms that compute both Deltas and Vegas in the three principal SMMs: ctSMM, cmSMM and ciSMM. The essential idea is that we decompose the one-step computation into a number of simpler vector operations. Each of these can be differentiated straightforwardly and then used for multiplication of the adjoint. A slight modification of the method of computing Deltas in generic market models will compute market Vegas with order $O(nF)$ per step.

In section 2, we review the efficient implementation of cmSMM and ciSMM. In section 3, we briefly discuss the efficient adjoint method in Joshi and Yang (2009). We then show how to estimate Deltas in the cmSMM and ciSMM in sections 4 and 5. In section 6, we extend the method to compute Vegas in generic market models. We run some timing tests to confirm the computational complexity of Delta and Vega estimation is $O(nF)$ per step in section 7. We conclude in section 8.

2. Generic market models

2.1. **Notations.** The tenor structure is a finite set of dates

$$0 = T_{-1} < T_0 < T_1 < \cdots < T_{n-1} < T_n,$$

where $\{T_i\}_{i=0}^n$ are spaced by a set of real numbers $\tau_{i-1} = T_i - T_{i-1}$, for all i. We let P_i denote the price of the zero-coupon bond maturing at time T_i. We let

$\mathrm{SR}_{i,j}$ denote the swap-rates associated to times $T_i, \ldots, T_j$. We let $A_{i,j}$ be the value of the annuity of $\mathrm{SR}_{i,j}$, then

$$\mathrm{SR}_{i,j} = \frac{P_i - P_j}{A_{i,j}}, \tag{2.1}$$

where $A_{i,j} = \sum_{k=i+1}^{j} \tau_{k-1} P_k$.

2.2. **Model set-up.** The n rates will be driven by F Brownian motions and will be evolved to each of the tenor dates step by step. We assume a piecewise constant volatility structure and therefore assign a pseudo-square root, $A = \{a_{i,k}\}$, of the covariance matrix, C, for each step to determine the evolution. We can therefore write across each step

$$\mathrm{dSR}_{i,j} = \mu_i^{(m)} \mathrm{d}t + \mathrm{SR}_{i,j} \sum_{k=1}^{F} a_{i,k} \mathrm{d}Z_k \tag{2.2}$$

where $\mu_i^{(m)}$ is the drift of $\mathrm{SR}_{i,j}$ under the measure associated with the bond P_m, and $\{Z_k\}$ is a vector of independent Brownian motions. We note that this formulation is general: if we take $F = n$, then any set of correlated Brownian motions can be decomposed as a linear combination of such independent Brownian motions.

2.2.1. *The cross variation derivative.* The cross variation derivative for two Itô processes

$$\begin{aligned} \mathrm{d}X_t &= \mu_X\big(X_t, Y_t, t\big)\mathrm{d}t + \sigma_X\big(X_t, Y_t, t\big)\mathrm{d}W_t^X, \\ \mathrm{d}Y_t &= \mu_Y\big(Y_t, Y_t, t\big)\mathrm{d}t + \sigma_Y\big(X_t, Y_t, t\big)\mathrm{d}W_t^Y \end{aligned}$$

is defined to be the coefficient of $\mathrm{d}t$ in $\mathrm{d}X_t \mathrm{d}Y_t$. If $\mathrm{d}W_t^X \mathrm{d}W_t^Y = \rho \mathrm{d}t$ then

$$\big\langle X_t, Y_t \big\rangle = \rho \sigma_X\big(X_t, Y_t, t\big)\sigma_Y\big(X_t, Y_t, t\big). \tag{2.3}$$

The cross variation derivative is linear in each term

$$\big\langle X_t, Y_t + Z_t \big\rangle = \big\langle X_t, Y_t \big\rangle + \big\langle X_t, Z_t \big\rangle,$$

and satisfies the following product rule

$$\big\langle X_t, Y_t Z_t \big\rangle = Z_t \big\langle X_t, Y_t \big\rangle + Y_t \big\langle X_t, Z_t \big\rangle,$$

where X_t, Y_t and Z_t are Itô processes. For detailed discussion of the cross variation derivative, we refer the reader to Joshi and Liesch (2007).

2.2.2. *General drift formulae.* The drifts of the rates are determined by no-arbitrage considerations to ensure that the ratio of every bond price to the *numeraire* bond P_m is a martingale. Since $\mathrm{SR}_{i,j} A_{i,j}$, $A_{i,j}$ are tradables, then $\mathrm{SR}_{i,j} A_{i,j} / P_m$

and $A_{i,j}/P_m$ are martingales under the the measure associated with bond P_m. Therefore the following stochastic differential equation

$$\begin{aligned}\mathrm{d}\Big(\frac{\mathrm{SR}_{i,j}A_{i,j}}{P_m}\Big) &= \frac{A_{i,j}}{P_m}\mathrm{dSR}_{i,j} + \mathrm{SR}_{i,j}\mathrm{d}\Big(\frac{A_{i,j}}{P_m}\Big) + \Big\langle \mathrm{SR}_{i,j}, \frac{A_{i,j}}{P_m}\Big\rangle \mathrm{d}t \\ &= \Big(\mu_i^{(m)}\frac{A_{i,j}}{P_m} + \Big\langle \mathrm{SR}_{i,j}, \frac{A_{i,j}}{P_m}\Big\rangle\Big)\mathrm{d}t \\ &\quad + \frac{A_{i,j}}{P_m}\mathrm{SR}_{i,j}\sum_{k=1}^{F} a_{i,k}\mathrm{d}Z_k + \mathrm{SR}_{i,j}\mathrm{d}\Big(\frac{A_{i,j}}{P_m}\Big) \end{aligned} \tag{2.4}$$

has no drift term, then

$$\begin{aligned}\mu_i^{(m)} &= -\frac{P_m}{A_{i,j}}\Big\langle \mathrm{SR}_{i,j}, \frac{A_{i,j}}{P_m}\Big\rangle \\ &= -\frac{P_m}{A_{i,j}}\mathrm{SR}_{i,j}\sum_{k=1}^{F} a_{i,k}\Big\langle Z_k, \frac{A_{i,j}}{P_m}\Big\rangle, \end{aligned} \tag{2.5}$$

where $\{Z_k\}$ is a vector of independent Brownian motions. In (2.5), we have a general formula for drifts of swap-rates in a market model.

2.3. **Derivatives Pricing.** An IRD will pay a stream of cash-flows until the product terminates, cancels or is triggered. Each cash-flow may be a function of the entire yield curve at the time it occurs and/or previous times. In practical terms, this means that each cash-flow is a function of the swap-rates underlying the market model and this function may also incorporate information from previous times.

With out loss of generality, we shall concentrate on the case where there is a single cash-flow which is a function, f, of the prevailing rates at the maturity of the IRD. However, if a product pays a stream of cash-flows, then these cash-flows will be aggregated along each path of the simulation. We make the (very mild) assumption that the cash-flow can be computed as a function of the yield curve at the time is determined with order n computations.

2.4. **Constant maturity swap-rate market model.** The cmSMM is characterized by the set of swap-rates

$$\mathrm{SR}_i^r = \mathrm{SR}_{i,i+r} = \frac{P_i - P_{i+r}}{A_i^r}, \quad i = 0, 1, \dots, n-1, \tag{2.6}$$

where r is a fixed integer number and $A_i^r = \sum_{k=i+1}^{i+r} \tau_{k-1} P_k$ is the annuity of $\mathrm{SR}_{i,i+r}$. We make the convention that if $i+r \geqslant n$ then we set $i+r = n$, thus the last r rates will be the co-terminal swap-rates. If we set $r = 1$ then the cmSMM becomes the LMM, and if we set $r = n$ then the cmSMM becomes the ctSMM.

2.4.1. *Recursive formula for the cross variation derivatives.* We choose P_n as the numeraire bond. Joshi and Liesch (2007) showed that the recursive formula of the cross variation derivatives in the co-terminal case is given by

$$\Big\langle Z_k, \frac{A_i}{P_n} \Big\rangle = \tau_i a_{i+1,k} \mathrm{SR}_{i+1} \frac{A_{i+1}}{P_n} + (1 + \tau_i \mathrm{SR}_{i+1}) \Big\langle Z_k, \frac{A_{i+1}}{P_n} \Big\rangle. \tag{2.7}$$

We can use this formula to compute the cross variation derivatives for the last r rates. We now derive the recursive formula for the cross variation derivatives of the first $n-r$ rates. For $i + r < n$,

$$A_i^r = A_{i+1}^r + \tau_i P_{i+1} - \tau_{i+r} P_{i+r+1},$$

therefore

$$\begin{aligned} \Big\langle Z_k, \frac{A_i^r}{P_n} \Big\rangle &= \Big\langle Z_k, \frac{A_{i+1}^r + \tau_i P_{i+1} - \tau_{i+r} P_{i+r+1}}{P_n} \Big\rangle \\ &= \Big\langle Z_k, \frac{A_{i+1}^r}{P_n} \Big\rangle + \tau_i \Big\langle Z_k, \frac{P_{i+1}}{P_n} \Big\rangle - \tau_{i+r} \Big\langle Z_k, \frac{P_{i+r+1}}{P_n} \Big\rangle. \end{aligned} \tag{2.8}$$

From (2.6)

$$\frac{P_i}{P_n} = \frac{P_{i+r}}{P_n} + \mathrm{SR}_i^r \frac{A_i^r}{P_n},$$

then

$$\Big\langle Z_k, \frac{P_{i+1}}{P_n} \Big\rangle = \Big\langle Z_k, \frac{P_{i+r+1}}{P_n} + \mathrm{SR}_{i+1}^r \frac{A_{i+1}^r}{P_n} \Big\rangle.$$

Using linearity and the product rule of cross variation derivatives, we have

$$\Big\langle Z_k, \frac{P_{i+1}}{P_n} \Big\rangle = \Big\langle Z_k, \frac{P_{i+r+1}}{P_n} \Big\rangle + a_{i+1,k} \mathrm{SR}_{i+1}^r \frac{A_{i+1}^r}{P_n} + \mathrm{SR}_{i+1}^r \Big\langle Z_k, \frac{A_{i+1}^r}{P_n} \Big\rangle. \tag{2.9}$$

Thus the recursive formula for the first $n - r$ rates is given by

$$\begin{aligned} \Big\langle Z_k, \frac{A_i^r}{P_n} \Big\rangle = \tau_i a_{i+1,k} \mathrm{SR}_{i+1}^r \frac{A_{i+1}^r}{P_n} + (1 + \tau_i \mathrm{SR}_{i+1}^r) \Big\langle Z_k, \frac{A_{i+1}^r}{P_n} \Big\rangle \\ + (\tau_i - \tau_{i+r}) \Big\langle Z_k, \frac{P_{i+r+1}}{P_n} \Big\rangle. \end{aligned} \tag{2.10}$$

If $\tau_i = \tau$, for $i = 0, 1, \dots, n-1$, then we can see that the final term in (2.10) will become zero so that the recursive formula can be simplified to

$$\Big\langle Z_k, \frac{A_i^r}{P_n} \Big\rangle = \tau a_{i+1,k} \mathrm{SR}_{i+1}^r \frac{A_{i+1}^r}{P_n} + (1 + \tau \mathrm{SR}_{i+1}^r) \Big\langle Z_k, \frac{A_{i+1}^r}{P_n} \Big\rangle, \tag{2.11}$$

with $\langle Z_k, A_{n-1}^r / P_n \rangle = 0$, for all k.

2.4.2. *Bond and annuity ratios.* Let $\{\widetilde{A}_i^r\}_{i=0}^{n-1}$ be a sequence of annuity ratios where $\widetilde{A}_i^r = A_i^r / P_n$, and let $\{\widetilde{P}_i\}_{i=0}^{n}$ be a sequence of bond ratios where $\widetilde{P}_i = P_i / P_n$. Then from (2.6):

(1) The last r terms of the sequence, $\{\widetilde{A}_i^r\}_{i=n-r}^{n-1}$, are given by setting

$$\begin{cases} \widetilde{A}_{n-1}^r = \tau_{n-1}, \\ \widetilde{A}_i^r = \widetilde{A}_{i+1}^r + \tau_i \widetilde{P}_{i+1}, \end{cases} \tag{2.12}$$

with

$$\widetilde{P}_i = 1 + \widetilde{A}_i^r \mathrm{SR}_i^r, \quad i = n-r, \ldots, n-1. \tag{2.13}$$

(2) The first $n-r$ terms of the sequence, $\{\widetilde{A}_i^r\}_{i=0}^{n-r-1}$, are given by setting

$$\widetilde{A}_i^r = \widetilde{A}_{i+1}^r + \tau_i \widetilde{P}_{i+1} - \tau_{i+r} \widetilde{P}_{i+r+1}, \tag{2.14}$$

with

$$\widetilde{P}_i = \widetilde{P}_{i+r} + \widetilde{A}_i^r \mathrm{SR}_i^r, \quad i = 0, 1, \ldots, n-r-1. \tag{2.15}$$

Under this algorithm, the computational order of deducing the bond and annuity ratios is $O(n)$ per step.

2.5. **Co-initial swap-rate market model.** The ciSMM is characterized by the set of swap-rates

$$\mathrm{SR}_{0,i} = \frac{P_0 - P_i}{A_{0,i}}, \quad i = 1, 2, \ldots, n, \tag{2.16}$$

where $A_{0,i} = \sum_{k=1}^{i} \tau_{k-1} P_k$ is the annuity of $\mathrm{SR}_{0,i}$.

2.5.1. *Recursive formula for the cross variation derivatives.* We choose P_0 as the numeraire bond. Since the annuity satisfies

$$A_{0,i} = A_{0,i-1} + \tau_{i-1} P_i,$$

then it follows that

$$\begin{aligned} \Big\langle Z_k, \frac{A_{0,i}}{P_0} \Big\rangle &= \Big\langle Z_k, \frac{A_{0,i-1} + \tau_{i-1} P_i}{P_0} \Big\rangle \\ &= \Big\langle Z_k, \frac{A_{0,i-1}}{P_0} \Big\rangle + \tau_{i-1} \Big\langle Z_k, \frac{P_i}{P_0} \Big\rangle. \end{aligned}$$

Joshi and Liesch (2007) showed that

$$\begin{aligned} \Big\langle Z_k, \frac{P_i}{P_0} \Big\rangle = \frac{-1}{1 + \tau_{i-1} \mathrm{SR}_{0,i}} \Big[a_{i,k} \mathrm{SR}_{0,i} \frac{A_{0,i-1}}{P_0} + \mathrm{SR}_{0,i} \Big\langle Z_k, \frac{A_{0,i-1}}{P_0} \Big\rangle \Big] \\ - \tau_{i-1} a_{i,k} \mathrm{SR}_{0,i} \frac{1 - \mathrm{SR}_{0,i} A_{0,i-1}/P_0}{(1 + \tau_{i-1} \mathrm{SR}_{0,i})^2}. \end{aligned}$$

Rearranging the terms gives the following recursive formula

$$\begin{aligned} \Big\langle Z_k, \frac{A_{0,i}}{P_0} \Big\rangle &= \frac{1}{1 + \tau_{i-1} \mathrm{SR}_{0,i}} \Big\langle Z_k, \frac{A_{0,i-1}}{P_0} \Big\rangle - \frac{\tau_{i-1} a_{i,k} \mathrm{SR}_{0,i}}{1 + \tau_{i-1} \mathrm{SR}_{0,i}} \Big(\frac{A_{0,i-1}}{P_0} + \tau_{i-1} \frac{P_i}{P_0} \Big) \\ &= \frac{1}{1 + \tau_{i-1} \mathrm{SR}_{0,i}} \Big[\Big\langle Z_k, \frac{A_{0,i-1}}{P_0} \Big\rangle - \tau_{i-1} a_{i,k} \mathrm{SR}_{0,i} \frac{A_{0,i}}{P_0} \Big], \end{aligned} \tag{2.17}$$

with $\langle Z_k, A_{0,0}/P_0 \rangle = 0$, for all k.

2.5.2. *Bond and annuity ratios.* Let $\{\widetilde{A}_i\}_{i=1}^n$ be a sequence of annuity ratios where $\widetilde{A}_i = A_{0,i}/P_0$, and let $\{\widetilde{P}_i\}_{i=0}^n$ be a sequence of bond ratios where $\widetilde{P}_i = P_i/P_0$. Then from (2.16):

$$\begin{cases} \widetilde{A}_1 = \tau_0 \widetilde{P}_1, \\ \widetilde{A}_i = \widetilde{A}_{i-1} + \tau_{i-1}\widetilde{P}_i, \quad i = 2, 3, \ldots, n, \end{cases} \tag{2.18}$$

with

$$\begin{cases} \widetilde{P}_1 = 1/\big(1 + \tau_0 \mathrm{SR}_{0,1}\big), \\ \widetilde{P}_i = \big(1 - \mathrm{SR}_{0,i}\widetilde{A}_{i-1}\big)/\big(1 + \tau_{i-1}\mathrm{SR}_{0,i}\big), \quad i = 2, 3, \ldots, n. \end{cases} \tag{2.19}$$

Under this algorithm, the computational order of deducing the bond and annuity ratios is $O(n)$ per step.

3. Delta estimation in swap-rate market models

3.1. **Set-up.** Assume that the market model has n underlying rates with the tenor structure $\{T_i\}_{i=0}^n$, and each rate is driven by F factors. Let $f(T_m)$ denote the pay-off of an IRD maturing at time T_m, $m \leqslant n$. The bond P_N is chosen as the numeraire, so that the discounted pay-off is given by

$$g = P_N(0)\frac{f(T_m)}{P_N(T_m)}. \tag{3.1}$$

The vector of Deltas of the IRD is given by

$$\underline{\Delta} = \frac{\partial g}{\partial \mathbf{SR}(0)} = \frac{\partial P_N(0)}{\partial \mathbf{SR}(0)}\mathbb{E}\big[\widetilde{f}(T_m)\big] + P_N(0)\frac{\partial}{\partial \mathbf{SR}(0)}\mathbb{E}\big[\widetilde{f}(T_m)\big], \tag{3.2}$$

where $\widetilde{f}(T_m) = f(T_m)/P_N(T_m)$. If f is Lipschitz-continuous, then the differentiation operator and the expectation operator in (3.2) can be interchanged

$$\underline{\Delta} = \frac{\partial P_N(0)}{\partial \mathbf{SR}(0)}\mathbb{E}\big[\widetilde{f}(T_m)\big] + P_N(0)\mathbb{E}\Big[\frac{\partial \widetilde{f}(T_m)}{\partial \mathbf{SR}(0)}\Big], \tag{3.3}$$

The first gradient vector in (3.3) is easy to compute, and we will compute the second gradient vector using the pathwise-adjoint method.

If we consider the following mappings

$$\mathbf{SR}(0) \xrightarrow{\mathbf{F}_0} \mathbf{SR}(T_0) \xrightarrow{\mathbf{F}_1} \cdots \xrightarrow{\mathbf{F}_m} \mathbf{SR}(T_m) \xrightarrow{\mathbf{F}} \widetilde{f}(T_m), \tag{3.4}$$

where $\mathbf{F}_i$ and $\mathbf{F}$ are vector functions, then it follows from the chain rule that

$$\frac{\partial \widetilde{f}(T_m)}{\partial \mathbf{SR}(0)} = \mathbf{F}'\mathbf{F}'_m\mathbf{F}'_{m-1}\cdots\mathbf{F}'_0, \tag{3.5}$$

where $\mathbf{F}'_i$ are Jacobian matrices and $\mathbf{F}$ is a gradient vector. If we define an adjoint relation as follows

$$\begin{cases} \mathbf{V}(T_m) = \mathbf{F}', \\ \mathbf{V}(T_{k-1}) = \mathbf{V}(T_k)\mathbf{F}'_k, \end{cases} \tag{3.6}$$

then the second gradient vector in (3.3) is given by setting

$$\frac{\partial \widetilde{f}(T_m)}{\partial \mathbf{SR}(0)} = \mathbf{V}(0). \tag{3.7}$$

The elements of $\mathbf{V}(T_{k-1})$ can be computed from $\mathbf{V}(T_k)$ via

$$\begin{aligned}\mathbf{V}_i(T_{k-1}) &= \sum_{j \geqslant i} \mathbf{V}_j(T_k) \frac{\partial \mathrm{SR}_j(T_k)}{\partial \mathrm{SR}_i(T_{k-1})} \\ &= \mathbf{V}_i(T_k) \frac{\mathrm{SR}_i(T_k)}{\mathrm{SR}_i(T_{k-1})} + \sum_{j \geqslant i} \mathbf{V}_j(T_k) \mathrm{SR}_j(T_k) \frac{\partial \mu_j^{(N)}\big(\mathbf{SR}(T_{k-1})\big)}{\partial \mathrm{SR}_i(T_{k-1})}.\end{aligned}$$

If we proceed naïvely, the computational complexity of the adjoint method will be $O(n^2)$ per step since we are multiplying a vector, $\mathbf{V}(T_k)$, by a Jacobian matrix, $\mathbf{F}'_k$. However, it is possible to reduce the complexity to $O(nF)$ per step. Thus if n is large and F is small, the reduction in computational order results in substantial time-saving.

3.2. **Efficient adjoint method.** Joshi and Yang (2009) decomposed the vector functions, $\mathbf{F}_k$, into a number of sub-mappings, $\mathbf{F}_{k,i}$, to an extent such that all vector operations consist of simple functions and the corresponding Jacobian matrices become sparse matrices. We then identify the non-zero elements of the Jacobian matrices $\mathbf{F}'_{k,i}$, and only carry out the multiplications of the corresponding non-zero entries in the operation

$$\mathbf{v} = \mathbf{w}\mathbf{F}'_{k,i}.$$

Thus the computational order of the adjoint relations in (3.6) can be reduced to $O(nF)$ per step.

In this paper, we consider the following sub-mappings for the various market models:

$$\mathbf{SR}(T_{k-1}) \xrightarrow{\mathbf{F}_{k,0}} \begin{pmatrix} \mathbf{SR}(T_{k-1}) \\ \widetilde{\mathbf{A}}(T_{k-1}) \end{pmatrix} \xrightarrow{\mathbf{F}_{k,1}} \begin{pmatrix} \mathbf{SR}(T_{k-1}) \\ \underline{\mu}^{(N)}(T_{k-1}) \end{pmatrix} \xrightarrow{\mathbf{F}_{k,2}} \mathbf{SR}(T_k), \tag{3.8}$$

where the sub-mappings satisfy

$$\mathbf{F}_k = \mathbf{F}_{k,0} \circ \mathbf{F}_{k,1} \circ \mathbf{F}_{k,2}, \quad k = 0, 1, \ldots, m.$$

Therefore the adjoint relation in (3.6) is equivalent to

$$\mathbf{V}(T_{k-1}) = \mathbf{V}(T_k)\mathbf{F}'_{k,2}\mathbf{F}'_{k,1}\mathbf{F}'_{k,0}, \quad k = 0, 1, \ldots, m. \tag{3.9}$$

We illustrate the efficient adjoint method under the cmSMM and ciSMM in the following two sections.

4. Delta estimation in the constant maturity swap-rate market model

For concreteness and readability, we will only show the computational order of the operations

$$\mathbf{V}(0) = \mathbf{V}(T_0)\mathbf{F}'_{0,2}\mathbf{F}'_{0,1}\mathbf{F}'_{0,0} = \mathbf{V}(T_0)\mathbf{F}'_0. \tag{4.1}$$

i.e. the adjoint operations for the first step. The adjoint operations when $k = 1, \ldots, m$, are essentially the same: the only real difference is that certain rates will have fixed and therefore will have zero volatility during these steps.

4.1. Computational order of $\mathbf{wF}'_{0,0}$.

We divide the mapping

$$\mathbf{SR}(0) \xrightarrow{\mathbf{F}_{0,0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix}$$

into the following sub-mappings

$$\begin{aligned}
\mathbf{SR}(0) &\xrightarrow{\mathbf{G}_{0,0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{n-1}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=n} \end{pmatrix} \xrightarrow{\mathbf{G}_{0,1}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{n-1}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=n-1} \end{pmatrix} \\
&\xrightarrow{\mathbf{G}_{0,2}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{n-2}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=n-1} \end{pmatrix} \xrightarrow{\mathbf{G}_{0,3}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{n-2}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=n-2} \end{pmatrix} \\
&\xrightarrow{\mathbf{G}_{0,4}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{n-3}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=n-2} \end{pmatrix} \cdots \xrightarrow{\mathbf{G}_{0,2n-2}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{A}^r_{0}(0) \\ \{\widetilde{\mathbf{P}}(0)\}^n_{i=1} \end{pmatrix} \\
&\xrightarrow{\mathbf{G}_{0,2n-1}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{P}}(0) \end{pmatrix} \xrightarrow{\mathbf{G}_{0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix}
\end{aligned} \tag{4.2}$$

The even-numbered mappings $\mathbf{G}_{0,j}$ update the annuity ratios using the given set of inputs, and the odd-numbered mappings $\mathbf{G}_{0,j}$ computes a new bond ratio using the given set of inputs.

4.1.1. *Jacobian matrix of $\mathbf{G}_{0,j}$ where j is even.*

From (2.12), we can see that $\widetilde{A}^r_i$, when $i \geqslant n-r$, only depends on $\widetilde{A}^r_{i+1}$ and $\widetilde{P}_{i+1}$, then

$$\frac{\partial \widetilde{A}^r_i}{\partial \widetilde{A}^r_{i+1}} = 1, \quad \frac{\partial \widetilde{A}^r_i}{\partial \widetilde{P}_{i+1}} = \tau_i.$$

The Jacobian matrix $\mathbf{G}'_{0,j}$ has 1s on the diagonal, one entry equal to τ on the last column, and 0s elsewhere. Therefore the operation $\mathbf{v} = \mathbf{wG}'_{0,j}$ has one computation.

From (2.14), we can see that $\widetilde{A}^r_i$, when $i < n-r$, depends on $\widetilde{A}^r_{i+1}$, $\widetilde{P}_{i+1}$ and $\widetilde{P}_{i+r+1}$, then

$$\frac{\partial \widetilde{A}^r_i}{\partial \widetilde{A}^r_{i+1}} = 1, \quad \frac{\partial \widetilde{A}^r_i}{\partial \widetilde{P}_{i+1}} = \tau_i, \quad \frac{\partial \widetilde{A}^r_i}{\partial \widetilde{P}_{i+r+1}} = -\tau_{i+r}.$$

The Jacobian matrix $\mathbf{G}'_{0,j}$ has 1s on the diagonal, one entry equal to τ on the last column, one entry equal to $-\tau$ and 0s elsewhere. Therefore the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,j}$ has two computations.

4.1.2. *Jacobian matrix of* $\mathbf{G}_{0,j}$ *where* j *is odd.* From (2.13), we can see that $\widetilde{P}_i$ when $i \geqslant n - r$ only depends on $\widetilde{A}_i^r$ and SR_i^r, then

$$\frac{\partial \widetilde{P}_i}{\partial \widetilde{A}_i^r} = \mathrm{SR}_i^r, \quad \frac{\partial \widetilde{P}_i}{\partial \mathrm{SR}_i^r} = \widetilde{A}_i^r.$$

The Jacobian matrix $\mathbf{G}'_{0,j}$ has 1s on the diagonal, two entries equal to SR_i and α_i on the last row. Therefore the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,j}$ has two computations.

From (2.15), we can see that $\widetilde{P}_i$ when $i < n - r$ only depends on $\widetilde{A}_i^r$, $\widetilde{P}_{i+r}$ and SR_i^r, then

$$\frac{\partial \widetilde{P}_i}{\partial \widetilde{A}_i^r} = \mathrm{SR}_i^r, \quad \frac{\partial \widetilde{P}_i}{\partial \widetilde{P}_{i+r}} = 1, \quad \frac{\partial \widetilde{P}_i}{\partial \mathrm{SR}_i^r} = \widetilde{A}_i^r.$$

The Jacobian matrix $\mathbf{G}'_{0,j}$ has 1s on the diagonal, three entries equal to SR_i^r, 1 and $\widetilde{A}_i^r$ on the last row. Therefore the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,j}$ has three computations.

4.1.3. *Jacobian matrix of* $\mathbf{G}_0$. For $i \geqslant n - r$,

$$\frac{\partial \widetilde{A}_i^r}{\partial \widetilde{P}_j} = \tau_{j-1}, \quad j > i.$$

However, for $i < n - r$,

$$\frac{\partial \widetilde{A}_i^r}{\partial \widetilde{P}_j} = \tau_{j-1}, \quad i + r \geqslant j > i.$$

Fortunately, the order of the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_0$ can be reduced to $O(n)$. We define a sum variable, and carry out the following algorithm:

- Set $\mathbf{v}_j = \mathbf{w}_j$ for $j = 0, \ldots, n-1$.
- Set sum equal to $\tau_0 \mathbf{w}_n$ in loop 0.
- Set $sum = sum + \tau_j \mathbf{w}_{n+j}$ in loop j for $j = 1, \ldots, r-1$.
- Set $sum = sum + \tau_j(\mathbf{w}_{n+j} - \mathbf{w}_{n+j-r})$ in loop j for $j = r, \ldots, n-1$.

If we add the updated sum to $\mathbf{v}_{2n-j}$ in the jth loop, then we have executed the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_0$. Hence, the order of the operation is $O(n)$.

4.1.4. *Total computational order of* $\mathbf{w}\mathbf{F}'_{0,0}$. Since the number of sub-mappings $\mathbf{G}_{0,j}$ depends on n, the computational order of the operation

$$\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,2n-1}\mathbf{G}'_{0,2n-2}\cdots\mathbf{G}'_{0,1}$$

is $O(n)$ since each operation has a constant computational order. We ignore the Jacobian matrix $\mathbf{G}'_{0,0}$ as $\widetilde{A}_{n-1}^r$ is a constant so that $\mathbf{G}'_{0,0}$ is equivalent to an identity matrix. Hence the operation

$$\mathbf{w}\mathbf{G}'_0\mathbf{G}'_{0,2n-1}\mathbf{G}'_{0,2n-2}\cdots\mathbf{G}'_{0,1} = \mathbf{w}\mathbf{F}'_{0,0}$$

has order $O(n)$.

4.2. **Computational order of $\mathbf{wF}'_{0,1}$.** We divide the mapping

$$\begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\end{pmatrix} \xrightarrow{\mathbf{F}_{0,1}} \begin{pmatrix}\mathbf{SR}(0)\\ \underline{\mu}^{(n)}(0)\end{pmatrix}$$

into the following sub-mappings

$$\begin{aligned}\begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\end{pmatrix} &\xrightarrow{\mathbf{H}_{0,0}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k, \widetilde{P}_i^r\rangle\big\}_{i=n-1}^{n-1}\end{pmatrix} \xrightarrow{\mathbf{H}_{0,1}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k, \widetilde{P}_i^r\rangle\big\}_{i=n-1}^{n-1}\\ \big\{\langle Z_k, \widetilde{A}_i^r\rangle\big\}_{i=n-2}^{n-2}\end{pmatrix}\\ &\xrightarrow{\mathbf{H}_{0,2}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k, \widetilde{P}_i^r\rangle\big\}_{i=n-2}^{n-1}\\ \big\{\langle Z_k, \widetilde{A}_i^r\rangle\big\}_{i=n-2}^{n-2}\end{pmatrix} \xrightarrow{\mathbf{H}_{0,3}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k, \widetilde{P}_i^r\rangle\big\}_{i=n-2}^{n-1}\\ \big\{\langle Z_k, \widetilde{A}_i^r\rangle\big\}_{i=n-2}^{n-3}\end{pmatrix}\\ \cdots &\xrightarrow{\mathbf{H}_{0,2(n-2)+1}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k, \widetilde{P}_i^r\rangle\big\}_{i=1}^{n-1}\\ \big\{\langle Z_k, \widetilde{A}_i^r\rangle\big\}_{i=0}^{n-2}\end{pmatrix} \xrightarrow{\mathbf{H}_{0}} \begin{pmatrix}\mathbf{SR}(0)\\ \underline{\mu}^{(n)}(0)\end{pmatrix}\end{aligned} \tag{4.3}$$

The even-numbered mappings $\mathbf{H}_{0,j}$ update the cross variation derivatives $\langle Z_k, \widetilde{P}_i^r\rangle$, and the odd-numbered mappings $\mathbf{H}_{0,j}$ update the cross variation derivatives $\langle Z_k, \widetilde{A}_i^r\rangle$.

4.2.1. *Jacobian matrix of $\mathbf{H}_{0,j}$ where j is even.* We rewrite (2.9) in terms of annuity and bond ratios. For $j \geqslant n - r$,

$$\big\langle Z_k, \widetilde{P}_j\big\rangle = \mathrm{SR}_j^r\Big(a_{j,k}\widetilde{A}_j^r + \big\langle Z_k, \widetilde{A}_j^r\big\rangle\Big),$$

we can see that $\big\langle Z_k, \widetilde{P}_j\big\rangle$ depend on SR_j^r, $\widetilde{A}_j^r$ and $\big\langle Z_k, \widetilde{A}_j^r\big\rangle$ so that we have the following non-zero partial derivatives

$$\begin{aligned}\frac{\partial}{\partial \mathrm{SR}_j^r}\big\langle Z_k, \widetilde{P}_j\big\rangle &= a_{j,k}\widetilde{A}_j^r + \big\langle Z_k, \widetilde{A}_j^r\big\rangle,\\ \frac{\partial}{\partial \widetilde{A}_j^r}\big\langle Z_k, \widetilde{P}_j\big\rangle &= a_{j,k}\mathrm{SR}_j^r,\\ \frac{\partial}{\partial \big\langle Z_k, \widetilde{A}_j^r\big\rangle}\big\langle Z_k, \widetilde{P}_j\big\rangle &= \mathrm{SR}_j^r,\end{aligned}$$

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

For $j < n - r$,

$$\big\langle Z_k, \widetilde{P}_j\big\rangle = \mathrm{SR}_j^r\Big(a_{j,k}\widetilde{A}_j^r + \big\langle Z_k, \widetilde{A}_i^r\big\rangle\Big) + \big\langle Z_k, \widetilde{P}_{j+r}\big\rangle.$$

we can see that $\langle Z_k, \widetilde{P}_j \rangle$ depend on SR^r_j, $\widetilde{A}^r_j$, $\langle Z_k, \widetilde{A}^r_i \rangle$ and $\langle Z_k, \widetilde{P}_{j+r} \rangle$. The first three partial derivatives are the same as those when $j \geqslant n - r$, the last partial derivative is given by

$$\frac{\partial}{\partial \langle Z_k, \widetilde{P}_{j+r} \rangle} \langle Z_k, \widetilde{P}_j \rangle = 1.$$

4.2.2. *Jacobian matrix of* $\mathbf{H}_{0,j}$ *where* j *is odd.* We rewrite the recursive formula of the cross variation derivatives in terms of annuity ratios. For $j \geqslant n - r$,

$$\langle Z_k, \widetilde{A}^r_j \rangle = \tau_j a_{j+1,k} \mathrm{SR}^r_{j+1} \widetilde{A}^r_{j+1} + (1 + \tau_j \mathrm{SR}^r_{j+1}) \langle Z_k, \widetilde{A}^r_{j+1} \rangle,$$

we can see that $\langle Z_k, \widetilde{A}^r_j \rangle$ depend on SR^r_{j+1}, $\widetilde{A}^r_{j+1}$ and $\langle Z_k, \widetilde{A}^r_{j+1} \rangle$, thus

$$\frac{\partial}{\partial \mathrm{SR}^r_{j+1}} \langle Z_k, \widetilde{A}^r_j \rangle = \tau_j \Big(a_{j+1,k} \widetilde{A}^r_{j+1} + \langle Z_k, \widetilde{A}^r_{j+1} \rangle \Big),$$

$$\frac{\partial}{\partial \widetilde{A}^r_{j+1}} \langle Z_k, \widetilde{A}^r_j \rangle = \tau_j a_{j+1,k} \mathrm{SR}^r_{j+1},$$

$$\frac{\partial}{\partial \langle Z_k, \widetilde{A}^r_{j+1} \rangle} \langle Z_k, \widetilde{A}^r_j \rangle = 1 + \tau_j \mathrm{SR}^r_{j+1},$$

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

For $j < n - r$,

$$\langle Z_k, \widetilde{A}^r_j \rangle = \tau_i a_{j+1,k} \mathrm{SR}^r_{j+1} \widetilde{A}^r_{j+1} + (1 + \tau_i \mathrm{SR}^r_{j+1}) \langle Z_k, \widetilde{A}^r_{j+1} \rangle + (\tau_j - \tau_{j+r}) \langle Z_k, \widetilde{P}_{j+r+1} \rangle,$$

we can see that $\langle Z_k, \widetilde{A}^r_j \rangle$ depend on an extra variable $\langle Z_k, \widetilde{P}_{i+r+1} \rangle$, thus we have an extra partial derivative with respect to $\langle Z_k, \widetilde{P}_{i+r+1} \rangle$ in addition to the three partial derivatives when $j \geqslant n - r$

$$\frac{\partial}{\partial \langle Z_k, \widetilde{P}_{j+r+1} \rangle} \langle Z_k, \widetilde{A}^r_j \rangle = \tau_j - \tau_{j+r}.$$

Therefore $\mathbf{H}'_{0,j}$ has 1s on the diagonals, and either 3 or 4 entries on each of the last F rows. Therefore each of the operations

$$\mathbf{v} = \mathbf{w} \mathbf{H}'_{0,j}, \quad j = 0, 1, \ldots, 2(n-2), 2(n-2)+1,$$

has computational order of $O(F)$.

4.2.3. *Jacobian matrix of* $\mathbf{H}_0$. According to (2.5), the drifts of the constant maturity swap-rates are given by

$$\mu^{(n)}_j = -\frac{1}{\widetilde{A}^r_j} \sum_{k=1}^{F} a_{j,k} \langle Z_k, \widetilde{A}^r_j \rangle. \tag{4.4}$$

We can see that each $\mu_j^{(n)}$ depends on $\widetilde{A}_j^r$ and $\langle Z_k, \widetilde{A}_j^r\rangle$, then

$$\frac{\partial}{\partial A_j^r}\mu_j^{(n)} = -\frac{1}{\widetilde{A}_j^r}\mu_j^{(n)},$$
$$\frac{\partial}{\partial\langle Z_k, \widetilde{A}_j^r\rangle}\mu_j^{(n)} = -\frac{1}{\widetilde{A}_j^r}a_{j,k},$$

for $j = 0, \ldots, n-2$ and $k = 1, \ldots, F$. The Jacobian matrix $\mathbf{H}_0'$ has $n-1$ partial derivatives equal to $\partial\mu_j^{(n)}/\partial A_j^r$, and $(n-1)F$ partial derivatives equal to $\partial\mu_j^{(n)}/\partial\langle Z_k, \widetilde{A}_j^r\rangle$. Therefore the operation

$$\mathbf{v} = \mathbf{w}H_0'$$

has order $O(nF)$.

4.2.4. *Total computational order of* $\mathbf{wF}_{0,1}'$. The number of mappings $\mathbf{H}_{0,j}$ depends on n, therefore the order of

$$\mathbf{v} = \mathbf{wH}_{0,2(n-2)+1}'\mathbf{H}_{0,2(n-2)}' \cdots \mathbf{H}_{0,1}'\mathbf{H}_{0,0}'$$

is $O(nF)$ since each operation has a constant order. Hence, the order of the operation

$$\mathbf{v} = \mathbf{wH}_0'\mathbf{H}_{0,2(n-2)+1}' \cdots \mathbf{H}_{0,0}' = \mathbf{wF}_{0,1}'$$

is $O(nF)$.

4.3. **Computational order of $\mathbf{wF}_{0,2}'$.** We have the following numerical scheme to simulate constant-maturity swap-rates

$$\mathrm{SR}_j^r(T_0) = \mathrm{SR}_j^r(0)\exp\left(\mu_j^{(n)} - \frac{1}{2}\sum_{k=1}^{F}a_{j,k}^2 + \sum_{k=1}^{F}a_{j,k}Z_k\right), \tag{4.5}$$

where $\mu_j^{(n)}$ is given in (4.4), for $j = 0, 1, \ldots, n-1$. The non-zero partial derivatives of $\mathbf{F}_{0,2}'$ are

$$\frac{\partial\mathrm{SR}_j^r(T_0)}{\partial\mathrm{SR}_j^r(0)} = \frac{\mathrm{SR}_j^r(T_0)}{\mathrm{SR}_j^r(0)}, \quad j = 0, \ldots, n-1$$
$$\frac{\partial\mathrm{SR}_j^r(T_0)}{\partial\mu_j^{(n)}} = \mathrm{SR}_j^r(T_0), \quad j = 0, \ldots, n-2.$$

Thus, $\mathbf{F}_{0,3}'$ has the general form

$$\begin{pmatrix} \frac{\mathrm{SR}_0^r(T_0)}{\mathrm{SR}_0^r(0)} & & & \mathrm{SR}_0^r(T_0) & & \\ & \ddots & & & \ddots & \\ & & \frac{\mathrm{SR}_{n-2}^r(T_0)}{\mathrm{SR}_{n-2}^r(0)} & & & \mathrm{SR}_{n-2}^r(T_0) \\ & & & \frac{\mathrm{SR}_{n-1}^r(T_0)}{\mathrm{SR}_{n-1}^r(0)} & & 0 \end{pmatrix},$$

where the blanks are zero and the 0 indicates that the swap-rate SR_{n-1} does not have drift under the terminal measure. The Jacobian matrix $\mathbf{F}'_{0,3}$ has $2n-1$ non-zero entries. Hence the operation $\mathbf{v} = \mathbf{w}\mathbf{F}'_{0,3}$ has $2n-1$ computations so that its order is $O(n)$.

4.4. **Computational order of $\mathbf{w}\mathbf{F}'$.** Similar to the ctSMM, the gradient $\mathbf{F}'$ can be computed in order $O(n)$. We divide the mapping F in (3.4) into the following sub-mappings:

$$\mathbf{SR}(T_m) \xrightarrow{\mathbf{I}_0} \begin{pmatrix} \mathbf{SR}(T_m) \\ \widetilde{\mathbf{A}}(T_m) \end{pmatrix} \xrightarrow{\mathbf{I}_1} \widetilde{f}(T_m). \tag{4.6}$$

We have shown in sections 4.1 that the operations $\mathbf{v} = \mathbf{w}\mathbf{I}'_0$ has order $O(n)$. The computation of the gradient $\mathbf{I}'_1$ is $O(n)$, so the order of computing $\mathbf{F}'$ is $O(n)$.

4.5. **Summary.** We have shown that the computational order of $\mathbf{w}\mathbf{F}'_{0,i}$ is either $O(n)$ or $O(nF)$. Hence the total computational order of

$$\mathbf{V}(0) = \mathbf{V}(T_0)\mathbf{F}'_{0,2}\mathbf{F}'_{0,1}\mathbf{F}'_{0,0} = \mathbf{V}(T_0)\mathbf{F}'_0$$

is $O(nF)$. Since the computational order of $\mathbf{w}\mathbf{F}'_{k,i}$ is similar to that of $\mathbf{w}\mathbf{F}'_{0,i}$, we have shown that the operation

$$\mathbf{V}(T_{k-1}) = \mathbf{V}(T_k)\mathbf{F}'_{k,2}\mathbf{F}'_{k,1}\mathbf{F}'_{k,0} = \mathbf{V}(T_k)\mathbf{F}'_k$$

is $O(nF)$.

5. Delta estimation in the co-initial swap-rate market model

The ciSMM is a one-period model, we consider the following mappings

$$\begin{aligned} \mathbf{SR}(0) &\xrightarrow{\mathbf{F}_{0,0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{F}_{0,1}} \begin{pmatrix} \mathbf{SR}(0) \\ \underline{\mu}^{(0)}(0) \end{pmatrix} \xrightarrow{\mathbf{F}_{0,2}} \mathbf{SR}(T_0) \\ &\xrightarrow{\mathbf{F}_{1,0}} \begin{pmatrix} \mathbf{SR}(T_0) \\ \widetilde{\mathbf{A}}(T_0) \end{pmatrix} \xrightarrow{\mathbf{F}} \widetilde{f}(T_0). \end{aligned} \tag{5.1}$$

To show that the adjoint method has order $O(nF)$, we need to show that the operation $\mathbf{V}(0) = \mathbf{F}'\mathbf{F}'_{1,0}\mathbf{F}'_{0,2}\mathbf{F}'_{0,1}\mathbf{F}'_{0,0}$ has order $O(nF)$.

5.1. **Computational order of $\mathbf{w}\mathbf{F}'_{0,0}$.** We divide the mapping

$$\mathbf{SR}(0) \xrightarrow{\mathbf{F}_{0,0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix}$$

into the following sub-mappings

$$\mathbf{SR}(0) \xrightarrow{\mathbf{G}_{0,0}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{P}_1(0) \end{pmatrix} \xrightarrow{\mathbf{G}_{0,1}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{P}_1(0) \\ \{\widetilde{\mathbf{A}}_i(0)\}_{i=1}^{1} \end{pmatrix}$$
$$\xrightarrow{\mathbf{G}_{0,2}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{P}_2(0) \\ \{\widetilde{\mathbf{A}}_i(0)\}_{i=1}^{1} \end{pmatrix} \xrightarrow{\mathbf{G}_{0,3}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{P}_2(0) \\ \{\widetilde{\mathbf{A}}_i(0)\}_{i=1}^{2} \end{pmatrix}$$
$$\cdots \xrightarrow{\mathbf{G}_{0,2n-2}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{P}_n(0) \\ \{\widetilde{\mathbf{A}}_i(0)\}_{i=1}^{n-1} \end{pmatrix} \xrightarrow{\mathbf{G}_{0,2n-1}} \begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \tag{5.2}$$

The even-numbered mappings $\mathbf{G}_{0,j}$ update a bond ratio using the given set of inputs, and the odd-numbered mappings $\mathbf{G}_{0,j}$ computes a new annuity ratio using the given set of inputs.

5.1.1. *Jacobian matrix of* $\mathbf{G}_{0,j}$ *where* j *is even.* From (2.19), we can see that $\widetilde{P}_i$ depends on $\widetilde{A}_{i-1}$ and $\mathrm{SR}_{0,i}$, then

$$\frac{\partial \widetilde{P}_i}{\partial \widetilde{A}_{i-1}} = -\frac{\widetilde{A}_{i-1} + \tau_{i-1}}{(1+\tau_{i-1}\mathrm{SR}_{0,i})^2}, \quad \frac{\partial \widetilde{P}_i}{\partial \mathrm{SR}_{0,i}} = \frac{\mathrm{SR}_{0,i}}{1+\tau_{i-1}\mathrm{SR}_{0,i}}.$$

The existence of two non-zero partial derivatives implies that the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,j}$ has two computations.

5.1.2. *Jacobian matrix of* $\mathbf{G}_{0,j}$ *where* j *is odd.* From (2.18), we can see that $\widetilde{A}_i$ only depends on $\widetilde{A}_{i-1}$ and $\widetilde{P}_i$, then

$$\frac{\partial \widetilde{A}_i}{\partial \widetilde{A}_{i-1}} = 1, \quad \frac{\partial \widetilde{A}_i}{\partial \widetilde{P}_i} = \tau_{i-1}.$$

Then the Jacobian matrix $\mathbf{G}'_{0,j}$ has 1s on the diagonal, one entry equal to τ on the last row. Therefore the operation $\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,j}$ has one computation.

5.1.3. *Total computational order of* $\mathbf{w}\mathbf{F}'_{0,0}$. Similar to the cmSMM: The number of sub-mappings $\mathbf{G}_{0,j}$ depends on n, thus the computational order of the operation

$$\mathbf{v} = \mathbf{w}\mathbf{G}'_{0,2n-1}\mathbf{G}'_{0,2n-2}\cdots\mathbf{G}'_{0,0} = \mathbf{w}\mathbf{F}'_{0,0}$$

is $O(n)$ since each operation has a constant computational order.

5.2. **Computational order of** $\mathbf{w}\mathbf{F}'_{0,1}$**.** We divide the mapping

$$\begin{pmatrix} \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{F}_{0,1}} \begin{pmatrix} \mathbf{SR}(0) \\ \underline{\mu}^{(0)}(0) \end{pmatrix}$$

into the following sub-mappings

$$\begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\end{pmatrix} \xrightarrow{\mathbf{H}_{0,0}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k,\widetilde{A}_i\rangle\big\}_{i=1}^{1}\end{pmatrix} \xrightarrow{\mathbf{H}_{0,1}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k,\widetilde{A}_i\rangle\big\}_{i=1}^{2}\end{pmatrix}$$
$$\cdots \xrightarrow{\mathbf{H}_{0,n-1}} \begin{pmatrix}\mathbf{SR}(0)\\ \widetilde{\mathbf{A}}(0)\\ \big\{\langle Z_k,\widetilde{A}_i\rangle\big\}_{i=1}^{n}\end{pmatrix} \xrightarrow{\mathbf{H}_{0}} \begin{pmatrix}\mathbf{SR}(0)\\ \underline{\mu}^{(0)}(0)\end{pmatrix} \tag{5.3}$$

In contrast to the cmSMM where we only need to compute $(n-1)F$ cross variation derivatives, we need to compute nF cross variation derivatives in the ciSMM since all rates have a positive drift under the spot measure.

5.2.1. *Jacobian matrix of* $\mathbf{H}_{0,j}$. We rewrite the recursive formula (2.17) in terms of annuity ratios,

$$\big\langle Z_k,\widetilde{A}_j\big\rangle = \frac{1}{1+\tau_{j-1}\mathrm{SR}_{0,j}}\Big[\big\langle Z_k,\widetilde{A}_{j-1}\big\rangle - \tau_{j-1}a_{j,k}\mathrm{SR}_{0,j}\widetilde{A}_j\Big].$$

We can see that $\big\langle Z_k,\widetilde{A}_j\big\rangle$ depend on $\mathrm{SR}_{0,j}$, $\widetilde{A}_j$ and $\big\langle Z_k,\widetilde{A}_{j-1}\big\rangle$, then

$$\begin{aligned}
\frac{\partial}{\partial \mathrm{SR}_{0,j}}\big\langle Z_k,\widetilde{A}_j\big\rangle &= -\frac{\big\langle Z_k,\widetilde{A}_{j-1}\big\rangle + a_{j,k}\tau_{j-1}\widetilde{A}_j\big(1+\mathrm{SR}_{0,j}(\tau_{j-1}-1)\big)}{1+\tau_{j-1}\mathrm{SR}_{0,j}},\\
\frac{\partial}{\partial \widetilde{A}_j}\big\langle Z_k,\widetilde{A}_j\big\rangle &= -\frac{\tau_{j-1}a_{j,k}\mathrm{SR}_{0,j}}{1+\tau_{j-1}\mathrm{SR}_{0,j}},\\
\frac{\partial}{\partial\big\langle Z_k,\widetilde{A}_{j-1}\big\rangle}\big\langle Z_k,\widetilde{A}_j\big\rangle &= \frac{1}{1+\tau_{j-1}\mathrm{SR}_{0,j}},
\end{aligned}$$

for $k=1,\dots,F$. Therefore $\mathbf{H}'_{0,j}$ has 1s on the diagonals, and 3 entries equal to the above partial derivatives on each of the last F rows. Therefore each of the operations

$$\mathbf{v} = \mathbf{w}\mathbf{H}'_{0,j}, \quad j=0,1,\dots,n-1$$

has $3F$ computations.

5.2.2. *Jacobian matrix of* $\mathbf{H}_0$. The drifts of $\mathrm{SR}_{0,j}$ under the spot measure are given by

$$\mu_j^{(0)} = -\frac{1}{\widetilde{A}_j}\sum_{k=1}^{F} a_{j,k}\big\langle Z_k,\widetilde{A}_j\big\rangle. \tag{5.4}$$

We can see that each $\mu_j^{(0)}$ depends on $\widetilde{A}_j$ and $\big\langle Z_k,\widetilde{A}_j\big\rangle$, then

$$\begin{aligned}
\frac{\partial}{\partial \widetilde{A}_j}\mu_j^{(0)} &= -\frac{\mu_j}{\widetilde{A}_j},\\
\frac{\partial}{\partial\big\langle Z_k,\widetilde{A}_j\big\rangle}\mu_j^{(0)} &= -\frac{a_{j,k}}{\widetilde{A}_j},
\end{aligned}$$

for $j = 1, \dots, n$ and $k = 1, \dots, F$. $\mathbf{H}_0'$ has n partial derivatives equal to $\partial\mu_j^{(0)}/\partial A_j$, and nF partial derivatives equal to $\partial\mu_j^{(0)}/\partial\langle Z_k, \widetilde{A}_j\rangle$. Therefore the operation $\mathbf{v} = \mathbf{w}H_0'$ has order $O(nF)$.

5.2.3. *Total computational order of* $\mathbf{wF}_{0,1}'$. The number of mappings $\mathbf{H}_{0,j}$ depends on n, therefore the order of

$$\mathbf{v} = \mathbf{w}\mathbf{H}_{0,n-1}'\mathbf{H}_{0,n-2}'\cdots\mathbf{H}_{0,1}'\mathbf{H}_{0,0}'$$

is $O(nF)$ since each operation has a constant order. Hence, the order of the operation $\mathbf{v} = \mathbf{w}\mathbf{H}_0'\mathbf{H}_{0,n-1}'\cdots\mathbf{H}_{0,0}' = \mathbf{wF}_{0,2}'$ is $O(nF)$.

5.3. **Computational order of $\mathbf{wF}_{0,2}'$.** We have the following numerical scheme to simulate co-initial swap-rates

$$\mathrm{SR}_{0,j}(T_0) = \mathrm{SR}_{0,j}(0)\exp\left(\mu_j^{(0)} - \frac{1}{2}\sum_{k=1}^{F} a_{j,k}^2 + \sum_{k=1}^{F} a_{j,k}Z_k\right), \tag{5.5}$$

where $\mu_j^{(0)}$ is given in (5.4), for $j = 1, 2, \dots, n$. The non-zero partial derivatives of $\mathbf{F}_{0,2}'$ are

$$\begin{aligned}\frac{\partial\mathrm{SR}_{0,j}(T_0)}{\partial\mathrm{SR}_{0,j}(0)} &= \frac{\mathrm{SR}_{0,j}(T_0)}{\mathrm{SR}_{0,j}(0)}, \quad j = 1, \dots, n\\ \frac{\partial\mathrm{SR}_{0,j}(T_0)}{\partial\mu_j^{(0)}} &= \mathrm{SR}_{0,j}(T_0), \quad j = 1, \dots, n.\end{aligned}$$

Thus, $\mathbf{F}_{0,3}'$ has the general form

$$\begin{pmatrix} \frac{\mathrm{SR}_{0,1}(T_0)}{\mathrm{SR}_{0,1}(0)} & & & \mathrm{SR}_{0,1}(T_0) & & \\ & \ddots & & & \ddots & \\ & & \frac{\mathrm{SR}_{0,n}(T_0)}{\mathrm{SR}_{0,n}(0)} & & & \mathrm{SR}_{0,n}(T_0) \end{pmatrix},$$

where the blanks are zero. The Jacobian matrix $\mathbf{F}_{0,3}'$ has $2n$ non-zero entries. Hence the operation $\mathbf{v} = \mathbf{wF}_{0,3}'$ has $2n$ computations so that its order is $O(n)$.

5.4. **Computational order of $\mathbf{wF}'$.** We divide the mapping F in (5.1) into the following sub-mappings:

$$\mathbf{SR}(T_0) \xrightarrow{\mathbf{I}_0} \begin{pmatrix}\mathbf{SR}(T_0)\\ \widetilde{\mathbf{A}}(T_0)\end{pmatrix} \xrightarrow{\mathbf{I}_1} \widetilde{f}(T_0). \tag{5.6}$$

We have shown in sections 5.1 that the operations $\mathbf{v} = \mathbf{wI}_0'$ has order $O(n)$. The computation of the gradient $\mathbf{I}_1'$ is trivial. Therefore, the order of computing $\mathbf{F}'$ is $O(n)$.

6. Vega estimation

6.1. **Model elementary Vegas.** Assume that the set of swap-rates at time T_r is given by a vector function with the following inputs: the rates at time T_{r-1}, the set of pseudo-root elements $\{a^r_{j,k}\}$ over the step (T_{r-1}, T_r), and a set of random variates $\mathbf{Z}$

$$\mathbf{SR}(T_r) = \mathbf{F}_r\big(\mathbf{SR}(T_{r-1}), \{a^r_{j,k}\}, \mathbf{Z}\big). \tag{6.1}$$

Suppose that the model is driven by F factors. We use the following numerical scheme to evolve the swap-rates

$$\mathrm{SR}_i(T_r) = \mathrm{SR}_i(T_{r-1}) \exp\left(\mu_i^{(N)}\big(\{a^r_{j,k}\}, \mathbf{SR}(T_{r-1})\big) - \frac{1}{2}\sum_{l=1}^{F}\big(a^r_{i,l}\big)^2 + \sum_{l=1}^{F} a^r_{i,l} Z_l\right), \tag{6.2}$$

where $\mu_i^{(N)}$ is the drift under the measure associated with P_N bond.

We define the model elementary Vegas of an IRD with discounted price g to be the following partial derivatives

$$\frac{\partial g}{\partial a^r_{j,k}} = P_N(0)\frac{\partial \widetilde{f}(T_m)}{\partial a^r_{j,k}},$$

for $j = 0, \dots, n-1$, $k = 1, \dots, F$ and $r = 0, \dots, m-1$. From (6.1) and using the chain rule,

$$\frac{\partial \widetilde{f}(T_m)}{\partial a^r_{j,k}} = \frac{\partial \widetilde{f}(T_m)}{\partial \mathbf{SR}(T_m)}\frac{\partial \mathbf{SR}(T_m)}{\partial \mathbf{SR}(T_{m-1})}\cdots\frac{\partial \mathbf{SR}(T_{r+1})}{\partial \mathbf{SR}(T_r)}\frac{\partial \mathbf{SR}(T_r)}{\partial a^r_{j,k}} \tag{6.3}$$

$$= \mathbf{V}(T_r)\frac{\partial \mathbf{F}_r}{\partial a^r_{j,k}}, \tag{6.4}$$

where $\mathbf{V}(T_r)$ is the adjoint vector defined in (3.6). In order to compute model elementary Vegas, we only need to compute the gradient $\partial \mathbf{F}_r/\partial a^r_{j,k}$ since the adjoint vectors have already been computed in Delta calculations.

Using (6.2), the ith entry of the gradient $\partial \mathbf{F}_r/\partial a^r_{j,k}$ is given by

$$\frac{\partial \mathrm{SR}_i(T_r)}{\partial a^r_{j,k}} = \mathrm{SR}_i(T_r)\left[\frac{\partial \mu_i^{(N)}}{\partial a_{j,k}} - a^r_{i,k}\delta_{i,j} + Z_k\delta_{i,j}\right], \tag{6.5}$$

where $\delta_{i,j}$ is Kronecker's delta. Similar to the computation of Deltas, there is no closed-form solutions for (6.5) due to the complicated form of the drifts.

However, if we modify the mappings in (3.8) to

$$\begin{pmatrix}\{a^r_{j,k}\}\\ \mathbf{SR}(T_{r-1})\\ \widetilde{\mathbf{A}}(T_{r-1})\end{pmatrix} \xrightarrow{\mathbf{J}_{r,1}} \begin{pmatrix}\{a^r_{j,k}\}\\ \mathbf{SR}(T_{r-1})\\ \underline{\mu}^{(N)}(T_{r-1})\end{pmatrix} \xrightarrow{\mathbf{J}_{r,2}} \mathbf{SR}(T_r), \tag{6.6}$$

then the first nF entries of the vector, $\mathbf{V}(T_r)\mathbf{J}'_{r,2}\mathbf{J}'_{r,1}$, will be equal to

$$\mathbf{V}(T_r)\frac{\partial \mathbf{F}_r}{\partial a^r_{j,k}},$$

for $j = 0, \ldots, n-1$ and $k = 1, \ldots, F$. If each of the vector multiplications has order $O(nF)$, then the order to compute model elementary Vegas will be $O(nF)$ per step. We show how to carry out efficient vega computations in the cmSMM and the ciSMM.

6.2. **Adjoint method in the cmSMM.** For concreteness and readability, we will only show the computational order of the operations

$$\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}. \tag{6.7}$$

i.e. the adjoint operations for the first step. We omit the superscript 0 in $a^0_{j,k}$ as it causes no confusion.

6.2.1. *Computational order of* $\mathbf{wJ}'_{0,1}$. We divide the mapping

$$\begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{J}_{0,1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \underline{\mu}^{(n)}(0) \end{pmatrix}$$

into the following sub-mappings

$$\begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{K}_{0,0}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{P}^r_i\rangle\}_{i=n-1}^{n-1} \end{pmatrix} \xrightarrow{\mathbf{K}_{0,1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{P}^r_i\rangle\}_{i=n-1}^{n-1} \\ \{\langle Z_k, \widetilde{A}^r_i\rangle\}_{i=n-2}^{n-2} \end{pmatrix}$$
$$\xrightarrow{\mathbf{K}_{0,2}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{P}^r_i\rangle\}_{i=n-2}^{n-1} \\ \{\langle Z_k, \widetilde{A}^r_i\rangle\}_{i=n-2}^{n-2} \end{pmatrix} \xrightarrow{\mathbf{K}_{0,3}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{P}^r_i\rangle\}_{i=n-2}^{n-1} \\ \{\langle Z_k, \widetilde{A}^r_i\rangle\}_{i=n-3}^{n-2} \end{pmatrix}$$
$$\cdots \xrightarrow{\mathbf{K}_{0,2(n-2)+1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{P}^r_i\rangle\}_{i=1}^{n-1} \\ \{\langle Z_k, \widetilde{A}^r_i\rangle\}_{i=0}^{n-2} \end{pmatrix} \xrightarrow{\mathbf{K}_0} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \underline{\mu}^{(n)}(0) \end{pmatrix} \tag{6.8}$$

The Jacobian matrices $\mathbf{K}_{0,i}$ and $\mathbf{K}_0$ have similar structures to the Jacobian matrices $\mathbf{H}_{0,i}$ and $\mathbf{H}_0$ in (4.3):

(1) In addition to the non-zero partial derivatives in section 4.2.1, $\mathbf{K}_{0,j}$ when j is even has an extra partial derivative with respect to $a_{j,k}$ equal to

$$\frac{\partial}{\partial a_{j,k}}\langle Z_k, \widetilde{P}_j\rangle = \mathrm{SR}^r_j \widetilde{A}^r_j.$$

(2) In addition to the non-zero partial derivatives in section 4.2.2, $\mathbf{K}_{0,j}$ when j is odd has an extra partial derivative with respect to $a_{j,k}$ equal to

$$\frac{\partial}{\partial a_{j,k}}\langle Z_k, \widetilde{A}_j^r\rangle = \tau_j \mathrm{SR}_{j+1}^r \widetilde{A}_{j+1}^r.$$

(3) In addition to the non-zero partial derivatives in section 4.2.3, $\mathbf{K}_0$ has an extra partial derivative with respect to $a_{j,k}$ equal to

$$\frac{\partial \mu_j^{(n)}}{\partial a_{j,k}} = -\frac{1}{\widetilde{A}_j^r}\langle Z_k, \widetilde{A}_j^r\rangle.$$

The number of mappings $\mathbf{K}_{0,j}$ depends on n, and each operation $\mathbf{v} = \mathbf{w}K'_{0,j}$, $j = 0, \ldots, 2(n-2)+1$, has a constant order. Similar to the Delta computations, the operation $\mathbf{v} = \mathbf{w}K'_0$ has order $O(nF)$. Hence, the order of the operation

$$\mathbf{v} = \mathbf{w}\mathbf{K}'_0\mathbf{K}'_{0,2(n-2)+1}\cdots\mathbf{K}'_{0,0} = \mathbf{w}\mathbf{J}'_{0,1}$$

is $O(nF)$.

6.2.2. *Computational order of* $\mathbf{w}\mathbf{J}'_{0,2}$. We use (4.5) to evolve the constant maturity swap-rates. The non-zero partial derivatives of $\mathbf{J}'_{0,2}$ are

$$\begin{aligned}
\frac{\partial \mathrm{SR}_j^r(T_0)}{\partial a_{j,k}} &= \mathrm{SR}_j^r(T_0)\big(-a_{j,k} + Z_k\big),\\
\frac{\partial \mathrm{SR}_j^r(T_0)}{\partial \mathrm{SR}_j^r(0)} &= \frac{\mathrm{SR}_j^r(T_0)}{\mathrm{SR}_j^r(0)},\\
\frac{\partial \mathrm{SR}_j^r(T_0)}{\partial \mu_j^{(n)}} &= \mathrm{SR}_j^r(T_0),
\end{aligned}$$

for $j = 0, \ldots, n-1$ and $k = 1, \ldots, F$. Hence the operation $\mathbf{v} = \mathbf{w}\mathbf{J}'_{0,2}$ has $2n - 1 + nF$ computations so that its order is $O(nF)$.

6.2.3. *Total computational order of* $\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}$. Given that the computational orders of

$$\mathbf{v} = \mathbf{w}\mathbf{J}'_{0,1} \quad \text{and} \quad \mathbf{v} = \mathbf{w}\mathbf{J}'_{0,2}$$

are $O(nF)$. Hence the adjoint operation

$$\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}$$

has order $O(nF)$.

6.3. **Adjoint method in the ctSMM.** If we set $r = n$ in the cmSMM, we then have the ctSMM. Thus we discuss Vega computations in the ctSMM no further as it is a special case of the more general cmSMM.

6.4. **Adjoint method in the ciSMM.** The ciSMM is a one-period model, we consider the following mappings

$$\begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{J}_{0,1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \underline{\mu}^{(0)}(0) \end{pmatrix} \xrightarrow{\mathbf{J}_{0,2}} \mathbf{SR}(T_0). \tag{6.9}$$

We will show that the following operations

$$\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}$$

has order $O(nF)$ to prove that the adjoint operations needed to compute model Vegas has order $O(nF)$.

6.4.1. *Computational order of* $\mathbf{wJ}'_{0,1}$. We divide the mapping

$$\begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{J}_{0,1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \underline{\mu}^{(0)}(0) \end{pmatrix}$$

into the following sub-mappings

$$\begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \end{pmatrix} \xrightarrow{\mathbf{K}_{0,0}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{A}_i\rangle\}_{i=1}^{1} \end{pmatrix} \xrightarrow{\mathbf{K}_{0,1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{A}_i\rangle\}_{i=1}^{2} \end{pmatrix}$$
$$\cdots \xrightarrow{\mathbf{K}_{0,n-1}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \widetilde{\mathbf{A}}(0) \\ \{\langle Z_k, \widetilde{A}_i\rangle\}_{i=1}^{n} \end{pmatrix} \xrightarrow{\mathbf{K}_{0}} \begin{pmatrix} \{a_{j,k}\} \\ \mathbf{SR}(0) \\ \underline{\mu}^{(0)}(0) \end{pmatrix} \tag{6.10}$$

The Jacobian matrices $\mathbf{K}_{0,i}$ and $\mathbf{K}_0$ have similar structures to the Jacobian matrices $\mathbf{H}_{0,i}$ and $\mathbf{H}_0$ in (5.3):

(1) In addition to the three non-zero partial derivatives in section 5.2.1, $\mathbf{K}_{0,i}$ has an extra partial derivative with respect to $a_{j,k}$ equal to

$$\frac{\partial}{\partial a_{j,k}}\langle Z_k, \widetilde{A}_j\rangle = -\frac{\tau_{j-1}\mathrm{SR}_{0,j}\widetilde{A}_j}{1+\tau_{j-1}\mathrm{SR}_{0,j}}.$$

Thus the operation $\mathbf{v} = \mathbf{wK}_{0,j}$ has $4F$ computations.

(2) In addition to the three non-zero partial derivatives in section 5.2.2, $\mathbf{K}_0$ has an extra partial derivative with respect to $a_{j,k}$ equal to

$$\frac{\partial \mu_j^{(0)}}{\partial a_{j,k}} = -\frac{1}{\widetilde{A}_j}\langle Z_k, \widetilde{A}_j\rangle.$$

Thus $\mathbf{K}'_0$ has an additional nF partial derivatives equal to $\partial\mu_j^{(0)}/\partial a_{j,k}$ compared with $\mathbf{H}'_0$. Therefore the operation $\mathbf{v} = \mathbf{w}K'_0$ has order $O(nF)$.

The number of mappings $\mathbf{K}_{0,j}$ is equal to n, therefore the order of

$$\mathbf{v} = \mathbf{w}\mathbf{K}'_{0,n-1}\mathbf{K}'_{0,n-2}\mathbf{K}'_{0,n-3}\cdots\mathbf{K}'_{0,1}\mathbf{K}'_{0,0}$$

is $O(nF)$ since each operation has a constant order. Hence, the order of the operation

$$\mathbf{v} = \mathbf{w}\mathbf{K}'_{0}\mathbf{K}'_{0,n-1}\cdots\mathbf{K}'_{0,0} = \mathbf{w}\mathbf{J}'_{0,1}$$

is $O(nF)$.

6.4.2. *Computational order of* $\mathbf{w}\mathbf{J}'_{0,2}$. We use (5.5) to evolve the co-initial swap-rates. The non-zero partial derivatives of $\mathbf{J}'_{0,2}$ are

$$\begin{aligned}
\frac{\partial \mathrm{SR}_{0,j}(T_0)}{\partial a_{j,k}} &= \mathrm{SR}_{0,j}(T_0)\big(-a_{j,k} + Z_k\big),\\
\frac{\partial \mathrm{SR}_{0,j}(T_0)}{\partial \mathrm{SR}_{0,j}(0)} &= \frac{\mathrm{SR}_{0,j}(T_0)}{\mathrm{SR}_{0,j}(0)},\\
\frac{\partial \mathrm{SR}_{0,j}(T_0)}{\partial \mu_j^{(0)}} &= \mathrm{SR}_{0,j}(T_0),
\end{aligned}$$

for $j = 1, \ldots, n$ and $k = 1, \ldots, F$. Hence the operation $\mathbf{v} = \mathbf{w}\mathbf{J}'_{0,2}$ has $2n+nF$ computations so that its order is $O(nF)$.

6.4.3. *Total computational order of* $\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}$. Similar to the cmSMM, given the adjoint vector $\mathbf{V}(T_0)$, the operation $\mathbf{V}(T_0)\mathbf{J}'_{0,2}\mathbf{J}'_{0,1}$ has order $O(nF)$.

6.5. **Market Vegas.** Traders and quants are not interested in the sensitivities with respect to model parameters such as model Vegas, they are interested in the sensitivities with respect to market observable interest rates and interest rate derivatives. The problem of converting model elementary Vegas to market Vegas in the LMM has been addressed in Joshi and Kwon (2009).

Suppose we evolve the model m steps in the simulation. The methods discussed in this section will produce an $n \times F$ matrix of model elementary Vegas at each step so that we have m matrices overall. Joshi and Kwon (2009) show that the market Vegas can be computed as linear combinations of the $m \times n \times F$ numbers, $\partial \mathbf{F}/\partial a^r_{j,k}$. Similar approaches will apply to generic market models, we leave this for future research.

7. Timing tests

We have shown that the order of the adjoint method is $O(nF)$ per step in the cmSMM and the ciSMM. If we carry out the algorithm for 1 step, we should obtain timings that are linear in n and F. If we carry out the the algorithm for n steps, we should obtain timings that are parabolic in n. In each of the following cases, we ran 163,840 paths on a European swaption and estimate Deltas and model Vegas using the efficient algorithms discussed in this paper. We ran the Monte Carlo simulations on a computer with an Intel Core 2 1.6GHz CPU and 1GB RAM, using single-threaded C++ code.

7.1. **Timing tests for Deltas.**

7.1.1. *Constant maturity swap-rate market model.* We have n constant maturity swap-rates, $\mathrm{SR}_{j,j+2}$, each with the reset date $T_j = j \times 0.5$, $j = 1, \ldots, n$. We fix the number of factors to be 3. We plot time required to compute the Deltas of a swaption maturing at T_n against n. We show the graph in figure 1, and we display the values of a fitted parabola through timings in table 7.1. Similar timing results are obtained for different values of r in $\mathrm{SR}_{j,j+r}$.

7.1.2. *Co-initial swap-rate market model.* We have n co-initial swap-rates, $\mathrm{SR}_{0,j}$, each with the reset date $T_j = j \times 0.5$, $j = 1, \ldots, n$. We fix the number of factors to be 3. We plot time required to compute the Deltas of a swaption maturing at T_0 against n. We show the graph in figure 2, and we display the values of a fitted line through timings in table 7.2.

7.2. **Timing tests for Vegas.**

7.2.1. *Model elementary Vegas.* Similar to the timing tests for Deltas, we compute model elementary Vegas in ctSMM, cmSMM and ciSMM. In each of the following cases, we have n swap-rates, each with the reset date $T_j = j \times 0.5$, $j = 1, \ldots, n$, and driven by F factors.

- We estimate $n \times n \times F$ model elementary Vegas in cmSMM. We fix $F = 3$ and plot time against n. We show the graph in figure 3, and we display the values of a fitted parabola through timings in table 7.3.
- Since the ctSMM is a special case of the cmSMM, similar results are obtained for the ctSMM.
- We estimate $n \times F$ model elementary Vegas in ciSMM. We fix $F = 3$ and plot time against n. We show the graph in figure 4, and we display the values of a fitted line through timings in table 7.4.

In particular, we can see that the time required to compute model Vegas only exceeds the time required to compute Deltas by a small proportion.

7.2.2. *Market Vegas.* As mentioned in section 6.4: we compute all the model elementary Vegas first, then we use linear combinations of those to calculate market Vegas. We compute market Vegas using the model elementary Vegas computed in a cmSMM with 60 rates across 60 steps. We compute the ratios of the time required to compute the market Vegas only to the time required to compute Deltas only in table 7.5.

Since we carry out the linear combinations outside the main simulation loop so that computing market Vegas costs little extra time compared with computing model elementary Vegas. In particular, if we compute 60 market Vegas, the ratio is no greater than 1.4.

n	Time	Parabolic Fit
3	0.51	0.52
5	1.34	1.36
10	5.24	5.25
15	11.86	11.72
20	20.59	20.76
25	32.42	32.36

TABLE 7.1. Timings for estimating Deltas of a T_n European swaption with $F = 3$ in the cmSMM.

n	Time	Linear Fit
3	0.55	0.54
5	0.81	0.81
10	1.48	1.47
15	2.09	2.13
20	2.81	2.80
25	3.47	3.46

TABLE 7.2. Timings for estimating Deltas of a T_0 European swaption with $F = 3$ in the ciSMM.

n	Time	Parabolic Fit
3	0.468	0.375
5	1.281	1.254
10	5.529	5.659
15	12.745	13.216
20	24.74	23.927
25	37.457	37.789

TABLE 7.3. Timings for estimating Vegas of a T_n European swaption with $F = 3$ in the cmSMM.

n	Time	Linear Fit
3	0.59	0.59
5	0.88	0.86
10	1.55	1.54
15	2.17	2.22
20	2.92	2.89
25	3.56	3.57

TABLE 7.4. Timings for estimating Vegas of a T_0 European swaption with $F = 3$ in the ciSMM.

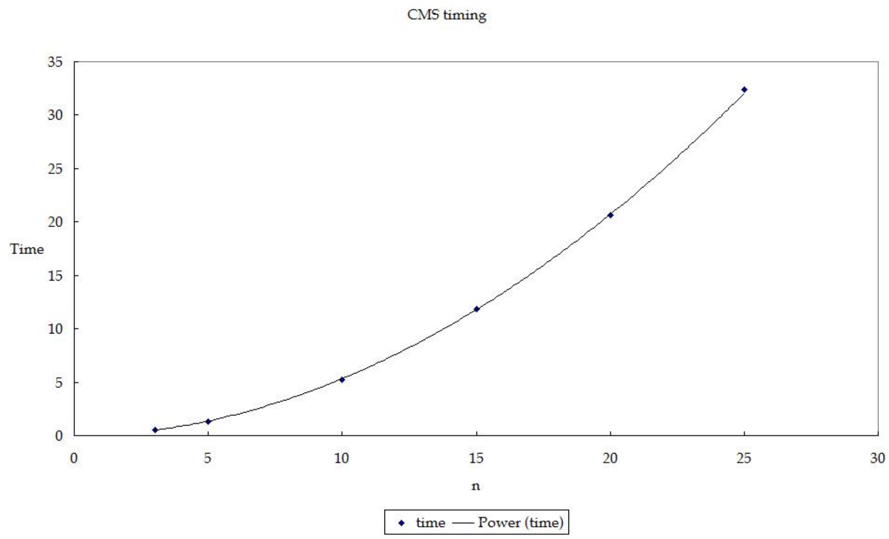

FIGURE 1. Graphs of n against time of estimating Deltas of a T_n European swaption with $F = 3$ in the cmSMM.

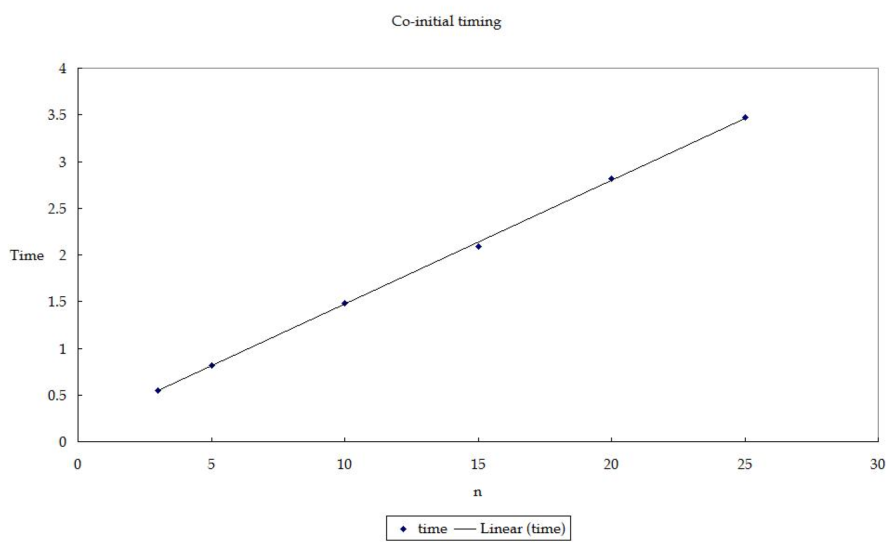

FIGURE 2. Graphs of n against time of estimating Deltas of a T_0 European swaption with $F = 3$ in the ciSMM.

8. CONCLUSION

We have presented efficient algorithms to implement the adjoint method to estimate Deltas of IRDs in the cmSMM and the ciSMM. We have also shown

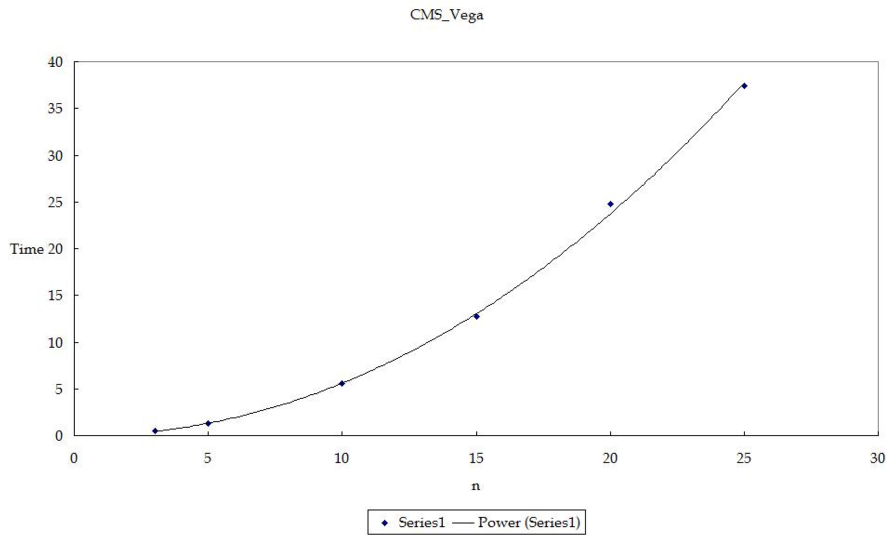

FIGURE 3. Graphs of n against time of estimating Vegas of a T_n European swaption with $F = 3$ in the cmSMM.

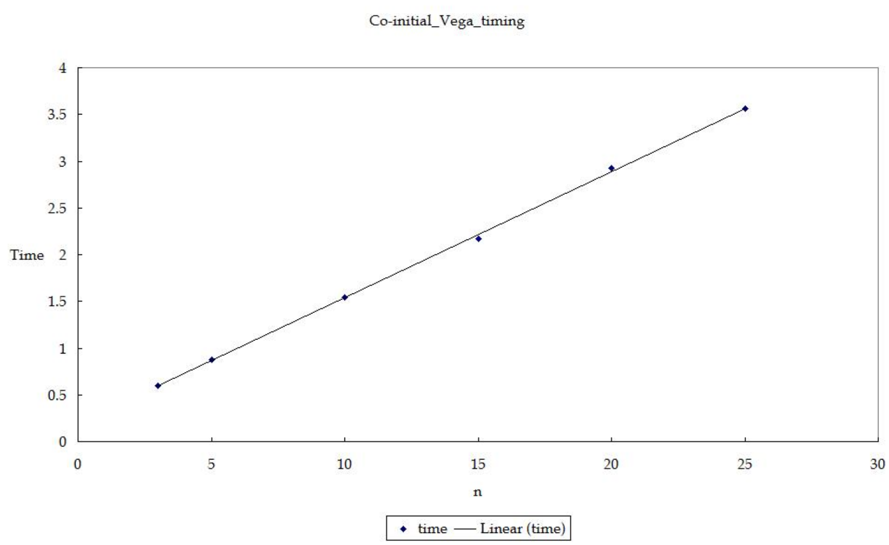

FIGURE 4. Graphs of n against time of estimating Vegas of a T_0 European swaption with $F = 3$ in the ciSMM.

how to extend the method to compute Vegas of IRDs in several generic market models including the coterminal-swap rate market model.

The timing tests confirm that the computational complexity is $O(nF)$ per step in generic market models, and that it does not take substantial additional time

Number of market Vegas	Ratio
1	1.298
5	1.380
10	1.366
20	1.347
40	1.355
60	1.387

TABLE 7.5. Ratios of the time required to compute market Vegas only to the time required to compute Deltas only in the cmSMM.

to compute Vegas once all the adjoint vectors have been computed in the Delta computations.

BIBLIOGRAPHY

[1] A. Brace, D. Gatarek & M. Musiela. (1997). The market model of interest rate dynamics. Mathematical Finance, 7, 127–155.
[2] A. Brace. (2009). *Engineering BGM,* Chapman & Hall/CRC Press.
[3] D. Brigo & F. Mercurio. (2006). *Interest Rate Models: Theory and Practice.* 2nd edn. Springer-Verlag, Berlin-Heidelberg-New York.
[4] S. Galluccio & C. Hunter. (2004). The co-initial swap market model. Economic Notes, vol. 33, no. 2, 209–232.
[5] S. Galluccio, J. M. Ly, Z. Huang & O. Scaillet. (2007). Theory and calibration of swap market models. Mathematical Finance, 17, 111–141.
[6] M. Giles & P. Glasserman. (2006). Smoking adjoints: fast Monte Carlo Greeks. Risk, January 2006, 92–96.
[7] P. Glasserman. (2004). *Monte Carlo Methods in Financing Engineering.* Springer-Verlag, Berlin-Heidelberg-New York.
[8] F. Jamshidian. (1997). LIBOR and swap market models and measures. Finance and Stochastics, 1, 293–330.
[9] M. Joshi. (2003). *The Concepts and Practice of Mathematical Finance.* Cambridge University Press.
[10] M. Joshi. (2003). Rapid drift computations in the LIBOR market model. Wilmott, May, 84–85.
[11] M. Joshi & L. Liesch. (2007). Effective implementation of generic market models. ASTIN Bulletin, 37(2), 453–473.
[12] M. Joshi & O. K. Kwon (2009). Monte Carlo market Greeks in the displaced diffusion LIBOR market model. *Forthcoming*.
[13] M. Joshi & C. Yang. (2009). Fast Delta computations in the swap-rate market model. SSRN working paper. Retrieved from: http://ssrn.com/abstract=1401094.
[14] R. Pietersz & M. van Regenmortel. (2005). Generic market models. Finance and Stochastics, vol. 10, issue 4, 507–528.
[15] J. Schoenmakers & B. Coffey (2003). Systematic generation of correlation structure for the LIBOR market model. International Journal of Applied Finance, 6, 1–13.

CENTRE FOR ACTUARIAL STUDIES, DEPARTMENT OF ECONOMICS, UNIVERSITY OF MELBOURNE, VICTORIA 3010, AUSTRALIA
E-mail address: mark@markjoshi.com
E-mail address: c.yang7@pgrad.unimelb.edu.au