Pricing Variance, Gamma and Corridor Swaps
Using Multinomial Trees

Honglei Zhao\textsuperscript{ab} Zhe Zhao\textsuperscript{ab} Rupak Chatterjee\textsuperscript{a} Thomas Lonon\textsuperscript{a}
Ionuț Florescu\textsuperscript{ab} (ifloresc@stevens.edu)

\textsuperscript{a}Financial Engineering Division, Stevens Institute of Technology
Castle Point on Hudson, Hoboken, NJ 07030, USA
\textsuperscript{b}Hanlon Financial Systems Lab, Stevens Institute of Technology
Babbio Center, 525 River St, Hoboken, NJ 07030, USA

Abstract

This article introduces a new methodology to approximate the prices of variance, gamma and corridor swaps in a stochastic volatility framework applicable to any given tree structure. The efficiency of this tree method is based on the decomposing the payoff structure into nested conditional expectations which may be calculated using a single pass through the tree. The total number of calculations is commensurable with the number of tree nodes making it substantially faster than Monte Carlo simulations. We exemplify the methodology using two different tree structures that approximate several types of stochastic volatility models. Furthermore, this methodology is general enough to be applied to any given tree structure. Extensive numerical tests show that the methodology introduced is fast, efficient and accurate. The method was applied to volatility instruments quoted on CBOE.

This article presents an efficient methodology to value variance, gamma and corridor swaps written on assets that are modeled using a stochastic volatility multinomial tree. Stochastic volatility models as well as more complex stochastic processes generally are not approximated using tree structures. Yet recent work (Florescu and Viens [2008]; Lo et al. [2017]; Stroock and Varadhan [2007]) create complex tree structures incorporating the stochastic volatility framework. Throughout the current article we use a generic name “multinomial tree structures” for these complex tree structures.

To the best of our knowledge, using trees to approximate the price of these types of path-dependent derivatives has not been attempted in a stochastic volatility framework. The fair
value of the floating leg is a sum of squared terms, which is typically difficult to implement in a recombining tree like structure. Furthermore, the actual swap contracts are defined using low frequency data, e.g., end of day returns. To obtain a good approximation for the contract price one needs to implement tree steps at a higher frequency than that of the sampled return and this further complicates the calculations. Our methodology is one of the first to address and efficiently solve these computational issues.

This methodology is designed to use recombining tree structures which are used in literature to approximate general types of Markovian processes. Despite the recombining structure and the path dependent nature of the derivative, the number of calculations in our method do not explode exponentially, rather linearly to the number of tree nodes.

The only other variance swap valuation with a tree structure which we could discover is presented in Wu [2008]. The author calibrates a local volatility model with an implied tree and uses the implied local volatilities on tree nodes to calculate the expected future realized variance. As we will show our method of calculating the expected realized variance throughout the tree is very different.

LITERATURE

Pricing variance swaps and other exotic swaps is a relatively recent problem. The floating leg of a swap contract is defined using discretely observed underlying prices. Therefore, the rate of the fixed leg should be calculated using these discrete future values. However, it is much easier mathematically to use continuously sampled observations (Howison et al. [2004]; Carr and Sun [2007]; Drimus [2012]; Cont and Kokholm [2013]). These continuously sampled approximating swap rates only perform well for swaps with short maturity and high frequency of calculating the discrete returns (Little and Pant [2001]; Sepp [2008]). The papers using a continuously sampled approximation typically use transformation methods (e.g., Fourier, Laplace, etc.).

In contrast, pricing discretely sampled swaps is a more complicated problem and analytical approaches have only been proposed for certain stochastic volatility models. Thus far, analytical formulas for the variance swap which are taking advantage of the specific form of the models have been proposed for Heston (Swishchuk [2004]; Zhu and Lian [2011]), Hull-White, Stein-Stein (Bernard and Cui [2014]), and 3/2 (Zheng and Kwok [2014]).

Merener and Vicchi [2015] introduce an efficient Monte Carlo method to price discretely sampled variance swaps. However, in general Monte Carlo methods are slower than tree methods. The recombining property of trees largely reduces the number of paths needed to obtain a good approximation. In literature, multinomial tree structures have been proposed to approximate many stochastic volatility models (Florescu and Viens [2008]; Vellekoop et al. [2009]; Costabile and Massabó [2010]; Yuen and Yang [2010]; Liu [2010]; Costabile et al. [2012a,b]; Jiang et al. [2016]; Lo et al. [2017]).

For our methodology we assume that an underlying multinomial tree structure is given
which approximates a stochastic volatility model (or indeed any stochastic process).

**VARIANCE SWAPS**

A variance swap is an exchange of a variable cash amount (called the floating leg) related to the volatility of an instrument with a fixed amount called the fixed leg or strike \( (K) \) in the case of a single payment (Carr and Lee [2009]). At the contract maturity \( T \) the fixed leg side pays the agreed upon strike price of the swap and receives a floating amount. The problem is to calculate the fair strike \( K \) of the contract.

The floating amount typically depends on squared returns over fixed intervals in time. Variance swaps are much more common than the volatility swaps due to the convexity correction needed to price the nonlinear nature of the volatility swaps. The realized variance is calculated in two ways either using log returns or simple returns (Carr and Lee [2009]). Both realized variance types use discretely sampled observations of the underlying asset price over the time interval of the swap \([t_0, T]\). The times of the observations are specified in the swap contract.

We denote the observation times \( t_0 < t_1 < \cdots < t_n = T \) and the corresponding underlying asset prices \( S_i = S_{t_i} \), with \( i \in \{0, \ldots, n\} \). The two realized variance types are denoted:

\[
\sigma^2_{\text{log}} := \frac{N}{n} \sum_{i=1}^{n} \ln^2 \left( \frac{S_i}{S_{i-1}} \right) \cdot 100^2 \\
\sigma^2_{\text{simple}} := \frac{N}{n} \sum_{i=1}^{n} \left( \frac{S_i - S_{i-1}}{S_{i-1}} \right)^2 \cdot 100^2
\]

\( N \) is the total number of observations per year that depends on the frequency of observations. For example, for daily returns \( N = 252 \), quarterly returns \( N = 4 \), etc. The variances are multiplied with \( 100^2 \) so that the results are quoted in variance points.

Recently, variations of these two contracts have been introduced to investors (Zheng and Kwok [2014]; Yuen et al. [2015]). These are called gamma swaps, corridor swaps, and downside swaps, and their floating leg are, respectively,

\[
\sigma^2_{\text{gamma}} := \frac{N}{n} \sum_{i=1}^{n} \frac{S_i}{S_0} \ln^2 \left( \frac{S_i}{S_{i-1}} \right) \cdot 100^2 \\
\sigma^2_{\text{corridor}} := \frac{N}{n} \sum_{i=1}^{n} \ln^2 \left( \frac{S_i}{S_{i-1}} \right) 1\{L \leq S_{i-1} \leq U\} \cdot 100^2 \\
\sigma^2_{\text{downside}} := \frac{N}{n} \sum_{i=1}^{n} \ln^2 \left( \frac{S_i}{S_{i-1}} \right) 1\{S_{i-1} \leq U\} \cdot 100^2
\]

where \( L \) and \( U \) are values specified in the respective swap contracts. A downside swap is a special case of a corridor swap with \( L = -\infty \).

For simplicity, throughout the rest of the paper we denote with \( \sigma^2_R \) the floating part of the swap contract. \( \sigma^2_R \) may be any one of the quantities above. If we denote with \( N \) the notional
amount of the swap in dollars per annualized volatility point squared, then the payoff of a variance swap at time $t = T$ is given by:

$$\text{Payoff} = (\sigma_R^2 - K) \cdot \mathcal{N}$$  \hspace{1cm} (6)

The floating leg $\sigma_R^2$ is not completely known until the time $t = T$, while the strike $K$ and the notional amount $\mathcal{N}$ are specified in the contract at time $t = t_0$.

Writing a swap contract requires calculating a fair value of the strike agreeable by both parties. Using standard no-arbitrage arguments the contract should have zero expected value at time $t_0$. Thus, the fair value of the strike price is:

$$K = E^Q \left[ \sigma_R^2 \mid \mathcal{F}_{t_0} \right]$$  \hspace{1cm} (7)

where $E^Q$ denotes the expectation under the risk neutral martingale measure $Q$ and $\mathcal{F}_{t_0}$ denotes the information up to time $t_0$. We assume all models in this paper are defined under the equivalent martingale measure $Q$, so we suppress the superscript $Q$ for the rest of the paper.

**SWAP PRICING METHODOLOGY**

Our methodology may be described as decomposing $E[\sigma_R^2 \mid \mathcal{F}_{t_0}]$ into components that may be evaluated using a single pass through the tree. We give our notations and present the methodology in Theorem 1.

**Iterative Decomposition of the Swap Valuation**

We rewrite (7) as:

$$E \left[ \sigma_R^2 \mid \mathcal{F}_{t_0} \right] = \frac{N}{n} E \left[ \sum_{i=1}^{n} g(\ln S_i, \ln S_{i-1}) \mid \mathcal{F}_{t_0} \right] \cdot 100^2$$

$$= \frac{N}{n} E \left[ g(\ln S_1, \ln S_0) + E \left[ \sum_{i=2}^{n} g(\ln S_i, \ln S_{i-1}) \mid \mathcal{F}_{t_1} \right] \mid \mathcal{F}_{t_0} \right] \cdot 100^2$$

$$= \frac{N}{n} E \left[ g(\ln S_1, \ln S_0) + \cdots + E \left[ g(\ln S_n, \ln S_{n-1}) \mid \mathcal{F}_{t_n} \right] \cdots \mid \mathcal{F}_{t_1} \right] \mid \mathcal{F}_{t_0} \right] \cdot 100^2$$  \hspace{1cm} (8)

The function $g$ in the expectation is one of the following,

$$g_{\log}(X_i, X_{i-1}) = (X_i - X_{i-1})^2$$  \hspace{1cm} (9)

$$g_{\text{simple}}(X_i, X_{i-1}) = (e^{X_i} - X_{i-1} - 1)^2$$  \hspace{1cm} (10)

$$g_{\text{gamma}}(X_i, X_{i-1}) = e^{X_i - X_0} (X_i - X_{i-1})^2$$  \hspace{1cm} (11)

$$g_{\text{corridor}}(X_i, X_{i-1}) = (X_i - X_{i-1})^2 1_{\{\ln L \leq X_{i-1} \leq \ln U\}}$$  \hspace{1cm} (12)

depending on the specific contract priced.
Notations

For simplicity we use an additive tree structure in $X_t = \ln S_t$. In the case of a multiplicative tree, we can simply take the logarithm of the node values and apply the same methodology. We call the tree steps corresponding to the times $\{t_0, t_1, \ldots, t_n\}$ at which the swap contract is calculated observation steps. Let $\Gamma_i$ denote the set of possible log prices that the underlying takes at each observation step $i$, that is, $X_i = \ln S_i \in \Gamma_i$, $i = 0, \ldots, n$.

Any step in the tree other than observation steps will be called an approximation step. We assume there are $\ell - 1$ time steps between any two consecutive observation steps, $\ell \geq 1$. When $\ell = 1$, there is no approximation step in the tree structure and every time corresponds to an observation step. When $\ell > 1$, we have $\ell - 1$ approximation steps between consecutive observation steps and we denote the corresponding times with $t_{i,j}$:

$$t_i = t_{i,0} < t_{i,1} < \cdots < t_{i,\ell - 1} < t_{i,\ell} = t_{i+1} = t_{i+1,0}$$

for $i \in \{1, 2, \ldots, n - 1\}$. Random variables $X_{i,j} = \ln S_{i,j}$ and the set $\Gamma_{i,j}$ of possible outcomes are defined correspondingly for every observation/approximation step. The method works as well in the case when there isn’t a uniform number of approximation steps (when $\ell_i \neq \ell$ for all $i$). However, for simplicity of notation we keep the same number of approximation steps between any two consecutive observation steps. Furthermore, we will show that increasing $\ell$ improves the swap pricing results.

For two nodes $x \in \Gamma_{i,j}$ and $y \in \Gamma_{i,j+m}$ we denote the the $m$-step probability from $x$ to $y$:

$$P_{xy}(m) := P\{X_{i,j} = x \mid X_{i,j+m} = y\}, m \leq \ell - j$$

If there is no path in the tree structure that connects $x$ and $y$, then $P_{x,y}(m) = 0$. In a recombining tree structure, nodes may be connected by more than one path. Exhibit 1 depicts different paths from $x$ to $y$ and all of them will be considered when calculating $P_{xy}(3)$.

The method will store four quantities at each and every node $x \in \Gamma_{i,j}$. We denote these quantities: $V_x$, $A_x$, $B_x$ and $C_x$. The $V_x$ number is only updated at observation steps (when $j = 0$). This quantity will store the value of the estimate of the expected floating leg aggregated from the terminal nodes to node $x$. Specifically:

$$V_x = \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \mid X_i = x \right]$$

Please note that we use $\hat{E}$ to emphasize that this is the approximate expectation calculated using the tree nodes. In Theorem 1, we will show that as the tree converges to the continuous time process when $\ell \to \infty$, the estimate $V_{x_0}$ will converge to $E [\sum_{i=1}^{n} g(X_i, X_{i-1}) | \mathcal{F}_0]$. 


Swaps Pricing

Calculating the strike involves calculating a nested conditional expectation. For example, in Exhibit 1, $V_x$ is calculated using

$$V_x = \hat{E} \left[ \sum_{s=i+1}^n g(X_s, X_{s-1}) \bigg| X_i = x \right]$$

$$= \hat{E} \left[ g(X_{i+1}, X_i) + \hat{E} \left[ \sum_{s=i}^n g(X_s, X_{s-1}) \bigg| X_{i+1} = y \right] \bigg| X_i = x \right]$$

$$= \sum_{\forall y \in \Gamma_{i+1}} P_{xy}(3) (g(y, x) + V_y)$$

For each pair $x$ and $y \in \Gamma_{i+1}$, $g(y, x)$ can be easily evaluated and the quantity $V_y$ is already calculated and stored at each $y$. However, determining the probability $P_{xy}(3)$ is complicated.

Exhibit 1: Six different paths need to be used to calculate the probability $P_{xy}(3)$.

The cannonical approach is finding all paths between $(x, y)$ and calculating the product of probabilities for each such path. This approach is inefficient since the computation grows exponentially with the number of approximating steps, and it does not take advantage of the recombining feature of the tree structure. The following theorem is proposing an efficient way to calculate $V_x$ by taking advantage of the specific form of the functions $g$ defining the swap contracts (equations (9) to (12)).

**Theorem 1** Let the log price process $X_t$ and a continuous function $g: \mathbb{R}^2 \to \mathbb{R}$ define the floating leg of the swap contract. Specifically, the fair strike price is:

$$\frac{N}{n} \hat{E} \left[ \sum_{i=1}^n g(X_i, X_{i-1}) \bigg| \mathcal{F}_0 \right] \times 100^2 \quad (14)$$

where $X_0, X_1, \ldots, X_n$ are log prices at time $t_0, t_1, \ldots, t_n$, respectively.
Suppose we are given a multinomial tree structure that approximates the process \( \{X_t\}_t \). The total number of steps in the tree is \( n \times \ell \) where \( n \) is the number of observation steps and \( \ell - 1 \) is the number of approximation steps as in (13). Assume that the function \( g(y, x) \) defining the swap contract can be written as:

\[
g(y, x) = h_1(y) f_1(x) + h_2(y) f_2(x) + h_3(y) + f_3(x)
\]

where the functions \( h_1, h_2, h_3, f_1, f_2 \) and \( f_3 \) are all defined on \( \mathbb{R} \to \mathbb{R} \).

We start at step \( T = t_{n,\ell} \) and initialize all quantities \( V_x, A_x, B_x, C_x \) with \( 0, \forall x \in \Gamma_n \). Working recursively from \( i + 1 \) to \( i \), we update the quantities \( A_x, B_x, C_x \) and \( V_x \) at each node \( x \in \Gamma_{i,j} \) using:

\[
A_x = \begin{cases} \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) h_1(y), & j = \ell - 1, \ell \geq 1 \\ \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) A_y, & j \in \{0, 1, 2, \ldots, \ell - 2\}, \ell > 1 \end{cases}
\]

\[
B_x = \begin{cases} \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) h_2(y), & j = \ell - 1, \ell \geq 1 \\ \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) B_y, & j \in \{0, 1, 2, \ldots, \ell - 2\}, \ell > 1 \end{cases}
\]

\[
C_x = \begin{cases} \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) (h_3(y) + V_y), & j = \ell - 1, \ell \geq 1 \\ \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1) C_y, & j \in \{0, 1, 2, \ldots, \ell - 2\}, \ell > 1 \end{cases}
\]

\[
V_x = \begin{cases} A_x f_1(x) + B_x f_2(x) + C_x + f_3(x), & j = 0 \\ 0, & j \in \{1, 2, \ldots, \ell - 2\} \end{cases}
\]

where \( i = n - 1, n - 2, \ldots, 0, j = \ell - 1, \ell - 2, \ldots, 0 \). Note that the quantity \( V_x \) is only updated at the observation steps (when \( j = 0 \)).

Then the resulting \( V_{x_0} \) is the estimate of:

\[
V_{x_0} = \hat{E} \left[ \sum_{i=1}^{n} g(X_i, X_{i-1}) \mid F_0 \right]
\]

and

\[
V_{x_0} \to E \left[ \sum_{i=1}^{n} g(X_i, X_{i-1}) \mid F_0 \right], \text{ as } \ell \to \infty
\]

where the convergence is a pointwise convergence. Finally, the estimated fair strike from the tree structure is \( \frac{N}{n} V_{x_0} \cdot 100^2 \), and it converges to the fair strike price:

\[
\frac{N}{n} V_{x_0} \cdot 100^2 \to K, \text{ as } \ell \to \infty
\]

The proof of this theorem is given in Appendix A.

**Remark 1** Formula (16) essentially rearranges the terms in a very efficient way that allows a single pass through the tree to calculate \( \hat{E}[\sigma^2|F_0] \). The number of calculations performed is commensurable with the number of nodes in the tree, not the number of paths in the tree. This is a significant advantage due to the recombining nature of the multinomial tree structures which generally have a polynomial number of nodes and an exponential number of paths.
Theorem 1 may be used to price simple and log variance swap as well as corridor and gamma swaps. The key in the calculation is decomposition (15) of the function $g$. Exhibit 2 summarizes this decomposition for all the swap types considered ((9) to (12)):

**Exhibit 2: Decomposition of function $g(y, x)$**

<table>
<thead>
<tr>
<th></th>
<th>$g_{log}$</th>
<th>$g_{simple}$</th>
<th>$g_{gamma}$</th>
<th>$g_{corridor}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1(y)$</td>
<td>1</td>
<td>$e^{2y}$</td>
<td>$e^{y-x_0}$</td>
<td>$y^2$</td>
</tr>
<tr>
<td>$h_2(y)$</td>
<td>$-2y$</td>
<td>$-2e^y$</td>
<td>$-2ye^{y-x_0}$</td>
<td>$-2y$</td>
</tr>
<tr>
<td>$h_3(y)$</td>
<td>$y^2$</td>
<td>1</td>
<td>$y^2e^{y-x_0}$</td>
<td>0</td>
</tr>
<tr>
<td>$f_1(x)$</td>
<td>$x^2$</td>
<td>$e^{-2x}$</td>
<td>$x^2$</td>
<td>$1_{{L \leq x \leq U}}$</td>
</tr>
<tr>
<td>$f_2(x)$</td>
<td>$x$</td>
<td>$e^{-x}$</td>
<td>$x$</td>
<td>$x1_{{L \leq x \leq U}}$</td>
</tr>
<tr>
<td>$f_3(x)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$x^21_{{L \leq x \leq U}}$</td>
</tr>
</tbody>
</table>

**Implementing the Methodology**

For the benefit of the reader who wants to implement the methodology we provide pseudo-code in Exhibit 3. Note that different expressions are used at observation steps and approximation steps.

**Exhibit 3 Pseudo-Code to implement Theorem 1**

```
for all $x \in \Gamma_n$, do $V_x = 0$
for $i = n - 1$ to 0 do
    for $j = l - 1$ to 0 do
        for all $x \in \Gamma_{i,j}$, do
            if $(j = l - 1, l > 1)$ or $(l = 1)$ then
                $A_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)h_1(y)$
                $B_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)h_2(y)$
                $C_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)(h_3(y) + V_y)$
            else
                $A_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)A_y$
                $B_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)B_y$
                $C_x = \sum_{y \in \Gamma_{i,j+1}} P_{xy}(1)C_y$
            if $j = 0$ then
                $V_x = A_x f_1(x) + B_x f_2(x) + C_x + f_3(x)$
        return $\sum_{n} V_{x_0} \cdot 100^2$
```
A Proof of Theorem 1

Proof. We firstly show that, at all observation steps, the quantities $V$ calculated from (16) satisfy

$$V_x = \sum_{y \in \Gamma_{i+1}} P_{xy}(\ell) (g(y, x) + V_y), \forall x \in \Gamma_i, i \in \{n-1, n-2, \ldots, 0\}$$

Then we use backward induction to prove that for every $x \in \Gamma_i, i \in \{n-1, n-2, \ldots, 0\}$

$$V_x = \hat{E} \left[ \sum_{s=i+1}^n g(X_s, X_{s-1}) \bigg| X_i = x \right]$$

Due to the fact that the discrete processes represented by the tree structure converge to the continuous stochastic process as $\ell \to \infty$ and that $\sigma_R^2$ in (1) to (5) are defined with continuous functions, it is enough to ensure that

$$V_{x_0} = \hat{E} \left[ \sum_{i=1}^n g(X_i, X_{i-1}) \bigg| \mathcal{F}_0 \right] \to E \left[ \sum_{i=1}^n g(X_i, X_{i-1}) \bigg| \mathcal{F}_0 \right], \text{ as } \ell \to \infty$$

If we look into the definitions of $A_x, B_x$ and $C_x, \forall x \in \Gamma_{i, \ell-1}$, it is easy to verify that they satisfy

$$\sum_{y \in \Gamma_{i, \ell}} P_{xy}(1) (g(y, x) + V_y) = A_x f_1(x) + B_x f_2(x) + C_x + f_3(x)$$

Now consider nodes in step $\Gamma_{i, \ell-2}, \forall x \in \Gamma_{i, \ell-2}$. The definitions of $A_x, B_x$ and $C_x$ in (16) may be easily checked to show that,

$$\sum_{y \in \Gamma_{i, \ell-1}} P_{xy}(1) \sum_{z \in \Gamma_{i, \ell}} P_{yz}(1) (g(z, x) + V_z) = A_x f_1(x) + B_x f_2(x) + C_x + f_3(x)$$

Similar equations can be built for all steps $\Gamma_{i, j}, j = \ell - 2, \ell - 3, \ldots, 1$. Finally, at step $\Gamma_{i, 0}$, $\forall x \in \Gamma_{i, 0}$ we have

$$V_x = \sum_{w} P_{xw}(\ell) (g_{\log}(w, x) + V_w)$$

$$= \sum_{y \in \Gamma_{i, 1}, z \in \Gamma_{i, 2}, \ldots, w \in \Gamma_{i, \ell-1}, u \in \Gamma_{i, \ell}} P_{xy}(1) P_{yz}(1) \cdots P_{uw}(1) (g_{\log}(w, x) + V_w)$$

$$= \sum_{y \in \Gamma_{i, 1}} P_{xy}(1) \sum_{z \in \Gamma_{i, 2}} P_{yz}(1) \cdots \sum_{u \in \Gamma_{i, \ell-1}} P_{uw}(1) \sum_{v \in \Gamma_{i, \ell}} P_{vw}(1) (g_{\log}(w, x) + V_w)$$

$$= \sum_{w \in \Gamma_{i, \ell}} P_{xw}(\ell) A_w f_1(x) + \sum_{w \in \Gamma_{i, \ell}} P_{xw}(\ell) B_w f_2(x) + \sum_{w \in \Gamma_{i, \ell}} P_{xw}(\ell) C_w + f_3(x)$$

$$= A_x f_1(x) + B_x f_2(x) + C_x + f_3(x)$$

Therefore, at all observation steps $t_{n-1}, t_{n-2}, \ldots, t_0$, the quantities $V$ calculated from (16) satisfy

$$V_x = \sum_{y \in \Gamma_{i+1}} P_{xy}(\ell) (g(y, x) + V_y), \forall x \in \Gamma_i, i \in \{n-1, n-2, \ldots, 0\}$$  (18)
We now use backward induction to prove that for every \( x \in \Gamma_i, i \in \{n - 1, n - 2, \ldots, 0\} \)

\[
V_x = \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_i = x \right. \right]
\]

For the initial step of the induction, \( i = n - 1, \forall x \in \Gamma_{n-1} \), using the defining relation (18),

\[
V_x = \sum_{y \in \Gamma_n} P_{xy}(\ell) (g(y, x) + 0) = \hat{E} \left[ g(X_n, X_{n-1}) \left| X_{n-1} = x \right. \right]
\]

For the induction step, assume \( \forall x \in \Gamma_i, i \in \{n - 1, n - 2, \ldots, 1\} \)

\[
V_x = \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_i = x \right. \right]
\]

Then, \( \forall x \in \Gamma_{i-1} \), we have

\[
V_x = \sum_{y \in \Gamma_i} P_{xy}(\ell) (g(y, x) + V_y)
\]

\[
= \sum_{y \in \Gamma_i} P_{xy}(\ell) \left( g(y, x) + \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_i = y \right. \right] \right)
\]

\[
= \sum_{y \in \Gamma_i} P_{xy}(\ell) g(y, x) + \sum_{y \in \Gamma_i} P_{xy}(\ell) \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_i = y \right. \right]
\]

\[
= \hat{E} \left[ g(X_i, X_{i-1}) \left| X_{i-1} = x \right. \right] + \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_{i-1} = x \right. \right]
\]

\[
= \hat{E} \left[ \sum_{s=i+1}^{n} g(X_s, X_{s-1}) \left| X_{i-1} = x \right. \right]
\]

Therefore, we can conclude the iterative argument to obtain:

\[
V_{x_0} = \hat{E} \left[ \sum_{i=1}^{n} g(X_i, X_{i-1}) \left| X_0 = x_0 \right. \right] = \hat{E} \left[ \sum_{i=1}^{n} g(X_i, X_{i-1}) \left| F_0 \right. \right]
\]

Now we prove the convergence in (17). We use the fact that the discrete process \( \{\hat{X}_{i,j}\}_{i,j} \) approximated by the tree structure converges to the continuous process \( \{X_t\}_t \) as the number of steps increases to infinity. In our case the total number of tree steps is \( n \times \ell \). We let \( \ell \to \infty \) since \( n \) is fixed by the swap contract. Because function \( g \) defining the floating leg is continuous, by the continuous mapping theorem (Florescu [2014], Theorem 7.33), we can get \( \sum_{i=1}^{n} g(\hat{X}_i, \hat{X}_j) \to \sum_{i=1}^{n} g(X_i, X_j) \) in distribution, as \( \ell \to \infty \). Therefore the expectation \( V_{x_0} \hat{E} \sum_{i=1}^{n} g(\hat{X}_i, \hat{X}_j) \to \sum_{i=1}^{n} g(X_i, X_j) \).

**ENDNOTES**

We thank Dr. Zhenyu Cui for his broad literature review on variance swaps and the idea of exotic variance swap products.
References


