

FX Double Window Double Barrier Option Model

An FX Double Window Double Barrier option can be seen as a special case of a complex barrier option, where we can have multiple single and double partial barrier (partial barrier meaning: the barrier is effective only on a subinterval of the full option term).

We offer a hybrid (trinomial tree plus semi-analytic formulas) pricing method for the product. Currently, the model uses spot implied volatility for the first time window, and forward implied volatility for the second time window. These are Black-Scholes implied volatilities from traded vanilla European options, but, while desired, usage of different volatilities based on strike or barrier moneyness directly within the tree is difficult to achieve.

We will start by defining **Single Knock Out Partial Barrier Option, Single Knock In Barrier Option, Double Knock Out Partial Barrier Option, and Double Knock In Barrier Option** contracts on whose features the **Double Window Double Barrier Option** contract definition is based. The exercise style is assumed European everywhere in this paper.

Assume an underlying price (FX rate in our case) process, S_t , an expiry date, $T > 0$, and a strike price (rate), K . Assume the underlying follows the SDE below:

$$dS_t = (r - q) \cdot S_t dt + \sigma \cdot S_t dW_t, \quad (1)$$

where r is the (continuously compounded) risk-free rate (domestic rate for FX options), q is the continuous dividend yield (foreign risk-free rate for FX options), σ is the volatility of the underlying, and W_t is standard one-dimensional Brownian process.

The **Single Knock Out Partial Barrier (SKOPB) Option** has one single partial knock out barrier. Let B be the single barrier and let $[T_1, T_2]$, $0 \leq T_1 \leq T_2 \leq T$, be the interval on which this barrier is effective. The possible types of Single Knock Out Partial Barrier Option contract are: up-out-call/put (UOC/P), down-out-call/put (DOC/P). If the barrier interval is effective over

the whole term of the option, we obtain the basic **Single Knock Out Barrier (SKOB) Option** contract. Define the minimum and maximum of the underlying process over a given time interval as follows:

$$m(T_1, T_2) = \min \{S_t : T_1 \leq t \leq T_2\}, \quad M(T_1, T_2) = \max \{S_t : T_1 \leq t \leq T_2\}.$$

The payoffs of such contracts are then defined as follows:

$$UOC/P - \text{payoff}(T) = (\beta \cdot (S_T - K))^+ \cdot 1_{\{M(T_1, T_2) < B\}},$$

$$DOC/P - \text{payoff}(T) = (\beta \cdot (S_T - K))^+ \cdot 1_{\{m(T_1, T_2) > B\}},$$

where $\beta = 1$ if call, $\beta = -1$ if put, and $1_{\{condition\}} = 1$ if *condition* is true, $1_{\{condition\}} = 0$ if *condition* is false.

The **Single Knock In Barrier (SKIB) Option** has one knock in barrier effective for the whole term of the option (we do not define here a general partial version for knock in). Let B be the single barrier. The possible types of Single Knock In Barrier Option contract are: up-in-call/put (UIC/P), down-in-call/put (DIC/P). The corresponding payoffs are then:

$$UIC/P - \text{payoff}(T) = (\beta \cdot (S_T - K))^+ \cdot 1_{\{M(0, T) > B\}},$$

$$DIC/P - \text{payoff}(T) = (\beta \cdot (S_T - K))^+ \cdot 1_{\{m(0, T) < B\}}.$$

The **Double Knock Out Partial Barrier (DKOPB) Option** contract has two partial knock out barriers. Let H and L be the two partial knock out barriers (**no order specified for H and L**), and the intervals on which they are effective be $[T_1^H, T_2^H]$, $0 \leq T_1^H \leq T_2^H \leq T$, and $[T_1^L, T_2^L]$, $0 \leq T_1^L \leq T_2^L \leq T$ (they may be overlapped or disjoint).

The possible types are: H-up-L-up-out-call/put (HULUOC/P), H-up-L-down-out-call/put (HULDOC/P), H-down-L-up-out-call/put (HDLUOC/P), and H-down-L-down-out-call/put (HDLDOC/P). If H is bigger than L , H is up, L is down, and their intervals cover the whole option term, then we obtain the basic **Double Knock Out Barrier (DKOB) Option**. The payoffs of such contracts are:

$$HULUOC/P - payoff(T) = (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{M(T_1^H, T_2^H) < H \text{ and } M(T_1^L, T_2^L) < L\}},$$

$$HULDOC/P - payoff(T) = (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{M(T_1^H, T_2^H) < H \text{ and } m(T_1^L, T_2^L) > L\}},$$

$$HDLUOC/P - payoff(T) = (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{m(T_1^H, T_2^H) > H \text{ and } M(T_1^L, T_2^L) < L\}},$$

$$HDLDOC/P - payoff(T) = (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{m(T_1^H, T_2^H) > H \text{ and } m(T_1^L, T_2^L) > L\}}.$$

The **Double Knock In Barrier (DKIB) Option** contract has two total knock in barriers (we do not define here a general partial version for knock in). Let H and L be the two knock in barriers, **with H bigger than L** . Only two types are considered: H-up-L-down-in-call/put (HULDIC/P).

The corresponding payoffs are then:

$$HULDIC/P - payoff(T) = (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{M(0,T) > H \text{ or } m(0,T) < L\}}.$$

We can now define the **Double Window Double Barrier (DWDB) Option** contract. The term of the option is split into two windows. Let $0 \leq T' \leq T$ be the end date of the first window. The first window, $[0, T']$, has the characteristics of a DKOPB (or SKOPB, if one of the barriers is not available), that is, we have partial knock out barriers H and L with effective time intervals $[T_1^H, T_2^H]$, $0 \leq T_1^H \leq T_2^H \leq T'$, and $[T_1^L, T_2^L]$, $0 \leq T_1^L \leq T_2^L \leq T'$ (or just one partial knock out barrier, B , with effective time interval $[T_1, T_2]$, $0 \leq T_1 \leq T_2 \leq T'$).

The second window, $[T', T]$, has the characteristics of a DKOB (SKOB) or DKIB (SKIB), that is, we have total (effective over the whole second window) knock out (knock in) barriers H_2 and L_2 , with $H_2 > L_2$ (or just one total knock out (knock in) barrier for the whole second window, B_2). For example, the payoff of a DWDB call with HULUOC/P characteristics for the first window, and HULDIC/P characteristics for the second window is:

$$\begin{aligned} DWDBCall - payoff(T) &= (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{M(T_1^H, T_2^H) < H \text{ and } M(T_1^L, T_2^L) < L\}} \cdot \mathbf{1}_{\{M(T', T) > H_2 \text{ or } m(T', T) < L_2\}} \\ &= (\beta \cdot (S_T - K))^+ \cdot \mathbf{1}_{\{M(T_1^H, T_2^H) < H \text{ and } M(T_1^L, T_2^L) < L\} \cap \{M(T', T) > H_2 \text{ or } m(T', T) < L_2\}}, \end{aligned}$$

that is, both the HULUOC/P condition **and** HULDIC/P condition, verified at option expiry date, determine the payoff. All other combinations permitted by the DWDB contract described above, can be presented in the same fashion.

There are analytic formulas for SKOB and SKIB options, and particular types of SKOPB and SKIPB options (either the start of the barrier interval is the valuation date or the end of the barrier interval is the expiry date). There are also semi-analytic formulas for DKOB and DKIB options. The DKOPB and SKOPB options can only be treated using a trinomial tree pricing. These two types of options are the main focus of this paper. We will use the above products that possess (semi) analytic formulas for testing the convergence of our lattice pricing.

- **Trinomial tree pricing for Multiple Single Partial Barrier (MSPB) Options**

The trinomial tree discretizing our underlying dynamics (1) is based on a common market practice. We parameterize our trinomial tree (shift it spatially in log scale) so that the barriers match underlying values. (In fact, it should be seen as a discretization of the Black-Scholes PDE associated to (1), thus allowing for negative “probabilities” for its branches.)

Tree state building

Assume a number of n time steps, and assume $H_i, i = 0, 1, \dots, n$ a series of barriers applicable at each time level. Let $\Delta t = \frac{T}{n}$ be the length of the time step. Let S_i be the value of the underlying at time

$i \cdot \Delta t$. Let $\Delta \log S = \lambda \cdot \sigma \cdot \sqrt{\Delta t}$ be the spatial step in log scale, where we choose $\lambda = \frac{1}{2} \sqrt{2\pi}$

(standard choice that assures the convergence of the standard trinomial tree). Consider the

underlying value at time level $i \in \{0, 1, 2, \dots, n\}$ and space level

$j \in \{-i, -i+1, \dots, 0, \dots, i-1, i\}$, denoted by $S_{i,j}$. If the underlying is in state $S_{i,j}$ at time level i , the

three values the underlying may take at next time level, $i+1$, are $S_{i+1,j+1}$, $S_{i+1,j-1}$, $S_{i+1,j}$. We define

them as follows:

$$\begin{aligned} S_{i+1,j+1} &= S_{i,j} \cdot \exp(\Delta \log S + \log \eta_i), \\ S_{i+1,j-1} &= S_{i,j} \cdot \exp(-\Delta \log S + \log \eta_i), \\ S_{i+1,j} &= S_{i,j} \cdot \exp(\log \eta_i), \end{aligned}$$

where η_i is a parameter which we will use to match barrier H_{i+1} .

Equivalently, from $S_{i,0}$ we construct $S_{i+1,0}$ via

$$S_{i+1,0} = S_{i,0} \eta_i, \quad (2)$$

and then all the other nodes at time level $i+1$ are built by

$$S_{i+1,j} = S_{i+1,0} \cdot \exp(j \cdot \Delta \log S). \quad (3)$$

The relevant states are for $j \in \{-i-1, -i, \dots, 0, \dots, i, i+1\}$, but we will use relation (3) beyond this restriction, in which case we talk about unrestricted underlying values. We want our barrier H_{i+1} to match one of the (possibly unrestricted) underlying values at time level $i+1$, that is, for some integer $j_{H_{i+1}}$ we want the following equality to hold:

$$H_{i+1} = S_{i+1, j_{H_{i+1}}}.$$

This is equivalent to:

$$H_{i+1} = S_{i+1, j_{H_{i+1}}} \Leftrightarrow H_{i+1} = S_{i,0} \cdot \eta_i \cdot \exp(j_{H_{i+1}} \cdot \Delta \log S). \quad (4)$$

Assuming we know $S_{i,0}$ (and automatically all other states at time level i), in (4) we are presented with one equation and two unknowns: real number η_i , and integer $j_{H_{i+1}}$.

We will choose first the integer $j_{H_{i+1}}$ so that H_{i+1} comes as close as possible to an (possibly unrestricted) underlying value at time level $i+1$, assuming that the central node at this time level is equal to the one at the precedent time level, that is, the distance

$$\left| H_{i+1} - S_{i,0} \cdot \exp(j_{H_{i+1}} \cdot \Delta \log S) \right|$$

is minimal. This minimization problem is easily solved by the integer:

$$j_{H_{i+1}} = \left\lfloor \frac{\log H_{i+1} - \log S_{i,0}}{\Delta \log S} + 0.5 \right\rfloor,$$

where $\lfloor x \rfloor$ is the integral part of the real number x (that is, the integer with the following property $x-1 < \lfloor x \rfloor \leq x$; consequently, $x-0.5 < \lfloor x+0.5 \rfloor \leq x+0.5$, which makes $\lfloor x+0.5 \rfloor$ the closest integer to the real number x).

Then η_i is chosen so that it solves exactly equation (4):

$$\eta_i = \frac{H_{i+1}}{S_{i,0}} \cdot \exp\left(- \left\lfloor \frac{\log H_{i+1} - \log S_{i,0}}{\Delta \log S} + 0.5 \right\rfloor \cdot \Delta \log S\right). \quad (5)$$

Note that η_i is very close to 1, which means the tree geometry does not change by much if the chosen mesh is dense enough.

Given $S_{0,0}$, from (2), (3), and (5) we can build, inductively, all possible states for our underlying at all time levels.

Tree probabilities

Let S_i be the value of the underlying at time $i \cdot \Delta t$. From our assumed dynamics (1) we have:

$$\begin{aligned} E[S_{i+1}|S_i = x] &= x \cdot \exp((r - q) \cdot \Delta t), \\ \text{Var}[S_{i+1}|S_i = x] &= x^2 \cdot \exp(2 \cdot (r - q) \cdot \Delta t) \cdot (\exp(\sigma^2 \cdot \Delta t) - 1). \end{aligned} \quad (6)$$

Let us introduce the following notation:

$$\begin{aligned} M &= \exp((r - q) \cdot \Delta t), \\ V &= \exp(2 \cdot (r - q) \cdot \Delta t) \cdot (\exp(\sigma^2 \cdot \Delta t) - 1). \end{aligned}$$

Consider now in our discretization the probabilities of state $S_{i,j}$ branching off to $S_{i+1,j+1}$, $S_{i+1,j-1}$, $S_{i+1,j}$, denoted respectively $p_{u,i}$ (up), $p_{d,i}$ (down), $p_{m,i}$ (middle). Denoting by $u = \exp(\Delta \log S)$, and then using (6), we obtain the following system of equations:

$$\begin{aligned} p_{u,i} \cdot x \cdot u \cdot \eta_i + p_{d,i} \cdot x \cdot u^{-1} \cdot \eta_i + p_{m,i} \cdot x \cdot \eta_i &= x \cdot M, \\ p_{u,i} \cdot x^2 \cdot u^2 \cdot \eta_i^2 + p_{d,i} \cdot x^2 \cdot u^{-2} \cdot \eta_i^2 + p_{m,i} \cdot x^2 \cdot \eta_i^2 - x^2 \cdot M^2 &= x^2 \cdot V, \\ p_{m,i} &= 1 - p_{u,i} - p_{d,i}. \end{aligned}$$

Equivalently:

$$\begin{aligned} p_{u,i} \cdot u + p_{d,i} \cdot u^{-1} + p_{m,i} &= \frac{M}{\eta_i}, \\ p_{u,i} \cdot u^2 + p_{d,i} \cdot u^{-2} + p_{m,i} &= \frac{V}{\eta_i} - \frac{M^2}{\eta_i^2}, \\ p_{m,i} &= 1 - p_{u,i} - p_{d,i}. \end{aligned}$$

Solving this linear system we obtain:

$$\begin{aligned} p_{u,i} &= \frac{u \cdot \left(\frac{V}{\eta_i^2} + \frac{M^2}{\eta_i^2} - \frac{M}{\eta_i} \right) - \left(\frac{M}{\eta_i} - 1 \right)}{(u-1) \cdot (u^2-1)}, \\ p_{d,i} &= \frac{u^2 \cdot \left(\frac{V}{\eta_i^2} + \frac{M^2}{\eta_i^2} - \frac{M}{\eta_i} \right) - u^3 \cdot \left(\frac{M}{\eta_i} - 1 \right)}{(u-1) \cdot (u^2-1)}, \\ p_{m,i} &= 1 - p_{u,i} - p_{d,i}. \end{aligned}$$

Pricing on the tree

Now that the full trinomial tree, states and ramification probabilities, is built, we can use it to price a multiple single barrier option. Let $C_{i,j}$ be the option price at time level $i \in \{0,1,2,\dots,n\}$ and space level $j \in \{-i,-i+1,\dots,0,1,2,\dots,i+1,i\}$. Let $B_{i,j}$ be the barrier condition at time level i and space level j (for example, if H_i is up and out type, then $B_{i,j} = \{H_i > S_{i,j}\}$).

At time level n we have:

$$C_{n,j} = (\beta \cdot (S_{n,j} - K))^+ \cdot 1_{B_{n,j}}.$$

To roll backwards, we use the averaging and discounting relation together with the barrier condition:

$$C_{i,j} = \exp(r \cdot \Delta t) \cdot (p_{u,i} \cdot C_{i+1,j+1} + p_{d,i} \cdot C_{i+1,j-1} + p_{m,i} \cdot C_{i+1,j}) \cdot 1_{B_{i,j}}.$$

The price of the option is then $C_{0,0}$.

- **Pricing SKOPB and DKOPB options**

For SKOPB option pricing, all we need to do is to set $H_i = B \forall i \in \{0,1,\dots,n\}$ and use $(B_{i,j} \cap \{i \cdot \Delta t \in [T_1, T_2]\}) \cup \{i \cdot \Delta t \notin [T_1, T_2]\}$ instead of the above barrier condition $B_{i,j}$.

For DKOPB, we first set $H_i = L \forall i \in \{0,1,\dots,n\}$ and then we modify slightly λ so that the other barrier, H , is matched automatically:

$$\lambda = \frac{\frac{D}{\sigma \sqrt{\Delta t}}}{\left[\frac{D}{(1/2)\sqrt{2\pi}\sigma\sqrt{\Delta t}} + 0.5 \right]},$$

where $D = \log\left(\frac{H}{L}\right)$.

This new λ is still very close to the old one so that the stability and convergence of the tree will not be affected, but also H gets matched by an underlying value. Indeed we have from (4)

$$\log L = \log S_{i,0} + \log \eta_i + j_L \cdot \Delta \log S.$$

Then we want an integer j_H such that

$$\log H = \log S_{i,0} + \log \eta_i + j_H \cdot \Delta \log S .$$

Subtracting we obtain:

$$D = j_H \cdot \Delta \log S - j_L \cdot \Delta \log S ,$$

which leads to:

$$j_H = \frac{D}{\Delta \log S} + j_L = \left\lfloor \frac{D}{(1/2)\sqrt{2\pi}\sigma\sqrt{\Delta t}} + 0.5 \right\rfloor + j_L .$$

- **Hybrid pricing for DWDB option: trinomial tree for first window, (semi) analytic formulas for second window**

The idea is to use the corresponding analytic or semi-analytic formulas for the second window, where we must have a product that admits such pricing (basically, the requirement is that the barriers involved in the second window should be total, that is, being effective from the end date of the first window to the expiry date of the option). We then build a tree only for the first window, using for the nodes at the end date of the first window the (semi) analytic prices we get from the second window. One advantage is that we can use the forward volatility for the second window (this way, we include some volatility term structure in our pricing). For a generalization of this product and its pricing see <https://finpricing.com/lib/EqBarrier.html>