Problems for the Quantathon: 2

The issue time for all options coincides with the initial time. The maturities, barrier, and exercise times are strictly greater than the initial time.

Callable capped floater

N : the notional.

C : the cap rate.

δt : the interval of time between the payments given as year fraction.

M : the total number of payments.

δL : the spread over Libor.

We assume that today is the issue time of the capped floater and denote this time by t_0. Let $L(s,t)$ be the float LIBOR computed at s for maturity t. The payment times are given by

$$t_m = t_0 + m\delta t, \quad m = 1, \ldots, M.$$

At payment time t_{m+1},

1. the holder *receives* the coupon

$$N\delta t \min(L(t_m, t_{m+1}) + \delta L, C),$$

2. the seller of the option *has the right to cancel* the contract. In this case, in addition to the above coupon he pays the notional. No payments will be made in the future. Note that the option can not be terminated at issue time.

If the contract has not been terminated before, then at maturity t_M the holder receives the coupon above plus the notional.

Algorithm. The event times are

$$\left\{t_0, (t_m)_{m=1,\ldots,M-1}\right\},$$

where t_0 is the initial time and $(t_m)_{m=1,\ldots,M-1}$ are all payment times except the last one. We divide the algorithm into 3 steps.

We multiply on the notional at the end (Step 3).

Step 1 (Boundary condition). We are one step away from the maturity. At maturity, we shall receive the notional and the coupon.

$$\begin{aligned} X(t_{M-1}) = \underbrace{X(t_{M-1})}_{>t_{M-1},>t_{M-1}} &= B(t_{M-1}, t_{M-1}+\delta t) \\ &+ \max(1 + B(t_{M-1}, t_{M-1}+\delta t)(\delta L\delta t - 1), B(t_{M-1}, t_{M-1}+\delta t)C\delta t). \end{aligned}$$

Step 2 (Loop). We enter the loop at t_{M-1} (included) and exit at t_0 (not included):

$$\underbrace{t_0}_{\text{end}} \longleftarrow \underbrace{t_{M-1}}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(t_m)}_{?} \longleftarrow \underbrace{X(t_{m+1})}_{\text{known}},$$

where

$$X(t_{m+1}) = \underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}$$

is the value to continue (the value of the payments made after t_{m+1} if no exercises were made before or at t_{m+1}). We have that

$$\begin{aligned} \underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} &= \min\big(1, \underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}\big), \\ \underbrace{X(t_m)}_{>t_{m+1},>t_m} &= \mathcal{R}_{t_m}\big(\underbrace{X(t_{m+1})}_{>t_{m+1},>t_m}\big). \end{aligned}$$

We now add the coupon paid at then next period.

$$\underbrace{X(t_m)}_{>t_m,>t_m} = \underbrace{X(t_m)}_{>t_{m+1},>t_m} + Y(t_m),$$

where

$$\begin{aligned} Y(t_m) &= B(t_m, t_{m+1})\delta t \min(L(t_m, t_{m+1}) + \delta L, C) \\ &= \max(1 + B(t_m, t_{m+1})(\delta L\delta t - 1), B(t_m, t_{m+1})C\delta t). \end{aligned}$$

Here $B(s,t)$ is the discount factor at s for maturity t and we used the identity:

$$B(s,t)(1 + L(s,t)(t-s)) = 1.$$

Step 3 (After the loop). We return $NX(t_0) = N\underbrace{X(t_0)}_{>t_0,>t_0}$.

Cap on swap rate

Parameters of cap:

$N:$ the notional.

$C:$ the cap rate.

$\delta t:$ the interval of time between payments given as year fraction.

$M:$ the total number of payments.

Parameters of swap rate:

$\delta s:$ the interval of time between payments given as year fraction.

$L:$ the total number of payments.

We assume that today is the issue time of the contract and denote this time by t_0. The payment times of the cap are given by

$$t_m = t_0 + m\delta t, \quad m = 1, \dots, M.$$

At payment time t_{m+1}, the owner of the option gets the caplet:

$$N \max(R^{swap}(t_m, L, \delta s)\delta t - C\delta t, 0),$$

where $R^{swap}(t, L, \delta s)$ is the market swap rate at t for L periods δs.

Algorithm. The event times are

$$\{t_0, (t_m)_{m=1,\dots,M-1}\},$$

where t_0 is the initial time and $(t_m)_{m=1,\dots,M-1}$ are all payment times except the last one. We divide the algorithm into 3 steps.

We multiply on the notional at the end (Step 3).

We recall that $R^{swap}(t, L, \delta s)$, the market swap rate at t for L periods δs, can be computed from the relation:

$$\text{value of fixed payments} \;=\; \text{value of float payments},$$

$$R^{swap}(t, L, \delta s)\delta s \sum_{k=1}^{L} B(t, t + k\delta s) = 1 - B(t, t + L\delta s),$$

where $B(t, T)$ is the discount factor at t for maturity T.

Step 1 (Boundary condition).

$$\underbrace{X(t_{M-1})}_{>t_{M-1}} = B(t_{M-1}, t_{M-1}+\delta t)\max(R^{swap}(t_{M-1}, L, \delta s)\delta t - C\delta t, 0).$$

Step 2 (Loop). We enter the loop at t_{M-1} (included) and exit at t_0 (not included):

$$\underbrace{t_0}_{\text{end}} \longleftarrow \underbrace{t_{M-1}}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(t_m)}_{?} \longleftarrow \underbrace{X(t_{m+1})}_{\text{known}},$$

where

$$X(t_{m+1}) = \underbrace{X(t_{m+1})}_{>t_{m+1}}$$

is the value to continue (the value of caplets paid after t_{m+1}). We have that

$$\underbrace{X(t_m)}_{>t_{m+1}} = \mathcal{R}_{t_m}\Big(\underbrace{X(t_{m+1})}_{>t_{m+1}}\Big)$$

and then that

$$\underbrace{X(t_m)}_{>t_m} = \underbrace{X(t_m)}_{>t_{m+1}} + B(t_m, t_{m+1})\max(R^{swap}(t_m, L, \delta s)\delta t - C\delta t, 0).$$

Step 3 (After the loop). We return $NX(t_0) = N\underbrace{X(t_0)}_{>t_0}$.

Callable Range Accrued Note (CRAN)

Note parameters :

- N : the notional.
- δt : the interval of time between the coupon payments given as year fraction.
- R : the coupon rate.

$M:$ the total number of coupon payments.

Lookup range :

$L:$ the total number of lookup times in the coupon period.

$U:$ the upper barrier for LIBOR.

$D:$ the lower barrier for LIBOR.

$\Delta:$ the period for LIBOR as year fraction.

Brief description: Each coupon period has several look up dates. For example, coupon period may be monthly with weekly lookups. At the end of coupon period the paid coupon is proportional to the number of lookup times inside of the interval, when the LIBOR is inside of the range specified by the barriers. After the coupon the issuer has the right to redeem the note for the notional N.

Denote by $(u_m)_{m=1,\dots,M}$ the payment times of the note:

$$u_m = t_0 + m\delta t, \quad m = 1, \dots, M.$$

Fix a payment time u_{m+1} and denote by $(s_l)_{l=1,\dots,L}$ the lookup dates in the interval (u_m, u_{m+1}):

$$s_l = u_m + \delta t \frac{l}{L+1}, \quad l = 1, \dots, L.$$

Note that the lookup dates are strictly inside of the coupon period (u_m, u_{m+1}):

$$u_m < s_1 < \cdots < s_L < u_{m+1}.$$

At time u_{m+1} the holder of the note receives the product of the fixed coupon $NR\delta t$ on the fraction of lookup times $(s_l)_{l=1,\dots,L}$, when Δ-period LIBOR rate was inside of the range (D, U). In other words the payment at u_{m+1} is given by

$$NR\delta t \frac{1}{L} \sum_{l=1}^{L} 1_{\{D < r(s_l, s_l+\Delta) < U\}},$$

where $r(s, s+\Delta)$ is the market float rate computed at s for maturity $s+\Delta$.

After the coupon payment the issuer of the note can terminate it by paying back to the holder the notional amount N.

If the note has not been terminated before, then at maturity u_M in addition to the coupon payment the holder of the note also receives the notional amount N.

Algorithm. We denote

$$\delta s = \frac{\delta t}{L+1}, \quad q = \frac{R\delta t}{L}.$$

The event times are

$$\{t_0, (t_j)_{j=1,\dots,J}\},$$

where t_0 is the initial time, $J = M(L+1)$, and

$$t_j = t_0 + j\delta s, \quad j = 1, \dots, J,$$

are all lookup times and all coupon times except the last one. We divide the algorithm into 3 steps. We multiply on the notional at the end (Step 3).

Step 1 (Boundary condition). We are at the last lookup time.

$$X(t_J) = \underbrace{X(t_J)}_{>t_J, >t_J} = B(t_J, t_J + \delta s).$$

Step 2 (Loop). We enter the loop at t_J (included) and exit at t_0 (not included):

$$\underbrace{t_0}_{\text{end}} \longleftarrow \underbrace{t_J}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(t_j)}_{?} \longleftarrow \underbrace{X(t_{j+1})}_{\text{known}},$$

where

$$X(t_{j+1}) = \underbrace{X(t_{j+1})}_{>t_{j+1}, >t_{j+1}}$$

is the value to continue (the value of coupons computed after t_{j+1} if we exercise after t_{j+1}). There are 2 possibilities:

Case 1: t_{j+1} is a lookup time. We have that

$$\begin{aligned}
\underbrace{X(t_{j+1})}_{>t_j,>t_j} = \underbrace{X(t_{j+1})}_{>t_j,>t_{j+1}} \\
&= \underbrace{X(t_{j+1})}_{>t_{j+1},>t_{j+1}} + B(t_{j+1}, u_{m+1}) q 1_{\{D<r(t_{j+1},t_{j+1}+\Delta)<U\}} \\
&= \underbrace{X(t_{j+1})}_{>t_{j+1},>t_{j+1}} + B(t_{j+1}, u_{m+1}) q 1_{\{\alpha<B(t_{j+1},t_{j+1}+\Delta)<\beta\}},
\end{aligned}$$

where u_{m+1} is the next payment time and

$$\alpha = \frac{1}{1+U\Delta}, \quad \beta = \frac{1}{1+D\Delta}$$

are the lower and upper barriers for the discount factors.

Case 2: t_{j+1} is a coupon time, $t_{j+1} = u_{m+1}$. We have that

$$\underbrace{X(t_{j+1})}_{>t_j,>t_j} = \underbrace{X(t_{j+1})}_{>t_{j+1},>t_j} = \min\big(\underbrace{X(t_{j+1})}_{>t_{j+1},>t_{j+1}}, 1\big).$$

Finally, we move one step backward:

$$\underbrace{X(t_j)}_{>t_j,>t_j} = \mathcal{R}_{t_j}\big(\underbrace{X(t_{j+1})}_{>t_j,>t_j}\big).$$

Step 3 (After the loop). We return $NX(t_0) = N \underbrace{X(t_0)}_{>t_0,>t_0}$.

Putable annuity

The following contract is typical for loans with a possibility of an early repayment such as mortgages.

$F:$ the face value (the initial notional),

$Q:$ the annuity rate,

$\delta t:$ the interval of time between the payments given as year fraction,

M : the total number of payments.

Assume that today is the issue time of the annuity. Denote this time by t_0. The payment times of the annuity are given by

$$t_m = t_0 + m\delta t, \quad m = 1, \ldots, M.$$

At every payment time t_m, the holder of the annuity receives the payment A given by

$$A \sum_{k=1}^{M} \frac{1}{(1+Q\delta t)^k} = F.$$

and then the *issuer* has the right to terminate the annuity by paying to the holder the remaining *notional amount* N_m computed by

$$A \sum_{k=1}^{M-m} \frac{1}{(1+Q\delta t)^k} = N_m.$$

Note that $N_0 = F$ and that the annuity can not be canceled at initial (issue) time t_0.

Algorithm. The event times are

$$\left\{t_0, (t_m)_{m=1,\ldots,M-1}\right\},$$

where t_0 is the initial time and we have included all payment times except the last one. We divide the algorithm into 3 steps. We multiply on the notional at the end (Step 3).

Step 1 (Boundary condition). We are at the last minus one payment time.

$$X(t_{M-1}) = \underbrace{X(t_{M-1})}_{>t_{M-1}, >t_{M-1}} = B(t_{M-1}, t_{M-1} + \delta t)A.$$

Step 2 (Loop). We enter the loop at t_{M-1} (included) and exit at t_0 (not included):

$$\underbrace{t_0}_{\text{end}} \longleftarrow \underbrace{t_{M-1}}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(t_m)}_{?} \longleftarrow \underbrace{X(t_{m+1})}_{\text{known}},$$

where

$$X(t_{m+1}) = \underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}$$

is the value to continue (the value of the annuity payments after t_{m+1} if we decided not to exercise before and at t_{m+1}). We have that

$$\begin{aligned}
\underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} &= \min\big(\underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}, N_m\big), \\
\underbrace{X(t_{m+1})}_{>t_m,>t_m} &= \underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} + A, \\
\underbrace{X(t_m)}_{>t_m,>t_m} &= \mathcal{R}_{t_m}\big(\underbrace{X(t_{m+1})}_{>t_m,>t_m}\big).
\end{aligned}$$

Step 3 (After the loop). We return $NX(t_0) = N\underbrace{X(t_0)}_{>t_0,>t_0}$.

Constant maturity swaption

T : the maturity of the option.

Parameters of underlying swap:

N : the notional.

R : the fixed rate.

δs : the interval of time between payments given as year fraction.

M : the total number of payments.

side : this parameter defines the side of the swap contract, i.e. whether one pays "fixed" and receives "float" or otherwise.

L : the number of periods in the standard swap contract (with the same period δs) that determines the floating rate in CMS (constant maturity swap).

At maturity T, a holder of the option can enter into the underlying CMS issued at T. The difference between CMS and standard "plain vanilla" swap is in the way the floating rate is computed:

1. In CMS, the float interest is paid according to the current swap rate for L periods δs. That is, the floating rate transacted at time $T+(m+1)\delta s$ is the swap rate determined at $T+m\delta s$ for L periods δs.

2. In standard swap, the float interest is paid according to the current LIBOR rate.

Algorithm. The event times are

$$\left\{t_0, T, (s_m)_{m=1,\ldots,M-1}\right\},$$

where t_0 is the initial time and $(s_m)_{m=1,\ldots,M-1}$ are all CMS payment times except the last one:

$$s_m = T + m\delta s, \quad m = 1, \ldots, M-1.$$

We use notations:

$B(s,t):$ the discount factor at s for maturity t.

$R^{swap}(t, L, \delta s):$ the market swap rate at t for L periods δs. It is computed from the relation:

$$\text{value of fixed payments} = \text{value of float payments},$$
$$R^{swap}(t, L, \delta s)\delta s \sum_{k=1}^{L} B(t, t+k\delta s) = 1 - B(t, t+L\delta s).$$

We divide the algorithm into 3 steps. We multiply on the notional at the end (Step 3). Until Step 3, we pay fixed rate and receive float swap rate.

Step 1 (Boundary condition).

$$\underbrace{X(s_{M-1})}_{>s_{M-1}} = B(s_{M-1}, s_{M-1}+\delta s)\left(R^{swap}(s_{M-1}, L, \delta s) - R\right)\delta s.$$

Step 2 (Loop). We enter the loop at s_{M-1} (included) and exit at $s_0 = T$ (not included):

$$\underbrace{s_0 = T}_{\text{end}} \longleftarrow \underbrace{s_{M-1}}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(s_m)}_{?} \longleftarrow \underbrace{X(s_{m+1})}_{\text{known}},$$

where

$$X(s_{m+1}) = \underbrace{X(s_{m+1})}_{>s_{m+1}}$$

is the value to continue (the value of payments made after s_{m+1}). We have that

$$\underbrace{X(s_m)}_{>s_{m+1}} = \mathcal{R}_{s_m}\Big(\underbrace{X(s_{m+1})}_{>s_{m+1}}\Big)$$

and then that

$$\underbrace{X(s_m)}_{>s_m} = \underbrace{X(s_m)}_{>s_{m+1}} + B(s_m, s_m + \delta s)\left(R^{swap}(s_m, L, \delta s) - R\right)\delta s.$$

Step 3 (After the loop). The current time is T, the maturity of the CMS option. At this moment, $X(T)$ is the value of CMS at its issue time, provided we pay fixed and notional = 1.

$$\text{If we pay float, then } X(T) = -X(T).$$

We have that

$$X(T) = \max(X(T), 0),$$
$$X(t_0) = \mathcal{R}_{t_0,T}\left(X(T)\right).$$

We return $NX(t_0)$.

American constant maturity swaption

$(t_m)_{m=1,\ldots,M}$: the exercise times.

Parameters of underlying swap:

N : the notional.

R : the fixed rate.

δs : the interval of time between the payments given as year fraction.

M : the total number of payments.

side : the side of the swap contract, i.e., whether one pays "fixed" and receives "float" or otherwise.

L : the number of periods δs that determines the floating rate in CMS.

A holder can enter into the underlying CMS at any exercise time t_m. This time then becomes the issue time of the CMS. The difference between CMS and the standard "plain vanilla" swap is in the way the floating rate is computed.

1. In CMS, the float interest at $t_{m+1} = t_m + \delta s$ is paid according to market swap rate $R^{swap}(t_m, L, \delta s)$ computed at t_m for the swap expiring after L periods of length δs.
2. In standard swap, the float interest at $t_{m+1} = t_m + \delta s$ is paid according to LIBOR rate $L(t_m, t_m + \delta s)$ computed at t_m for period δs.

Algorithm. Event times $(u_k)_{k=0,1,\ldots,K}$ are a sorted union of

t_0 : the initial time;

$(t_m)_{m=1,\ldots,M}$: the exercise times;

$(s_{ml})_{m=1,\ldots,M,\, l=1,\ldots,L-1}$: all payment times except the last one for CMS issued at t_m.

If the union contains several identical elements, we keep only one.

We denote

$Y^{CMS}(t, L, \delta s)$: the value of CMS for L payments of periods δs issued at t. We shall show how to compute it at the end of the algorithm.

We divide the algorithm into 3 steps. In this algorithm, we only use initial time t_0 and exercise times $(t_m)_{m=1,\ldots,M}$.

Step 1 (Boundary condition).

$$\underbrace{X(t_M)}_{>t_M} = 0.$$

Step 2 (Loop). We enter the loop at t_M (included) and exit at t_0 (not included):

$$\underbrace{t_0}_{\text{end}} \longleftarrow \underbrace{t_M}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{X(t_m)}_{?} \longleftarrow \underbrace{X(t_{m+1})}_{\text{known}},$$

where

$$X(t_{m+1}) = \underbrace{X(t_{m+1})}_{>t_{m+1}}$$

is the value to continue (the value at t_{m+1} if we decided to exercise later). We have that

$$\underbrace{X(t_{m+1})}_{>t_m} = \max\big(Y^{CMS}(t_{m+1}, L, \delta s), \underbrace{X(t_{m+1})}_{>t_{m+1}}\big)$$

and then that

$$\underbrace{X(t_{m+1})}_{>t_m} = \mathcal{R}_{t_m}\big(\underbrace{X(t_{m+1})}_{>t_{m+1}}\big).$$

Step 3 (After the loop). We return $\underbrace{X(t_0)}_{>t_0}$.

Value of CMS

We now show how to compute

$$Y(t) = Y^{CMS}(t, L, \delta s),$$

the value of CMS with L periods δs issued at t. We denote

$B(t,T)$: the discount factor at t for maturity T.

$R^{swap}(t, L, \delta s)$: the market swap rate in the contract issued at t with L periods δs. It is computed from the relation:

$$\text{value of fixed payments} \;=\; \text{value of float payments},$$

$$R^{swap}(t, L, \delta s)\delta s \sum_{l=1}^{L} B(t, t + l\delta s) = 1 - B(t, t + L\delta s).$$

We divide the algorithm into 3 steps. We multiply on the notional at the end (Step 3). Until Step 3, we pay fixed rate and receive floating swap rate.

We use issue time $r_0 = t$ and all payment times except the last one:

$$r_l = t + l\delta s, \quad l = 0, 1, \dots, L-1.$$

Step 1 (Boundary condition).

$$Y(r_{L-1}) = B(r_{L-1}, r_{L-1} + \delta s)\left(R^{swap}(r_{L-1}, L, \delta s) - R\right)\delta s.$$

Step 2 (Loop). We enter the loop at r_{L-1} (included) and exit at $t = r_0$ (not included):

$$\underbrace{r_0 = t}_{\text{end}} \longleftarrow \underbrace{r_{L-1}}_{\text{begin}}.$$

We consider the iteration:

$$\underbrace{Y(r_l)}_{?} \longleftarrow \underbrace{Y(r_{l+1})}_{\text{known}},$$

where

$$Y(r_{l+1}) = \underbrace{Y(r_{l+1})}_{>r_{l+1}}$$

is the value to continue (the value of payments made after r_{l+1}). We have that

$$\underbrace{Y(r_l)}_{>r_{l+1}} = \mathcal{R}_{r_l}\Big(\underbrace{Y(r_{l+1})}_{>r_{l+1}}\Big)$$

and then that

$$\underbrace{Y(r_l)}_{>r_l} = \underbrace{Y(r_l)}_{>r_{l+1}} + B(r_l, r_l + \delta s)\left(R^{swap}(r_l, L, \delta s) - R\right)\delta s.$$

Step 3 (After the loop). We account for our side:

$$\text{if we pay float, then } Y(t) = -Y(t),$$

and return $NY(t)$.