Examples for the Quantathon

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

Interest rate cap

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.

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

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

At payment time t_m a holder receives the *caplet*

$$N \max(L(t_{m-1}, t_m)\delta t - C\delta t, 0),$$

where $L(s,t)$ is the LIBOR computed at time s for maturity t.

Algorithm. The event times are

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

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).

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

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

$$\begin{aligned}\underbrace{X(t_m)}_{>t_m} &= \underbrace{X(t_m)}_{>t_{m+1}} + B(t_m, t_{m+1}) \max(L(t_m, t_{m+1})\delta t - C\delta t, 0) \\ &= \underbrace{X(t_m)}_{>t_{m+1}} + \max(1 - B(t_m, t_{m+1})(1 + C\delta t), 0),\end{aligned}$$

where $B(s, t)$ 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 $NX(t_0) = N\underbrace{X(t_0)}_{>t_0}$.

Interest rate swap

N : the notional.

R : the fixed rate.

δt : the interval of time between the 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.

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

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

At time t_{m+1},

1. One side pays "float" interest

$$NL(t_m, t_{m+1})\delta t,$$

   where $L(s, t)$ is the float (LIBOR) rate computed at s for maturity t.

2. Another side pays "fixed" interest

$$NR\delta t.$$

We need to compute the present value of the swap. Write the algorithm with just one event time t_0.

Algorithm. We need just one event time:

$$\{t_0\},$$

which is the initial time. Assume first that we pay float LIBOR $L(s, t)$ and receive fixed rate R. Then

$$\underbrace{X(t_0)}_{\text{pay float}} = \underbrace{\text{swap}}_{\text{pay float}} = \text{coupon bond} - \text{bank account} = Y(t_0) - Z(t_0).$$

1. The coupon bond pays fixed coupons

$$NR\delta t \quad \text{at} \quad t_m = t_0 + m\delta t, \quad m = 1, \dots, M,$$

   and notional N at maturity t_M. Its value is given by

$$Y(t_0) = N\left(R\delta t \sum_{m=1}^{M} B(t_0, t_m) + B(t_0, t_M)\right).$$

2. The bank account pays float interest

$$NL(t_{m-1}, t_m)\delta t \quad \text{at} \quad t_m = t_0 + m\delta t, \quad m = 1, \dots, M,$$

and notional N at maturity t_M. By replication, its value is given by the notional:

$$Z(t_0) = N.$$

If we pay fixed, then

$$\underbrace{X(t_0)}_{\text{pay fixed}} = -\underbrace{X(t_0)}_{\text{pay float}}.$$

Swaption

T : the maturity.

Parameters of underlying swap:

N : the notional.

R : the fixed rate.

δt : the interval of time between the 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.

At maturity T, the holder has the right to enter into the swap contract with the parameters defined above and issue time T.

Algorithm. The event times have the form:

$$\{t_0, T\},$$

where t_0 is the initial time. We have that

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

where $V(T)$ is the value of the swap issued at T. To compute $V(T)$, assume first that we pay LIBOR $L(s,t)$ and receive fixed rate R. Then

$$V(T) = \underbrace{V(T)}_{\text{pay float}} = \underbrace{\text{swap}}_{\text{pay float}} = \text{coupon bond} - \text{bank account} = Y(T) - Z(T).$$

1. The coupon bond pays fixed coupons

$$NR\delta t \quad \text{at} \quad t_m = T + m\delta t, \quad m = 1, \ldots, M,$$

and notional N at maturity t_M. Its value is given by

$$Y(T) = N\left(R\delta t \sum_{m=1}^{M} B(T, t_m) + B(T, t_M)\right).$$

2. The bank account pays float interest

$$NL(t_{m-1}, t_m)\delta t \quad \text{at} \quad t_m = T + m\delta t, \quad m = 1, \ldots, M,$$

and notional N at maturity t_M. By replication, its value at T is given by the notional:

$$Z(T) = N.$$

If we pay fixed, then

$$V(T) = \underbrace{V(T)}_{\text{pay fixed}} = -\underbrace{V(T)}_{\text{pay float}}.$$

Cancellable interest rate collar

N : the notional.

C : the cap rate.

F : the floor rate ($F < C$).

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

M : the total number of payments.

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

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

At payment time t_m:

1. If LIBOR rate $L(t_{m-1}, t_m)$ is greater than cap rate C, then a holder *receives*
$$N\delta t(L(t_{m-1}, t_m) - C).$$
2. If LIBOR rate $L(t_{m-1}, t_m)$ is less than floor rate F, then a holder *pays*
$$N\delta t(F - L(t_{m-1}, t_m)).$$
3. After the payment is either received or paid, a holder *has the right to cancel* the contract. No payments will be made after that.

Algorithm. The event times are
$$\left\{t_0, (t_m)_{m=1,\dots,M-1}\right\},$$
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).

Step 1 (Boundary condition).
$$\begin{aligned} X(t_{M-1}) = \underbrace{X(t_{M-1})}_{>t_{M-1}} &= \max(1 - B(t_{M-1}, t_M)(1 + C\delta t), 0) \\ &- \max(B(t_{M-1}, t_M)(1 + F\delta t) - 1, 0) \end{aligned}$$

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 collarets paid after t_{m+1} if we cancel after t_{m+1}). We have that

$$\underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} = \max(\underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}, 0)$$

and then that

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

and

$$\begin{aligned}
\underbrace{X(t_m)}_{>t_m,>t_m} = &\underbrace{X(t_m)}_{>t_{m+1},>t_m} + B(t_m, t_{m+1}) \max(L(t_m, t_{m+1})\delta t - C\delta t, 0) \\
&- B(t_m, t_{m+1}) \max(F\delta t - L(t_m, t_{m+1})\delta t, 0) \\
= &\underbrace{X(t_m)}_{>t_{m+1},>t_m} + \max(1 - B(t_m, t_{m+1})(1 + C\delta t), 0) \\
&- \max(B(t_m, t_{m+1})(1 + F\delta t) - 1, 0),
\end{aligned}$$

where $B(s,t)$ 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 $NX(t_0) = N \underbrace{X(t_0)}_{>t_0,>t_0}$.

Down-and-out cap

Underlying cap :

> N : the notional.
>
> R : the cap rate.
>
> δt : the interval of time between the payments given as year fraction.
>
> M : the total number of payments.

L : the lower bound for float (LIBOR) rate.

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

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

The down-and-out cap generates the same cash flow as the interest rate cap up to (and including) the payment time, when the float rate drops below L. After that the option is terminated. In other words, if we denote by τ the first payment time t_m, when float rate $r(t_m, t_m + \delta t)$ between t_m and $t_m + \delta t$ is less then L, then for a payment time t_j:

1. If $t_j \leq \tau$, then the holder gets standard cap payment

$$N \max(r(t_{j-1}, t_j)\delta t - R\delta t, 0).$$

2. If $t_j > \tau$, then the holder gets nothing.

Algorithm. The event times are

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

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). We denote by $B(s,t)$ and $r(s,t)$ the discount factor and the float rate, respectively, at s for maturity t. We recall the identity:

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

From this identity we deduce that

$$\begin{aligned} r(t, t+\delta t) > L &\iff 1 + r(t, t+\delta t)\delta t > 1 + L\delta t \\ &\iff 1 > B(t, t+\delta t)(1 + L\delta t) \\ &\iff U > B(t, t+\delta t), \end{aligned}$$

where $U = 1/(1 + L\delta t)$.

Step 1 (Boundary condition).

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

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

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

is the value to continue (the value of caplets paid after t_{m+1} if possible barrier events happen after t_{m+1}).

We have that

$$\begin{aligned}\underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} &= \underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}} 1_{\{r(t_{m+1},t_{m+1}+\delta t)>L\}} \\ &= \underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}} 1_{\{U>B(t_{m+1},t_{m+1}+\delta t)\}}\end{aligned}$$

and then that

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

and

$$\begin{aligned}\underbrace{X(t_m)}_{>t_m,>t_m} &= \underbrace{X(t_m)}_{>t_{m+1},>t_m} + B(t_m,t_{m+1}) \max(r(t_m,t_{m+1})\delta t - C\delta t, 0) \\ &= \underbrace{X(t_m)}_{>t_{m+1},>t_m} + \max(1 - B(t_m,t_{m+1})(1+C\delta t), 0),\end{aligned}$$

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

Futures on LIBOR

The futures contracts of these types are traded, for example, on EUREX, where the underlying is 3 month EURO LIBOR.

Input: the parameters of futures contract.

> Δ : the period for LIBOR given as year fraction ($\Delta = 0.25$ for 3 month LIBOR).
>
> T : the maturity of futures contract.
>
> M : the number of settlement times between today and the maturity.

Output: futures price $F(t_0)$ computed at initial time t_0.

We assume that the settlement times are given by

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

where t_0 is the initial time and

$$\delta t = \frac{T - t_0}{M}.$$

Notice that the settlement times include T, but do not contain t_0.

The futures contract on LIBOR involves the following transactions:

1. It costs nothing to enter into either a long or a short position in the futures contract

2. At time t_m before maturity, $m = 1, \dots, M - 1$,

   (a) the buyer (long position) pays futures price $F(t_{m-1})$ established at the previous trading day,

   (b) the seller (short position) pays futures price $F(t_m)$ established at the current trading day.

3. At maturity $T = t_M$

   (a) the buyer (long position) pays futures price $F(t_{M-1})$ established at previous trading day,

(b) the seller (short position) pays

$$F(t_M) = F(T) = 1 - L(T, T+\Delta),$$

and $L(T, T+\Delta)$ is the LIBOR computed at time T for maturity $T+\Delta$.

Algorithm. The event times are

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

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

Step 1 (Boundary condition).

$$F(t_M) = 1 - L(t_M, t_M + \Delta),$$

where

$$\begin{aligned} L(t_M, t_M+\Delta) &= \frac{1}{\Delta}\left((1 + L(t_M, t_M+\Delta)\Delta) - 1\right) \\ &= \frac{1}{\Delta}\left(\frac{1}{B(t_M, t_M+\Delta)} - 1\right) \end{aligned}$$

and $B(s,t)$ is the discount factor computed at s for maturity t.

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:

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

We have that

$$F(t_m) = \frac{1}{B(t_m, t_{m+1})} \mathcal{R}_{t_m}\left(F(t_{m+1})\right),$$

because

$$0 = \mathcal{R}_{t_m}\left(F(t_{m+1}) - F(t_m)\right) = \mathcal{R}_{t_m}\left(F(t_{m+1})\right) - B(t_m, t_{m+1})F(t_m).$$

Step 3 (After the loop). We return $F(t_0)$.

Drop-lock swap

Brief description: a swap in which, the first time the market swap rate is above the upper barrier or below the lower barrier, the fixed rate is reset to the upper or lower barriers, respectively, and then remains constant.

Parameters of underlying swap:

N : the notional.

R : the initial fixed rate in the swap.

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

M : the total number of payments.

side: the side of the swap contract. It defines whether the holder pays "fixed" rate and receives "float" rate or otherwise.

Reset rates:

L : the lower value for the swap rate after reset.

U : the upper value for the swap rate after reset, $L < U$.

We denote by $(t_m)_{m=1,\ldots,M}$ the payment times of the swap:

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

and by $Q(t_m; \delta t, M)$ the market swap rate at t_m for the contract with the same period δt and the number of payments M as in the original swap. Let

$$\tau = \min\{(t_m)_{m=1,\ldots,M} : \ Q(t_m; \delta t, M) > U \text{ or } Q(t_m; \delta t, M) < L\}.$$

Up to and including time τ, the payments in the swap contract are determined by initial fixed rate R. After τ, the fixed payments are given by U if $Q(\tau; \delta t, M) > U$ and by L if $Q(\tau; \delta t, M) < L$.

If neither upper nor lower barriers are crossed, then all payments are determined by original fixed rate R. Note that the initial time is *not* a reset time.

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 denote

$Y(t,k,r)$: the value of interest rate swap issued at t with notional N, swap rate r, period δt, and the number of payments k. If we pay float and received fixed payments, then

$$Y(t,k,r) = N\left(r\delta t \sum_{i=1}^{k} B(t, t+i\delta t) + B(t, t+k\delta t) - 1\right).$$

If we pay fixed, then we need to multiply the previous expression by -1.

$Q(t)$: the market swap rate computed at t for period δt and the number of payments M.

$$Q(t) = \frac{1 - B(t, t+M\delta t)}{\delta t \sum_{m=1}^{M} B(t, t+m\delta t)}.$$

Step 1 (Boundary condition).

$$\underbrace{X(t_{M-1})}_{>t_{M-1}, >t_{M-1}} = Y(t_{M-1}, 1, R).$$

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 payments after t_{m+1} if there were no resets before and at t_{m+1}). We have that

$$\begin{aligned}\underbrace{X(t_{m+1})}_{>t_{m+1},>t_m} = 1_{\{Q(t_{m+1}>U)\}}Y(t_{m+1}, M-m, U)\\ + 1_{\{L>Q(t_{m+1})\}}Y(t_{m+1}, M-m, L)\\ + 1_{\{L\le Q(t_{m+1})\le U\}}\underbrace{X(t_{m+1})}_{>t_{m+1},>t_{m+1}}\end{aligned}$$

and then that

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

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

Auto interest rate cap

t_0 : the initial time.

Parameters of the cap:

> N : the notional.
>
> C : the cap rate.
>
> δt : the interval of time between the payments given as year fraction.
>
> M : the total number of periods.

K : the maximal number of paid caplets, $K \le M$.

The holder receives first K in-the-money caplets and then the contract is terminated. Recall that the caplet paid at time

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

is given by

$$N \max(L(t_{m-1}, t_m)\delta t - C\delta t, 0),$$

where $L(s,t)$ is the LIBOR computed at time s for maturity t.

Algorithm. The event times are

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

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).

For payment time t_m we denote by $Y(t_m)$ the value of the next caplet:

$$Y(t_m) = \max(1 - B(t_m, t_m + \delta t)(1 + C\delta t), 0),$$

where $B(s,t)$ 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 1 (Boundary condition).

$$\underbrace{X_k(t_{M-1})}_{>t_{M-1}} = Y(t_{M-1}), \quad k = 0, 1, \ldots, K-1.$$

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

$$X(t_{m+1}) = \underbrace{X_k(t_{m+1})}_{>t_{m+1}}, \quad k = 0, \ldots, K-1,$$

and $\underbrace{X_k(t_{m+1})}_{>t_{m+1}}$ is the value to continue (the value of caplets paid after t_{m+1}) if k caplets were in the money before and at t_{m+1}. We have that

$$\underbrace{X_k(t_m)}_{>t_{m+1}} = \mathcal{R}_{t_m}\Big(\underbrace{X_k(t_{m+1})}_{>t_{m+1}}\Big), \quad k = 0, \ldots, K-1,$$

and then that

$$\begin{aligned}
\underbrace{X_k(t_m)}_{>t_m} &= \underbrace{X_k(t_m)}_{>t_{m+1}} + 1_{\{Y(t_m)>0\}}\Big(\underbrace{X_{k+1}(t_m)}_{>t_{m+1}} + Y(t_m) - \underbrace{X_k(t_m)}_{>t_{m+1}}\Big),\\
&\quad k = 0, \dots, K-2,\\
\underbrace{X_{K-1}(t_m)}_{>t_m} &= \underbrace{X_{K-1}(t_m)}_{>t_{m+1}} + 1_{\{Y(t_m)>0\}}\Big(Y(t_m) - \underbrace{X_{K-1}(t_m)}_{>t_{m+1}}\Big).
\end{aligned}$$

Step 3 (After the loop). We return $N\,\underbrace{X(t_0)}_{>t_0}[0]$.