Exercise 1

Consider a tenure structure

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

(a) Write a class with:

- One method that converts a zero coupon bond curve (given as an array of `doubles`) with tenure structure given above in a Libor rate curve with same tenure structure, and returns the Libor rate curve again as an array of `doubles`.
- One method that does the opposite, i.e., converts a Libor rate curve (given as an array of `doubles`) with tenure structure given above in a zero coupon bond curve, and returns the zero coupon bond curve with same tenure structure again as an array of `doubles`.

Note that:

- You can suppose the valuation time to be $T_0 = 0$.
- The tenor structure must be given as a `TimeDiscretization` object, either directly to the methods or as a field of the class (this is your choice).
- You can decide that the first value of your Libor curve is $L(0, T_1; 0)$. In this case, to compute it you will also use the value of $P(0; 0) = 1$: you can avoid to include it in the zero coupon bond curve, so the first value of the zero coupon bond curve can be $P(T_1; 0)$.

(b) Verify, with a tenor discretization and a zero coupon curve at your choice, that if you convert your zero coupon bond curve in Libors, and then back in bonds, then you obtain the initial curve.

Exercise 2

Suppose you have a tenure structure

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

of evenly distributed points (i.e., $T_i - T_{i-1} = \Delta$ for every $i = 1, \ldots, n$), and call CB_m, $m = 2, \ldots, n$, the value at zero of the Coupon Bond for the partial tenure structure

$$T_0 = 0 < \cdots < T_m,$$

i.e., the value

$$\sum_{i=1}^{m-1} C_i (T_{i+1} - T_i) P(T_{i+1}; 0) + P(T_m; 0)$$

where $C_1, \ldots, C_{m-1}$ are the associated coupons, see also Theorem 127 of the script.

(a) Write a class in order to get the value $P(T_2; 0)$ knowing CB_2 and C_1 and then the value of $P(T_{m+1}; 0)$ knowing the value of $P(T_2; 0), \ldots, P(T_m; 0)$, CP_m and $C_1, \ldots, C_{m-1}$, for any $m > 2$. Try to give an implementation of this class such that these calculations can be done recursively, i.e., the newly computed values of the bonds can be stored and used, together with the value of CP_m and of the coupons, to compute the next ones. For example: you first know CP_2, C_1 and you compute $P(T_2; 0)$. Then you use this value, along with CP_3 and C_2, to compute $P(T_3; 0)$. And so on.

(b) Write a test where you use this class in order to compute the value of the bonds $P(T_i; 0)$, $i \geq 2$ when you have $\Delta = 0.5$,

$$CP_2 = 1.93, \quad CP_3 = 2.77, \quad CP_4 = 3.55, \quad CP_5 = 4.45,$$
$$CP_6 = 5.2, \quad CP_7 = 5.9, \quad CP_8 = 6.55, \quad CP_9 = 7.15$$

and

$$C_1 = 2.1, \quad C_2 = 1.9, \quad C_3 = 1.8, \quad C_4 = 2.2,$$
$$C_5 = 2.1, \quad C_6 = 1.95, \quad C_7 = 2, \quad C_8 = 2.05.$$

You should get the following results:

$$P(1;0) = 0.9415, \qquad P(1.5;0) = 0.9136, \qquad P(2;0) = 0.8914, \qquad P(2.5;0) = 0.8530$$
$$P(3;0) = 0.7820, \qquad P(3.5;0) = 0.7504, \qquad P(4;0) = 0.7002, \qquad P(4.5;0) = 0.6421.$$