Introduction to Mathematical Finance

Exercise sheet 6

Exercise 6.1

(a) Recall and prove the *numeraire change theorem*.

(b) Show that the forward Libor rate $(L(t,T,T+\delta))_t$[1] is a martingale under the *forward probability* $P^{T+\delta}$. Derive the arbitrage free price of a caplet with payoff $\mathrm{C}^{caplet} = \delta(L(t,T,T+\delta)-K)_+$.

(c) Using the formula

$$S_{T_0,T_N}(t) = \frac{P(t,T_0)-P(t,T_N)}{\sum_{k=0}^{N} \delta P(t,T_k)}$$

show that the swap rate $(S_{T_0,T_N}(t))_t$ is a martingale under the *annuity probability* P^A where the *annuity* is $A(t) = \sum_{k=0}^{N} \delta P(t,T_k)$. Derive the arbitrage free price of a swaption with payoff $\mathrm{C}^{swaption} = A(T_f)(S_{T_0,T_N}(t)-K)_+$.

Exercise 6.2
Derive a formula for the arbitrage free price of following contingent claims.

(a) *up-and-out call option*, $\mathrm{C}^{call}_{u\&o} = (S_T-K)_+ 1_{\max\limits_{0\le t\le T} S_t < B}$

(b) *down-and-in put option*, $\mathrm{C}^{put}_{d\&i} = (K-S_T)_+ 1_{\min\limits_{0\le t\le T} S_t \le B}$

(c) *lookback put option*, $\mathrm{C}^{put}_{max} = \max\limits_{0\le t\le T} S_t - S_T$

(d) *lookback call option*, $\mathrm{C}^{call}_{min} = S_T - \min\limits_{0\le t\le T} S_t$

Exercise 6.3 We consider a binomial market model with N periods on a period of time of length T. The riskless asset grows at a rate $r = \frac{RT}{N}$, where R is the (constant) instantaneous interest rate, and the risky asset's price goes up by a factor $1+u$ and down by a factor $1+d$ such that

$$\log\left(\frac{1+u}{1+r}\right) = -\log\left(\frac{1+d}{1+r}\right) = \sigma\sqrt{\frac{T}{N}},$$

for some constant σ. The starting values (at time t=0) of both the assets is 1 $\mathbb{P}$-a.s. The unique equivalent martingale measure $\mathbb{P}^*$ for S^1 is such that the $(Y_i)_{i\in\{1,2\dots,N\}}$ are i.i.d. and given by

$$\mathbb{P}^*\left[Y_i = 1+d\right] = 1-\mathbb{P}^*\left[Y_i = 1+u\right] = \frac{u-r}{u-d} = p^*$$

We study the limiting case for $N\to\infty$.

(a) Let $(Z_n)_{n\in\mathbb{N}}$ be a sequence of random variables of the form :

$$Z_n = \sum_{i=1}^{n} X_i^n$$

[1] the forward zero-coupon $P(t,T,T+\delta) = \frac{1}{1+\delta L(t,T,T+\delta)}$ and $P(t,T,T+\delta) = \frac{P(t,T+\delta)}{P(t,T)}$.

for $n \in \mathbb{N}$, $X_i^n \in \{-\sigma\sqrt{\frac{T}{n}},\ \sigma\sqrt{\frac{T}{n}}\}$ and the variables $(X_i^n)_{i\in\{1,2,\dots n\}}$ are independent identically distributed with mean μ_n. The constants μ_n are such that $\lim_{n\to\infty} n\mu_n = \mu$.

Prove that the sequence $(Z_n)_{n\in\mathbb{N}}$ converges in law to a gaussian random variable with mean μ and variance $\sigma^2 T$.

Hint: Use the fact that point-wise convergence of the characteristic functions of a sequence of random variables (if the limiting function ϕ is continuous at 0) implies the convergence in law of this sequence of random variables to a random variable whose characteristic function is ϕ.

(b) We consider a European put option, with strike K and maturity T. Show that its value at time 0 is given by

$$V_0^{P,N} = \mathbb{E}^*\left[\left(\frac{K}{(1+r)^N} - S_0^1 \exp(Z_N)\right)^+\right],$$

where $\mathbb{E}^*$ denotes the expectation under $\mathbb{P}^*$, and Z_N is a random variable that you will define.

(c) Use part a) to prove the following asymptotic price :

$$\lim_{N\to\infty} V_0^{P,N} = Ke^{-RT}\Phi(-d_2) - S_0^1\Phi(-d_1),$$

where $d_1 = \dfrac{\log\left(\frac{S_0}{K}\right) + RT + \frac{\sigma^2 T}{2}}{\sigma\sqrt{T}}$, $d_2 = d_1 - \sigma\sqrt{T}$ and Φ is the cumulative distribution function of a standard normal random variable.

Hint: Use the value of p^* and of u to prove that $\lim_{N\to\infty} N\mathbb{E}^*\left[\log\left(\frac{Y_i}{1+r}\right)\right] = -\frac{\sigma^2 T}{2}$

Exercise 6.4 Python - Trinomial model Inspire yourself from binomial price array to complete the trinomial price function.

```python
def trinomial_price(maturity, spot, strike, rate, vol, steps_number, payoff_fct=None, graph_name=None):
  """Compute the trinomial price. Draw graph if graph_name is given.
  """
  proba_up = 0  # TODO
  proba_down = 0  # TODO
  proba_middle = 0  # TODO
  steps = range(steps_number)
  spot_prices = [] # TODO
  option_prices = []  # TODO

  # The following two list are only needed to display the graph:
  spot_prices_history = [spot_prices]
  option_prices_history = [option_prices]

  # TODO

  if graph_name:
    create_graph(graph_name, spot_prices_history, option_prices_history)

  return option_prices[0]
```