Mathematical Finance

Solution sheet 11

Solution 11.1 Let X be a uniformly integrable martingale with respect to the filtration $(\mathcal{F}_t)_{t\in[0,\infty]}$. Then

$$\widetilde{X}_t = \begin{cases} X_{\frac{t}{1-t}}, & t < 1, \\ X_\infty, & t \geqslant 1, \end{cases}$$

is a martingale with respect to the filtration

$$\widetilde{\mathcal{F}}_t = \begin{cases} \mathcal{F}_{\frac{t}{1-t}}, & t < 1, \\ \mathcal{F}_\infty, & t \geqslant 1. \end{cases}$$

Let H be a bounded predictable process such that $(H \bullet X)$ is not a uniformly integrable $(\mathcal{F}_t)$-martingale. Since

$$(\widetilde{H} \bullet \widetilde{X})_t = (H \bullet X)_{\frac{t}{1-t}},\ t < 1,$$

the stochastic integral of the process $\widetilde{H}_t = H_{\frac{t}{1-t}} \mathbb{1}_{\{t<1\}}$ with respect to $\widetilde{X}$ is not an $(\widetilde{\mathcal{F}}_t)$-martingale. Indeed, there exists a martingale X and a predictable bounded H such that $(H \bullet X)$ is not uniformly integrable. Let

$$a_n = 2n,\ b_n = \frac{2n}{2n^2 - n + 1},\ p_n = \frac{n-1}{2n^2},\ n \in \mathbb{N}.$$

Define recursively a martingale $(X_n)_{n\in\mathbb{N}}$ and $(A_n)_{n\in\mathbb{N}}$ as

$$\begin{aligned}
&X_0 = 1, X_1 = 1,\ A_1 = \Omega, \\
&P(X_{n+1} = a_2 \cdots a_{n+1} \mid A_n) = p_{n+1}, \\
&P(X_{n+1} = a_2 \cdots b_{n+1} \mid A_n) = 1 - p_{n+1}, \\
&P(X_{n+1} = X_n \mid A_n^c) = 1, \\
&A_{n+1} = \{X_{n+1} = a_2 \cdots a_{n+1}\}.
\end{aligned}$$

Define $X_t = X_n$, $t \in [n, n+1[$ and $H_t = \sum_{n=1}^\infty \mathbb{1}_{]2n-1,2n]}(t)$ for $t \in [0,\infty[$ and $n \in \mathbb{N}$. Then X is a uniformly integrable martingale w.r.t. $\mathcal{F}_t = \mathcal{F}_t^X$, $t \in [0,\infty[$. Indeed, for $m > n$, we have

$$E[|(X_m - X_n)\mathbb{1}_{A_{n+1}}|] = E[(X_m - X_n)\mathbb{1}_{A_{n+1}}] = E[(X_{n+1} - X_n)\mathbb{1}_{A_{n+1}}] = E[|(X_{n+1} - X_n)\mathbb{1}_{A_{n+1}}|],$$

so,

$$\begin{aligned}
E[|X_m - X_n|] &= E[|(X_m - X_n)\mathbb{1}_{A_n}|] = E[|(X_m - X_n)\mathbb{1}_{A_{n+1}}|] + E[|(X_m - X_n)\mathbb{1}_{A_n}\mathbb{1}_{A_{n+1}^c}|] \\
&= E[|(X_{n+1} - X_n)\mathbb{1}_{A_{n+1}}|] + E[|(X_{n+1} - X_n)\mathbb{1}_{A_n}\mathbb{1}_{A_{n+1}^c}|] = E[|(X_{n+1} - X_n)\mathbb{1}_{A_n}|] \\
&= a_2 \cdots a_n(a_{n+1} - 1)p_2 \cdots p_n p_{n+1} + a_2 \cdots a_n(1 - b_{n+1})p_2 \cdots p_n(1 - p_{n+1}) \\
&\leqslant a_2 \cdots a_n p_2 \cdots p_n(a_{n+1}p_{n+1} + 1) = \frac{1}{n}\left(\frac{n}{n+1} + 1\right) \leqslant \frac{2}{n},
\end{aligned}$$

therefore, $(X_n)_{n\in\mathbb{N}}$ converges in L^1, i.e., $(X_n)_{n\in\mathbb{N}}$ is uniformly integrable. However, for $n \leqslant m$, we have

$$\begin{aligned}
E[|\mathbb{1}_{A_{2n}}\mathbb{1}_{A_{2n+1}^c}(H \bullet X)_{2m}|] &= E[\mathbb{1}_{A_{2n}}\mathbb{1}_{A_{2n+1}^c}\sum_{k=1}^n (X_{2k} - X_{2k-1})] \geqslant E[\mathbb{1}_{A_{2n}}\mathbb{1}_{A_{2n+1}^c}(X_{2n} - X_{2n-1})] \\
&= p_2 \cdots p_{2n}(1 - p_{2n+1})a_2 \cdots a_{2n-1}(a_{2n} - 1) \geqslant \frac{1}{4}p_2 \cdots p_{2n}a_2 \cdots a_{2n} = \frac{1}{8n},
\end{aligned}$$

so,

$$E[|(H \bullet X)_{2m}|] \geqslant \sum_{n=1}^{m} E[|\mathbb{1}_{A_{2n}} \mathbb{1}_{A^c_{2n+1}} (H \bullet X)_{2m}|] \geqslant \sum_{n=1}^{m} \frac{1}{8n} \to \infty \text{ as } m \to \infty,$$

i.e., $(H \bullet X)$ is not uniformly integrable.

Solution 11.2 For $\Omega = \{\omega_1, \ldots, \omega_N\}$, denoting $X_T(\omega_n) := \xi_n$, $p_n = P(\omega_n)$ and $q_n = Q(\omega_n)$, the utility maximization problem is

$$\text{maximize } E_P[U(X_T)] = \sum_{n=1}^{N} p_n U(\xi_n) \tag{1}$$

subject to

$$E_Q[X_T] = \sum_{n=1}^{N} q_n \xi_n \leqslant x \tag{2}$$

for which the Lagrangian is

$$L(\xi_1, \ldots, \xi_N, y) = \sum_{n=1}^{N} p_n U(\xi_n) - y \left(\sum_{n=1}^{N} q_n \xi_n - x \right) = \sum_{n=1}^{N} p_n \left(U(\xi_n) - y \frac{q_n}{p_n} \xi_n \right) + yx. \tag{3}$$

We write

$$\Phi(\xi_1, \ldots, \xi_N) := \inf_{y>0} L(\xi_1, \ldots, \xi_N, y) \text{ and } \Psi(y) := \sup_{\xi_1, \ldots, \xi_N} L(\xi_1, \ldots, \xi_N, y).$$

The value function for the utility maximization problem (1)-(2) is

$$u(x) := \sup_{\xi_1, \ldots, \xi_N} \Phi(\xi_1, \ldots, \xi_N)$$

whilst for the dual value function v we have

$$\Psi(y) = \sum_{n=1}^{N} p_n V\left(y \frac{q_n}{p_n} \right) + yx = E_P \left[V\left(y \frac{dQ}{dP} \right) \right] + yx =: v(y) + yx.$$

For the dual optimizer $\widehat{y}(x)$, i.e., the solution to

$$\inf_{y>0} \Psi(y) = \inf_{y>0} \{v(y) + yx\},$$

the maximizer $\widehat{X}_T := (\widehat{\xi}_1, \ldots, \widehat{\xi}_N)$ for (3) with $y = \widehat{y}(x)$ fixed is given by

$$\widehat{\xi}_n := [U']^{-1} \left(\widehat{y}(x) \frac{q_n}{p_n} \right), \; n = 1, \ldots, N,$$

and the pair $\left(\widehat{X}_T, \widehat{y}(x) \right)$ is the unique saddle-point of the Lagrangian, i.e.,

$$u(x) = \sup_{X_T} \inf_{y>0} L(X_T, y) = L\left(\widehat{X}_T, \widehat{y}(x) \right) = \inf_{y>0} \sup_{X_T} L(X_T, y) = \inf_{y>0} \{v(y) + yx\}.$$

Solution 11.3 Let Q denote the risk-neutral measure and W^Q the corresponding Brownian motion. Under Q, the *undiscounted* stock price process $\widetilde{S}$ is given by

$$\widetilde{S}_t = e^{rt} S_t := e^{rt} S_0 \exp(\sigma W^Q_t - \frac{1}{2}\sigma^2 t), \; t \in [0, T],$$

where $S_t := S_0 \exp(\sigma W_t^Q - \frac{1}{2}\sigma^2 t)$ represents the *discounted* stock price at time $t \in [0,T]$ under the measure Q. We have

$$\widetilde{S}_T = e^{r(T-t)}\widetilde{S}_t \exp\left(\sigma(W_T^Q - W_t^Q) - \frac{1}{2}\sigma^2(T-t)\right),\ t \in [0,T].$$

The *discounted* value V_t of a power option at time t with undiscounted payoff $h(\widetilde{S}_T) = \widetilde{S}_T^p$ is the payoff's discounted $\mathcal{F}_t$-conditional Q-expected value, i.e.,

$$V_t = E_Q[e^{-rT}h(\widetilde{S}_T)|\mathcal{F}_t] = E_Q[e^{-rT}\widetilde{S}_T^p|\mathcal{F}_t].$$

We have

$$e^{-rT}\widetilde{S}_T^p = e^{prT-rT}\left(e^{-rT}\widetilde{S}_T\right)^p = e^{r(p-1)T}S_T^p, \tag{4}$$

where

$$\begin{aligned} S_T^p &= S_t^p \exp\left(\sigma p(W_T^Q - W_t^Q) - \frac{1}{2}\sigma^2 p(T-t)\right) \\ &= S_t^p \exp\left(\sigma p(W_T^Q - W_t^Q) - \frac{1}{2}\sigma^2 p^2(T-t)\right)\exp\left(\frac{1}{2}\sigma^2 p(p-1)(T-t)\right). \end{aligned}$$

The middle factor has $\mathcal{F}_t$-conditional Q-expectation 1; so we get

$$\begin{aligned} V_t &= S_t^p \exp\left(\frac{1}{2}\sigma^2 p(p-1)(T-t) + r(p-1)T\right) \\ &= e^{-rt}\widetilde{S}_t^p \exp\left(\left(\frac{1}{2}\sigma^2 p + r\right)(p-1)(T-t)\right), \end{aligned}$$

where we used that $S_t^p = e^{-rt}\widetilde{S}_t^p e^{-r(p-1)t}$; c.f. (4). The *undiscounted* value at time t is

$$\widetilde{V}_t = e^{rt}V_t = \widetilde{S}_t^p \exp\left(\left(\frac{1}{2}\sigma^2 p + r\right)(p-1)(T-t)\right).$$

Solution 11.4 For the value process

$$V_t = v(t, S_t),$$

the hedging strategy is

$$\vartheta_t = \frac{\partial v}{\partial x}(t, S_t),\ \eta_t = V_t - \vartheta_t S_t.$$

Since

$$V_t = S_t^p \exp\left(\frac{1}{2}\sigma^2 p(p-1)(T-t) + r(p-1)T\right),$$

we can compute

$$\vartheta_t = pS_t^{p-1}\exp\left(\frac{1}{2}\sigma^2 p(p-1)(T-t) + r(p-1)T\right)$$

and then obtain

$$\eta_t = (1-p)S_t^p \exp\left(\frac{1}{2}\sigma^2 p(p-1)(T-t) + r(p-1)T\right).$$

```
import numpy
from pylab import hist, show
from matplotlib.pyplot import subplot

from brownian import brownian
```

```python


#Function computes the forward integral of rows of a mxN-matrix w.r.t. another
def integral(x,y,m,N,out=None):

    if out is None:
        out = numpy.empty(x.shape)

    for i in range(m):
  for j in range(N):
    out[i,j+1]=out[i,j]+x[i,j]*(y[i,j+1]-y[i,j])

    return out


def main():

    # The Wiener process parameter.
    delta = 1
    # Total time.
    T = 1.0
    # Number of steps.
    N = 10000
    # Time step size
    dt = T/N
    # Number of realizations to generate.
    m = 1
    # Create empty arrays to store the realizations and integrals.
    x = numpy.empty((m,N+1))
    y = numpy.empty((m,N+1))
    z = numpy.empty((m,N+1))
    # Initial values of x,y,z,w.
    x[:, 0] = 0
    y[:, 0] = 0
    z[:, 0] = 0

    # Simulate the paths
    brownian(x[:,0], N, dt, delta, out=x[:,1:])

    # Volatility
    sigma = 0.4
    # Interest rate
    r = 0.06
    # Exponent for the power option
    p = 2.0

    # Form the geometric Brownian motion
    y = numpy.exp(sigma*x-.5*sigma**2.*numpy.cumsum(dt*numpy.ones((m,N+1))))

    # Compute the integral
    integral(p*y**(p-1)*numpy.exp( (.5*sigma**2.*p+r)*(p-1.)*T - .5*sigma**2*p*(p-1)*numpy.cumsum(dt*numpy.ones((m,N+1))) ),y,m,N,out=z)
```

```python

    # Print the terminal values
    print numpy.exp(r*(p-1)*T)*y[:,N]**p,numpy.exp( (.5*sigma**2.*p+r)*(p-1.)*T)+z[:,N]

if __name__ == "__main__":
    main()
```