Mathematical Finance

Solution sheet 8

Solution 8.1

(a) It is sufficient to show that the requirement that the martingale M in the sigma-martingale representation $X = H \bullet M$ with a predictable $H > 0$ can be relaxed to that M is a local martingale. Let $\tau_0 = 0$ and $(\tau_n)_{n\in\mathbb{N}}$ be the localizing sequence for M in $\mathcal{H}^1$. For each n, set $N^n := \mathbb{1}_{]\tau_{n-1},\tau_n]} \bullet M^{\tau^n}$ and choose $\alpha_n > 0$ such that $\sum_n \alpha_n ||N^n||_{\mathcal{H}^1} < \infty$. Then $N := \sum_n \alpha_n N^n$ is an $\mathcal{H}^1$-martingale and, for $J := \mathbb{1}_{\{0\}} + H \sum_n \alpha_n^{-1} \mathbb{1}_{]\tau_{n-1},\tau_n]} > 0$, we have $X = J \bullet N$.

(b) Let $X = M + A$, where M is a local martingale and A is a predictable FV process with $A_0 = 0$. It is sufficient to to show that $A = 0$. There exists predictable $H > 0$ such that $H \bullet X$ is a (local) martingale and without loss of generality we may assume that H is bounded. Indeed, for $H := \widetilde{H}^{-1} > 0$ given by the sigma-martingale decomposition $X = \widetilde{H} \bullet \widetilde{M}$, we have

$$H \bullet X = \widetilde{H}^{-1} \bullet (\widetilde{H} \bullet M) = \widetilde{M}$$

and

$$(H \wedge 1) \bullet X = \left(\frac{H \wedge 1}{H}\right) \bullet (H \bullet X)$$

which is a local martingale, since $H \bullet X$ is. Remark that

$$H \bullet A = (H \bullet A)_- + \Delta(H \bullet A) = (H \bullet A)_- + (H \bullet \Delta A),$$

so, the process $H \bullet A$ is predictable. Consequently, the process $H \bullet A = H \bullet X - H \bullet M$ is a predictale FV local martingale, so, $H \bullet A = 0$. For $J = \operatorname{sign}(A)$, we have

$$\int H|dA| = H \bullet (J \bullet A) = J \bullet (H \bullet A) = 0.$$

Since $H > 0$, we have $A = 0$.

Solution 8.2 Let τ be a random variable, uniformly distributed on $[0, 1]$, independent of a random variable ξ with $P(\xi = 1) = P(\xi = -1) = \frac{1}{2}$, and set

$$X_t := \mathbb{1}_{\{t \geqslant \tau\}} \xi.$$

The process X is a martingale w.r.t. its (augmented) natural filtration. We observe that all stopping times $\sigma \leqslant \tau$ on this filtration are of the form $\sigma = \tau \wedge t$, t fixed. Indeed, let $t := \sup\{s : P(\sigma > s) > 0\}$. Then, for any $s < t$, the random variables $\{X_u : u \leqslant s\}$ are all zero when restricted to the set $\{\tau > s\}$ and, therefore, any $A \in \mathcal{F}_s$ satisfies $P(A \mid \tau > s) = 0$ or 1. In particular, $P(\sigma > s \mid \tau > s) = 1$, and, therefore, $\{\tau > s \geqslant \sigma\}$ has zero probability. This holds for all $s < t$, giving $\sigma \geqslant \tau \wedge t$. By construction, $\sigma \leqslant t$ almost surely, giving $\sigma = \tau \wedge t$. Consider the sigma-martingale

$$Y_t = \int_0^t s^{-1} dX_s = \mathbb{1}_{\{t \geqslant \tau\}} \tau^{-1} \xi.$$

For any stopping time σ with $P(\sigma > 0) > 0$, we have $\sigma \wedge \tau = t \wedge \tau$ for some $t > 0$. So,

$$E|Y_\sigma| = E[\mathbb{1}_{\{\tau \leqslant t\}} \tau^{-1} |\xi|] = \int_0^{t \wedge 1} s^{-1} ds = \infty.$$

Therefore Y is not locally integrable, and cannot be a local martingale.

Solution 8.3 We consider the standard probability space $[0,1]$ endowed with the Lebesgue measure P and the filtration generated by the assets $(S_t^n)_{t\in[0,1]}$, $S_t^n = 0$ for $0 \leqslant t < 1$ and

$$S_1^n(\omega) = \begin{cases} -\frac{1}{\sqrt{\omega}} & \text{if } \omega \in [0, \varepsilon_n[, \\ \frac{1}{(1-\omega)^{\frac{1}{n+1}}} & \text{if } \omega \in [\varepsilon_n, 1], \end{cases}$$

where $(\varepsilon_n) \subset]0,1[$ is such that $E[S_1^n] = 1$ for all $n \in \mathbb{N}$ and $\varepsilon_n \to 0$ as $n \to \infty$. Indeed, such sequence (ε_n) exists; we have

$$E[S_1^n] = 1 \iff 1 + 2\sqrt{\varepsilon_n} = \frac{n+1}{n}(1-\varepsilon_n)^{\frac{n}{n+1}}$$

and the solution ε_n is approaching zero as $n \to \infty$. We next show that P is a separating measure for this LFM. By Fatou's lemma, it is sufficient to show that P is a separating measure for each sub-market consisting of the first n assets. Every terminal wealth X_1 on the n^{th} sub-market is of the form

$$X_1 = \sum_{k=1}^{n} c_k S_1^k$$

and $E[X_1] \leqslant 0$ for every 1-admissible X_1. Indeed, we have

$$E[X_1] = \sum_{k:c_k>0} c_k E[S_1^k] + \sum_{k:c_k<0} c_k E[S_1^k] = \sum_{k:c_k>0} c_k - \sum_{k:c_k<0} |c_k|$$

and since, for $\omega \in [0, \varepsilon_n]$, we have

$$X_1(\omega) = -\frac{1}{\sqrt{\omega}} \left(\sum_{k:c_k>0} c_k - \sum_{k:c_k<0} |c_k| \right)$$

we must have

$$E[X_1] \leqslant 0 \text{ for every } X \in \mathcal{X}_1.$$

On the other hand, if Q is an equivalent (sigma-)martingale measure for P, then

$$\int_0^{\varepsilon_n} \frac{1}{\sqrt{\omega}} dQ(\omega) = \int_{\varepsilon_n}^1 \frac{1}{(1-\omega)^{\frac{1}{n+1}}} dQ(\omega),$$

and, due to the equivalence, by DCT, the term on the left should go to zero as $n \to \infty$, but since

$$\int_{\varepsilon_n}^1 \frac{1}{(1-\omega)^{\frac{1}{n+1}}} dQ(\omega) \geqslant Q([\varepsilon_n, 1]),$$

this would in turn imply $P([\varepsilon_n, 1]) \to 0$, a contradiction.

Solution 8.4 Let W denote the driving Brownian motion for S. We note that $g(S_T) = \sigma W_T$. Indeed,

$$\log \frac{S_T}{S_0} + \frac{1}{2}\sigma^2 T = \sigma W_T.$$

We have

$$\sigma W_T = \sigma W_0 + (\sigma \bullet W)_T = 0 + (S^{-1} \bullet S)_T,$$

so that the self-financing strategy whose initial capital is $V_0 = 0$ and which at $0 \leqslant t \leqslant T$ holds $H_t = S_t^{-1}$ shares of stock and $\varphi_t^0 = V_t - H_t S_t = V_t - 1$ units of cash on the bank account replicates the payoff $g(S_T)$. Here,

$$V_t = V_0 + (H \bullet S)_t = (S^{-1} \bullet S)_t = \sigma W_t = \log \frac{S_t}{S_0} + \frac{1}{2}\sigma^2 t.$$

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

from brownian import brownian


#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:])

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

```python

    # Compute the integral
    integral(1./y,y,m,N,out=z)

    # Print the terminal values
    print x[:,N],z[:,N]

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