Mathematical Finance

Solution sheet 9

Solution 9.1 Remark, for a process X with independent increments, the following are equivalent:

(i) X is locally integrable,

(ii) X is integrable,

(iii) $\int_{[0,t]\times\mathbb{R}^d} \mathbb{1}_{\{||x||\geqslant 1\}}|x|d\nu(t,x) < \infty$ for every $t \geqslant 0$,

where ν is the jump measure for X.

(a) A non-negative sigma-martingale $\exp(L)$ is a local martingale, and since it has independent increments, it is a martingale; the implication (i) $\Longrightarrow$ (ii).

(b) Since the process L is a sigma-martingale, there exists a sequence $(A^n)_{n\in\mathbb{N}} \subset \mathcal{P}$ such that $\bigcup_{n=1}^{\infty} A^n = \Omega \times \mathbb{R}_+$ and $M^n := (\mathbb{1}_{A^n} \bullet L)$ is a local martingale for each n. The jump measure for M^n is $\nu^n(\omega,t,dx) = \mathbb{1}_{A^n}(\omega,t)\nu(dx)$, where ν is the jump measure for L, and since M^n is a local martingale, it satisfies (iii) for almost every ω. Consequently, ν satisfies (iii) and the process L is a martingale.

Solution 9.2 Denote by f the density for S_T and recall the Leibniz integration rule

$$\frac{\partial}{\partial K}\int_{a(K)}^{b(K)} f(x,K)dx = \frac{db(K)}{dK}f(b(K),K) - \frac{da(K)}{dK}f(a(K),K) + \int_{a(K)}^{b(K)} \frac{\partial}{\partial K}f(x,K)dx.$$

For a call option

$$C(K) = \int_0^\infty (x-K)^+ f(x)dx = \int_K^\infty (x-K)f(x)dx$$

we get

$$\frac{\partial C}{\partial K} = 0 - (K-K)f(K) - \int_K^\infty f(x)dx = -\int_K^\infty f(x)dx.$$

Since

$$1 = \int_0^\infty f(x)dx = \int_0^K f(x)dx + \int_K^\infty f(x)dx,$$

by the Fundamental Theorem of Calculus, we get

$$\frac{\partial^2 C}{\partial K^2}(K) = \frac{\partial}{\partial K}[\int_0^K f(x)dx - 1] = f(K)$$

and similarly for a put option

$$\frac{\partial^2 P}{\partial K^2}(K) = f(K).$$

So,

$$
\begin{aligned}
E[w(S_T)] &= \int_0^\infty w(K)f(K)dK = \int_0^{S_0} w(K)\frac{\partial^2 P}{\partial K^2}(K)dK + \int_{S_0}^\infty w(K)\frac{\partial^2 C}{\partial K^2}(K)dK \\
&= [w(K)\int_0^K f(x)dx]\Big|_0^{S_0} - \int_0^{S_0} w'(K)\frac{\partial P}{\partial K}(K)dK + [w(K)(\int_0^K f(x)dx - 1)]\Big|_{S_0}^\infty - \int_{S_0}^\infty w'(K)\frac{\partial C}{\partial K}(K)dK \\
&= w(S_0) - \int_0^{S_0} w'(K)\frac{\partial P}{\partial K}(K)dK - \int_{S_0}^\infty w'(K)\frac{\partial C}{\partial K}(K)dK \\
&= w(S_0) - [w'(K)P(K)]\Big|_0^{S_0} + \int_0^{S_0} w''(K)P(K)dK - [w'(K)C(K)]\Big|_{S_0}^\infty + \int_{S_0}^\infty w''(K)C(K)dK \\
&= w(S_0) + \int_0^{S_0} w''(K)P(K)dK + \int_{S_0}^\infty w''(K)C(K)dK.
\end{aligned}
$$

Solution 9.3 The process V is a supermartingale of class $(\mathcal{D})$, so, it admits a Doob-Meyer decomposition, which is

$$V_t = V_0 + M_t^V - A_t^V,\ t \in [0,T],$$

where M^V is a martingale vanishing at zero, and A^V is an integrable predictable increasing process vanishing at zero. We have

$$\sup_{0\leqslant t\leqslant T} |M_t^V| \leqslant \sup_{0\leqslant t\leqslant T} E[\sup_{0\leqslant s\leqslant T} |U_s| \mid \mathcal{F}_t] + |V_0| + A_T^V,$$

so, $M^V \in \mathcal{H}_0^1$. Since

$$U_t \leqslant V_t = V_0 + M_t^V - A_t^V,$$

we have

$$
\begin{aligned}
\inf_{M\in\mathcal{H}_0^1} E[\sup_{0\leqslant t\leqslant T}(U_t - M_t)] &\leqslant E[\sup_{0\leqslant t\leqslant T}(U_t - M_t^V)] \\
&\leqslant E[\sup_{0\leqslant t\leqslant T}(V_t - M_t^V)] \\
&\leqslant E[\sup_{0\leqslant t\leqslant T}(V_0 - A_t^V)] \\
&= V_0.
\end{aligned}
$$

On the other, for any $M \in \mathcal{H}_0^1$, we have

$$V_0 = \sup_{0\leqslant \tau\leqslant T} EU_\tau = \sup_{0\leqslant \tau\leqslant T} E[U_\tau - M_\tau] \leqslant E[\sup_{0\leqslant t\leqslant T}(U_t - M_t)],$$

i.e.,

$$V_0 \leqslant \inf_{M\in H_0^1} E[\sup_{0\leqslant t\leqslant T}(U_t - M_t)].$$

Solution 9.4

```python
import numpy
from scipy.stats import norm
from math import exp, log, pi, sqrt

from brownian import brownian


# The Wiener process parameter.
```

```
delta = 1
# Total time.
T = 0.5
# Number of steps.
N = 50
# Time step size
dt = T/N
# Number of realizations to generate.
m = 5000
# Create an empty array to store the realizations.
x = numpy.empty((m,N+1))
# Initial values of x.
x[:, 0] = 0

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

# The terms j=1,2 for the Black-Scholes formula
def d_j(j, S, K, r, sigma, T):
    return (log(S/K) + (r + ((-1)**(j-1))*0.5*sigma*sigma)*T)/(sigma*(T**0.5))

# The Black-Scholes price for the European put option
def european(S, K, r, sigma, T):
    return -S*norm.cdf(-d_j(1, S, K, r, sigma, T))+K*exp(-r*T) * norm.cdf(-d_j
    (2, S, K, r, sigma, T))

# The initoal value for the stock
S0 = 100.
# The strike price
K = 100.
# The volatility
sigma = 0.4
# The interest rate
r = 0.06

# The upper bound for the price of the American put option
v = european(S0,K,r,sigma,T)
for i in range(m):
  for j in range(N):
    x[i,j] = exp(-r*j*dt)*max((K-S0*exp(sigma*x[i,j]+(r-.5*sigma**2)*j*dt)),0)
    - (european(S0*exp(sigma*x[i,j]+(r-.5*sigma**2)*j*dt),K,r,sigma,T-j*dt)-v)

print numpy.mean(numpy.amax(x, axis=0))
```