Mathematical Finance

Solution sheet 7

Solution 7.1 First let $p : X \to \mathbb{R}$ be a sublinear functional on a vector space. If p is linear and $f(x) \leqslant p(x)$ for all $x \in X$ for some linear functional $f : X \to \mathbb{R}$, then $-f(x) = f(-x) \leqslant p(-x) = -p(x)$, so $p(x) \leqslant f(x)$ for all $x \in X$, i.e., $f = p$. Now assume that p dominates exactly one linear functional on X and that p is not linear. Then there exists some $x_0 \neq 0$ such that $-p(-x_0) < p(x_0)$. Let $M := \{\lambda x_0 : \lambda \in \mathbb{R}\}$ be the vector space generated by x_0 and define the linear functionals $f, g : M \to \mathbb{R}$ by $f(\lambda x_0) = \lambda p(x_0)$ and $g(\lambda x_0) = -\lambda p(-x_0)$. From $f(x_0) = p(x_0)$ and $g(x_0) = -p(-x_0)$ we see that $f \neq g$. Next, notice that $f(z) \leqslant p(z)$ and $g(z) \leqslant p(z)$ for each $z \in M$, that is, p dominates both f and g on the subspace M. Now, by the Hahn-Banach theorem, the two distinct linear functionals f and g have linear extensions to all of X dominated there by p, a contradiction.

Solution 7.2 Let Q be an equivalent martingale measure for P which exists by (NA). By Doob's maximal inequality, for every 1-admissible H, we have

$$Q((|H \circ X)^*_T| > K) \leqslant \frac{2E_Q[(H \circ X)^-_T]}{K} \leqslant \frac{2}{K}$$

for $K > 0$, so, the family $\{(H \circ X)_T : H \text{ is 1-admissible}\}$ is bounded in Q and consequently in P, i.e., $(NUPBR)$ holds.

Solution 7.3 Let P be the Lebesgue measure on $\Omega :=]0, 1]$. Define

$$X^{n,k} := n^2 \mathbb{1}_{](k-1)/n, k/n]}, \ k = 1, \ldots, n \in \mathbb{N}.$$

We have

$$P(X^{n,k} > K) = \begin{cases} 1/n & \text{if } n > \sqrt{K}, \\ 0 & \text{otherwise,} \end{cases}$$

so, the family $\{X^{n,k}\}$ is bounded in probability, but since

$$\sum_{k=1}^{n} \frac{1}{n} X^{n,k} = n \in \mathrm{co}(X^{n,k}), \ n \in \mathbb{N},$$

the convex hull of $\{X^{n,k}\}$ is not bounded in probability.

Solution 7.4 Let B and C be two independent Brownian motions. Define

$$L_t := \exp\left(B_t - \frac{1}{2}t\right) \text{ and } N_t := \exp\left(C_t - \frac{1}{2}t\right)$$

and

$$\tau := \inf\left\{t \geqslant 0 : L_t = \frac{1}{2}\right\} \text{ and } \sigma := \inf\left\{t \geqslant 0 : N_t = 2\right\}.$$

By the properties of Brownian motion, we get

$$P(\tau = \infty) = 0 \text{ and } P(\sigma = \infty) = P(\sigma < \infty) = \frac{1}{2}.$$

Let $X = L^{\tau\wedge\sigma}$ and $Y = N^{\tau\wedge\sigma}$. From above, we conclude that $X_\infty > 0$ a.s. and $Y_\infty > 0$ a.s. Moreover, we have

$$E[X_{\tau\wedge\sigma}] = E[L_{\tau\wedge\sigma}] = E[L_\tau 1_{\{\sigma=\infty\}}] + E[L_{\tau\wedge\sigma} 1_{\{\sigma<\infty\}}].$$

For the first term, we get

$$E[L_\tau 1_{\{\sigma=\infty\}}] = P(\sigma = \infty)E[L_\tau] = \frac{1}{4}$$

by independence, and for the second term,

$$E[L_{\tau\wedge\sigma} 1_{\{\sigma<\infty\}}] = \int_0^\infty P(\sigma \in dt)E[L_{\tau\wedge t}] = \int_0^\infty P(\sigma \in dt) = P(\sigma < \infty) = \frac{1}{2}$$

by independence and the optional stopping theorem. Hence,

$$E[X_\infty] = E[X_{\tau\wedge\sigma}] = \frac{1}{4} + \frac{1}{2} = \frac{3}{4} < 1,$$

and so we conclude that X is a strict local martingale. However, the process $Y = (N^\sigma)^\tau$ is a uniformly bounded martingale (since N^σ is) and the product XY is a uniformly integrable martingale. Indeed, we have

$$E[X_\infty Y_\infty] = E[L_{\tau\wedge\sigma} N_{\tau\wedge\sigma}] = E[L_{\tau\wedge\sigma} N_\sigma] = 2E[L_{\tau\wedge\sigma} \mathbb{1}_{\{\sigma<\infty\}}] = 2P(\sigma < \infty) = 1.$$

Finally, define the process S as $dS = dM + d\langle M, M\rangle$, where $X = \mathcal{E}(-M)$. The measure Q defined as

$$dQ = X_\infty Y_\infty dP$$

is an equivalent local martingale measure for S. The density X_∞ does not define a probability measure.

Solution 7.5 The discounted asset prices X^1 and X^2 satisfy

$$dX^1 = X^1 dB + \frac{1}{2}X^1 dt \text{ and } dX^2 = \frac{1}{2}X^2 dB + \frac{1}{8}X^2 dt,$$

so, we try

$$H^1 = \frac{X^2}{X^1} \text{ and } H^2 = -2.$$

```
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])
```

```python

    return out


def main():

    # The Wiener process parameter.
    delta = 1
    # Total time.
    T = 1.0
    # Number of steps.
    N = 1000
    # Time step size
    dt = T/N
    # Number of realizations to generate.
    m = 5000
    # 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:])

    # Compute the integrals
    integral(numpy.exp(.5*x)/numpy.exp(x),numpy.exp(x),m,N,out=y)
    integral(-2.0*numpy.ones((m,N+1)),numpy.exp(.5*x),m,N,out=z)

    # Plot the terminal distribution
    hist(y[:,N]+z[:,N],normed=True,bins='auto')
    show()

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