ECG790C - Problem Set 4 - Spring 2009

1. Write a path generator for the general multivariate SDE

   $$dS = \mu(S)dt + \sigma(S)dW$$

   where $S \in \mathbf{R}^d$, $\mu : \mathbf{R}^d \to \mathbf{R}^d$ and $\sigma : \mathbf{R}^d \to \mathbf{R}^{d\times d}$. As an added feature allow the user to pass d-vectors of lower and upper bounds on S (by default these can be $-\inf$ and inf). As an additional feature, allow it to compute and return only the end period path values.

   Test this code on the process from question 2 of the exam and use it to produce histograms of the terminal values of Z (use a time horizon that is large enough so this approximates the long-run behavior of Z).

2. This problem leads you through a statistical application using simulation. Consider estimating the parameters of the process

   $$dS = \alpha(\mu - S)dt + \sigma S dW$$

   Define the vector of parameters

   $$\theta = [\alpha\ \mu\ \sigma]^\top$$

   Given data on S_t, we would like to estimate θ.

   One estimation method uses simulation to match moments of the data to moments obtained from simulated data. Suppose that β is a vector of statistics (such as sample means or standard deviations) and we estimate $\hat{\beta}$ from the data. We can then simulate S for a given value of θ and estimate $\tilde{\beta}(\theta)$ from the simulated data. We would like to choose θ so that $\tilde{\beta}(\theta)$ and $\hat{\beta}$ are as close as possible; specifically we will use the following objective

   $$\min_\theta\ f(\theta) = \left(\tilde{\beta}(\theta) - \hat{\beta}\right)^\top \left(\tilde{\beta}(\theta) - \hat{\beta}\right)$$

   (note that β must have at least as many elements as θ).

   First write a function

   ```
   s=spath(theta,s0,Delta,dW)
   ```

   that generates a single path (the inputs should be clear at this point).

Next write a function that computes β

`beta=moments(y)`

For the example I have in mind consider the regression function

$$s_{t+\Delta} = c_1 + s_t c_2 + e_t$$

In general the coefficients of a regression equation $y = Xc + e$ can be obtained using `c=X\y` in MATLAB (and estimates of e can be obtains using `e=y-X*c`). Hence the regression equation I'm looking for treats y as observations 2 through n of the s path and X as a 2-column matrix consisting of a vector of ones and the first $n-1$ elements of the s path. Have the `moments` function define

$$\beta = [c_0 \; c_1 \; std(e) \; mean(y) \; std(y)]^\top$$

Next write a function that computes the objective function $f(\theta)$. It should have the form

`f=smmobjfun(theta,pathf,momf,s0,Delta,dW,betahat)`

Here `pathf` and `momf` are function handles that call the path simulator and the moment function. The function should first generate simulated values using `pathf` and then compute $\tilde{\beta}$ using `momf`. Finally it should compute and return the value of the objective function $f(\theta)$.

To use all of this to obtain an estimate of θ first compute $\hat{\beta}$ using observed data. Then obtain a set of simulated values of the Brownian increments (`dW`) and pass `smmobjfun` to an optimization solver along with all of the necessary information needed by `smmobjfun`.

Run this code on the data that I send by email. Δ for this data is 1/50. For this procedure to work you should use paths with at least 50000 time steps and more is better. This is a computationally intensive method of estimation and it is essential that the functions to generate the paths and to compute β be as efficient as possible; otherwise the procedure may take a very long time to run. It is also hard to obtain accurate derivative values so a derivative free method such as `direct` or `genetic` is recommended. Both algorithms require the specification of a search area; $[0,1]^3$ will work for this example.