ECG790C - Homework 2

1. Consider the hybrid process

$$dS_t = \mu(z_t)dt + \sigma(z_t)dw$$

where S is a scalar diffusion process and z_t is a Poisson process taking values in the set $\{1, \ldots, m\}$. Jumps in z_t are governed by the $m \times m$ intensity matrix Λ, where $\lambda_{ii} = 0$ and $\lambda_{ij} \geq 0$. Specifically, the probability that, if $x_t = i$, no jump occurs within a period Δ is $\exp(-\Delta \sum_j \lambda_{ij})$ and the probability that a jump occurs to z_j is

$$Prob(z_{t+\Delta} = j | z_t = i) = \frac{\lambda_{ij}}{\sum_j \lambda_{ij}}(1 - \exp(-\Delta \sum_j \lambda_{ij}))$$

This is correct to the extent that Δ is small enough that the probability of more than one jump in a given period is vanishingly small.

   (a) Write a MATLAB function that accepts a matrix Λ and a time period Δ and returns the probability matrix P where

$$P_{ij} = Prob(z_{t+\Delta} = j | z_t = i)$$

   (b) Write a MATLAB function that simulates values of the hybrid process given above over the period $[0, T]$. It should accept m-vectors μ and σ, an $m \times m$ matrix Λ, the starting values S_0 and z_0, the time horizon T, the number of time steps, n. If $z_t = i$, the value of $S_{t+\Delta}$ can be approximated using

$$S_{t+\Delta} \approx S_t + \mu_i \Delta + \sigma_i \sqrt{\Delta} e_t$$

   where e is standard normal. The value of $z_{t+\Delta}$ can be approximated by taking a draw from the discrete distribution with probability vector $[p_{i1}\ p_{i2} \ldots p_{im}]$. This can be accomplished using a random draw u from a Uniform(0,1) variable (using MATLAB function `rand`) and setting

$$z = 1 + \sum_{k=1}^{m} 1_{u > \sum_{j=1}^{k} p_j}$$

   where 1_c is 1 if c is true and 0 otherwise.

2. Consider the function

$$C(u, v, a) = \frac{a}{\ln\left(e^{a/u} + e^{a/v} - e^{a}\right)}$$

where $u, v \in [0, 1]$ and $a \in (0, \infty)$. The limiting cases are

$$C(u, v, 0) = \frac{uv}{u + v - uv}$$

and $C(u, v, \infty) = \min(u, v)$. Also $C(0, v, a) = C(u, 0, a) = 0$, $C(1, v, a) = v$ and $C(u, 1, a) = u$. Write a MATLAB function that accepts vectors of u and v and scalar a and returns vectors of C. It should be written to handle limiting cases of u, v and a and to address potential overflow and other numerical problems. To do this you will need to write the function in a mathematically equivalent form. Test your code on a grid of values for u, v and a. Provide a script file that conducts the tests you perform.