*M.Phil. in Mathematical Finance*

Numerical Methods in Finance 2020

Tutorial 11

---

## Problems

1. A van der Corput sequence is a deterministic sequence of numbers (on the interval (0,1)) that is used in Monte Carlo simulations instead of uniform random numbers.

   Given a natural number $m$, represented in base $b$ by

   $$m = \sum_{i=0}^{l-1} a_i b^i,$$

   where $l$ is the number of digits, the corresponding van der Corput number is given by

   $$g_b(m) = \sum_{i=0}^{l-1} a_i b^{-i-1}. \tag{1}$$

   The algorithm for generating a sequence of $n$ numbers for base $b$ is as follows:

   Your function `vanderCorput.m` should take inputs $(n, b)$

   - **Step 1:** Create a vector $m$ with the first $n$ natural numbers (Recall that natural numbers start from 0);
   - **Step 2:** By using the Matlab function `dec2base` convert the vector $m$ into its base $b$ representation.
   - **Step 3:** Convert the matrix of characters formed in the previous step to a matrix of integers, called a, by subtracting `double('0')` (**NB**, this is the number zero not the letter O) from every element.
   - **Step 4:** Now, for each row generate a corresponding van der Corput number using equation (1).
     **Hint:** If you have vectorized this correctly, you should not need to use any loops.
   - **Step 5:** Return the vector of van der Corput numbers you have generated.

   If you have implemented the algorithm correctly, >> `vanderCorput(5,2)` should return the vector

   $$\begin{bmatrix} 0 & 0.5 & 0.25 & 0.75 & 0.125 \end{bmatrix}^T.$$

   Now use your function to generate 10 numbers with $b = 3$.
   **Challenge:** rewrite your function so that it does not use `dec2base`.

2. A cash-or-nothing call option has the payoff function $X\mathbb{I}_{S_T > K}$, where $X$ is the amount of cash and $K$ is the strike price. A closed form pricing formula for the option is given by

   $$c = Xe^{-rT}\Phi(d) \qquad \text{where} \qquad d = \frac{\log(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}.$$

   With parameters

   $$T = 2, \qquad S_0 = 100, \qquad \sigma = 40\%, \qquad r = 10\%, \qquad K = 110, \qquad X = 10,$$

   compute and plot crude Monte Carlo estimates for the option price as a function of sample sizes in the range $n = 1000, 2000, \ldots, 50000$. Compute and plot the Black-Scholes price and plot three-standard deviation bounds for the Monte Carlo estimates around the Black-Scholes price.

   Now, with the same sample sizes, use van der Corput numbers, generated with base $b = 3$, to price the option. Remember to exclude the first van der Corput number (0). In order to compute error bounds for your estimate, note that expressions for the discrepancies are given by

   $$\lim_{n\to\infty} D_{n,b} = \lim_{n\to\infty} D^*_{n,b} = \begin{cases} \frac{\log n}{n} \frac{b^2}{4(b+1)\log b} & \text{for even } b, \\ \frac{\log n}{n} \frac{b-1}{4\log b} & \text{for odd } b, \end{cases}$$

   where $b$ is the base used to generate the van der Corput sequence. Use this expression in conjunction with the variation for the payoff function (which is easily computed in closed form) to plot the Koksma-Hlawka error bound for the Quasi-Monti Carlo estimate.

   For debugging purposes, initialize the seed at the beginning of your program with the Matlab command `rng(0);`.