M.Phil. in Mathematical Finance

Numerical Methods in Finance 2020

Tutorial 10

Problems

1. Consider an additive (vanilla) cliquet call option (also called a ratchet option) with $N = 4$ equi-spaced start dates, which has a **discounted** payoff of

$$\sum_{i=1}^{N} e^{-rih}(S_{ih} - S_{(i-1)h})^{+},$$

where $h = T/N$, $T = 1$ is the terminal time and $r = 10\%$ is the risk free rate. This is a compound option which is essentially a sum of forward starting call options struck at the money. The analytical price for this option is given by

$$c = NS_0(\Phi(d_1) - e^{-rh}\Phi(d_2)),$$

where

$$d_1 = \frac{(r + \frac{1}{2}\sigma^2)h}{\sigma\sqrt{h}}, \qquad d_2 = d_1 - \sigma\sqrt{h},$$

and $S_0 = 100$ and $\sigma = 30\%$.

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

Hint: when coding the discounted payoff function vectorise by using matrix multiplication and the Matlab `diff` function. When generating the normal variates, use the command `randn(n,N)` to aid with debugging (as usual set the seed at the beginning of your script using `rng(0);`). Also note that paths are rows, so modify your sum command appropriately or use matrix multiplication with your discount factors.

Now, compute and plot end point stratified Monte Carlo estimates (and three standard deviation boundaries) for the same sample size range. Use $d = 50$ equally probable strata for the terminal stock prices. You can do this (for each stratum $1 \le i \le d$) by generating n/d uniform variates from $\mathscr{U}\left[\frac{i-1}{d}, \frac{i}{d}\right]$ and using the inverse method to generate the normal variates and, in turn, the terminal Weiner process values. Before calling your function to generate the Brownian bridges, generate the other $(n/d, N-1)$ random numbers required (for the increments prior to the terminal time) by using the command `randn(n/d,N-1)` — this will help with debugging.

2. In this problem you will produce approximations to the equidistributed sequences formed by multiples of the irrational numbers $\sqrt{3}$ and $\sqrt{7}$. Note that computers are not able to store representations of irrational numbers, but with a bit of careful manipulation we can avoid egregious machine precision errors.

Consider the sequence that makes use of $\sqrt{3}$: $0, \{\sqrt{3}\}, \{2\sqrt{3}\}, \{3\sqrt{3}\}, \ldots$. The ith term in this sequence can be written as

$$\begin{aligned} z_i(\sqrt{3}) &= \left\{(i-1)\sqrt{3}\right\} \\ &= \left\{\sqrt{3(i-1)^2}\right\} \\ &= \sqrt{3(i-1)^2} - \left\lfloor \sqrt{3(i-1)^2} \right\rfloor, \end{aligned} \tag{1}$$

where $\lfloor \cdot \rfloor$ can be implemented by the Matlab `floor` function. Thus, as long as we implement the method using this last expression and keep the number of terms relatively low (say 50000) we won't run into serious machine precision issues.

Generate the first 50000 terms in the equidistributed sequences formed using $\sqrt{3}$ and $\sqrt{7}$ as irrational numbers. Side by side plot histograms (with 100 bins) for the two sequences. In a separate plot, produce a scatter plot of the two dimensional sequence $S_n = \left\{ x_i(\sqrt{3}), x_i(\sqrt{7}) \middle| 1 \le i \le 1000\right\}$

Now, compute and graph (in a separate figure) estimates of the star discrepancy for the two dimensional equidistributed sequence $S_n = \left\{ x_i(\sqrt{3}), x_i(\sqrt{7}) \middle| 1 \le i \le n\right\}$ as a function of sample sizes $n = 1000, 2000, \ldots, 50000$. To compute the estimates of the star discrepancy, consider only the rectangles with one vertex at the origin that have areas $0.1i \times 0.1j$ for $1 \le i, j \le 10$ $(i, j \in \mathbb{N})$.