Chapter 2 Monte Carlo Simulation (v.17.1)

The most common use of Monte Carlo Simulation in finance is when one needs to calculate the expected value of a functional $\mathbb{E}f(X)$. Assume $g(\cdot)$ is the density function of random variable X. Then we can express the expectation as an integral:

$$\mathbb{E}f(X) = \int_{-\infty}^{\infty} f(t)g(t)dt.$$

If this integral can't be computed explicitly, then Monte Carlo simulation techniques are adopted to estimate it. The idea is to use the Law of large Numbers (LLN) to estimate the integral.

Suppose $\{X_i\}^n{}_{i=1}$ is a sample of $i.i.d.$ random variables with the same distribution as X and $\mathbb{E}(X) = \mu,\ Var(X) = \sigma^2 < \infty$.

For the Sample Mean for random sample $\{f(X_1), f(X_2), \dots, f(X_n)\}$ defined as

$$\overline{fX}_n = \frac{f(X_1)+f(X_2)+ \dots + f(X_n)}{n}$$

we have

$$\lim_{n\to\infty} \mathbb{E}\left(\frac{f(X_1) + f(X_2) + \ \dots + \ f(X_n)}{n} - \mathbb{E}f(X)\right)^2 = \lim_{n\to\infty} \mathbb{E}\big(\ \overline{fX}_n - \mathbb{E}f(X)\big)^2 = 0.$$

That is, $\overline{fX}_n \approx \ \mathbb{E}f(X)$ for large enough n.

Thus, the simple Monte Carlo method is enough to estimate the expected values of complicated functionals of random variables. The method becomes even more appealing as the dimensionality of the problem increases.

Naturally, the larger the size of the sample (n) is , the closer the approximation would be to the actual value. A natural question arises then: what is the relative error of the Monte Carlo method? Or, what is the convergence rate of the method?

In this section we will study the error of Monte Carlo method, and a few methods that would help to reduce those errors and thus make the algorithms converge to the actual number faster.

Define the Sample Mean $\bar{X}_n$ and the Sample Variance S_n^2 of the sample $\{X_i\}^n{}_{i=1} \sim i.i.d.$ with the same distribution as the random variable X, where

$$\mathbb{E}(X) = \mu, \qquad Var(X) = \sigma^2 < \infty.$$

$$\bar{X}_n = \frac{X_1 + X_2 + \ldots + X_n}{n}, \quad S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \bar{X}_n)^2.$$

Assume we are trying to estimate the unknown mean μ. Notice that the sample mean is an unbiased estimator for μ:

$$\mathbb{E}(\bar{X}_n) = \mu$$

Then, we have

$$Var(\bar{X}_n) = \mathbb{E}(\bar{X}_n - \mu)^2 = \frac{\sigma^2}{n}.$$

It can be seen from the latter that the square-distance of the Sample Mean and the actual expected value converges to 0 as n gets larger.

We have that $\boldsymbol{P}\left(-z_{\frac{\alpha}{2}} < Z < z_{\frac{\alpha}{2}}\right) = 1 - \alpha$ and we know that $\frac{\bar{X}_n - \mu}{\frac{\sigma}{\sqrt{n}}} \to Z \sim N(0,1)$ as $n \to \infty$, due to the Central Limit Theorem.

Therefore, we can write that $\boldsymbol{P}\left(-z_{\frac{\alpha}{2}} < \frac{\bar{X}_n - \mu}{\frac{\sigma}{\sqrt{n}}} < z_{\frac{\alpha}{2}}\right) \approx 1 - \alpha$.

This will give us the $(1 - \alpha)$ confidence interval for the unknown mean μ:

$$\bar{X}_n \pm \frac{z_{\alpha/2}}{\sqrt{n}}\sigma$$

However, in most realistic cases σ is unknown and needs to be estimated. Replacing the standard deviation σ by the sample standard deviation S_n, we derive the $(1 - \alpha)$ confidence interval for the unknown mean μ , which will be given by:

$$\bar{X}_n \pm \frac{t_{\alpha/2}}{\sqrt{n}} S_n$$

Notice, that when the sample standard deviation is used instead of σ, the distribution of the standardized sample mean is no longer normal, but it is t-distributed with (n-1) degrees of freedom. However, for large sample sizes and using the Central Limit Theorem, we can still use the z-values in the confidence interval derivation.

Thus, the error of the approximation is given by $\frac{z_{\alpha/2}}{\sqrt{n}} S_n$, which implies the following:

Lemma: Assume ε is the desired error of the Monte Carlo simulation. Then, the size of the sample $\hat{n}$ should be chosen so that $\frac{z_{\alpha/2}}{\sqrt{n}} S_n \leq \varepsilon$.

Assume that S_n is bounded. Then the error term of the approximation is bounded by $\frac{K}{\sqrt{n}}$, where $K < \infty$ is a constant. That is, the convergence rate of Monte Carlo method is $n^{-1/2}$. It implies that, if we want to achieve an accuracy of 0.001 in our approximation, then the number of simulations should be about $n = 10^6$.

One can increase n and achieve any accuracy in the Monte Carlo simulation. However, this may be computationally costly and not applicable in some cases. There are a few alternative ways to increase the accuracy of the approximation that result in faster convergence of the algorithm. These methods are designed to reduce the variance of the approximation and are titled variance reduction techniques in the literature.

Example: Suppose we want to price a European call option by simulation.

First, to simulate W_T, notice that $W_T \approx \sqrt{T}\,Z$ where $Z \sim N(0,1)$. We have $c = e^{-rT}\mathbb{E}^*(S_T - X)^+$, where $S_T = S_0 exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma W_T\right)$. Then, we would follow these steps to estimate c by simulations:

STEP 1. Generate $\{Z_i\}_{i=1}^n \sim iid\ N(0,1)$,

STEP 2. Compute the payoff of the call option price in scenario i:

$$payoff_i = \left(S_0\ e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z_i\right)} - X\right)^+$$

STEP 3. Take $\tilde{c} = e^{-rT}\ \frac{1}{n}\sum_{i=1}^n payoff_i = e^{-rT}\ \frac{1}{n}\sum_{i=1}^n \left(S_0\ e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z_i\right)} - X\right)^+$

Then $\tilde{c}$ is an estimated value of c.

2.1 Variance Reduction Techniques

In this section we will discuss some techniques for improving the speed and efficiency of a Monte Carlo simulation algorithm, usually called "Variance Reduction Techniques".

The success of any variance reduction technique is dependent on the problem of interest. Since there is no theory to show which method should be used for what type of problems, preliminary runs are necessary to assess a variance reduction technique's effectiveness.

Suppose we want to estimate a parameter θ and that we can generate a sequence of random variates to estimate: $\{X_n\}_{n=1}^{N} \sim i.i.d.$ with $\mathbb{E}(X_n) = \theta$ and $Var(X_n) = \sigma^2$.

If we take the Sample Mean $\bar{X}_n = \frac{X_1 + X_2 + \ldots + X_n}{n}$ of the sequence, then we know that $\mathbb{E}\bar{X}_n = \theta$. That is, the Sample Mean is an *unbiased estimator* of θ.

It follows from the Central Limit Theorem (CLT) that for large enough n we have

$$\frac{\bar{X}_n - \theta}{\sigma/\sqrt{n}} \sim N(0,1)$$

That is, the Sample Mean not only is an unbiased estimator for the unknown actual mean θ, but it converges (in distribution) to it. It follows from the above that it would be desired to have as low variance (variation) in our estimates (for θ) as possible.

Thus, having lower variance of the estimators would help the algorithm converge faster. That is, if we have two sequences of unbiased estimators of θ, then we'll take the one that has smaller variance.

2.1.1 Antithetic Variates

This method is widely used in problems of pricing financial instruments. Suppose we want to estimate $\theta = \mathbb{E}(X)$ and that we can generate X_1 and X_2 so that $\mathbb{E}X_1 = \mathbb{E}X_2 = \theta$. Both $X_1$and X_2 are unbiased for θ, and so is $\frac{X_1 + X_2}{2}$. We have

$$Var\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}(VarX_1 + VarX_2 + 2Cov(X_1, X_2))$$

The variance of the estimator $\frac{X_1 + X_2}{2}$ would be reduced if the variates X_1 and X_2 were negatively correlated (they are not independent). That is, if $Cov(X_1, X_2) < 0$ then $\frac{X_1 + X_2}{2}$ will have lower variance than both X_1 and X_2. This is the basic idea behind the Antithetic Variates variance reduction technique.

Example: Suppose we want to compute $\theta = \mathbb{E}(e^U)$ for $U \sim U[0,1]$. The actual value of the integral is $e - 1$.

Define $X = e^U$. Take $X_1 = e^{U_1}, X_2 = e^{U_2}$. If $U_2 = 1 - U_1$ then $Cov(X_1, X_2) = -0.2342 < 0$. Also, $Var\ e^U = 0.2420$. Thus, $Var\left(\frac{e^{U_1} + e^{U_2}}{2}\right) = 0.1210$ if U_1 is independent of U_2. However, $Var\left(\frac{e^U + e^{1-U}}{2}\right) = 0.0039$. This is a reduction of a variance by about 97%.

In general, suppose we are interested in estimating $E\ h(X_1, X_2, \ldots, X_n)$ where $\{X_i\}_{i=1}^n$ are independent random variables. Assume we are able to simulate a sequence $\{X_i\}_{i=1}^n$ by the inverse-transformation method. $\{X_i\}_{i=1}^n$are generated by $\{U_i\}_{i=1}^n \sim i.i.d.\ U[0,1]$, so that $F^{-1}(U_i)$ has the same distribution as X_i.

Define $Y_1 = h(\, F^{-1}(U_1), \ldots, F^{-1}(U_n))$ and $Y_2 = h(\, F^{-1}(1-U_1),\ \ldots, F^{-1}(1-U_n))$. Then Y_1 and Y_2 have the same distribution. If $h(\cdot)$ is a monotone function, then Y_1 and Y_2 are negatively correlated and the antithetic variates estimate $\frac{Y_1+Y_2}{2}$ has a lower variance than either one of Y_i. The following result helps to justify the method:

Lemma: If $h(x_1, x_2, \ldots, x_n)$ is a monotone function of its arguments, then for $\{U_i\}^n{}_{i=1} \sim i.i.d.\ U[0,1]$, we have $Cov(\, h(U_1, U_2, \ldots, U_n),)\ h(1-U_1, 1-U_2, \ldots, 1-U_n) \leq 0$.

In general, for each simulated sample path, we obtain a second one by using the random numbers as follows: If U_i was used in one path, then use $1-U_i$ in the second path.

Example: Suppose we want to price a European call option by simulation and implement the Antithetic Variates method of variance reduction. Using the risk-neutral option pricing techniques, one can say that the price of a European call option is the present value of the expected future payoff at maturity, under the risk-neutral measure.

That is, $c = e^{-rT}\mathbb{E}^*(S_T - X)^+$, where $S_T = S_0 exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma W_T\right)$.

Then, we would follow these steps:

STEP 1. Generate $\{Z_i\}_{i=1}^n \sim iid\ N(0,1)$,

STEP 2. Compute the following two realizations of the call option price:

$$c^+{}_i = e^{-rT}\left(S_0\ e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z_i\right)} - X\right)^+ \text{ and}$$

$$c^{-}{}_i = e^{-rT}\left(S_0\ e^{\left(\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}(-Z_i)\right)} - X\right)^{+}$$

STEP 3. Define $c_i = \frac{c^{+}{}_i + c^{-}{}_i}{2}$, and take $\tilde{c} = \frac{1}{n}\sum_{i=1}^{n} c_i$.

Then $\tilde{c}$ is an approximation of the actual c.

Comment:

Notice that if we generate $\{Z_i\}_{i=1}^{n} \sim iid\ N(0,1)$, then the odd moments of the random sample $\{\{Z_i\}_{i=1}^{n}, \{-Z_i\}_{i=1}^{n}\}$ are equal to zero by construction. If we had generated a random sample $\{Z_i\}_{i=1}^{2n} \sim iid\ N(0,1)$, then the odd moments are not exactly equal to zero.

2.1.2 Control Variates

Suppose we want to estimate $\theta = E(X)$ and that we know that $\mathbb{E}(Y) = \delta$, where Y is a control variate, that is correlated with X, and will be used in the estimation of θ. For any constant γ define $T = X - \gamma(Y - \delta)$. Then T is an unbiased estimator for θ: $ET = \mathbb{E}X - \gamma(\mathbb{E}Y - \delta) = \theta$.

The idea is to choose the free parameter γ in such a way that T has a lower variance than X.

$$Var(T) = \gamma^2 Var(Y) - 2\gamma Cov(X,Y) + Var(X).$$

The variance of T is the lowest if we take $\gamma = \frac{Cov(X,Y)}{Var(Y)}$. With this choice of γ, we get $Var(T) = (1-\rho^2)Var(X)$. It is easy to see that the greater the correlation ρ between X and Y, the more the variance reduction.

In practice, it is not known what $Cov(X,Y)$ and $Var(Y)$ are, so a few test simulations must be run to estimate them and find the optimal γ.

The $Cov(X,Y)$ will be estimated as follows:

$$\widehat{Cov(X,Y)} = \frac{\sum(Y_i - \bar{Y})(X_i - \bar{X})}{n-1}$$

Comment: One may use more than one control variable in simulations as follows:

$$T = X - \sum_{i=1}^{m} \gamma_i \, (Y_i - \delta_i)$$

where $\mathbb{E}Y_i = \delta_i$.

Example

Suppose we want to estimate $\theta = \mathbb{E}\left[e^{(U+W)^2}\right]$, where $U, W \sim \ iid \ U(0,1)$. We can use the usual Monte Carlo simulation technique to estimate the parameter as follows:

1. Generate $U_1, \dots, U_n, W_1, \dots, W_n$ all $iid \ U(0,1)$;
2. Compute $X_1 = e^{(U_1+W_1)^2}, \dots, X_n = e^{(U_n+W_n)^2}$
3. Take $\hat{\theta}_n = \frac{\sum_{j=1}^{n} X_j}{n}$ to estimate θ.

If we want to use the Control Variate technique in the above estimation, the first step is to choose an appropriate control variate Z. Good control variates have two main properties:

1. their expected values are known, and
2. the correlation between the control variate and the original random variable is high.

In this example, there are several candidates. For example, we could choose

1. $Z_1 = U + W, \mathbb{E}[Z_1] = 1$

2. $Z_2 = (U + W)^2, \mathbb{E}[Z_2] = \frac{7}{6}$
3. $Z_3 = e^{U+W}, \mathbb{E}[Z_3] = (e-1)^2$

In all cases we expect the covariance between our control variate Z and $Y = e^{(U+W)^2}$ to be positive. Numerical calculations show that the correlation coefficient between Y and Z_3 is the highest, closely followed by the correlation coefficient between Y and Z_2. We suggest studying the performance of the technique in the following section.

Suggested problems

1. Estimate $\mathbb{E}[Y]$ with and without control variates.
2. Determine the variance reduction in the example above. That means, compare the variance of T_i and the variance of Y, where $T_i = Y + \gamma(Z_i - E[Z_i], i = 1, \dots, 3$.
3. Can you find other control variates which result in a bigger variance reduction?
4. You want to price an Asian call option. What are possible candidates for the control variate?

2.1.3 Conditioning

We now consider a simple, but powerful generalization of the control variate technique. Computing expected values by conditioning is a common technique in probability theory. When we want to estimate $\mathbb{E}(X)$, it is sometimes useful to condition the random variable X on another random variable Y. The following result will enable us to reduce the variance of our estimation:

Theorem

(a) $\mathbb{E}(X) = \mathbb{E}\{\mathbb{E}\,[\,X|Y]\}$

(b) $Cov(X,Y) = \mathbb{E}\{cov(X,Y|Z)\} + cov\{\mathbb{E}[X|Z], \mathbb{E}[Y|Z]\}$

(c) $Var(X) = \mathbb{E}\{Var[X|Y]\} + Var\{\mathbb{E}[X|Y]\}$

Comment: The double expectation, assuming discrete distributions, can be expanded to be written as follows:

$$\mathbb{E}\{\mathbb{E}[X|Y]\} = \sum_i \mathbb{E}(X|Y = y_i)P(Y = y_i) = \sum_i \sum_k x_k P(X = x_k|Y = y_i)P(Y = y_i)$$

The above theorem can be used as follows to reduce the variance of estimation. Suppose we are estimating $\theta = \mathbb{E}(X)$. Property (a) of the above result implies that for any random variable Y, $\mathbb{E}[X|Y]$ is an unbiased estimator for θ. It follows from part (c) of the above Theorem that $Var(X) = \mathbb{E}\{Var[X|Y]\} + Var\{\mathbb{E}[X|Y]\} \geq Var\{\mathbb{E}[X|Y]\}$.

That, is $\mathbb{E}[X|Y]$ and X are unbiased estimators for θ, but $\mathbb{E}[X|Y]$ has a lower variance then X, and therefore is preferable as an estimator.

In general, the search for an appropriate Y requires searching for a random variable Y such that:

(a) The conditional expectation $\mathbb{E}[X|Y]$ is computable,

(b) $Var\{\mathbb{E}[X|Y]\}$ is substantially smaller than $Var(X)$.

Example

We want to estimate $\theta = P(U + Z > 4)$ where $U \sim Exp(1)$ and $Z \sim Exp(0.5)$. Notice that $P(U + Z > 4) = \mathbb{E}[1_{(U+Z>4)}]$. Define $Y = 1_{(U+Z>4)}$. Then, we can use the standard Monte Carlo simulation method to estimate θ as follows:

1. Generate $U_1, \ldots, U_n \sim$ iid $Exp(1), Z_1, \ldots, Z_n \sim$ iid $Exp(0.5)$

2. Define $\mathrm{Y_i} = 1_{(\mathrm{U_i}+\mathrm{Z_i}>4)}$ for $\mathrm{i} = 1, \dots, \mathrm{n}$.
3. Use the following estimate: $\hat{\theta}_\mathrm{n} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{Y_i}$

If we want to use the conditioning approach, set $V = \mathbb{E}(Y|Z)$. Then

$$\mathbb{E}[Y|Z = z] = P(U + Z > 4|Z = z) = P(U > 4 - z) = 1 - F_U(4 - z)$$

where $F_u(\cdot)$ is the cumulative distribution function of U. Therefore

$$1 - F_u(4 - z) = \begin{cases} e^{-(4-z)} \; if \; 0 \le z \le 4, \\ 1 \quad if \; z > 4. \end{cases}$$

Hence

$$V = \mathbb{E}[Y|Z] = \begin{cases} e^{-(4-Z)} \; if \; 0 \le Z \le 4, \\ 1 \; if \; Z > 4. \end{cases}$$

Now the conditional algorithm for estimating $\theta = \mathbb{E}[V]$ is:

1. Generate $Z_1, \dots, Z_n$ all independent, $Exp(0.5)$ distributed,,
2. Set $V_i = \mathbb{E}[Y|Z_i]$ for $i = 1, \dots, n$,
3. Set $\hat{\theta} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{V_i}$.

Problems

1. Implement the two algorithms in the example above. Is there a difference in their convergence rates?
2. How can we use the conditioning technique to price a chooser option? A chooser option is European style option with maturity T_2. At time $T_1 < T_2$ you may choose if the option is a call or a put; the strike price K is fixed at time $t = 0$.

2.1.4 Stratified Sampling

Suppose we are interested in estimating $\theta = \mathbb{E}(X)$ and that X is somehow "dependent" on the value of another random variable Y, which may take a finite set of values with known probabilities: $P(Y = y_k) = p_k$ for $k = 1, \dots, N$.

Then, we can write the expression for θ in terms of distribution of Y:

$$\theta = \mathbb{E}\,X = \sum_{k=1}^{N} \mathbb{E}(X|Y = y_k)P(Y = y_k)$$

If Y is chosen in such a way that the conditional expectations are easy to calculate, then based on the conditional variance results of the last Theorem, we have an estimator with lower variance.

More generally, if a population can be separated into several sub-populations ("strata") according to their variability, then we can take more samples in the sub-populations with high variability than in the sub-populations with low variability to estimate the population mean. This idea is illustrated by the following example.

Example

Suppose we want to estimate the double integral

$$\mu = \int_{-1}^{1}\int_{-1}^{1} \exp\left(\sqrt[3]{x_1} + \sqrt[3]{x_2}\right) dx_1 dx_2$$

using the idea of the stratified sampling. In order to be able to use $U(0,1) -$ distributed random variables to estimate the integral, we apply the following scaling transformation:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

This transformation defines a function $x = g(u)$. The determinant of its Jacobian is given by

$$|Jg(u)| = \begin{vmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_2} & \frac{\partial g_2}{\partial u_2} \end{vmatrix} = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4$$

Thus, we obtain

$$\mu = \int_{-1}^{1}\int_{-1}^{1} \exp\left(\sqrt[3]{x_1} + \sqrt[3]{x_2}\right) dx_1 dx_2 = \int_{0}^{1}\int_{0}^{1} 4\exp\left(\sqrt[3]{2u_1 - 1} + \sqrt[3]{2u_2 - 1}\right) du_1 du_2.$$

To determine the strata, we plot the function $\boldsymbol{h(u_1, u_2) = 4\ \exp\left(\sqrt[3]{2u_1 - 1} + \sqrt[3]{2u_2 - 1}\right)}$ $\boldsymbol{for\ 0 \le u_1 \le 1, 0 \le u_2 \le 1.}$

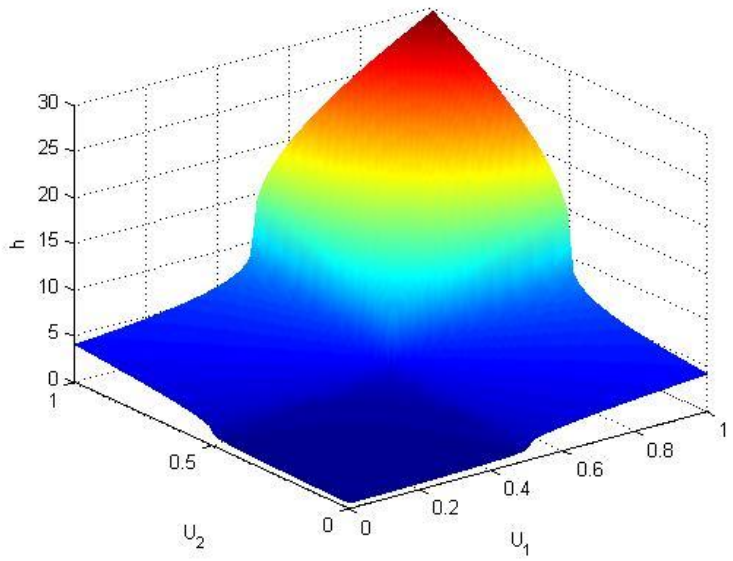

Figure 1: *Plot of Function* $\boldsymbol{h}$.

The graph shows us that the function $h(u_1, u_2)$ is most variable for $0.5 \le u_1, u_2 \le 1$.

For a given sample size, a Monte Carlo estimation in that sub-region would produce a higher standard error than a Monte Carlo estimation in the remaining area. In order to produce a more accurate result, we use stratified sampling.

Define the three sub-regions of $[0,1] \times [0,1]$:

1. $\Omega_1 = \{(u_1, u_2): 0 \leq u_1 \leq .5 \; and \; 0 \leq u_2 \leq .5\}$
2. $\Omega_2 = \{(u_1, u_2): (0 \leq u_1 \leq .5 \; and \; .5 \leq u_2 \leq 1) \; or \; (.5 \leq u_1 \leq 1 \; and \; 0 \leq u_2 \leq .5)\}$
3. $\Omega_3 = \{(u_1, u_2): 0.5 \leq u_1 \leq 1 \; and \; 0.5 \leq u_2 \leq 1\}$

From the graph it is obvious that we want to sample more often in Ω_3 than in Ω_1 and Ω_2. The number of samples in a region should depend on its size (probability weight) and the variability of h in this region relative to the overall (weighted) variability. Define the probability weight $p_i = P(\Omega_i)$. In order to measure the variability in the different regions, let U_{Ω_i} be the restriction of $U = (U_1, U_2)$ on the set Ω_i. Thus, the variability of h restricted to the set Ω_i can be measured by $\sigma_i = \sqrt{Var\left(h\left(U_{\Omega_i}\right)\right)}$. Assuming we want to take n overall samples, the sample size n_i for $\Omega_i, 1 \leq i \leq 3$ is given by

$$n_i = \frac{p_i \times \sigma_i}{p_1 \times \sigma_1 + \; p_2 \times \sigma_2 + \; p_3 \times \sigma_3}$$

Notice that the standard deviations $\sigma_i, 1 \leq i \leq 3$ need to be estimated. Let ${U^k}_{\Omega_i}$ be the k-th realization of the random variable U_{Ω_i}. For example, for $i = 1, \dots, 3$ generate 50 U_{Ω_i} distributed random variables and compute $h\left({U^1}_{\Omega_i}\right), \dots, h\left({U^{50}}_{\Omega_i}\right)$. Then compute the sample standard deviation s_i of this sample. Thus,

$$n_i = \frac{p_i \times s_i}{p_1 \times s_1 + p_2 \times s_2 + p_3 \times s_3}.$$

Estimating the standard deviation results in $s_1 = .281, s_2 = .893, s_3 = 4.478$. Since we will need estimates for $E\left[h\left(U_{\Omega_i}\right)\right], i = 1, \dots, 3$, we compute these as well: $\hat{\mu}_1 = .930$, $\hat{\mu}_2 = 4.119$ and $\hat{\mu}_3 = 18.040$. The probability weights are $p_1 = .25, p_2 = .5$ and $p_3 = .25$.

Assuming that the overall sample size is $n = 1000$, the sample sizes for the three regions are given by $n_1 = 43$, $n_2 = 273$ and $n_3 = 684$. Since our derivation involved a Monte Carlo simulation to estimate the standard deviations, your result may differ slightly. Without stratified sampling we would on average sample 250 times from Ω_1, 500 times from Ω_2 and 250 times from Ω_3.

We can now construct our stratified sampling estimator. It is given by

$$\hat{\theta}_{st} = \frac{\sum_{k=1}^{m_1} h\left(U^k{}_{\Omega_1}\right)}{m_1} p_1 + \frac{\sum_{k=1}^{m_2} h\left(U^k{}_{\Omega_2}\right)}{m_2} p_2 + \frac{\sum_{k=1}^{m_3} h\left(U^k{}_{\Omega_3}\right)}{m_3} p_3$$

By how much does the stratified sampling approach reduce the variance? We can compare the standard deviation of the stratified sampling estimator with the standard deviation of a crude Monte-Carlo estimator. Since the random variables

$$U^1{}_{\Omega_1}, \dots, U^{m_1}{}_{\Omega_1}, \; U^1{}_{\Omega_2}, \dots, U^{m_2}{}_{\Omega_2}, U^1{}_{\Omega_3}, \dots, U^{m_3}{}_{\Omega_3}$$

are all independent, the standard deviation of $\hat{\theta}_{st}$ is given by

$$\sigma_{\hat{\theta}_{st}} = \sqrt{\left(\frac{p_1}{m_1}\right)^2 \sum_{k=1}^{m_1} \sigma_1^2 + \left(\frac{p_2}{m_2}\right)^2 \sum_{k=1}^{m_2} \sigma_2^2 + \left(\frac{p_3}{m_3}\right)^2 \sum_{k=1}^{m_3} \sigma_3^2}$$

Replacing the standard deviations σ_i by the sample standard deviations s_i, we obtain

$$s_{\widehat{\theta}_{st}} = \sqrt{\left(\frac{p_1}{m_1}\right)^2 \sum_{k=1}^{m_1} s_1^2 + \left(\frac{p_2}{m_2}\right)^2 \sum_{k=1}^{m_2} s_2^2 + \left(\frac{p_3}{m_3}\right)^2 \sum_{k=1}^{m_3} s_3^2}$$

If we use the values calculated above, we have

$$s_{\widehat{\theta}_{st}} = 0.517$$

The standard deviation of the crude Monte Carlo estimator is given by

$$\sigma_{MC} = \frac{\sigma}{\sqrt{n}},$$

where $\sigma$is the population standard deviation of $h(U)$. We could now run another Monte-Carlo Simulation to find an estimate for σ. This is legit, but it is not computationally efficient. We can also treat $h(U)$ as a mixture of the distributions of the random variables $h\left(U_{\Omega_1}\right), h\left(U_{\Omega_2}\right)$ and $h\left(U_{\Omega_3}\right)$. This means that the probability density function $f_{h(\boldsymbol{U})}$ of $h(U)$ is given by

$$f_{h(\boldsymbol{U})}(u_1, u_2) = \sum_{i=1}^{3} f_{h\left(\boldsymbol{U}_{\Omega_i}\right)}(u_1, u_2) p_i.$$

Then

$\hat{\mu} = p_1\ \hat{\mu}_1 + p_2\ \hat{\mu}_2 + p_3\ \hat{\mu}_3 = 6.802$

and

$s = \sqrt{(s_1^2 + \hat{\mu}_1^2)p_1 + (s_2^2 + \hat{\mu}_2^2)p_1 + (s_3^2 + \hat{\mu}_3^2)p_1 - \hat{\mu}^2} = 7.016$

Thus the standard deviation of the crude Monte Carlo estimator is

$$\sigma_{MC} = \frac{7.016}{\sqrt{1000}} = .222$$

This means, if we wanted to achieve a standard error of the same size as the stratified sampling estimator for the crude Monte Carlo simulation, we would have to sample

$$n = \left(\frac{s}{s_{\widehat{\theta}_{st}}}\right)^2 = 18{,}417$$

realizations of the random variable U.

Back to our original task, the stratified sampling estimator returns 6.928 as the value of the integral.

Suggested Exercise

1. Implement the previous example to analyze the stratified sampling method.

2.1.6 Importance Sampling

The idea here is to distort the underlying probability measure to estimate the parameter under consideration, with less variance. This is achieved by using Girsanov's Theorem to "change" the underlying measure in such a way that is computationally less expensive to estimate the unknown parameter. That is, we will attempt to reduce the variance by changing the probability distribution from which the values are generated. Suppose we are interested in estimating

$$\theta = \mathbb{E}\, g(X) = \int_{-\infty}^{\infty} g(x) f(x) dx,$$

where $f(\cdot)$ is the probability density function (pdf) of X.

The idea is to choose a function $t(.)$ so that $\frac{g(x)f(x)}{t(x)}$ is as smooth as possible (ideally, we would like it to be a constant, but it is impossible in reality).

Assume $Support(f) \in Support\ (t)$, where $t(\cdot)$ is a density function. Then,

$\theta = \int_{-\infty}^{\infty} \frac{g(x)f(x)}{t(x)} t(x)dx = \mathbb{E}_t\left(\frac{g(Y)f(Y)}{t(Y)}\right)$, where Y is a random variable with density function $t(\cdot)$.

The algorithm for estimating θ is as follows:

1. Generate $Y_1, Y_2, \dots, Y_n$ from the density $t(\cdot)$,
2. Compute $\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{g(Y_i)f(Y_i)}{t(Y_i)}$ and use it as an estimator for θ.

Then, we can claim that:

(a) $\hat{\theta}_n$ is an unbiased estimator for θ: $\mathbb{E}\hat{\theta}_n = \theta$.

(b) $Var\hat{\theta}_n = \frac{1}{n}\left(\mathbb{E}_t\left(\frac{g(Y)f(Y)}{t(Y)}\right)^2 - \theta^2\right)$.

Exercise:

Relate this algorithm to the Acceptance-Rejection algorithm.

Consider the following examples.

Example 1

Assume that the stock price at time T has $N(100,1)$ distribution. You want to price a digital call option, struck at 110, by Monte Carlo simulation. Let the initial stock price be $S_0 = 100$.

The main problem here is that, if we simulate paths of the stock price, almost all paths will end up out of the money for the option we are trying to price. The probability of a ten standard deviation move of the stock price is very small. Therefore, the use of a Monte-Carlo Simulation will result in a call price of zero. We can apply importance sampling to this problem.

Our goal is to evaluate

$$\int_{-\infty}^{\infty} f(S_0 + z)\phi(z)dz$$

where f is the payoff function of the digital call

$$f(x) = \begin{cases} 0, & x < 110 \\ 1, & x \geq 110 \end{cases}$$

and $\phi(z)$ is the probability density function of a standard normal random variable. Let ψ be the probability density function of a normal random variable with mean 10 and standard deviation 1. Then we can write the above integral as

$$\int_{-\infty}^{\infty} f(S_0 + z)\frac{\phi(z)}{\psi(z)}\psi(z)dz = \mathbb{E}_{\psi}\left[f(S_0 + Z)\frac{\phi(Z)}{\psi(Z)}\right]$$

If we run a Monte Carlo simulation to estimate the expected value, then we draw realizations from a normal distributed random variable with mean 10 and standard deviation 1 and evaluate $f(S_0 + z)\frac{\phi(z)}{\psi(z)} = f(S_0 + z)e^{\frac{10^2}{2} - 10z}$. Notice that we sample paths where the payoff of the digital call jumps. Using importance sampling we actually have paths which end up in the money, whereas a crude Monte Carlo simulation does not produce any paths which end up in the money. An implementation of this problem shows that a crude Monte Carlo Simulation results in a price of zero, whereas the Monte Carlo simulation using the importance sampling procedure gives a price of 7.6485e-024. This is still very small, but it is more accurate.

Although very stylized, this example demonstrates the power of importance sampling.

It is now a very natural question to ask which choice of $t(\cdot)$ reduces the variance of the importance sampling estimator

$$\hat{\theta}_{IS} = \frac{1}{n}\sum_{i=1}^{n}\frac{g(X_i)f(X_i)}{t(X_i)}$$

Consider the variance of the importance sampling estimator given by

$$Var(\hat{\theta}_{IS}) = \frac{1}{n}\mathbb{E}\left[\frac{g(X)f(X)}{t(X)} - \mathbb{E}\left[\frac{g(X)f(X)}{t(X)}\right]\right]^2$$

Looking at the last expression and assuming that g is non-negative, we see that we can find an importance sampling estimator with zero variance if we choose $t(x) = \alpha g(x)f(x)$, where α is a normalizing constant, such that, t is a probability density function. Unfortunately, the normalizing constant α is nothing else than the expected value we are trying to estimate.
But, it illustrates that it is desirable to choose $t(\cdot)$ as close as possible to being proportional to $|g(\cdot)|f(\cdot)$.
Furthermore, a good importance sampling function $t(x)$ should have the following additional properties:

I. $t(x) \geq 0\ \forall x$ for which $g(x)f(x) \neq 0$.

II. It should be easy to simulate values from $t(x)$.

III. It should be easy to compute the density $t(x)$ for any value x which can be realized.

General examples of the use of the Importance Sampling method are also given by the **Esscher Transform**, in which the function $t(\cdot)$ is given by

$$t(x) = \frac{e^{ux}f(x)}{E_f(e^{uX})}$$

Examples:

1. **Exponential:** $f(x) = \alpha e^{\alpha x}$. Then, $t(x) = (\alpha - u)e^{(\alpha-u)x}$, which is exponential provided that $\alpha > u$.
2. **Bernoulli:** $f(x) = p^x(1-p)^{1-x}$. for $x = 0, 1$. Then, $t(x) = \left(\frac{pe^u}{pe^u+1-p}\right)^x \left(\frac{1-p}{pe^u+1-p}\right)^{1-x}$

Example 2 (By Wei Cai. Used by permission)

The Importance Sampling idea is well demonstrated by using one of the classical integral calculations: $\int_0^1 \sqrt{1-x^2}dx = \frac{\pi}{4}$.

To illustrate the use of the class of functions in importance sampling, consider the following family of density functions:

$$t(x) = \begin{cases} \dfrac{1-ax^2}{1-\frac{a}{3}} & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

It can be shown that with this function $t(.)$, the variance of $\frac{g(Y)f(Y)}{t(Y)}$ will be given by:

$$Var\left(\frac{g(Y)f(Y)}{t(Y)}\right) = \left(1-\frac{a}{3}\right)\left(\frac{1}{a} - \frac{1-a}{a^{3/2}}\, tanh^{-1}(a^{-\frac{1}{2}})\right) - \left(\frac{\pi}{4}\right)^2.$$

Since our goal is to find $t(.)$ that makes the variance of our estimate as small as possible, we will choose the constant a in such a way that the above variance is the smallest. It turns out that $a = 0.74$ is the number (numerical calculations) that minimizes the variance. Notice, that $a = 0$ is the original case in which we just use the uniform distribution.

Suggested exercises: Try a few other possibilities of $g(x)$:

(1) $g(x) = \frac{(1-\alpha x)}{(1-\frac{\alpha}{2})}$. What is the optimal α to minimize the computational cost here?

(2) $g(x) = \frac{(1-\alpha x^2)}{(1-\frac{\alpha}{3})}$. What is the optimal α to minimize the computational cost here? (Try $\alpha = 0, 0.35, 0.48, 0.65, 0.74, 0.86, 1$).

Comments

1. To gain some more intuitive insight in important sampling, refer to the paper on "Importance sampling: an illustrative introduction" by P H Borcherds.
2. For an application of importance sampling to pricing barrier options, see *"Using Monte Carlo Simulation and Importance Sampling to Rapidly Obtain Jump-Diffusion Prices of Continuous Barrier Options"*, by Mark S. Joshi and Terrence Leung.

Simulation Efficiency

As was seen earlier, we can build confidence intervals for parameter estimators as follows: $\bar{X}_n \pm \frac{z_{1-\alpha/2}}{\sqrt{n}} S_n$. One way to measure the efficiency of our estimator $\bar{X}_n$, is to consider the length of the confidence interval: $\frac{2z_{1-\alpha/2}}{\sqrt{n}} S_n = 2z_{1-\alpha/2}\sqrt{\frac{S_n}{n}}$.

We would like to have a relatively small interval (less error), but it is not always possible to achieve that goal, with a "reasonable computational cost". Of course, by increasing the number of simulations, one can decrease the error term and achieve any accuracy. Therefore, the question is: how to estimate the quality of the estimation, by taking into account the computational cost, and the accuracy of the approximation.

One measure of simulation efficiency would be consider the combination of the two: computational cost and accuracy.

Let cc be the computational cost to simulate one realization of a random variable under consideration, n be the number of simulations, and let w be the e be the error term, measured by the width of the confidence interval. We have:

$$n = \left(\frac{2S_n z_{1-\alpha/2}}{w}\right)^2$$

Then, the total computational "cost" of simulations –TCC - will be

$$TCC = n * cc = cc * \left(\frac{2S_n z_{1-\alpha/2}}{w}\right)^2$$

Thus, two methods – i and j - will be compared in terms of their efficiency as follows:

If:

$$TCC_i = cc_i * \left(\frac{2S_n^i z_{1-\alpha/2}}{w}\right)^2 < TCC_j = cc_j * \left(\frac{2S_n^j z_{1-\alpha/2}}{w}\right)^2$$

then, the method i is more efficient than j.

In other words, if $cc_i * \left(S_n^i\right)^2 < cc_j * \left(S_n^j\right)^2$ then method i is more efficient than j.

Notice, that we can replace $\boldsymbol{cc}$ by t (computing time) and measure and compare the efficiency of simulation methods.

2.2 Quasi Monte Carlo Simulation

The Monte Carlo simulation method is based on the idea that the random variates we generate are truly (or close to) random. That is, the uniformly distributed random numbers we obtain are truly random, so that a sample of uniforms will be "evenly" spread on the [0.1] interval. However, the random numbers produced by any of the algorithms we considered are not random and when used to estimate integrals or expectations of random variables, may produce errors in certain situations.

When generating random numbers, the idea is to generate a sequence of numbers that are uniformly spread over the [0,1] interval in a one-dimensional case (unit square, cube, etc. in multi-dimensional cases). If we create a large number of uniform variates, then one may argue that the distribution of those numbers is reasonably close to the one of the uniformly-distributed random variable. However, in some instances, one needs to control the computational time and control the program by terminating it when certain accuracy of convergence is achieved. Then, there is a need for a sequence of numbers that are uniformly distributed over the [0,1] interval, even if we have not many of them. That is, no matter how many variates we take, they are uniformly (or close to uniformly) distributed over the unit interval. That is, we would like to use an algorithm, in which the next generated number is placed on the unit interval in such a way that the distribution of the sequence is as close to uniform as possible.

This idea may be made more precise by defining the discrepancy of a sequence of numbers. Assume we want to generate a sequence of N random vectors $(X_1, X_2, \dots, X_N)$ on the n-dimensional hypercube

$$I^n = [0,1] \times [0,1] \times \ \dots \times [0,1] \subset \mathbb{R}^n.$$

If the vectors are uniformly distributed on I^n, then by intuition, the number of variates in any subset A of I^n should be roughly proportional to the volume of A, relative to the volume of I^n.

Define the set of all rectangular subsets of I^n:

$$A = \{[a_1, b_1] \times [a_2, b_2] \times \ \dots \times [a_n, b_n], for\ any\ 0 \le a_i \le b_i \le 1, i = 1,2, \dots, n\}$$

Define the discrepancy $D(x, A)$ of a set of points $x = (x_1, x_2, \dots, x_m)$ with respect to set A as follows:

$$D(x, A) = \sup_{B \in A} \left| \frac{Number\ of\ x_i\ in\ B}{m} - Volume\ of\ B \right|$$

Here $x_i \in \mathbb{R}^n$. A sequence would be "close-to-uniform", if the discrepancy of the sequence is small. The lower the discrepancy, the closer the sequence is to uniformity.

When computing integrals, it is natural to use sequences of low-discrepancy numbers as opposed to the pseudo-random sequences. Some theoretical results suggest that low-discrepancy sequences may perform better than pseudorandom sequences.

The use of the low-discrepancy sequences would be justified if we can see (or prove) their faster convergence or lower variance, than the standard Monte Carlo method. It is known that the convergence rate for the Monte Carlo method with pseudo-random numbers used is in the order of $n^{-1/2}$. Is has been shown that the use of certain low-discrepancy sequences result in better convergence rate --- $\frac{\ln(n)^N}{n}$, where N is the dimension of the problem. As it is easy to see, this is better rate of convergence than $n^{-(1-\varepsilon)}$ for any small $\varepsilon > 0$.

Definition: A sequence $x_1, x_2, \dots, x_m, \dots \in \mathbb{R}^n$ is said to be a Low Discrepancy Sequence (LDS), if there exists a finite constant K_m so that the Discrepancy of the Sequence satisfies

$$D_n \leq K_m \frac{(\ln n)^m}{n} \text{ for any } n.$$

Using Quasi Monte Carlo methods, we can approximate the integrals as follows:

Suppose $h_1, h_2, \dots, h_k, \dots$ are from a Low Discrepancy Sequence. Then, we set

$$\int_0^1 f(x)\, dx \approx \frac{1}{k} \sum_{i=1}^{k} f(h_i)$$

We will implement one of the popular LDS - the Halton's low discrepancy sequence here. We refer interested readers to the book by P. Glasserman, for more details about other sequences (such as Sobol, Feure, Van der Corput, etc.) and for proofs of convergence results.

Comment: To compute multi-dimensional integrals using the LDS we follow the same technique:

$$\int_0^1 \int_0^1 f(x,y) dx\, dy \approx \frac{f(H_1^1, H_2^1) + f(H_1^2, H_2^2) + \cdots + f(H_1^n, H_2^n)}{n}$$

where $H_1^k \; and H_2^k$ are the k-th numbers of Halton sequences with different bases.

2.5.1 Halton's Low-Discrepancy Sequences

Define $H(k,m)$ to be the Halton's $k-th$ number in the sequence with base m. The algorithm for generation of $H(k,m)$ is as follows:

STEP 1: For any $k \geq 1$ integer, write $k = a_0 + a_1 m + \cdots + a_r m^r$, where $a_i \in \{0,1, \dots, m-1\}$, and r is so that $m^r \leq k < m^{r+1}$.

STEP 2: Define $H(k,m) = \frac{a_0}{m} + \frac{a_1}{m^2} + \cdots + \frac{a_r}{m^{r+1}}$ -- the $k\ th$ member of the Halton's sequence with base m.

Comment: m is chosen as a prime number.

For estimation of multidimensional integrals, sometimes one might need to generate multidimensional Halton sequences. To generate n −dimensional Halton sequences, we follow the following steps:

STEP 1: Choose the first n prime numbers: $m_1, m_2, \ldots, m_n$.

STEP 2: For any $k \geq 1$ integer, and m_i, using the method above, generate the vector of Halton's numbers: $X_k = (H(k,m_1), H(k,m_2), \ldots, H(k,m_n))$

The following Matlab function generates a one-dimensional Halton sequence (the code is adopted from Brandimarte's "Numerical methods in finance and economics: a MATLAB-based introduction")":

```
function Seq = GetHalton(HowMany, Base)
Seq = zeros(HowMany,1);
NumBits = 1+ceil(log(HowMany)/log(Base));
VetBase = Base.^(-(1:NumBits));
WorkVet = zeros(1,NumBits);
for i=1:HowMany
  j=1;
  ok = 0;
  while ok == 0
    WorkVet(j) = WorkVet(j)+1;
    if WorkVet(j) < Base
      ok = 1;
    else
      WorkVet(j) = 0;
      j = j+1;
    end
  end
```

```
  Seq(i) = dot(WorkVet,VetBase);
end
```

Comment:

If we generate a Halton sequence for base $m = 2$, then we see that the first few numbers are

$$\left(\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \dots\right)$$

Plotting these numbers shows that they evenly fill out the interval $(0,1)$. But, notice that these numbers would be a bad choice for a symmetric random walk because every step to the right would be followed by a step to the left.

2.6 Exercises

1. Generate a series (X_i, Y_i) for $i = 1, \dots, n$ Bivariate-Normally distributed random vectors, with mean of $(0,0)$ and the variance – covariance matrix of $\begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix}$. Compute the following:

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

   where $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, and $n = 1000$.

2. Evaluate the following expected value by using Monte Carlo simulation:

$$E(X^3 + \sin(Y) + X^2Y)$$

where X and Y have $N(0,1)$ distribution and a covariance of 0.65.

3. (a) Estimate the following expected values:

$$E({W_5}^2 + \sin(W_5),$$

$$E\left(e^{\frac{t}{2}}\cos(W_t)\right) for\ t = 0.5, 4, 8.$$

where W_t is a Wiener Process.

(b) How are the values of the last three integrals (for the cases $t = 0.5, 4, 8$) related?

(c) Now use a variance reduction technique (whichever you want) to compute the expected value in part (a). Do you see any improvements?

4. Let S_t be a Geometric Brownian Motion process: .

$$S_t = S_0\ e^{\left(\sigma W_t + \left(r - \frac{\sigma^2}{2}\right)t\right)}$$

where $r = 0.04, \sigma = 0.2, S_0 = 88$, and W_t is a Standard Brownian Motion process (Wiener process).

(a) Estimate the price c of a European Call option on the stock with $T = 10, X = 100$ *by using Monte Carlo simulation*.

(b) Now use the variance reduction techniques to compute the price in part (a) again. Did the accuracy improve? You may compute the exact value of the option c by the Black-Scholes formula, by using Excel. Now estimate c by crude Monte Carlo simulation, and then by using different variance reduction techniques to see if there is an improvement in convergence.

5. Simulate 4 paths of $S_t for\ 0 \leq t \leq 10$ (defined in the Problem 4) by dividing up the interval [0, 10] into 1,000 equal parts. Then, for each integer number n from 1 to 10, compute ES_n. Plot all of this in one graph.

6. Consider a method of computing the number π: $\int_0^1 \sqrt{1-x^2}dx = \frac{\pi}{4}$

(a) This can be done by simple numerical integration using, say Euler's (or any other) scheme. Compute it numerically. Now compute it by Monte Carlo simulation.

(b) Now try the Importance Sampling method to improve the estimate of π of part (a).

7. (*Popular interview question on stochastic calculus*) What can you say about the following integrals?

$$\int_0^T W_t dt, \text{ and } \int_0^T W_t dW_t$$

Use the simulation techniques learned to simulate the integrals and empirically find their distributions.

8. (*Popular Interview question*) We (two of us) are to play on a table in the next room. We each have bags of an infinite number of identical quarters (American 25-cents). We will take it in turns to put a quarter on the table. Quarters may not overlap on the table. When there is no

more room left on the table to put another quarter, then the winner is the last person to put a quarter on the table. Does there exist a winning strategy for the starter of the game? What about the second player? Assume the table is of rectangular shape. What if the table was of a circular shape?

9. Evaluate the following expected values and probabilities:

$$E\left(X_2^{\frac{1}{3}}\right), \qquad E(Y_3), \qquad E\big(X_2 Y_2\, \mathbf{1}(X_2 > 1)\big), \qquad P(Y_2 > 5).$$

where the Ito's processes X and Y evolve according to the following SDEs:

$$dX_t = \left(\frac{1}{5} - \frac{1}{2}X_t\right)dt + \frac{2}{3}dW_t,\ X_0 = 1,$$

$$dY_t = \left(\left(\frac{2}{1+t}\right)Y_t + \frac{1+t^3}{3}\right)dt + \frac{1+t^3}{3}dZ_t,\ Y_0 = \frac{3}{4}$$

$W\ and\ Z$ are independent Wiener processes, and $\mathbf{1}(X_2 > 1) = 1\ \ if\ X_2 > 1, and\ 0\ if\ X_2 \leq 1.$

10. Estimate the following expected values and compare:

$$E(1 + X_3)^{1/3},\ and\ \ E(1 + Y_3)^{1/3}$$

where

$$dX_t = \frac{1}{4}X_t dt + \frac{1}{3}X_t dW_t - \frac{3}{4}X_t dZ_t,\ \ X_0 = 1$$

$$Y_t = e^{-0.08t + \frac{1}{3}W_t + -\frac{3}{4}Z_t}$$

and $W\ and\ Z$ are independent Wiener processes.

11. Compute the probability (by simulation) that a European *put* option will expire in the money. Use these parameters: $S_0 = 20$, $X = 20$, $\sigma = 0.25$, $r = 0.04$ and $T = 0.5$ years.

12. Compare the pseudorandom sample with the quasi MC sample of $Uniform[0,1]x[0,1]$:

    a) Generate 100 vectors of $Uniform$ $[0,1]x[0,1]$ by using MATLAB (or the software you are using) random number generator.

    b) Generate 100 points of the 2-dimentional Halton sequences, using bases 2 and 7.

    c) Generate 100 points of the 2-dimentional Halton sequences, using bases 2 and 4. (4 is non-prime!).

    d) Draw all on separate graphs and see if there are differences in the three (visual test only).

    e) Use 2-dimensional Halton sequences and compute the following integral:

    (use $N = 10{,}000$. Try different pairs of bases: $(2,4), (2,7), (5,7)$.)

$$\int_0^1 \int_0^1 e^{-xy} \left(\sin(6\pi x) + cos^{\frac{1}{3}} (2\pi y) \right) dxdy$$