Estimating Number of Upcrossing of Correlated Brownian Motions

Saran Ahuja, Sarita Bunsupha
Stanford University

June 1, 2011

Abstract

In this paper, we explain how to implement Interacting Particle System (IPS) method and Important Sampling (IS) method to estimate a number of up crossing of correlated Brownian motions when the barrier level is high relative to the variance. We use the case of one-dimensional problem to illustrate how the studied methods work. Through numerical result, we compare the performance of these two methods.

1 Introduction

Consider a Barrier option on N assets whose prices are denoted by

$$S_t^{(1)}, S_t^{(2)}, \dots, S_t^{(N)}$$

The probability of crossing a barrier must be computed to price such option. However, when the assets are correlated, the joint distribution of running maxima cannot be computed explicitly. This problem requires an understanding of the law of a Brownian path, and it has been known to be extremely difficult to analyze even when $N = 2$. Some progresses has been made in this direction. In [1], the authors computed explicitly the covariance of running maxima of two correlated Brownian motion. However, for our purpose, we need to know more than the covariance. We need to compute the value

$$\mathbb{P}\{\sum_i \mathbf{1}_{M_T^{(i)} > b} = k\}$$

where

$$M_T^{(i)} = \max_{0 \le t \le T} S_t^{(i)}$$

This type of calculation also finds application in credit portfolio where we need to compute the number of defaults of firms whose dynamic are correlated. In that case, the problem is to find the number of down crossing. However, all the methods described here will apply by reflection. As explained earlier, this problem cannot be solved explicitly even with the simplest assumption that the dynamics are just Brownian motion. The correlation makes things even more difficult. As a result, the best we can do is to apply numerical simulation to estimate the probability.

Our interest is to consider the case when the upper barrier level is high relative to the variance of the Brownian motion. This assumption turns our problem into the problem of estimating a rare event. Various methods can be used to tackle the problem. Our focus here is to use two well known methods - Interacting Particle Systems and Importance Sampling. Although different in detail, these methods share the same goal, which is to generate more paths that yield non-zero value to the estimation. The Interacting Particle System (IPS) method employs the resampling at each step that the new sample paths are generated. By weighting more on paths that are more likely to contribute to non-zero values, more "good" paths will be kept, while relatively "bad" paths will be thrown away. In this method, the dynamics of the path is, nevertheless, unchanged. In contrast, the Importance Sampling method exploits the change of measure to achieve this goal. By applying Girsanov Theorem. we can change the measure to concentrate more on the set of "good" paths. Generating samples from this new measure will yield more path that are "importance", hence the name of the method.

The paper is organized as follows; Section 2 described briefly the theoretical justification of both methods. Section 3 presents our problems of estimating number of up crossing of correlated Brownian motion and describe how to apply these two methods step-by-step. Section 4 presents numerical results for both one-dimensional and multi-dimensional problem. Section 5 is a conclusion of our study.

2 Methods of Estimation

In this section, we describe briefly the general idea of Interacting Particle System and Importance Sampling method.

2.1 Interacting Particle Systems

As explained in Section 1, the IPS algorithm use the resampling idea to generate more paths that contribute to our estimation. As a result, we get more accurate estimator for a rare event of interest. Since the Gibb resampling originates from a problem of simulating a Markov Chain, we will describe the method in that context. Also, as we shall see in the next section, our problems at hands is, in fact, in a Markov Chain setting since we will need to discretize the time step in order to do resampling. This section mainly follows [3], so please refer to [3] for any clarification.

Suppose we have a Markov Chain $X_k \in \mathbb{R}^d$ with transition probability $p(x, dy)$. That is, $\mathbb{P}[X_{k+1} \in A | X_k = x] = p(x, A) = \int_A p(x, dy)$. Let Y_k denote the path of X_i up to time k, that is,

$$Y_k = (X_1, X_2, \ldots, X_k)$$

In practice, X_k would represent all information we need at time k and Y_k represents the history of all that information up to time k. To resample, we will need to have a weighted function. Let $G_k : \mathbb{R}^{dk} \to [0, \infty)$ be a potential function, which we need to select appropriately. Suppose we would like to estimate $\mathbb{E}[f(Y_n)]$, then the main idea of using IPS method can be described as follows; suppose we have generated M samples of paths up to time k. Denote them by $Y'_{k,1}, Y'_{k,2}, \ldots, Y'_{k,M}$. Before we proceed to extend the paths to time $k+1$, we apply the resampling. That is, we select from these M paths independently to get M paths $Y_{k,1}, \ldots, Y_{k,M}$. But instead of resample uniformly, we put the weight on each $Y'_{k,m}$ according to the Boltzmann-Gibbs measure, that is.

$$\mathbb{P}[Y = Y'_{k,m}] = \sum_{j=1}^{M} \frac{G_k(Y'_{k,j})}{\sum_{i=1}^{M} G_k(Y'_{k,i})}$$

By selecting $G_k(\cdot)$ appropriately so that it gives more weight to $Y'_{k,j}$ that is "good", our resulting M sample paths at time k will have more "good" paths than those without resampling. After we proceed to time n and get M sample paths $Y_{n,1}, \ldots, Y_{n,M}$, our estimator is then given by

$$\gamma_n^M(f) = \eta_n^M\left(\frac{f}{\prod_{p=1}^{n-1} G_p}\right) \prod_{p=1}^{n-1} \eta_p^M(G_p) \tag{1}$$

where for any g_p on $\mathbb{R}^{dp}$, η_p^M is given by

$$\eta_p^M(g_p) = \frac{1}{M} \sum_{j=1}^{M} g_p(Y_{p,j})$$

It can be shown (see [2]) that under some technical condition on G_p and f,

$$\mathbb{E}[\gamma_n^M(f)] = \mathbb{E}[f(Y_n)]$$

That is, γ_n^M is an unbiased estimator.

2.2 Importance Sampling

Consider the simplest case. Suppose we would like to estimate $\mathbb{E}[f(X)]$ for some real-valued random variable X with density $g(x)$. Then suppose Y has density h such that $h(x) \neq 0$ whenever $f(x)g(x) \neq 0$, then it follows that

$$\mathbb{E}[f(X)] = \int f(x)g(x)dx = \int \frac{f(x)g(x)}{h(x)} h(x)dx = \mathbb{E}[\frac{f(Y)g(Y)}{h(Y)}]$$

Therefore, $\mathbb{E}[f(X)]$ can be estimated by

$$\frac{1}{M}\sum_{k=1}^{M}\frac{f(Y_k)g(Y_k)}{h(Y_k)}$$

where $Y_1,\ldots,Y_M$ are iid with density $h(y)$. In other words, we have change our measure from $f(x)$ to $h(x)$. Thus, for this method to be useful, we need to choose h to concentrated around set of samples that is good for our estimation.

Now, let's consider the same idea in the context of continuous sample path. Suppose we would like to estimate $\mathbb{E}[f(\{\mathbf{B}_s\})]$, where $\mathbf{B}_s = (B_{1,s},, B_{d,s})$ denote the standard multi-dimensional Brownian motion and $\{\mathbf{B}_s\}$ denote its path from 0 to some fixed time T. Then by applying Girsanov change of measure, we have, for $\mu \in \mathbb{R}^d$,

$$\mathbb{E}[f(\{\mathbf{B}_s\})] = \mathbb{E}^*[f(\{\mathbf{B}_s + \mu s\})] = \mathbb{E}^*[f(\{\mathbf{B}_s + \mu s\})] = \mathbb{E}[f(\{\mathbf{B}_s + \mu s\})\mathbf{Z}_T]$$

where

$$Z_T = e^{-\mu\cdot\mathbf{B}_T-\frac{1}{2}\mu^T\mu T^2}$$

Thus, by choosing $\mu = (\mu_1,,\mu_d)$ appropriately so that $f(\{\mathbf{B}_s+\mu s\})$ yields more non-trivial value, we get a more accurate estimator.

3 Estimating Number of Up Crossing

3.1 One dimensional problem

Since the exact value is only known in the case of one dimensional, we will first apply two methods described in Section 2 to this case. Denote a Brownian motion by B_t, the running maxima by M_t, and an upper barrier by $b >> 1$. Since in this case, the number of up crossing is either zero or one, the problem is equivalently to estimating

$$\mathbb{P}[M_T > b]$$

By reflection principle, this value can be computed explicitly and is given by

$$\mathbb{P}[M_T > b] = 2\mathbb{P}[B_T > b] = 2(1-\Phi(\frac{b}{\sqrt{T}}))$$

Now we would like to apply the two methods described in Section 2 to estimate $\mathbb{P}[M_T > b]$. This can be done as explained below

3.1.1 IPS method

To apply IPS method to our problem, we need to specify the information we would like to collect, i.e. the Markov Chain X_k, and the potential function $G_k(\cdot)$ (Please refer to Section 2 for notation.) First of all, since we can resample only for a finite number of time, we will need to discretize the time. Choose $n, \Delta t$ so that $n\Delta t = T$ and let $t_k = k\Delta t$. Since we only need to know the current values and running maxima of a Brownian motion, we will let $X_k = (B_{t_k}, M_{t_k})$ and, thus, $Y_k = (B_{t_1}, M_{t_1}, \dots, B_{t_k}, M_{t_k})$. For a weighted function, we will let

$$G_k(Y_k) = \exp(\alpha(M_{t_k} - M_{t_{k-1}}))$$

The reason for this potential function is that we would like to collect more paths that are running toward the barrier, so that there are more potentially "good" paths in the next step. Let M denote the number of Monte Carlo samples. Now, with everything specified, let's summarize step-by-step how to estimate $\mathbb{P}[M_T > b]$. Let M denote the

- *Initialization*: We first initialize all M paths. That is, we set $Y_{0,i} = (0, 0), i = 1, \dots, M$.

- *Resampling*: Suppose we have generate M sample paths up to time t_p. That is, we have $Y'_{p,i}, i = 1, \dots, M$. We first calculate

  $$\eta_p^M(G_p) = \frac{1}{M}\sum_{i=1}^{M} G_p(Y'_{p,i}) = \frac{1}{M}\sum_{i=1}^{M} e^{\alpha(M'_{t_k,i} - M'_{t_{k-1},i})}, i = 1, \dots, M$$

  and the resampling weight

  $$q_i = \frac{G_p(Y'_{p,i})}{M\eta_p^M(G_p)}, \qquad i = 1, \dots, M$$

  Next, we assign probability q_i to $Y'_{i,p}$, then resample (with replacement) M paths from $\{Y'_{1,p}, \dots, Y'_{M,p}\}$. Denote the resulting paths by $\{Y_{1,p}, \dots, Y_{M,p}\}$. We store both η_p^M and our paths and move on to the next step. Note that we need η_p^M in the final step when we compute the estimate.

- *Mutation*: We now simply generate the Brownian motion at the next time step t_{p+1}.

- *Estimate*: After we have $Y_{i,p}$ for $i = 1, \dots, M, p = 1, \dots, n$, then the Monte Carlo estimate under IPS algorithm is given by (1). Note that we have stored $\eta_p^M(G_p)$ along the process and

  $$\prod_{p=1}^{n-1} G_p(Y_p) = e^{\alpha M_{t_n}}$$

  Thus, the estimate can be written explicitly as follows,

  $$\hat{\mathbb{P}}[M_T > b] = \left(\frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{M_{i,t_n} > b} e^{-\alpha M_{i,t_n}}\right) \prod_{p=1}^{n-1} \eta_p^M(G_p)$$

3.1.2 IS method

Based on the Girsanov Theorem as explained in Section 2, it is straightforward to implement IS method in the problem. We simply generate M sample paths of Brownian motion by usual discretization. Denote the samples by $(B_{i,t_0}, B_{i,t_1}, \ldots, B_{i,t_n})$, then the estimate is given by

$$\hat{p} = \frac{1}{M}\left(\sum_{i=1}^{M} \mathbf{1}_{\{\max_s B_{i,s}+\mu s > b\}} e^{-\mu B_{i,T} - \frac{\mu^2}{2}T}\right)$$

The only part that we need to consider is choosing μ. However, in this case, we can simply pick μ to be any positive value such that $\mu T > b$ so that $\mathbb{E}[W_T] = \mu T > b$. As a result, most of the paths will cross the barrier.

3.2 Multi-dimensional problem

Now suppose that we have d-dimensional Brownian motion, which we will denote by $\mathbf{B}_t = (B_t^1, \ldots, B_t^d)$. Assume further that there are correlated, that is,

$$\mathbb{E}[B_t^i B_s^j] = \rho_{ij} s \wedge t$$

Let $\Sigma = \{\rho_{ij}\}$ be the covariance matrix and C be the matrix such that $CC^T = \Sigma$, i.e. C is the Cholesky decomposition of Σ. Then $\mathbf{B}_t$ can be written as

$$\mathbf{B}_t = C\tilde{\mathbf{B}}_t \tag{2}$$

where $\tilde{\mathbf{B}}_t$ is the standard $d-$dimensional Brownian motion, i.e. $\tilde{B}_{i,t}$ are independent Brownian motions. Thus, to generate a path of $\mathbf{B}_t$, we simply generate a path of $\tilde{\mathbf{B}}_t$ and use (2). Now, we let

$$N(t) = \sum_{i=1}^{d} \mathbf{1}_{M_T^i > b}$$

We would like to estimate

$$\mathbb{P}[N(t) = k], \quad 0 \leq k \leq d$$

Similar to what we explained in Section 3.1, we can apply IPS and IS method to estimate such probability.

3.2.1 IPS method

The IPS method can be extended to multi-dimensional version of the problem easily. Since we need to keep track of the value and running maxima of every Brownian motion at each time step, we let

$$X_k = (B_{t_k}^1, M_{t_k}^1, B_{t_k}^2, M_{t_k}^2, \ldots, B_{t_k}^d, M_{t_k}^d)$$

For the same reason as explained in Section 3.1.1, we let

$$G_k(Y_k) = \exp(\sum_{i=1}^{d} \alpha(M^i_{t_k} - M^i_{t_{k-1}}))$$

With $X_k, G_k(\cdot)$ specified, the rest follows through in the exact same way as described in Section 3.1.1.

3.2.2 IS method

The IS method can be extended from one dimensional problem to multidimensional problem easily. The only non trivial issue is how to select appropriate μ. First, recall that

$$\mathbf{B}_t = C\tilde{\mathbf{B}}_t$$

where $\tilde{\mathbf{B}}_t$ is the standard $d-$dimensional Brownian motion. Thus, suppose we change the drift so that under the new measure $\tilde{\mathbf{B}}_t + \mu t$ is the standard $d-$dimensional Brownian motion. Then under the new measure

$$\mathbf{B}_t = C(\tilde{\mathbf{B}}_t + \mu t) = C\tilde{\mathbf{B}}_t + (C\mu)t$$

Thus, we would like to select μ so that $C\mu$ has a positive drift. Fix $\alpha > 0$, preferably $\alpha T > b$, then we let

$$\mu = \alpha C^{-1}\vec{1} \tag{3}$$

so that $\mathbf{B}_t$ has drift $\alpha\vec{1}$ under the new measure. To summarize, the IS method can be applied as follows,

- *Generating Sample Path*: Suppose we have generated M sample paths up to time t_p, denote them by $\mathbf{B}_{i,t_p}$. To extend each path to the next step, we first generate d independent normal random variables with mean 0 and variance 1, denote them by $\mathbf{Z}$. Then we get

  $$\tilde{\mathbf{B}}_{i,t_{p+1}} = \tilde{\mathbf{B}}_{i,t_p} + \sqrt{\Delta t}\mathbf{Z} + \mu\Delta t$$

  $$\mathbf{B}_{i,t_{p+1}} = \mathbf{B}_{i,t_p} + C(\tilde{\mathbf{B}}_{i,t_{p+1}} - \tilde{\mathbf{B}}_{i,t_p})$$

  We need to do this step for all $i = 1, 2, \ldots, M$ independently.

- *Estimation* After generating the path up to time $t_n = T$, our estimate is given by

  $$\hat{\mathbb{P}}(N(t) = k) = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}_{\{f(\mathbf{B}_{i,T})=k\}} e^{-\mu\tilde{\mathbf{B}}_T - \frac{\mu^T\mu}{2}T}$$

  where

  $$f(\mathbf{B}_T) = \sum_{j=1}^{d} \mathbf{1}_{M^j_T > b}$$

4 Numerical Result

4.1 One Dimensional Problem

With $T = 1$ and $b = 5$, we have the exact value of $\mathbb{P}[M_T > b] = 0.5733 \times 10^{-6}$. Let $n = 100, M = 10,000, \alpha = 5, \mu = 3$ in our estimation, we have the following results in Table I.

Method of Estimation	Estimator	Standard Deviation	CPU Running Time
Plain Monte Carlo	0	0	0.06
IPS	0.3551×10^{-6}	0.0084×10^{-6}	130.75
IS	0.3568×10^{-6}	0.142×10^{-6}	0.06

Table 1: Estimator of $\mathbb{P}[M_T > b]$

From Table 1 above, we have that plain Monte Carlo fails completely in estimating the value of $\mathbb{P}[M_T > b]$. Each Monte Carlo simulation gives value 0. In contrast, the IPS and IS method yield non-zero results. Notice that for the same number of path and the same number of discretization, the IPS estimator has lower standard deviation but requires more CPU running time.

Note that the CPU running varies greatly between between IPS and IS method. Thus, it is necessary to take that into account when comparing the performance of both methods. Table

M	500	1,000	5,000	10,000	50,000	100,000	500,000
Plain Monte Carlo							
CPU Running Time	0.02	0.03	0.04	0.06	0.54	1.52	12.73
Estimator	0	0	0	0	0	0	0
Standard Deviation	0	0	0	0	0	0	0
Interacting Particle System							
CPU Running Time	4.28	8.77	54.32	130.75	1,646.99	7,052.26	-
Estimator($\times 10^{-6}$)	0.1331	0.4084	0.3429	0.3551	0.3043	0.3493	-
Standard Deviation ($\times 10^{-6}$)	0.0140	0.0259	0.0122	0.0084	0.0034	0.027	-
Important Sampling							
CPU Running Time	0.01	0.02	0.04	0.06	0.69	1.76	24.01
Estimator($\times 10^{-6}$)	0.2439	0.5924	0.3299	0.3568	0.3843	0.3421	0.3682
Standard Deviation ($\times 10^{-6}$)	9.2650	0.1407	0.0427	0.0302	0.0142	0.0109	0.0048

Table 2: Estimators from different methods for varying number of simulation (M)

From Table 2, we have that plain Monte Carlo fails completely at estimating $\mathbb{P}[M_T > b]$ regardless of the number of simulation (M). For IPS and IS, we have that, as the number of simulation increases, the variance of estimator is reduced as a trade off with longer CPU running time. In case of one dimensional problem, IPS seems to have poorer performance than IS. IPS requires significantly larger amount of time. Note that the IS estimator with $M = 500,000$ gives the estimator of 0.3682×10^{-6} with standard deviation of 0.0048×10^{-6} which is lower than the standard deviation of 0.0084×10^{-6} of the IPS estimator with $M = 50,000$ after 24.01 CPU running time versus the 130.75 CPU running time in IPS case.

N	50	100	500	1,000	10,000
Plain Monte Carlo					
CPU Running Time	0.03	0.06	0.34	0.60	62.34
Estimator	0	0	0	0	0
Standard Deviation	0	0	0	0	0
Interacting Particle System					
CPU Running Time	64.66	133.98	763.89	1,901.77	-
Estimator($\times 10^{-6}$)	0.2208	0.3663	0.3741	0.2754	-
Standard Deviation ($\times 10^{-6}$)	0.5538	0.8976	1.002	0.7779	-
Important Sampling					
CPU Running Time	0.06	0.07	0.38	0.9	116.13
Estimator($\times 10^{-6}$)	0.2778	0.3518	0.5523	0.4699	0.5827
Standard Deviation ($\times 10^{-6}$)	2.9063	2.8281	4.765	3.5172	0.0420

Table 3: Estimators for different methods for varying number of discretization (N)

From Table 3, we have that plain Monte Carlo fails completely at estimating regardless of the number of discretization (N). For IPS and IS, we have that, as N increases, CPU running time increases, but with some improvement in bias. However, the estimators from IPS and IS are not very stable. As N is bigger than 100, it is not clear that as we keep increasing N, the bias will be reduced. From Table 3, it is clear that IS method is better in this case. Note that the IS estimator with $N = 10,000$ gives the estimator of 0.5827×10^{-6} with standard deviation of 0.0420×10^{-6} which is lower than the standard deviation of 0.7779^{-6} of the IPS estimator with $N = 1,000$ after 116.13 CPU running time versus the 1,901.77 CPU running time in IPS case.

4.2 Multi-Dimensional Problem

In this problem, we will work with $d = 2$, i.e. two Brownian motions. We pick the following covariance matrix,

$$\Sigma = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

From Table 4, again plain Monte Carlo fails completely. In contrast, the IPS and IS method yields non-zero results. Notice that while IS requires longer CPU running time of 82.25 seconds, the variance of the IS estimator is still greater than that of the IPS method. This shows that, as we move to higher dimensional, the conclusion in section (4.1) is not carried. Due to the complexity of the problem, it might be easier to select appropriate weighting function than to pick the right measure to capture the event of interest.

Method	Estimator ($\times 10^{-7}$)	S.D. ($\times 10^{-7}$)	M	CPU Running Time
Plain Monte Carlo	0	0	100,000	257.74
IPS	6.148	0.8614	10,000	70.49
IS	5.757	1.0745	30,000	82.25

Table 4: Estimator of $\mathbb{P}[N_T = 1]$ with $N = 50, T = 1, b = 5, d = 2, \alpha = 2$

Next, we consider a higher dimension by letting $d = 4$ with covariance matrix

$$\Sigma = \begin{bmatrix} 1 & 0.6 & -0.3 & 0.7 \\ 0.6 & 1 & -0.5 & 0.2 \\ -0.3 & -0.5 & 1 & -0.1 \\ 0.7 & 0.2 & -0.1 & 1 \end{bmatrix}$$

Method	Estimator ($\times 10^{-7}$)	S.D. ($\times 10^{-7}$)	M	CPU Running Time
Plain Monte Carlo	0	0	100,000	261.51
IPS	3.5181	0.3786	10,000	72.27
IS	4.0157	1.3998	50,000	137.77

Table 5: Estimator of $\mathbb{P}[N_T = 1]$ with $N = 50, T = 1, b = 5, d = 4, \alpha = 2$

From Table 5, we observe a similar result to the previous case. That is, plain Monte Carlo cannot capture the rare event and give zero probability, while both IPS and IS method yield non-zero result. IPS method performs better with lower standard deviation and use only half amount of running time compared to the IS method.

5 Conclusion

In this paper, we have explained in detail how to estimate a number of up crossing of correlated Brownian motions using Interacting Particle System (IPS) method and Important Sampling (IS) method. When a barrier level is high relative to the variance of the Brownian motions, this event is considered to be rare, and plain Monte Carlo fails completely in estimating the probability of this event.

Since IPS method involves resampling in every step, this method require longer CPU running time, when running with the same number of simulation, compared to that of IS or plain Monte Carlo. However, it is easier to select an appropriate weighting function to capture the event of interest than to select a Girsanov change of measure, especially in high dimensional problem. As a result, IPS requires less number of simulation to generate the same level of standard deviation. Thus, with these two opposing factors, it is not clear a priori which method will perform better at estimating this rare event.

The main goal of this paper is to answer the above question through numerical result. In one-dimensional problem ($d = 1$) , the numerical result suggests that the IS method is a superior estimator with lower running time and lower standard deviation (Section 4.1). However, in higher dimensional problem ($d = 2, 4$), the result is reversed. Even with a carefully chosen change of measure (Section 3.2.2), IS performs worse with higher running time and higher standard deviation (Section 4.2).

References

[1] Larry Shepp L.C.G. Rogers, *The correlation of the maxima of correlated brownian motions*, Journal of Applied Probability **43** (2006), no. 3.

[2] P.Del Moral, *Feynman-kac formulae: Genealogical particle analysis of rare events*, Annals of Applied Probability **15** (2005), 2496–2534.

[3] Douglas Vestal Rene Carmona, Jean-Pierre Fouque, *Interacting particle systems for the computation of rare credit portfolio losses*, Finance and Stochastics **13** (2009), 613–633.