A Comparative Study of Randomized Algorithms for Multidimensional Integration

Zinan Zhao and Mrinal Kumar

Abstract—This paper presents a comparative study of randomized algorithms for computation of a class of high dimensional Gaussian weighted integrals. The work is an extension of past research by Keister et al. and later by Papageorgiou et al. who used non-product (grid-less) multidimensional quadrature rules and Quasi Monte Carlo respectively for integration in up to 100 dimensions. In the present paper, the same integrals are computed using Markov chain Monte Carlo (MCMC) and a comparison is made. It is shown that the MCMC technique is significantly more accurate for the problem of interest and is also robust in implementation. Moreover, by virtue of its information-centric approach, MCMC can be adapted to efficiently compute integrals weighted by general weight functions (besides Gaussian weights).

I. INTRODUCTION AND PROBLEM STATEMENT

Very high dimensional integrals frequently arise in numerous applications of engineering, for example in the problem of Bayesian data fusion involving multiple sensors and/or targets. The number of dimensions involved can become very high in certain applications of computational physics, e.g. the calculation of elements of the quantum mechanical matrix in atomic, nuclear and particle physics (Ref.[1]). The well known Monte Carlo (MC) method [2] is known to break the curse of dimensionality associated with numerical integration. For this result to hold however, the domain of integration must be the unit hypercube. The expected error associated with MC integration decreases at the rate $N^{-1/2}$, where $N$ is the number of uniformly distributed samples used in the domain of integration. For problems involving very high dimensional integrals, this convergence rate is unfortunately deemed inadequate (Refs.[1], [3], [4]). Consider a general weighted integration problem given as follows:

$$I = \int_{\mathbb{R}^d} f(x) \pi(x) dx$$

(1)

where $x \in \mathbb{R}^d$, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is integrable and $\pi(x)$ is the weight function.

In the quest for accurate computation of the above integrals with low computational effort, two different non-product (i.e. grid-less) deterministic quadrature rules were employed in papers by Capstick et al. (Ref.[1]) and Keister (Ref.[3]). These quadrature laws are significantly different from standard continued tensor-product based rules such as Gauss quadratures because no grids are involved. The two techniques are due to McNamee and Stenger (Ref.[5]) and Genz and Patterson (Refs.[6], [7]). The McNamee-Stenger scheme uses the zeros of orthogonal polynomials in a single dimension as quadrature points and thus can only be used if the weight function appearing in the integral is separable (see Eq.1 below). This aspect severely reduces its scope of application. The Genz-Patterson (GP) quadrature scheme uses the zeros of multidimensional Lagrange interpolation polynomials as quadrature points and thus has a broader scope of application. Reasonable results for Gaussian weighted integrals in 4, 9 and 25 dimensions were reported with the GP method in Ref.[1]. These results far exceed the accuracy possible with grid-based techniques. It was additionally claimed that randomization schemes such as Monte Carlo based methods would fail for these integrals.

In Ref.[4], Papageorgiou et al. explored the use of quasi randomization for evaluation of the integral in Eq.1. They utilized the generalized Faure sequence version of quasi Monte Carlo integration (QMC-GF) (Ref.[8]). The selection of QMC-GF was motivated by its success in fast valuation of very high dimensional financial derivatives (Ref.[9]). It is known that in relatively low dimensions (approximately $< 10$) all QMC constructions are equivalent. However, special care is required for higher dimensions and the use of generalized sequences (such as generalized Halton sequence or generalized Faure sequence) becomes important in order to prevent quadrature points from forming patterns (Ref.[8]). Theoretically QMC is known to provide error convergence at the rate $(\log N)^d/N$ on the unit hypercube, which is better than MC in lower dimensions. However when dimensionality $(d)$ increases QMC is expected to become worse than MC. Papageorgiou et al. (Refs. [4], [10]) have reported results on the contrary, which is likely due to the use of generalized QMC sequences (QMC-GF).

The current paper revisits the randomization paradigm for evaluating integrals of the form in Eq.1. However, instead of taking the Monte Carlo approach, the information-based Markov chain Monte-Carlo (MCMC) paradigm is adopted (Refs.[11], [12], [13], [14]). MCMC has already been successfully used in a variety of applications involving high dimensionality, including Bayesian inference[14], nonlinear filtering (Ref.[15]) and indeed, numerical integration ([16], [17]), and new applications continue to emerge. The key distinguishing factor (what we refer to as the “information-centric property”) of MCMC is that by virtue of its foundation in Markov chain theory (Refs.[14]), it automatically exploits the information contained in the integrand (even if...
the integrand does not include a weighting function). The resulting outcome is that a single MCMC algorithm can generate quadrature points that are equivalent in measure to any integrand of concern (with standard regularity properties). Therefore, the method does not require pre-processing steps such as the use of transformations that convert the integrand into a more convenient form. For the integral in Eq.1, two comparisons are made: (i) in Refs.[1], [3] and [4], a transformation was used to express \( I \) as a function of the cumulative probability density of the Gaussian weight. Results obtained for this transformed integral using MCMC are obtained and compared to QMC-GF as well as the GP non-product quadrature rule. (ii) MCMC is used to evaluate \( I \) without the use of any simplifying transformation and results are compared between MCMC and QCMC. It is shown that in the former case with the transformation, MCMC is comparable to QMC whereas in the case of no transformation, MCMC outperforms QMC considerably. In other words, the outcome of MCMC does not depend on the use of simplifying transformations, making it a robust approach. This is important also because it is not always possible to obtain useful transformations and this case is illustrated via an example.

The rest of the paper is organized as follows. In Sec.II the details of the above described transformation are provided. Sections II-A and II-B describe the QMC and MCMC schemes used in this paper and comparative results are provided in Sec.III. Finally, conclusions are drawn in Sec. IV.

II. QUASI RANDOMIZED AND MCMC RANDOMIZED INTEGRATION

Most quadrature based numerical methods for evaluation of integrals such as the GP scheme, MC or QMC require a finite sized domain for implementation. Unfortunately, this is not the case for the integral in Eq.1. One way to resolve this issue is to use a conservatively sized domain outside of which the integral mass is negligible. For the Gaussian weighted integral of Eq.1, the finite domain can be sized using the eigenvalues of the covariance matrix, e.g. in \( d = 1 \), a conservative domain could be \([-12\sigma, 12\sigma]\), where \( \sigma \) is the standard deviation of the Gaussian weight. Two test cases are studied in this work: \( f_1(x) = \cos(||x||) \) and \( f_2(x) = \sqrt{1 + ||x||^2} \) along with a Gaussian weight, i.e. \( \pi(x) = \exp\left(-||x||^2\right) \). Note that this is an example of an isotropic integral, i.e. one in which all dimensions contribute equally and thus the use of special techniques that spend less computational effort on “unimportant dimensions” is not an option. In Refs.[1], [4], the following simplifying transformation was employed to convert the domain of integration to \([0,1] \times [0,1] \times \ldots \times [0,1]\), i.e. the hypercube \([0,1]^d\):

\[
I = \int_{[0,1]^d} \cos(||x||) \exp\left(-||x||^2\right) \, dx \\
= 2^{-d/2} \int_{\mathbb{R}^d} \cos\left(\frac{||y||}{\sqrt{2}}\right) \exp\left(-\frac{||y||^2}{2}\right) \, dy \\
= \pi^{d/2} \int_{\mathbb{R}^d} \cos\left(\frac{||y||}{\sqrt{2}}\right) \left(\frac{1}{2\pi} \right)^{d/2} \exp\left(-\frac{||y||^2}{2}\right) \, dy \\
= \pi^{d/2} \int_{[0,1]^d} \cos\left(\sum_{j=1}^{d} (\phi^{-1}(t_j)^2)/2\right) \, dt
\]

where \( \phi \) is the cumulative standard normal distribution function given as:

\[
\phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-s^2/2} \, ds, \, u \in [-\infty, \infty].
\]

It will be shown in the current paper that the above transformation plays a crucial role in the success of QMC-GF. On the other hand, MCMC integration is unaffected by its use and thus provides more robust estimates. In more general problems, e.g. when the weight function is not Gaussian, it may not be possible to perform manipulations shown above. It is claimed that the MCMC method would fair better in such situations.

A. Quasi Monte Carlo

Quasi Monte Carlo is a suite of “quasi-randomized” algorithms that is based on “low-discrepancy” sequences. The basic idea behind QMC is to deterministically generate a sequence of points that are (approximately) equivalent in measure to a uniform random distribution. The “discrepancy” is thus understood as the difference in measure over the domain of interest between the generated sequence and a sequence of numbers belonging to a true uniform random distribution (Ref.[18]). The motivation behind QMC is to be able to exploit the power of randomization when faced with high dimensional problems while also retaining the advantages of a deterministic approach, e.g. deterministic error bounds. Common low discrepancy sequences are the Halton sequence, Faure sequence and Sobol sequence, and they all come in two varieties - regular and generalized, e.g. regular Faure sequence and generalized Faure sequence (Ref.[19]). Among the regular variety of QMC sequences, there is no strong evidence suggesting the superiority of one over the other. On the unit hypercube, QMC theoretically provides a deterministic error bound of \(O((\log N)^{d-1}/N)\) in contrast to MC’s probabilistic error bound of \(O(N^{-1/2})\) for the problem of numerical integration. However, this result is strongly dependent on the ability of QMC sequences to retain their property of low discrepancy in higher dimensions. Unfortunately this does not happen for the regular variety of sequences and they start showing patterns in dimensions as low as 10. The problem of pattern formation is very effectively resolved by the generalized variety of sequence, e.g. the generalized Faure sequence (QMC-GF), which has been used in this paper for comparison purposes. In Ref.[4] it has been shown that for the class of integrals considered here,
QMC-GF performs much better than the theoretical error convergence rate of $O((\log N)^{d-1}/N)$. We confirm this result in the current paper, however discover that the underlying reason is likely the use of the transformation described in Eq.2. When the transformation is not used, QMC-GF error characteristics deteriorate significantly.

For the integral in Eq.1, assuming a general integration domain of $\Omega \subseteq \mathbb{R}^d$ with volume $V(\Omega)$, a sequence of QMC points $\{x_i\}_{i=1}^N$ provide the following estimate:

$$I_{\text{QMC}} \approx \frac{V(\Omega)}{N} \sum_{i=1}^N \cos(\|x_i\|) \exp(-\|x_i\|^2)$$  \hspace{1cm} (4)

Note that the above expression is exactly the same as that for an MC estimate. The only difference is that instead of being samples drawn from a true uniform distribution on $\Omega$, i.e. $U(\Omega)$, $x_i$ are QMC samples of low discrepancy. It is interesting to observe that in practice, it is actually not possible to generate truly random numbers on a computer. Both MC and QMC are use algorithmically generated samples, and MC samples realized on a computer use time-varying seeds whereas QMC samples use fixed seeds.

The QMC estimate in Eq.4 does not make use of the transformation of Eq.2. If the transformation is used, the sampling is performed on the unit hypercube, $\{t_i\}_{i=1}^N$ and the resulting estimate is:

$$I_{\text{QMC}} = \frac{\pi^{d/2}}{N} \sum_{i=1}^N \cos(\sqrt{\sum_{j=1}^d (\phi^{-1}(t_{ij})/2}$$  \hspace{1cm} (5)

B. Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a powerful suite of methods that is used to generate discrete samples from any given continuous probability measure, i.e. $\{x_i\}_{i=1}^N \sim \pi(x)$. Since the set $\{x_i\}_{i=1}^N$ is equivalent in measure to $\pi(x)$, the integral of Eq.1 can be approximated as (Ref.[14]):

$$I = \int_{\mathbb{R}^d} f(x)\pi(x)\,dx = E_x[f(x)] \approx \frac{1}{N} \sum_{i=1}^N f(x_i)$$  \hspace{1cm} (6)

The only assumption in the above relationship is that the weight $\pi(\cdot)$ must be a normalized probability density function over the domain of integration, i.e. $\int_{\mathbb{R}^d} \pi(x)\,dx = \eta = 1$. However, for the actual procedure of generating samples $\{x_i\}_{i=1}^N$, the weight is not required to be normalized. In other words, the samples approximate the integral of Eq.6 with respect to a normalized weight even if the constant of normalization ($\eta$) is unknown. Also, Eq.6 is valid for both finite and indefinite domains of integration. In fact, one of the greatest advantages of MCMC is that it completely eliminates integration domain from the picture by automatically sampling more points from regions where the weight is “heavy”. In other words, the relative number of points sampled from different regions of the domain is (approximately) in proportion to the weight’s measure in those regions. For probability density functions, the “weight-measure” is the same as the probability mass. There exist a variety of MCMC algorithms for generating samples from weight functions. In the current work, we have used the Metropolis-Hastings (MH) algorithm (Refs. [11], [12], [20]) with adaptive proposal selection. A comprehensive introduction can be found in Müller (Ref.[21]) and Tierney (Ref.[22]). Here, we mention only the key ideas and some important implementation issues.

MCMC samples are generated by building a Markov chain and allowing it to evolve through a period of initial transience. This is called “chain burn-in” and all samples generated during this period are discarded. Once transience is completed, the chain exhibits stationary behavior, with a distribution which by design is chosen to be the same as the weight ($\pi(x)$) for which samples are desired. The chain is constructed by generating samples from a proposal distribution and then accepting or rejecting them based on their relative likelihood of belonging to the target weight function. Fig.1 illustrates how this is done. The red circle depicts the current state of the chain, $x_c$. A new sample, $x_s$ (marked by the blue cross) is drawn from a proposal density, $q(x)$.

As in other sequential Monte Carlo methods [23], $q(x)$ is a density that is easy to sample from, e.g. a Gaussian pdf centered at $x_c$. In the Metropolis-Hastings algorithm, the new sample is admitted into the chain with a probability of acceptance $= \min\left(1, \frac{\pi(x_s)}{\pi(x_c)}\right)$. If the new sample causes the chain to move towards a mode of the target density, it is accepted with certainty (see Fig.1(a)). On the other hand, if $x_s$ moves away from a mode, it is accepted with a lower probability, as illustrated in Fig.1(b).

Below, we discuss some important implementation issues in the M-H algorithm:

- The proposal must satisfy two conflicting properties: it must be easy to sample from, yet cause the chain to burn-in quickly. Various types of proposals are used in the existing literature and design of the ideal proposal continues to be an area of active research. We mention here that while important, choice of a very good proposal is not a critical issue in the MH algorithm. We mention here that while important, choice of a very good proposal is not a critical issue in the MH algorithm. In the current work, we use a Gaussian proposal with mean located at the current state of the chain (see Fig.1). This leaves only the covariance matrix for tuning. We use the following adaptive rule: (see Ref.[21]) (i) begin with an initial standard deviation of $\sigma = 1$ along all dimensions; (ii) if the average acceptance rate over the 10 most recent states is greater than 0.8, then $\sigma$ is increased by 20% (to increase the exploratory tendency of the chain and avoid getting trapped near a particular mode); while if this average is less than 0.2, $\sigma$ is decreased by 30%.
- Another subject of research in MCMC literature is the simultaneous execution of multiple chains. Although running multiple chains is possible, in this paper we only execute a single chain. Currently, there is no consensus about the relative merits of running one long chain versus multiple short chains in parallel (Ref.[14]).
- The final important issue is determination of the burn-in time, which depends on the starting location of the chain and the proposal densities used. In the current
paper, the chains were initiated at arbitrary locations (no criteria used) and used a relatively long burn-in time of 5000 states for dimensions 1–100. In an ideal case, the starting value would be chosen close to the dominant features of the target weight, which would lead to fast burn-in.

III. RESULTS

Consider the integral of Eq.1 and note that the weight function \( \pi(x) \) is not normalized. We therefore perform a simple re-scaling as follows:

\[
I = \eta \int_{\mathbb{R}^d} f(x) \tilde{\pi}(x) \, dx
\]

where

\[
\eta = (2\pi)^{d/2} |\Sigma|^{1/2}, \quad |\Sigma| = 1/2,
\]

\[
\tilde{\pi}(x) = \frac{\pi(x)}{\eta} = \exp\left(\frac{-||x||^2}{2|\Sigma|}\right)
\]

Additionally, we use \( f(x) = f_1(x) = \cos(||x||) \). The true value of this integral is known to be: \( I(d=25) = -1.356914 \times 10^{26} \) and \( I(d=100) = 4.57024396 \times 10^{24} \). Accuracy of results is measured in terms of absolute relative error: \( e_{rel} = \left| \frac{I - \hat{I}}{I} \right| \), where \( \hat{I} \) represents the numerically computed estimate of the true value, \( I \). Using MCMC samples described in Sec.II-B for \( \pi(x) \), an estimate for Eq.8 can be given as (see Eq.6):

\[
\hat{I}_{\text{MCMC}} \approx \eta \frac{N}{N} \sum_{i=1}^{N} f(x_i)
\]

Comparisons are presented below for two cases: with and without the use of the transformation described in Eq.2 for the QMC-GF method. Note that for all the results below, the transformation was not used in the MCMC method because MCMC sampling thrives on the presence of a weight function to exploit information from. When the integral of Eq.1 is converted to Eq.2 via the change of variables, the weight function no longer explicitly appears in the integrand.

There could be two possible routes to take: (i) assume that the weight function is a uniform distribution on the domain of integration. However, this assumption causes MCMC to degenerate to the simple MC approach, in which samples are drawn from a uniform distribution. (ii) treat the entire integrand as the weight function (in case there are negative parts, take its absolute value). This would reduce the problem to one of finding the normalization constant of the new weight function in the domain of integration. MCMC cannot be used directly for this problem and there exist importance sampling type techniques for dealing with it. The authors of this paper are also currently involved in research towards developing an alternate method for this problem.

A. Case 1: Transformation of Eq.2 Used for QMC-GF

For comparison purposes, we concentrate on two dimensions, \( d = 25 \) and \( d = 100 \). Graphical results are shown in Fig.2. Since there are an infinite number of ways to implement the generalized Faure sequence (Ref.[8]), two graphs have been shown: one for the QMC-GF sequence used in this paper (dash-dot black line) and one for which data was taken from Ref.[4] (unit weight blue line with data points shown using filled circles). Finally, results from the MCMC approximation have been shown using a solid bold red line. Since MCMC is a randomized technique, its relative error results shown in these figures are an average of 100 runs of the algorithm. Also, a finite domain was not specified for the MCMC technique and it was allowed to construct Markov chains in \( \mathbb{R}^d \) which is the desired domain of integration.

Table I shows the relative errors for Genz-Patterson (GP), QMC and MCMC methods for the case of \( d = 25 \). GP results were taken from Ref.[1] in which a somewhat different number of quadrature points was used (1251 points for a relative error of 2.04 and 19751 for a relative error of 0.446). The corresponding data for \( d = 100 \) is shown in Table II (GP results are not available for this case). The following observations can be made:
1) MCMC method draws upon the information provided by the weight function to estimate the integral whereas the QMC method, even after the variable transformation and with the use of the generalized Faure sequence, assumes uniformity and does not make use of the information.

2) For $d = 25$, MCMC performance is comparable with that of QMC-GF when the transformation is used.

3) On average, MCMC outperforms the QMC-GF method by an order of magnitude in terms of relative error for $d = 100$.

4) Comparable performance is observed for the two QMC-GF algorithms as well as QMC-Halton for both $d = 25$ and $d = 100$. While the first result (similar performance of two separate QMC-GF algorithms) is not a surprise, the second result is. QMC-Halton performs almost as well as QMC-GF despite pattern formations in the dimensions of interest.

5) For both dimensions, MCMC method achieves a relative error $O(10^{-3})$ using 500 sample points, which is an improvement over the two QMC-GF techniques.

### Table I

<table>
<thead>
<tr>
<th>Number of Points →</th>
<th>500</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC-Halton</td>
<td>$3 \cdot 10^{-2}$</td>
<td>$3 \cdot 10^{-2}$</td>
<td>$10^{-3}$</td>
<td>$5 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>QMC-GF (Ref.[4])</td>
<td>$10^{-4}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>QMC-GF (present paper)</td>
<td>$4 \cdot 10^{-2}$</td>
<td>$2 \cdot 10^{-4}$</td>
<td>$2 \cdot 10^{-3}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>MCMC</td>
<td>$2 \cdot 10^{-3}$</td>
<td>$2 \cdot 10^{-3}$</td>
<td>$2 \cdot 10^{-3}$</td>
<td>$2 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>Genz-Patterson</td>
<td>-</td>
<td>2.04</td>
<td>-</td>
<td>0.446</td>
</tr>
</tbody>
</table>

#### Table I

**QMC AND MCMC ESTIMATION RESULTS FOR $d = 25$**

### Table II

<table>
<thead>
<tr>
<th>Number of Points →</th>
<th>500</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMC-Halton</td>
<td>$2 \cdot 10^{-4}$</td>
<td>$2 \cdot 10^{-4}$</td>
<td>$1.5 \cdot 10^{-5}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>QMC-GF (Ref.[4])</td>
<td>$5 \cdot 10^{-4}$</td>
<td>$5 \cdot 10^{-3}$</td>
<td>$3 \cdot 10^{-3}$</td>
<td>$3 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>QMC-GF (Current Paper)</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>MCMC</td>
<td>$6 \cdot 10^{-3}$</td>
<td>$6 \cdot 10^{-3}$</td>
<td>$2 \cdot 10^{-3}$</td>
<td>$2 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

#### Table II

**QMC AND MCMC ESTIMATION RESULTS FOR $d = 100$**

Tests for other dimensions were also carried out, and the results are similar to the ones we show with 25 and 100 dimensions.

**B. Case 2: Transformation of Eq.2 Not Used for QMC-GF**

For many problems, a transformation of the type given in Eq.2 may not be available. In this section, we will work with the original integral. Since a finite size domain must be specified for QMC to work, we use $[-10,10]^d$. To get a feel for the extent of this domain, consider that $6\sigma \approx 4.25$ along each dimension. For MCMC, as in the above case, results are provided as the mean of 100 runs. Simulation results for the test integral are shown in Fig. 3: Fig.3(a) shows the relative errors obtained for MC, QMC-GF, QMC-Halton and MCMC methods when 5000 points are used for integration in dimensions 1-30. Several points are worth noting:

1) For lower dimensions ($\leq 5$), MC, QMC-GF and QMC-Halton all perform relatively well. However, the errors climb very fast as 5000 points are not sufficient to fill up the total volume of integration. Note that this leads to the failure of all three methods: MC, QMC-Halton and QMC-GF. When the transformation was used, the domain of integration was converted to the unit hypercube. The volume of a unit hypercube does not grow with dimensionality and hence claims about breaking the curse can be made. However, in a more general setting, the volume grows exponentially with dimensionality and these claims no longer hold up. The number of points required to maintain accuracy must therefore also increase in sync with the volume of the
domain of integration, which unfortunately implies an exponential growth.

2) After about \( d = 10 \), QMC and MC completely fail and the relative error reaches a value of 1, which implies an absolute error of 100%.

3) On the other hand, the performance of MCMC is consistently good for all dimensions, even though the integration is being performed in \( \mathbb{R}^d \). The simple reason is that the MCMC chain automatically follows the regions of importance of the weight function and does not venture into areas where there is no activity of the integrand.

4) Not only does MCMC provide better mean error characteristics, it is also highly confident about its estimates (see confidence bounds in Fig.3(b)). The reason is the same - since the quadrature points are specially tailored to suit the integral being computed, there is small uncertainty about the mean value of error computed. This translates to fewer number of runs required to compute the expected value of integral accurately.

IV. CONCLUSIONS

In this paper, we compared the use of randomization schemes for computation of a class of isotropic multidimensional integrals. Monte Carlo, quasi Monte Carlo (Halton and generalized Faure) and Markov chain Monte Carlo methods were compared. Monte Carlo methods have traditionally been deemed inadequate for these integrals and the results in this paper confirm this fact. In existing literature, typically a transformation is employed to evaluate these integrals which converts the indefinite integral to a definite integral on the unit hypercube. The MCMC approach was shown to be highly successful even when this transformation is not in use, as might be the case with more general problems. On the other hand, QMC (generalized Faure) fails to replicate its exceptional performance in the absence of the transformation. On the other hand when the transformation is in place, the two methods give comparable results, with MCMC faring better if a very small number of points is used (few hundreds).

It should be stressed that the usage of MCMC requires that the integration must be performed with respect to a normalized weight, even though the knowledge of the constant of normalization is not required. There exist importance sampling methods to relax this assumption and research is currently under progress for the development of alternate methods. Theoretical work on the error characteristics of QMC are available for the class of integrals concerned. For MCMC however, this is still not available and represents another direction for further research.

REFERENCES


