Adaptive Monte Carlo algorithm for Wigner kernel evaluation

Article *in* Neural Computing and Applications · November 2019

DOI: 10.1007/s00521-019-04519-9

CITATIONS
0

READS
39

4 authors, including:

Venelin Todorov
Bulgarian Academy of Sciences
26 PUBLICATIONS **28** CITATIONS
SEE PROFILE

Ivan Dimov
Bulgarian Academy of Sciences
251 PUBLICATIONS **1,720** CITATIONS
SEE PROFILE

Stoyan Dimitrov
University of Illinois at Chicago
5 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04519-9

ORIGINAL ARTICLE

Adaptive Monte Carlo algorithm for Wigner kernel evaluation

Venelin Todorov[1,2] · Ivan Dimov[2] · Rayna Georgieva[2] · Stoyan Dimitrov[3]

Received: 28 June 2018 / Accepted: 5 October 2019

Abstract
In this paper, we study numerically various approaches, namely an adaptive Monte Carlo algorithm, a particular rank-1 lattice algorithm based on generalized Fibonacci numbers and a Monte Carlo algorithm based on Latin hypercube sampling for computing multidimensional integrals. We compare the performance of the algorithms over three case studies—multidimensional integrals from Bayesian statistics, the so-called Genz test functions and the Wigner kernel—an important issue in quantum mechanics represented by multidimensional integrals. A comprehensive study and an analysis of the computational complexity of the algorithms under consideration has been presented. Adaptive strategy is well-established as an efficient and reliable tool for multidimensional integration of integrands functions with computational peculiarities like peaks. The presented adaptive Monte Carlo algorithm gives reliable results in computing the Wigner kernel by a stochastic approach that has significantly lower computational complexity than the existing deterministic approaches.

Keywords Multidimensional integration · Adaptive Monte Carlo algorithm · Fibonacci lattice sets · Latin hypercube sampling · Wigner kernel

Mathematics Subject Classification 65C05 · 65U05 · 65F10 · 65Y20

1 Introduction

The Monte Carlo (MC) methods are widely used in solving different multidimensional problems by performing realizations of random processes or random variables [8, 12, 26].

Various mathematical formulations of quantum mechanics exist, among which the ones suggested by E. Schrodinger, E. Wigner, R. Feynman, L.V. Keldysh, K. Husimi, D. Bohm are more frequently used nowadays [34]. The Wigner formulation of quantum mechanics allows the comprehension and prediction of quantum mechanical phenomena in terms of quasidistribution functions. One of the best known physicist Richard Feynman formulated the problem of finding an effective and fast algorithm with linear or polynomial computational complexity for computing multidimensional integrals that represent Wigner kernel [13].

Up to now the Wigner kernel is calculated with deterministic methods [2, 35, 36, 42] which suffer from the ,,curse of

Venelin Todorov is supported by the Bulgarian National Science Fund under Project KP-06-PM32/4—2019 "Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics" and by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)", contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria. Stoyan Dimitrov is supported by the Project KP-06-H22/3 financed by the National Scientific Fund of Bulgaria at the Ministry of Education and Science. The work is also partially supported by the Bulgarian National Science Fund under Project DN 12/5-2017. The authors are grateful to Dr. Jean Michel Sellier for the valuable discussions concerning quantum mechanics and especially Wigner function.

✉ Venelin Todorov
vtodorov@math.bas.bg; venelin@parallel.bas.bg

Ivan Dimov
ivdimov@bas.bg

Rayna Georgieva
rayna@parallel.bas.bg

Stoyan Dimitrov
sdimit6@uic.edu

Extended author information available on the last page of the article

Published online: 02 November 2019

dimensionality" and that leads to computational times growing exponentially with the problem dimension. It is well known that MC and quasi-Monte Carlo (QMC) methods are not affected by the „curse of dimensionality" [7, 8]. The Plain MC method [39] has a rate of convergence $O(N^{-1/2})$, where N is the number of samples, which is independent of the dimension d of the integral, and that is why Monte Carlo integration is the only practical method for many high-dimensional problems. Adaptive Monte Carlo approach [3, 9, 10, 15] is widely used method for evaluation of integrals, which gives the best results when the integrand function has peculiarities and peaks. On the other hand, lattice rules are a special type of the so-called low-discrepancy sequences and give best results when the integrand is sufficiently regular [38]. Latin hypercube sampling (LHS) may lead to smaller variance compared to the pseudorandom sampling [39]. The algorithms under consideration have been applied to compute the Wigner kernel in the case of single-body quantum systems in terms of signed particle formulation of quantum mechanics. This is the first time to study stochastic algorithms for computing Wigner kernel in the case of single-body quantum system for higher dimensions. The case study of lower dimensions is presented in [32].

The layout of the rest of this paper is as follows. The next Sect. 2 the MC and QMC methods that we are using in the paper are discussed. Section 3 describes the Wigner kernel. The numerical experiments and results are presented in Sect. 4. Finally, some conclusions have been made concerning the advantages and disadvantages of the algorithms studied.

2 Monte Carlo and quasi-Monte Carlo methods

2.1 Plain (crude) Monte Carlo approach

Plain (Crude) Monte Carlo is the most famous and easy to implement MC approach for solving multidimensional integrals [8]. We are interested in the approximate computation of the integral $I[f] = \int_\Omega f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$. Let the random variable $\theta = f(\xi)$ is such that $\mathbf{E}\theta = \int_\Omega f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$, where the random points $\xi_1, \xi_2, \ldots, \xi_N$ are independent realizations of the random point ξ with probability density function (p.d.f.) $p(\mathbf{x})$ and $\theta_1 = f(\xi_1), \ldots, \theta_N = f(\xi_N)$. We obtain that the approximate value of the integral I is $\overline{\theta}_N = \frac{1}{N}\sum_{i=1}^{N}\theta_i$, which defines the Plain MC algorithm. One can see that the computational complexity of the Plain MC is linear, because we should choose N random points from the domain and every such choice is at the cost of $\mathcal{O}(1)$ operations.

2.2 Quasi-Monte Carlo strategy

Random MC techniques can often be improved by replacing random numbers with a more uniformly distributed deterministic sequences. Quasi-random numbers (QRNs) are constructed to minimize a measure of their deviation from uniformity called discrepancy [19]. The QMC based on well-distributed Sobol sequences [40] is a good alternative to the standard Monte Carlo algorithms. It is very appropriate for smooth integrands and not very high *effective dimension* (up to $d = 15$) [41]. We implement the algorithm using the Sobol sequence (SOBOL) that is an adaption of the INSOBL and GOSOBL routines in ACM TOMS Algorithm 647 [14] and ACM TOMS Algorithm 659 [4, 18]. The routine adapts the ideas of Antonov and Saleev [1].

2.3 Adaptive Monte Carlo algorithm

Adaptive Monte Carlo methods proposed by Lautrup [7] use a "sequence of refinements" of the original area selected so that the computations to be concentrated in subdomains with computational difficulties. There are various adaptive strategies depending on the technique of adaptation [8]. Our adaptive algorithm (simple adaptive Monte Carlo algorithm) does not use any a priori information about the smoothness of the integrand, but it uses a posteriori information for the variance obtained during calculations. The main idea is a concentration of random points in the subregions where the variance is large (in terms of a preliminary given accuracy), i.e. the approach is based on a recursive partitioning of the integration area using a posteriori error information for the current partition. Denote by p_j and I_{Ω_j} the following expressions: $p_j = \int_{\Omega_j} p(\mathbf{x})\,\mathrm{d}\mathbf{x}$ and $I_{\Omega_j} = \int_{\Omega_j} f(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$. Consider now a random point $\xi^{(j)} \in \Omega_j$ with a density function $p(\mathbf{x})/p_j$. In this case $I_{\Omega_j} = \mathbf{E}\left[\frac{p_j}{N}\sum_{i=1}^{N} f(\xi_i^{(j)})\right] = \mathbf{E}\theta_N$. This adaptive algorithm gives an approximation with an error $\varepsilon \le c\, N^{-1/2}$, where $c \le 0.6745\sigma(\theta)$ ($\sigma(\theta)$ is the standard deviation). From the estimation of the error, it can be concluded that, in general, the simple adaptive Monte Carlo algorithm gives an error less than the error of the Plain Monte Carlo algorithm, but the order is the same. The adaptive MC algorithm applied here is described below.

2.3.1 Description of the adaptive algorithm

1. **Input data** *number of points N, number of subintervals Mon every dimension coordinate (a relatively small integer number $M \ge 2$), constant ε (estimation of the variance, "stop criterion"), constant δ (maximum number of "bad" subdomains where the "stop criterion" is not satisfied).*

2. **Initial division** of the integration domain Ω into subdomains with identical volume, i.e. $\Omega = \sum_j \Omega_j, \quad j = 1, M^d$.
3. **For** $j = 1, M^d$ $(M \geq 2 for\ initial\ division\ and\ M = 2 otherwise)$
   - 3.1. **Calculate** *the approximation of* I_{Ω_j} *and the variance* $\mathbf{D}_{\Omega_j}$ *in subdomain* Ω_j *based on* N *independent realizations of random variable* θ_N.
   - 3.2. **If** $(\mathbf{D}_{\Omega_j} \geq \varepsilon)$ **then**
     - 3.2.1. **Add** *the current subdomain in the database of subdomains with a variance greater than* ε.
     - 3.2.2. **Choose** *the subdomain from the database with maximum variance from all levels of division.*
     - 3.2.3. **Divide** *the current domain into* 2^d *subdomains (into two in all directions)* **then go to** *step 3.1.*
   - 3.3. **Else** $(\mathbf{D}_{\Omega_j} < \varepsilon)$ **then**
     - 3.3.1. **Remove** *the current domain from the domain database.*
     - 3.3.2. **If** *the database has not been exhausted and the fixed maximum number of "bad" subdomains* δ*has not been reached* **then go to** *step 3.2.2.*
   - 3.4. **Accumulation** *in the approximation* I_N *of* I.
4. **End**

2.3.2 Discussion about computational complexity of the adaptive MC algorithm

The adaptive MC algorithm under consideration overcomes the main disadvantage of deterministic approaches to multidimensional integration—the problem is known as "the curse of dimensionality". The computational cost in this case is proportional to the number of points of the partition and it also grows exponentially to integral dimension. For our algorithm the computational cost is defined in the following way:

- Initially (at zero level of division), the original integration domain is divided into M^d $(M \geq 2)$ subdomains with identical volumes, where N number of samples is generated in each of these subdomains.
- On the next step of the algorithm k_0 $(1 \leq k_0 \leq M^d)$ subdomains are chosen, where the variance is greater than ε and this choice is independent of the dimension. At first and next levels of division each of the chosen subdomains is divided into 2^d subdomains with identical volumes. Obviously on every step of the algorithm, the adaptiveness is not in all subdomains, but only in $\mathcal{O}(1)$ subdomains.
- The following estimation for computational complexity is obtained:

$$M^d N + k_0 2^d N + \cdots + k_l 2^d N = N(M^d + 2^d \sum_{i=0}^{l} k_i) \approx 2^d \mathcal{O}(N),$$

where k_i is the number of subdomains for which the variance is greater than ε after the i-th level of division. The numbers k_i $(i = 0, \ldots, l)$ do not depend on the dimension of the integration area and, in practice, we assume that each k_i is much smaller than M^d. That is why we estimate it as $\mathcal{O}(1)$. We should emphasize here that the efficiency of the adaptive MC algorithm under consideration is high when computational peculiarities of the integrand occur only in comparatively small subregion of the initial integration domain. This fact defines the assumption about k_i to be much smaller than M^d as natural and reasonable. The efficiency of the adaptive MC algorithm also depends on the magnitude of the additive term $M^d N$. If the integer M is much larger than 2, then it is possible this term to define the main impact into the computational cost of the algorithm. That is why we recommend M to be comparatively small integer and close to 2 to take advantages of the algorithm in high extent.

- A simple example to compare the computational complexity of the adaptive MC algorithm under consideration and deterministic approaches for numerical integration is presented. If we would like to compute approximately a 10-dimensional integral with a preliminary given accuracy 0.01 by the proposed adaptive MC approach and the composite rectangular or the composite trapezoidal rule, then the initial data is as follows: $d = 10, N = 10,000, M = 2, m + 1 = 10$, where m is the number of subintervals for each component partition for the corresponding composite quadrature rules. Then, the integrand should be computed at $(m + 1)^d$ points to complete the particular quadrature rule. Therefore, for our stochastic approach we have $2^{10}.10^4 \approx 10^7$ integrand calculations whereas for deterministic approach 10^{10} points are necessary.

The main conclusion here is the adaptive MC algorithm under consideration has significantly smaller computational complexity than deterministic approaches for multidimensional numerical integration. This advantage is demonstrated especially for higher dimensions.

2.4 Lattice point sets

Lattice point sets are a special type of low-discrepancy sequences, and are based on the use of deterministic sequences rather than random sequences. The monographs of Sloan and Kachoyan [37] and Sloan and Joe [38]

provide comprehensive expositions of the theory of good lattice point sets for multidimensional integration.

We use the following a particular rank-1 lattice sequence [43]:

$$\mathbf{x}_k = \left\{\frac{k}{N}\mathbf{a}\right\},\ k = 1,\ldots,N,$$

where N is an integer, $N \geq 2$, $\mathbf{a} = (a_1, a_2, \ldots a_s)$ is an integer vector modulo N of dimensionality d called a generator of the set and $\{z\}$ denotes the fractional part of z. The choice of a good generating vector, such that the points to be well distributed over the integration domain that leads to a small integration error, is not trivial.

Let $F_n^{(d)}$ is the n-th term of the corresponding generalized Fibonacci sequence of dimensionality d. Therefore, it is a sum of previous d terms from this sequence:

$$F_n^{(d)} = \sum_{i=n-d}^{n-1} F_i^{(d)}, \quad \text{where} \quad n \quad \text{is an integer and} \quad n \geq d$$

and the following initial conditions hold:

$$F_0^{(d)} = F_1^{(d)} = \cdots = F_{d-2}^{(d)} = 0,\ F_{d-1}^{(d)} = 1.$$

Consider the following generating vector [43]:

$$\mathbf{a} = (1, F_n^{(d)}(2), \ldots, F_n^{(d)}(d)), \tag{1}$$

where the following notation is used

$$F_n^{(d)}(j) := F_{n+j-1}^{(d)} - \sum_{i=0}^{j-2} F_{n+i}^{(d)}$$

and $F_{n+l}^{(d)}$ $(l = 0, \ldots, j-1, j$ is an integer , $2 \leq j \leq d)$ is the corresponding term of the generalized Fibonacci sequence of dimensionality d.

One can easily obtain that in this case each component of the generating vector $\mathbf{a}$ is defined by a sum of some terms of the generalized Fibonacci sequence with dimensionality d. For example:

$$F_n^{(d)}(2) = F_{n+1}^{(d)} - F_n^{(d)} = (F_n^{(d)} + F_{n-1}^{(d)} + \cdots + F_{n-d+1}^{(d)}) - F_n^{(d)} = F_{n-1}^{(d)} + \cdots + F_{n-d+1}^{(d)}.$$

Therefore, the generating vector (1) is transformed into:

$$\mathbf{a} = (1, F_{n-1}^{(d)} + F_{n-2}^{(d)} + \cdots + F_{n-d+1}^{(d)}, \ldots, F_{n-1}^{(d)} + F_{n-2}^{(d)}, F_{n-1}^{(d)}).$$

The discrepancy of the corresponding lattice point set with $N = F_n^{(d)}$ points obtained by using the vector described above is asymptotically estimated by the following result proved by Hua and Wang (1981, [16]): The set

$$\left(\left\{\frac{1}{F_n^{(d)}}k\right\}, \left\{\frac{F_n^{(d)}(2)}{F_n^{(d)}}k\right\}, \ldots, \left\{\frac{F_n^{(d)}(d)}{F_n^{(d)}}k\right\}\right), 1 \leq k \leq F_n^{(d)},$$

has discrepancy $D(F_n^{(d)}) = \mathcal{O}\left(F_n^{(d)^{-\frac{1}{2}-\frac{1}{2^{d+1}\cdot\ln 2}-\frac{1}{2^{2d+3}}}}\right)$.

The number of calculations required to obtain the generating vector $\mathbf{a}$ is $\mathcal{O}(\ln N)$ [43], because it is well known that the Fibonacci numbers can be obtained with logarithmic computational complexity [6]. A constant number of operations are necessary for generating a new point. Therefore, to obtain a lattice set consisting of N points, $\mathcal{O}(\ln N)$ number of operations are necessary. As in the case for Plain MC, it can be easily seen that the algorithm based on generalized Fibonacci numbers has linear computational complexity, because we should choose N random points in the domain at the cost of $\mathcal{O}(1)$ operations. The advantage of the algorithm described above is its time efficiency.

2.5 Latin hypercube sampling

Latin hypercube sampling (LHS) is a type of stratified sampling [8]. More information for LHS can be found in [11, 25, 27, 28].

In the case of stratified sampling one must simply divide the domain $[0,1]^d$ into M^d disjoint subdomains, each of volume $\frac{1}{M^d}$, and to sample one point at each subdomain. It is well known that stratified sampling can never result in higher variance than plain random sampling [39]. In the background of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. Note that such configuration is similar to having N rooks on a chess board without threatening each other. We can sample using such configuration in order to make the sampling point distribution close to the target probability density function. The Latin hypercube method generalises this concept to an arbitrary number of dimensions. When sampling from a multivariate distribution, a sample size n from multiple variables is drawn such that for each variable the sample is marginally maximally stratified, i.e. the probability of falling in each of the strata is n^{-1} (McKay et al. [25]).

More specifically, LHS works in the following way:

Given m random variables $X_1, X_2, \ldots, X_m$, we divide the range set of each variable into n equally probable intervals, using it's cumulative distribution function. Then, a random sample is taken at each interval and for each variable. As a result, we have n random values for each variable, that are going to be the coordinates of the final n points in the corresponding dimension. The n values obtained for each of the variables are then paired with each other. This may happen randomly, i.e. by taking a random permutation of $\{1, 2, \ldots, n\}$ for each of the dimensions $l = 2, 3, \ldots, m$. For example, if $m = 2$, then to do the pairing, we will need only one random permutations, a bijection $\sigma : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$, that will tell us that for each $i = 1, 2, \ldots, n$, the value in the interval i for variable

X_1 will be paired with the value in the interval $\sigma(i)$ for variable X_2. Similarly, for $m = 3$ variables(and dimensions), one would need 2 random permutations to determine the pairing of the values sampled, respectively for variables X_2 and X_3 and the other variables.

The described version of the algorithm is usually called "random LHS". We should note that the random LHS does not ensure to fill the input space as evenly as possible. To improve the space-filling property, an efficient way is to adopt some additional criterion for constructing LHS. These methods are called optimized LHS methods (OLHS). An example of a criterion that could be used is the maximin criterion which aims to maximize the minimal distance among sample points. In general, the space-filling criteria can be classified into three categories: distance criteria, uniformity criteria and the entropy criterion. For more details on OLHS, the interested reader may see the relatively recent work of Li et al. ([22]). In this paper, for the purpose of our numerical comparisons, we stick to the random LHS algorithm, that was already explained above.

The formal description of the LHS algorithm is the following:

- Divide the distribution of each variable X into n equiprobable intervals.
- For each interval, choose one uniform random number $r \in [0, 1]$. Then, let
$$\text{Prob}_i = \frac{i-1}{n} + r * \frac{1}{n}$$
- Transform the probability into the sampled value x using the inverse of the cumulative distribution function:
$$x = F^{-1}(\text{Prob})$$
- The n values obtained for each variable X are paired randomly(e.g. by generating a random permutation of $\{1, 2, \ldots, n\}$ for each variable except one) with the n values of the other variables.

Examples of random, stratified and Latin hypercube samplings with 16 points are presented on Fig. 1 [17].

Since LHS is a type of stratified sampling, it directly leads that it has linear computational complexity, because we are doing the same number of operations as the Plain MC approach. Moreover, the computational time for the two approaches are very similar as can be seen in the next section.

3 Wigner kernel

The necessity of a quantum theory was raised by a series of experimental observations with some extremely small objects such as electrons and other elementary particles, atoms and molecules, which could not be explained with the help of classical mechanics. After the work of several other theoretical physicist, E. Schrödinger finally formulated his equation, summarizing the description of quantum systems in terms of probability amplitudes, or wave-functions. He was awarded a Nobel prize for physics in 1933, mainly because of this result of him.

Right after the birth of the Schrödinger equation, other formulations and rules trying to adequately describe quantum mechanics appeared. Here, the work of E. Wigner stands out among these alternative (but completely equivalent) formulations, as an intuitive model which provides a direct connection between classical and quantum physics. Indeed, the theory of Wigner has many similarities with the classical statistical mechanics. The Wigner formalism has been successfully applied to problems related to fields like nuclear physics and comprehension of chemical reactions. The reader is referred to [34] for a more complete list of references and more thorough introduction to the subject. Central for the Wigner's theory is the Wigner equation which represented an incredibly challenging mathematical task for years, being a partial integro-differential equation where the unknown is a function defined over a $2 \times d \times n$ dimensional phase-space, where d is the dimensionality of the space and n is the number of involved particles.

The highly multidimensional Wigner kernel is a major component in the Wigner equation. It can be interpreted as the probability for a particle to jump in the phase-space and have a momentum p; this jump happens discontinously and in discrete amounts equivalent to half the momenta of light quanta, as if the potential were composed of light [34].

A new formulation of quantum mechanics in terms of signed classical field-less particles is presented in [31, 34]. The theory is based on a novel generalized physical interpretation of the mathematical Wigner Monte Carlo method which is able to simulate the time-dependent single- and many-body Wigner equation [34].

The three postulates which completely define the new mathematical formulation of quantum mechanics in terms of signed particles in a single-body context are given below [31] and they are enough to replicate the results of more conventional quantum theories.

Postulate I Physical systems can be described by means of (virtual) Newtonian particles, i.e. provided with a position $\mathbf{x}$ and a momentum $\mathbf{p}$ simultaneously, which carry a sign which can be positive or negative.

Postulate II A signed particle, evolving in a potential $V = V(x)$, behaves as a field-less classical point-particle which, during the time interval dt, creates a new pair of signed particles with a probability $\gamma(\mathbf{x}(t))\mathrm{d}t$, where

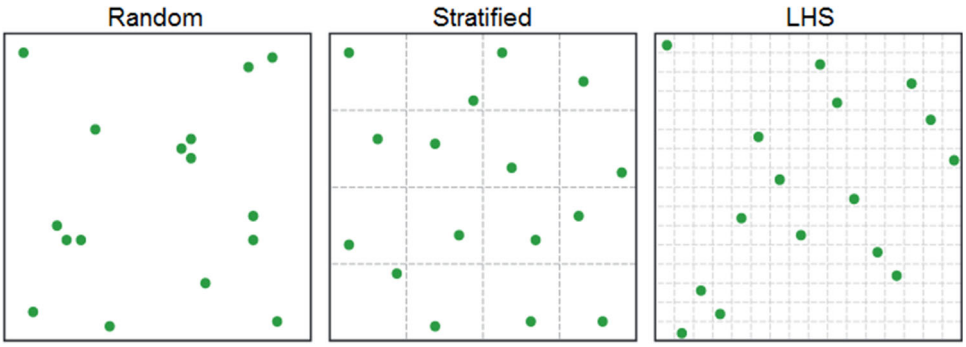

Fig. 1 Comparison of random, stratified ($d = 2, M = 4$) and Latin hypercube samplings with 16 points

$$\gamma(\mathbf{x}) = \int_{-\infty}^{+\infty} \mathrm{D}\mathbf{p}' V_W^+(\mathbf{x};\mathbf{p}') \equiv \lim_{\triangle \mathbf{p}' \to 0^+} \sum_{\mathbf{M}=-\infty}^{+\infty} V_W^+(\mathbf{x};\mathbf{M} \triangle \mathbf{p}'),$$

where $\hbar = \frac{h}{2\pi}$ is the reduced Planck constant (h) or Dirac constant, $\mathbf{M} = (M_1, M_2, \ldots, M_d)$ is a set of d integers and $V_W^+(\mathbf{x};\mathbf{p})$ is the positive part of the quantity

$$V_W(\mathbf{x};\mathbf{p}) = \frac{i}{\pi^d \hbar^{d+1}} \int_{-\infty}^{+\infty} \mathrm{d}\mathbf{x}' e^{-\frac{2i}{\hbar}\mathbf{x}'\mathbf{p}} [V(\mathbf{x}+\mathbf{x}') - V(\mathbf{x}-\mathbf{x}')], \tag{2}$$

known as the Wigner kernel (in a d-dimensional space) [44]. If, at the moment of creation, the parent particle has sign s, position $\mathbf{x}$ and momentum $\mathbf{p}$, the new particles are both located in $\mathbf{x}$, have signs $+s$ and $-s$, and momentum $\mathbf{p}+\mathbf{p}'$ and $\mathbf{p}-\mathbf{p}'$ respectively, with $\mathbf{p}'$ chosen randomly according to the (normalized) probability $\frac{V_W^+(\mathbf{x};\mathbf{p})}{\gamma(\mathbf{x})}$.

Postulate III Two particles with opposite sign and same phase-space coordinates $(\mathbf{x}, \mathbf{p})$ annihilate.

More information about the signed particle formulation of a single-body and many-body system can be found in [32–34].

The calculation of the Wigner kernel is the bottleneck of the Wigner MC algorithm, because it is equivalent to the calculation of a function defined over a space which dimensions increase exponentially with the number of bodies involved [32]. We will show how the new adaptive MC approach for the Wigner kernel introduces crucial advantages in terms of numerical efficiency and computational complexity. Then the problem of simulating time-dependent quantum systems reduces to a problem in which computational complexity now increases linearly with the typical dimensions involved. Such an improvement opens a whole new world of opportunities and enables the simulation of relatively big quantum systems by means of cost-effective computational resources.

4 Numerical experiments

An efficient MC algorithm for numerical integration should be recognized if a smaller relative error is attained for comparable or smaller computational time in comparison with other Monte Carlo algorithms. On the other hand, the main problem here is to compute an unknown value (multidimensional integral) with a preliminary fixed accuracy. This requirement holds for the number of samples N. The computational time necessary for the corresponding MC estimate depends on the number of samples and the function under integration. Therefore, the initial problem can be formulated as approximate computation of multidimensional integrals for a preliminary fixed computational time following the rule for balancing of approximate error and computational complexity.

For all case studies computational time is presented in seconds. In the tables, the notation FIBO is used for the algorithm based on generalized Fibonacci numbers. Also note, that in all tables below the relative error for the Adaptive algorithm is normalized with the ratio *TNP* / *N*, where TNP denotes the total number of points used. For this reason, we have put the name Adaptive* for this method, in all the tables comparing our computational methods.

4.1 Multidimensional integrals from Bayesian statistics

A fundamental problem in Bayesian statistics is the accurate evaluation of multidimensional integrals. Shaowei Lin [23, 24] considers the problem of evaluating multidimensional integrals of two kinds in Bayesian statistics which are used in neural networks and computational biology. The first kind of integrals considered by Lin in [23] has the form

$$\int_{\Omega} p_1^{u_1}(\mathbf{x}) \ldots p_k^{u_k}(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{3}$$

where $\Omega \in \mathcal{R}^d$, $\mathbf{x} = (x_1, \ldots, x_d)$, $p_i(\mathbf{x})$ are polynomials and u_i are integers. The second kind has the form

$$\int_{\Omega} e^{-kf(\mathbf{x})}\phi(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{4}$$

where $f(\mathbf{x})$ and $\phi(\mathbf{x})$ are polynomials and k is a natural number.

We considered three different examples of 5, 15 and 30-dimensional integrals, respectively, where their referent values are given:

Example 1 ($d = 5$):

$$\int_{[0,1]^5} \exp(-100x_1x_2x_3)(\sin(x_4) + \cos(x_5))\,\mathrm{d}\mathbf{x} \approx 0.1854297367. \tag{5}$$

Example 2 ($d = 15$):

$$\int_{[0,1]^{15}} \left(\sum_{i=1}^{10} x_i^2\right)(x_{11} - x_{12}^2 - x_{13}^3 - x_{14}^4 - x_{15}^5)^2\,\mathrm{d}\mathbf{x} \approx 1.96440666. \tag{6}$$

Example 3 ($d = 30$):

$$\int_{[0,1]^{30}} \frac{4x_1x_3^2e^{2x_1x_3}}{(1+x_2+x_4)^2} e^{x_5+\ldots+x_{20}}x_{21}\ldots x_{30}\,\mathrm{d}\mathbf{x} \approx 3.244540. \tag{7}$$

Numerical results about relative errors and computational times are presented in two types of tables: according to a fixed sample size and a fixed computational time. Firstly, numerical results show advantage in terms of relative error for the lattice algorithm based on Fibonacci generalized numbers (in 1–5 orders) in comparison with all the other algorithms under consideration. The advantage is essential towards Plain MC algorithm, the adaptive MC algorithm and the MC algorithm based on LHS for lower dimensions (see Tables 1 and 2, $d = 5, 15$). The results for relative errors corresponding to FIBO and SOBOL are similar especially for higher sample size (see Tables 1 and 2, $d = 15, 30$). One should note that the increase of sample size leads to a significant decrease of the relative error corresponding to FIBO in the case of lower dimensions. For higher dimensions LHS outperforms FIBO and SOBOL (in 1–3 orders, see Tables 1 and 2, $d = 30$). If the computational time is fixed, FIBO gives the smallest relative errors for lower dimensions (see Tables 1 and 2, $d = 5$). Otherwise, the most reliable and also efficient approach is LHS. Its computational time is comparable with the computational times of Plain MC and FIBO, but it leads to smaller relative errors (in 1–2 orders).

4.2 Genz test functions

Numerical experiments with this function family

$$f(\mathbf{x}) = \left(1 + \sum_{i=1}^{d} a_i\, x_i\right)^{-(d+1)} \tag{8}$$

have been performed and the results for various dimensions are presented in Table 3. This class of functions belongs to a test package given by Genz (see [15]). Every distinct function family in the package is characterized by special peculiarity in computational aspect [30]. The set of functions that we are considering have a unique local maximum near one of the corners of the unit hypercube. Some functions describing the change in the concentration of the air pollutants have the same property. The parameters a_i are evaluated using temporarily values a_i', uniformly distributed in $[\frac{1}{20}; 1 - \frac{1}{20}]$, and $\mathbf{a} = c\ \mathbf{a}'$. The constant c is the parameter of the computational complexity, which is chosen [3], so that "the sharpness" of the local maximum to be controlled by the following norm $||\mathbf{a}||_1 = \frac{600}{d^2}$. Two case studies in the current tests have been considered: $d = 5$, $\mathbf{a} = (5, 5, 5, 5, 4)$, $I[f] \approx 2.1200e-06$, and $d = 18$, $\mathbf{a} = \left(\frac{1}{9}, \frac{2}{27}, \frac{2}{27}, \frac{1}{9}, \frac{2}{27}, \frac{1}{9}, \frac{1}{9}, \frac{4}{27}, \frac{2}{27}, \frac{1}{9}, \frac{1}{9}, \frac{2}{27}, \frac{2}{27}, \frac{1}{9}, \frac{1}{9}, \frac{4}{27}, \frac{1}{9}, \frac{1}{9}\right)$, $I[f] \approx 9.9190e-06$.

The numerical results are presented according to a preliminary fixed computational time to be easier to compare the efficiency of the algorithms in terms of relative error. The sample size for each of the algorithms corresponding to the chosen computational time is also given in the tables. One should note that for the adaptive MC algorithm the number of samples in every subdomain is given, but not the total number of samples in the entire integration domain. Here, the sample size corresponding to FIBO algorithm is chosen to be a generalized Fibonacci number of corresponding dimension to provide optimal results for this algorithm.

The numerical results confirm the assertion that the adaptive technique is effective for such class of functions that have computational speciality in the local subarea of the integration domain. The adaptive algorithm leads to smaller relative errors (even in 1–3 orders) for a fixed computational time in comparison to other stochastic algorithms under consideration. The advantage of the adaptive MC algorithm is essential and it holds for both cases for different dimensions here.

4.3 Wigner kernel

Numerical results obtained via a mid rectangular deterministic method, the Plain MC algorithm, Sobol QMC

Table 1 Relative error and computational time for multidimensional integrals from Bayesian statistics according to sample size

d	*N*	Plain		Adaptive*		FIBO		SOBOL		LHS	
		Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
5	10^3	2e-2	0.007	1e-2	0.27	2e-4	0.007	5e-4	0.0083	9e-3	0.007
	10^4	5e-3	0.07	4e-3	2.43	1e-5	0.06	1e-4	0.082	3e-3	0.07
	10^5	1e-3	0.64	1e-3	22.2	9e-6	0.61	2e-5	0.78	2e-3	0.69
	10^6	8e-4	6.06	5e-4	220	5e-7	5.98	6e-6	7.17	2e-4	6.17
	10^7	2e-4	60	1e-4	2043	9e-9	58.4	2e-6	71.39	2e-5	61
15	10^3	6e-2	0.09	4e-2	8.24	5e-2	0.08	2e-3	0.126	1e-2	0.12
	10^4	4e-2	0.95	9e-3	68	1e-3	0.93	3e-4	1.41	7e-3	1.07
	10^5	3e-2	9.7	3e-3	547	1e-4	9.7	1e-5	13.94	2e-4	10
	10^6	7e-3	96	4e-4	5235	6e-6	97	6e-6	136.9	1e-5	100
30	10^3	9e-1	0.02	7e-1	2.27	9e-1	0.02	1e-1	0.03	2e-2	0.02
	10^4	7e-1	0.1	3e-2	20.1	1e-2	0.18	9e-2	0.16	7e-3	0.19
	10^5	4e-1	1.12	2e-2	229	3e-2	1.56	2e-2	1.74	2e-3	1.54
	10^6	2e-1	11	1e-2	2271	1e-2	13.61	9e-3	17.31	1e-4	14

Table 2 Relative error for multidimensional integrals from Bayesian statistics according to computational time

d	Time	Plain	Adaptive*	FIBO	SOBOL	LHS
5	0.1	3.16e-3	3.48e-3	1.09e-5	1.34e-4	3.21e-3
	1	1.08e-3	2.08e-3	5.58e-6	7.21e-5	8.54e-4
	5	8.79e-4	8.20e-4	8.71e-7	1.54e-5	3.25e-4
	10	5.85e-4	7.51e-4	4.15e-7	9.32e-6	8.65e-5
	20	3.99e-4	6.95e-4	8.37e-8	7.39e-6	5.02e-5
15	1	4.16e-2	6.30e-2	1.10e-3	2.04e-3	7.89e-3
	5	3.72e-2	1.68e-2	2.45e-4	7.32e-4	6.78e-4
	10	2.33e-2	2.89e-3	9.48e-5	1.94e-4	1.64e-4
	20	1.03e-2	1.66e-3	9.87e-6	4.05e-5	5.67e-5
30	1	5.01e-1	2.32e-1	2.38e-2	1.01e-1	4.38e-3
	5	3.21e-1	9.21e-2	1.81e-2	7.76e-2	8.16e-4
	10	1.13e-1	7.05e-2	9.48e-3	5.71e-2	3.11e-4
	20	9.06e-2	6.99e-2	7.87e-3	1.28e-2	8.63e-5

Table 3 Relative error and computational time for Genz test functions

d	Adaptive*			LHS			FIBO		
	N	Error	Time	*N*	Error	Time	*N*	Error	Time
5	10^2	5.12e-02	0.33	10^5	7.23e-02	0.27	1546352	9.71e-02	0.38
	10^3	1.43e-02	1.44	10^6	3.25e-02	1.22	3040048	6.75e-02	1.32
	10^4	1.34e-03	11	10^7	2.52e-03	12	11749641	1.54e-02	15
	10^5	9.93e-05	142	10^8	1.66e-03	124	175514464	2.92e-03	135
18	10	7.21e-03	16	10^7	8.64e-03	14	14930352	7.16e-02	15
	10^2	4.35e-03	142	10^8	5.12e-03	140	102334155	5.11e-02	144
	10^3	5.56e-04	1408	10^9	1.63e-03	1354	1134903170	2.89e-02	1344

algorithm and the adaptive MC algorithm are presented in Table 4. The different dimensions of the integrals are defined by the dimensions of the three variables: $\mathbf{x}, \mathbf{x}'$ and $\mathbf{p}$. The infinite domain of integration can be mapped into the d-dimensional unit hypercube using the following transformation $\frac{1}{2}+\frac{1}{\pi}\arctan(x)$ which maps $(-\infty, \infty)$ into (0, 1).

It is well known that Wigner kernel has real values [44]. From the numerical results it can be seen that the stochastic method under consideration outperforms the deterministic method when the dimensionality of the Wigner kernel increases. The adaptive MC approach outperforms the rest of the algorithms under consideration by at least 1–5 orders (see Table 4). It can be seen that the computational time for the adaptive MC approach is better than the deterministic method when the dimensionality increases.

The more interesting part is to compare the adaptive MC algorithm with MC alogirhtms using generalized Fibonacci sequences and LHS. In Table 5 we have shown the relative error of the stochastic algorithms for preliminary given number of samples. In Table 6 we report the relative error of the algorithms for preliminary given computational time for numerical integration. Numerical results show that the Fibonacci lattice algorithm and Sobol algorithm produce similar results for the same number of samples, but FIBO is significantly faster. LHS is better than Fibonacci lattice, because for the same time it produces lower relative errors with 1–2 orders. This is the first time the adaptive MC algorithm is applied to a single-body problem in multidimensional case. It gives superior results in comparison with the other stochastic algorithms under consideration. To acquire a better intuition about the Wigner kernel we have plotted it as a function of $\mathbf{x}$ and $\mathbf{p}$ in one dimensional case

Table 4 Relative error for the Wigner kernel with deterministic and stochastic algorithms

d	N	determ.		SOBOL		Plain		Adaptive*	
		Error	Time	Error	Time	Error	Time	Error	Time
3	$32^2 \times 50$	8.5e-03	0.2	8.5e-03	0.04	8.4e-03	0.003	2.7e-03	0.6
	$32^2 \times 100$	8.2e-03	0.5	8.1e-03	0.01	8.2e-03	0.008	3.4e-04	1.1
	$64^2 \times 50$	5.8e-03	1	5.3e-03	0.13	5.1e-03	0.1	7.5e-05	2.3
	$64^2 \times 100$	4.9e-03	1.9	4.6e-03	0.37	4.8e-03	0.3	1.2e-05	4.5
6	$8^4 \times 25^2$	7.1e-02	11	2.3e-03	1.75	5.3e-03	1.1	3.3e-04	10.1
	$8^4 \times 50^2$	1.2e-02	41	8.7e-04	6.07	9.1e-04	4	1.2e-04	38
	$8^4 \times 100^2$	9.8e-03	161	5.7e-04	29.66	6.4e-04	18	1.5e-05	148
9	$6^6 \times 16^3$	1.8e-03	836	5.5e-04	73.	9.4e-04	47	7.6e-05	501
	$6^6 \times 32^3$	1.4e-03	5544	2.4e-04	323.09	8.3e-04	225	2.7e-05	3980
	$6^6 \times 40^3$	1.1e-03	10684	1.1e-04	722.1	5.1e-04	492	8.1e-06	7501

Table 5 The relative error for 3, 6 and 9-dimensional integral with different stochastic methods

d	N	LHS		FIBO		SOBOL		Adaptive*	
		Error	Time	Error	Time	Error	Time	Error	Time
3	10^3	4.38e-03	0.01	3.72e-02	0.02	1.07e-02	0.01	5.36e-03	0.4
	10^4	7.94e-04	0.06	7.06e-03	0.07	8.77e-03	0.07	4.84e-04	3.3
	10^5	2.51e-04	0.41	3.40e-03	0.43	8.57e-04	0.48	2.51e-05	32.6
	10^6	8.20e-05	3.52	1.01e-03	4.4	6.73e-04	4.11	1.76e-05	302
6	10^3	1.54e-03	0.01	7.82e-03	0.01	2.42e-02	0.01	6.72e-03	0.5
	10^4	6.34e-04	0.06	5.01e-03	0.07	5.02e-03	0.08	9,10e-04	4.1
	10^5	4.22e-04	0.44	6.88e-03	0.43	4.60e-04	0.57	5.26e-05	37
	10^6	8.57e-05	3.7	7.68e-04	5.97	3.59e-04	4.89	2.70e-06	351
9	10^3	6.11e-03	0.04	2.03e-02	0.06	5.42e-02	0.05	4.92e-02	0.5
	10^4	1.02e-03	0.06	2.02e-03	0.07	6.02e-03	0.08	9.09e-04	4.7
	10^5	4.69e-04	0.43	9.16e-04	0.53	3.57e-03	0.55	3.32e-05	40
	10^6	8.08e-05	3.8	7.13e-04	3.7	8.02e-04	4.9	6.46e-06	381

after integrating on $\mathbf{x}'$ using the algorithms mentioned above (see Fig. 2). The behavior of the Wigner kernel explains the advantage and the efficiency of the adaptive MC algorithm for this case study.

Table 6 Relative errors for the 3, 6 and 9 dimensional integrals

d	Time	LHS	FIBO	SOBOL	Adaptive*
3	1	1.14e-04	5.42e-03	7.27e-03	8.73e-04
	10	7.21e-05	2.11e-03	7.83e-04	5.62e-05
	100	9.11e-06	9.50e-04	2.18e-04	3.43e-06
6	1	1.14e-04	9.25e-04	3.31e-03	4.51e-04
	10	6.15e-05	5.11e-04	9.34e-04	2.57e-05
	100	1.65e-05	1.05e-04	1.27e-04	2.72e-06
9	1	6.54e-04	8.72e-04	5.59e-03	1.23e-03
	10	5.41e-05	6.51e-04	5.84e-03	3.82e-05
	100	9.22e-06	3.70e-04	6.39e-04	3.09e-06

5 Conclusions

A deterministic (mid rectangular rule) and various stochastic algorithms (Plain MC approach with pseudo-random sequences, Sobol QMC algorithm, adaptive MC algorithm, a particular rank-1 lattice rule based on generalized Fibonacci numbers, a MC algorithm based on Latin hypercube sampling) are presented and discussed. A number of numerical tests with integrands of various dimensions and various applied fields have been performed. We consider multidimensional integrals from Bayesian statistics and Genz test functions of various dimensions as case studies. The efficiency of an adaptive Monte Carlo algorithm for the calculation of the Wigner kernel from the Wigner equation in the case of a single-body problem in a presence of a single barrier is also studied. Adaptive approach appears to be an efficient stochastic solution to solve this kind of problem, because of its established reliability for calculating integrals depending on integrands with peculiarities (such as local and

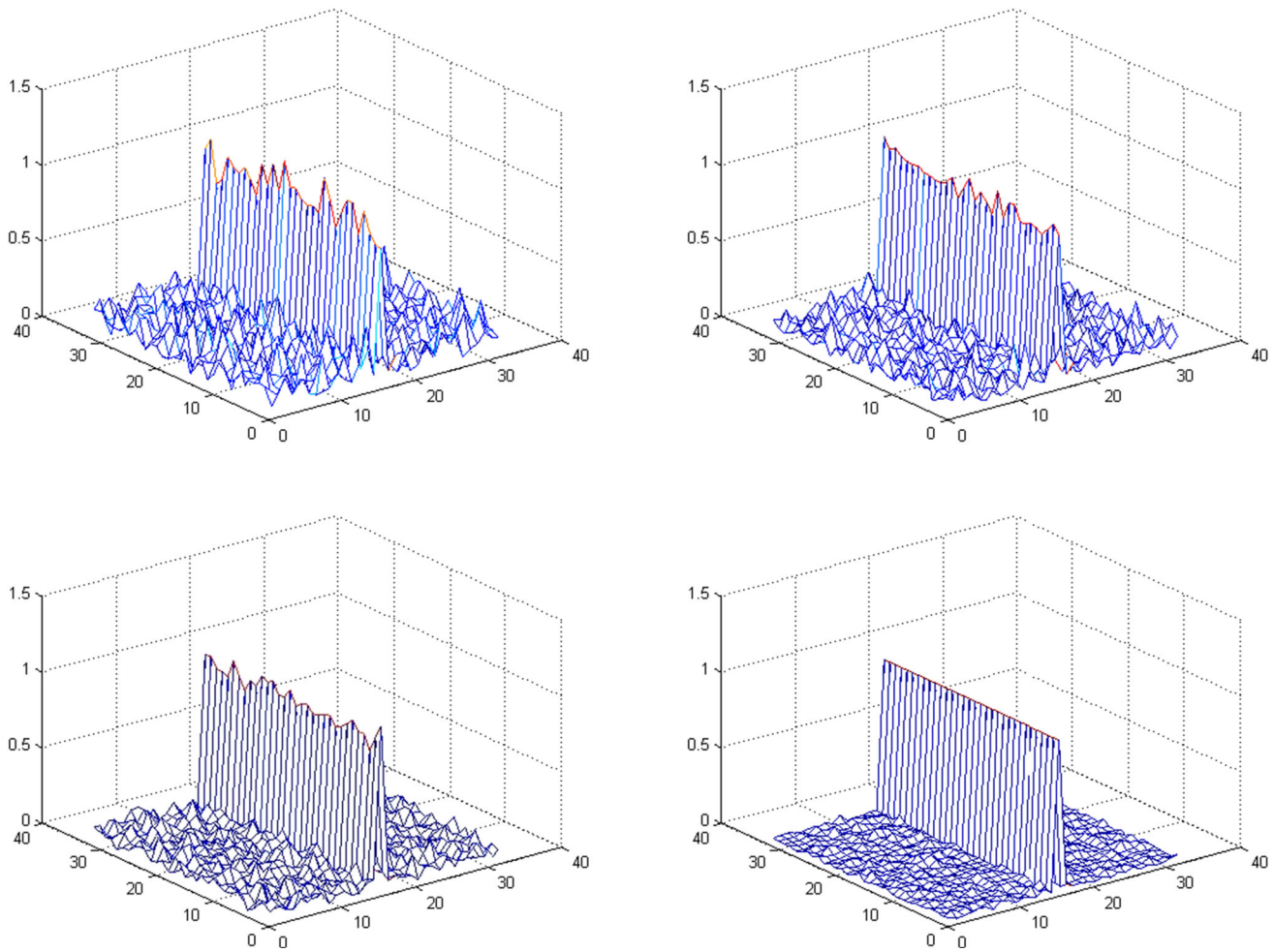

Fig. 2 Wigner kernel $V = V_W(\mathbf{x}, \mathbf{p})$ obtained with deterministic method, FIBO (first row), LHS and Adaptive Monte Carlo (second row) with 10^4 points

global maxima) in a relatively small subregion of the original integration domain. We should note here, that some Bayesian integration methods established by Larkin in [20, 21] could also be a relevant computational approach to the considered tasks and might give comparable results. Those methods recently regained interest(see Hagan [29] and Briol et al.([5])) and a new comparison of these methods with the currently considered methods could be an object of a further investigation.

The numerical results including relative errors and computational times corresponding to the algorithms under consideration are presented, and the algorithms efficiency is discussed. For smooth integrands without computational peculiarities Fibonacci lattice algorithm gives better results for lower dimensions, but it requires more random points when the dimensionality increases. The MC algorithm based on Latin hypercube sampling has higher accuracy for this case study with increasing the dimensionality of the integral. The adaptive MC algorithm is slower, but it requires smaller number of random points to achieve better accuracy even for higher dimensions. The efficiency of the adaptive MC algorithm is clearly shown in the second and the third case studies where the integrands have computational specialty in the local subarea of the integration domain. Five stochastic algorithms have been applied to Wigner kernel case study. The studied algorithms have linear or significantly lower computational complexity in comparison to deterministic approaches utilized for such kind of problems. It is of great importance for the problems in quantum mechanics with high dimensions. The approximate calculation of Wigner kernel with the presented adaptive Monte Carlo algorithm has a significant advantage over previously applied deterministic approaches to its computational complexity that is demonstrated especially in larger dimensions. We should emphasize here that the efficiency of the adaptive MC algorithm under consideration is high when computational peculiarities of the integrand occur only in comparatively small subregion of the initial integration domain. The adaptive MC algorithm under consideration has significantly lower computational complexity than existing deterministic approaches for multidimensional numerical integration. This advantage is demonstrated especially for higher dimensions. Therefore, the presented adaptive MC algorithm is one possible successful solution (in terms of robustness and reliability) of Richard Feynman's problem for Wigner kernel evaluation.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Antonov I, Saleev V (1979) An economic method of computing LP_τ-sequences. USSR Comput Math Phys 19:252–256
2. Baraniuk RG, Jones DL (1993) A signal-dependent time-frequency representation: optimal kernel design. IEEE Trans Signal Process 41(4):1589–1602
3. Berntsen J, Espelid TO, Genz A (1991) An adaptive algorithm for the approximate calculation of multiple integrals, ACM Trans. Math Softw 17:437–451
4. Bratley P, Fox B (1988) Algorithm 659: implementing Sobol's Quasirandom sequence generator. ACM Trans Math Softw 14(1):88–100
5. Briol F-X, Oates CJ, Girolami M, Osborne MA, Sejdinovic D (2019) Probabilistic integration: a role in statistical computation? Stat Sci 34(1):1–22
6. Cull P, Holloway JL (1989) Computing Fibonacci numbers quickly. Inf Process Lett 32(3):143–149
7. Davis PJ, Rabinowitz P (1984) Methods of numerical integration, 2nd edn. Academic Press, London
8. Dimov I (2008) Monte Carlo methods for applied scientists. World Scientific, New Jersey, p 291
9. Dimov I, Karaivanova A, Georgieva R, Ivanovska S, (2003) Parallel Importance Separation and Adaptive Monte Carlo Algorithms for Multiple Integrals, 5th Int. conf. on NMA, (August 2002) Borovets, Bulgaria, Springer Lecture Notes in Computer Science, 2542. Springer-Verlag, Berlin, Heidelberg, New York, pp 99–107
10. Dimov I, Georgieva R (2010) Monte Carlo algorithms for evaluating Sobol' sensitivity indices. Math Comput Simul 81(3):506–514
11. Eglajs V, Audze P (1977) New approach to the design of multifactor experiments: problems of dynamics and strengths. 35 (in Russian). Riga: Zinatne Publishing House pp 104–107
12. Ermakov SM (1985) Monte Carlo methods and mixed problems. Nauka, Moscow
13. Feynman RP (1948) Space-time approach to non-relativistic quantum mechanics. Rev Mod Phys: 20
14. Fox B (1986) Algorithm 647: implementation and relative efficiency of Quasirandom sequence generators. ACM Trans Math Softw 12(4):362–376
15. Genz A (1984) Testing multidimensional integration routines. Tools, Methods and Languages for Scientific and Engineering Computation, pp 81–94
16. Hua LK, Wang Y (1981) Applications of number theory to numerical analysis. Springer, Berlin
17. Jarosz W (2008) Efficient Monte Carlo methods for light transport in scattering media, PhD dissertation, UCSD
18. Joe S, Kuo F (2003) Remark on algorithm 659: implementing Sobol's Quasirandom sequence generator. ACM Trans Math Softw 29(1):49–57
19. Karaivanova A, Dimov I, Ivanovska S (2001) A quasi-monte carlo method for integration with improved convergence. LNCS 2179:158–165
20. Larkin FM (1972) Gaussian measure in Hilbert space and applications in numerical analysis. Rocky Mt J Math 2(3):379–421
21. Larkin FM (1974) Probabilistic error estimates in spline interpolation and quadrature. In: IFIP Congress, pp 605–609
22. Li Wei, Lingyun Lu, Xie Xiaotian, Yang Ming (2017) A novel extension algorithm for optimized Latin hypercube sampling. J Stat Comput Simul 87(13):2549–2559. https://doi.org/10.1080/00949655.2017.1340475
23. Lin S (2011) Algebraic methods for evaluating integrals in Bayesian statistics, Ph.D. dissertation, UC Berkeley

24. Lin S, Sturmfels B, Xu Z (2009) Marginal likelihood integrals for mixtures of independence models. J Mach Learn Res 10:1611–1631
25. McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245. https://doi.org/10.2307/1268522
26. Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44(247):335–341
27. Minasny BB, McBratney AB (2006) A conditioned Latin hypercube method for sampling in the presence of ancillary information. J Comput Geosci Arch 32(9):1378–1388
28. Minasny B, McBratney AB (2010) Conditioned Latin hypercube sampling for calibrating soil sensor data to soil properties, Chapter: Proximal Soil Sensing, Progress in Soil Science pp 111–119
29. O'Hagan A (1991) Bayes–Hermite quadrature. J Stat Plan Inference 29(3):245–260
30. Schurer R (2001) Parallel high-dimensional integration: Quasi-monte carlo versus adaptive cubature rules, Computational Science, ICCS 2001. Springer, Berlin
31. Sellier JM (2015) A signed particle formulation of non-relativistic quantum mechanics. J Comput Phys 297:254–265
32. Sellier JM, Dimov I (2016) On a full Monte Carlo approach to quantum mechanics. Phys A Stat Mech Appl 463:45–62
33. Sellier JM, Dimov I (2014) The many-body Wigner Monte Carlo method for time-dependent ab-initio quantum simulations. J Comput Phys 273:589–597
34. Sellier JM, Nedjalkov M, Dimov I (2015) An introduction to applied quantum mechanics in the Wigner Monte Carlo formalism. Phys Rep 577:1–34
35. Shao S, Lu T, Cai W (2011) Adaptive conservative cell average spectral element methods for transient Wigner equation in quantum transport. Commun Comput Phys 9:711–739
36. Shao S, Sellier JM (2015) Comparison of deterministic and stochastic methods for time-dependent Wigner simulations. J Comput Phys 300:167–185
37. Sloan IH, Kachoyan PJ (1987) Lattice methods for multiple integration: theory, error analysis and examples, SIAM. J Numer Anal 24:116–128
38. Sloan IH, Joe S (1994) Lattice methods for multiple integration. Oxford University Press, Oxford
39. Sobol I (1973) Numerical methods Monte Carlo. Nauka, Moscow
40. Sobol IM (1967) "Distribution of points in a cube and approximate evaluation of integrals". Zh. Vych. Mat. Mat. Fiz. 7: 784–802 (in Russian); U.S.S.R Comput. Maths. Math. Phys. 7:86–112
41. Sobol IM (1989) Quasi-Monte Carlo methods. In: Dimov IT, Sendov BI (eds) International youth workshop on Monte Carlo methods and parallel algorithms. World Scientific, Singapore, pp 75–81
42. Xiong Y, Chen Z, Shao S (2016) An advective-spectral-mixed method for time-dependent many-body Wigner simulations. SIAM J Sci Comput, to appear, [arXiv:1602.08853]
43. Wang Y, FJ Hickernell (2002) An historical overview of lattice point sets
44. Wigner E (1932) On the quantum correction for thermodynamic equilibrium. Phys Rev 40:749

Affiliations

Venelin Todorov[1,2] · Ivan Dimov[2] · Rayna Georgieva[2] · Stoyan Dimitrov[3]

[1] Information Modeling Department, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Block 8, 1113 Sofia, Bulgaria

[2] Department of Parallel Algorithms, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Block 25 A, 1113 Sofia, Bulgaria

[3] Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, 851 S Morgan St, Chicago, IL 60607, USA