On the use of Sobol' sequence for high dimensional simulation*

Emanouil Atanassov and Sofiya Ivanovska

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria
{emanouil,sofia}@parallel.bas.bg

Abstract. When used in simulations, the quasi-Monte Carlo methods utilize specially constructed sequences in order to improve on the respective Monte Carlo methods in terms of accuracy mainly. Their advantage comes from the possibility to devise sequences of numbers that are better distributed in the corresponding high-dimensional unit cube, compared to the randomly sampled points of the typical Monte Carlo method. Perhaps the most widely used family of sequences are the Sobol' sequences, due to their excellent equidistribution properties. These sequences are determined by sets of so-called direction numbers, where researches have significant freedom to tailor the set being used to the problem at hand. The advancements in scientific computing lead to ever increasing dimensionality of the problems under consideration. Due to the increased computational cost of the simulations, the number of trajectories that can be used is limited. In this work we concentrate on optimising the direction numbers of the Sobol' sequences in such situations, when the constructive dimension of the algorithm is relatively high, compared to the number of points of the sequence being used. We propose an algorithm that provides us with such sets of numbers, suitable for a range of problems. We then show how the resulting sequences perform in numerical experiments, compared with other well known sets of direction numbers. The algorithm has been efficiently implemented on servers equipped with powerful GPUs and is applicable for a wide range of problems.

Keywords: quasi-Monte Carlo, high-dimensional simulation, low-discrepancy sequences

1 Introduction

The quasi-Monte Carlo methods are built upon the idea to replace the random numbers, typically produced by pseudorandom number generators, by specially crafted deterministic sequences. Most of the Monte Carlo algorithms can

* This work has been financed in part by a grant from CAF America. We acknowledge the provided access to the e-infrastructure of the Centre for Advanced Computing and Data Processing, with the financial support by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014-2020) and co-financed by the European Union through the European structural and investment funds.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

be thought of multi-dimensional integral approximations by average of random samples

$$\int_{E^s} f(x)\,dx \approx \frac{1}{N}\sum_{i=0}^{N-1} f(x_i),$$

therefore the corresponding quasi-Monte Carlo method would use a special s-dimensional sequence to compute the same integral. In this setting the dimension s is called the "constructive dimension" of the algorithm. One justification for the use of low-discrepancy sequences is the Koksma-Hlawka inequality (see, e.g., [3]), which connects the accuracy of the algorithm with one measure of equidistribution of the sequence, called star-discrepancy. For a fixed dimension s, the best possible order of the star-discrepancy is believed to be $N^{-1}\log^s N$, and sequences that achieve this order are called low-discrepancy sequences. Other measures of the quality of distribution of sequences are also used, for example the diaphony [10] or the dyadic diaphony [1].

The Sobol' sequences are one of the oldest known families of low-discrepancy sequences [9], but they are still the most widely used, partially because of their close relation to the binary number system, but mostly because of their good results in high-dimensional settings, see, e.g., [8]. As an example, the price of a financial asset can be modelled by sampling its trajectories along time, where each time step requires one or more random numbers in order to advance. This means that the constructive dimension of the algorithm is a multiple of the number of time steps. Even though the dimension rises fast in such situations, the Sobol' sequences retain their advantage compared to Monte Carlo. Other notions of dimension have been proposed, in order to explain and quantify such effects, e.g., the effective dimension or the average dimension, see [4]. Although these quantities can explain the observed advantage of a quasi-Monte Carlo method based on the Sobol' sequences, they do not change the necessary dimension.

Although initially the low-discrepancy sequences used in quasi-Monte Carlo methods were fully deterministic, there are many theoretical and practical reasons to add some randomness to a quasi-Monte Carlo algorithm. Such a procedure, which modifies a low-discrepancy sequence in a random way, while retaining its equidistribution properties, is usually called "scrambling". In this work we are going to use only the simplest and computationally inexpensive procedure of Matoušek [5], which is also one of the most widely used and provides equal grounds for comparison. Essentially, it consist of performing a bitwise **xor** operation with a fixed random vector for each dimension.

Even though there are more complex scrambling schemes, e.g., the scrambling proposed by Owen in [6], they do not solve one important problem that arises when the constructive dimension of the algorithm becomes high compared with the number of points. Consider a problem where the constructive dimension is s and the number of points is $N = 2^n$ and $s > N/2$. The first bit of the term of the Sobol' sequence in each dimension is determined by the index and the first column of the matrix of direction numbers $A_i = a^i_{jk}$. Since we have s such columns and the index is between 0 and $2^s - 1$ inclusively, only the first n

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

bits of the column are important. Moreover, the matrix of direction numbers is triangular with ones over the main diagonal, so we have only $n-1$ bits that can take values of either 0 or 1, or 2^{n-1} possibilities. Therefore the Dirichlet principle ensures that if $s > N/2$, there will be two dimensions where the first bits of the sequence will be exactly equal. Even if we add scrambling, the situation does not improve, because the bits will be either exactly equal for all terms of the sequence, or exactly inverted. Thus the correlation between these two dimensions will be much higher in absolute value, compared with Monte Carlo sampling. If the algorithm uses only the first bit of the terms of the sequence, e.g., for making some binary choice, choosing subsets, etc., then such correlation is obviously problematic. There are also many algorithms where a discrete value is to be sampled. In such case, the subsequent bits reveal even worse problems, since the available choices become even less. Thus, even if $s < N/2$, we can still have dimensions where certain bits always coincide in the original Sobol' sequence. Owen scrambling would alleviate such a problem, but only to an extent. We should also mention that following the typical definition of direction numbers related to primitive polynomials, the available choices are even more restricted, especially in the first dimensions.

In the next section we describe our approach to quantify this problem and define measures that have to be optimised. Then we describe our optimisation algorithm. In the numerical experiments we show how these sets perform on typical integration problems.

2 Optimisation framework

We define measures that should lead to optimal directional numbers in view of the problems described in the previous section. Because of computational costs, we try to obtain numbers that are reusable in many settings. Thus we to cover all dimensions up to some maximal dimension and number of points in a range from 2^m up to and including 2^n. We define several stages. The first stage performs filtering. Since we must avoid coincidences in the columns of the matrices A_i, for a given column of order j in dimension i, we define a measure of coincidences as follows:

$$P(a) = \sum_{k=m}^{n} \sum_{r=1}^{i-1} 2^k c_k(a, a_j^r),$$

where $c_k(a, a_j^r)$ is one if the columns a and $a_j^r = \{a_{ij}^r\}$ coincide in their upper k bits, otherwise zero.

We are going to search for direction numbers column by column, filtering the possible choices to only those, which attain minimum of the function above. Optionally, we can restrict those choices to only columns that follow the usual construction of the Sobol' sequences using primitive polynomials. This seems to improve the results for lower number of dimensions, but is very restrictive when dimensions are higher than 64, so in our experiments we are not using this option. In the next stage we minimise a measure of the equidistribution

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

properties of the sequence, related to Walsh functions. It is very similar to the notion of the diadic diaphony, adjusted to be easily computable for the Sobol' sequence. Consider

$$R(\sigma, n) = \sum_{(m_1,\ldots,m_s)\in L} w(m_1, \ldots, m_s)\, r(\sigma, n; m_1, \ldots, m_s),$$

where $w(m_1, \ldots, m_s) = \prod_{i=1}^{s} 2^{-\alpha k_i}$ if $2^{k_i-1}\ m_i \le 2^{k_i}$. The weight of $m_i = 0$ is taken to be 1. The set L consists of those integer tuples $(m_1, \ldots, m_s)$, that have at least one non-zero value and $0 \le m_i < 2^n$. The quantity $r(\sigma, n, m_1, \ldots, m_s)$ depends on the Sobol sequence and the multidimensional Walsh function corresponding to $(m_1, \ldots, m_s)$ and is 0 or 1 depending whether the integral computes exactly to 0 when using the sequence σ with 2^n terms or not. We obtained good results when limiting the set L to only contain tuples with up to 4 non-zero integers, since quasi-Monte Carlo methods have difficulty outperforming Monte Carlo when the effective dimension is higher than 4.

When the number of dimensions s gets higher, the value of this measure grows, especially if the power α is lower. Rounding errors accumulate too. Thus for reasonable values of the dimension s and range of number of points between 2^m and 2^n, we obtain multiple values of the column of direction numbers with measures that either equal or close. Thus the outcome of the optimisation is usually a set of acceptable numbers for the given column. In order to cover multiple possibilities for the number of points we take the measure R computed for different values of the number of points, between m and n, weighted by the number of points used. We usually set $\alpha = 1$, although the choice $\alpha = 2$ is also logical due to the connection with the diadic diaphony. With $\alpha = 1$ we get reproducible results, while choices like $\alpha = 1.5$ may produce different results for different runs due to accumulation of rounding errors and non-deterministic order of summation when computing in parallel. The last stage is a validation stage, where we test the obtained direction numbers on a suitably chosen integral and select those that gave the best results. As these computations are expensive, we do not test all possibilities. Our approach to the validation stage is explained more in details in the next section. The outcome of the optimisation is a set of direction numbers that can be used for all dimensions less than some maximal dimension and for number of points that is less than some fixed power-of-two.

3 Optimisation algorithm

We proceed to compute direction numbers column by column, initially computing all first columns dimension by dimension and then proceeding to the second columns and so on. In this way the choice of a column from one dimension is only dependent on the columns that are at the same position or ahead in previous dimensions. The filtering stage is straightforward - computing the measure F for each possible value of the column. For a column at position $j \ge 1$ in the dimension $i \ge 1$ we have 2^{n-j} possibilities since the main diagonal consists of

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

ones only. We form an array M of all columns for which F attains its minimum and proceed to the next stage, where we compute the value of R for all columns $x \in M$, selected at the previous stage. If the value of n is too high and this stage becomes prohibitively expensive, we can select random subset of the array of possible columns. In order to take into account possible rounding errors, we select those values for the column, for which $R(\sigma) \leq (1+\varepsilon) \min R(\sigma)$, forming a set Q, and proceed to the next stage. As we only know the first few columns, we only compute terms in R that correspond to these columns, i.e., m_i have only as many bits as is the number of columns. This computation, when done on GPUs, is relatively fast, provided we maintain an array of weights and use dinamic programming. The set Q can be large, but we choose randomly only a small subset for validation. In our experiments we used up to 10 values. For these values of the column we compute the etalon integral of

$$f(x) = \left(\frac{1}{\sqrt{s}} \sum_{i=1}^{s} (2x_i - 1) \right)^2$$

which quantifies the amount of correlation between dimensions. We select the column that produces the best result (performing multiple computations, e.g., 40, randomly filling the yet unfilled positions in the binary matrices). The validation step allows to obtain direction numbers with smoother behaviour.

4 Numerical results

We tested direction numbers obtained by using our algorithm in comparison with the direction numbers, provided at [7], using criterion D_7, following [2]. In all cases we consider integration problems in a high-dimensional unit cube E^s and we apply the Matoušek scrambling, obtaining 100 different estimates for the integral, which assures us that the RMS error computed is representative. The first subintegral function is used regularly for such kinds of tests:

$$f_1(x) = \left(\frac{1}{\sqrt{s}} \sum_{i=1}^{s} \Phi^{-1}(x_i) \right)^2,$$

where Φ^{-1} is the inverse of the c.d.f. for the normal distribution,

$$f_2(x) = \max \left(0, S \exp \left(\frac{1}{\sqrt{s}} \sum_{i=1}^{s} \Phi^{-1}(x_i) - \frac{\sigma^2}{2} \right) - K \right).$$

The function f_2 corresponds to an approximation of the value of an European call option, such that the exact value of the integral can be computed from the Black-Scholes formula. We used $S = 1, K = 1.05, T = 1, \sigma = 0.10, r = 0$. In order to better compare the results across the dimensions, we always normalise by dividing by the exact value of the integral. On Fig. 1-4 we see how the direction numbers produced by the algorithm compare with the fixed direction

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

numbers. The number of points of the sequence is 2^{10} and 2^{12} and the maximum number of dimensions is 1024 or 16384 respectively. One can see how the accuracy evolves with the increase in the constructive dimension. Our algorithm seems to produce suboptimal results for small dimensions, up to 64. However, it starts to outperform when the number of dimensions becomes larger. It is possible to use fixed direction numbers for the first dimensions and extend the set following the algorithm, but we observe worse results for the larger dimensions in this way. For brevity we do not present results for other integrals, but our experience is that for such integrals where the constructive dimension is high and there are many interactions between dimensions without clear domination of a small number of variables the direction numbers produced by our algorithm are promising. This is a typical situation for all the integrals that we have tested. When the number of dimensions becomes bigger than 64, the new direction numbers show clear advantage.

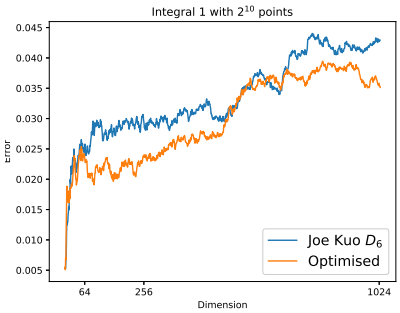

Fig. 1. Estimate the integral 1 for different dimensions using 2^{10} points

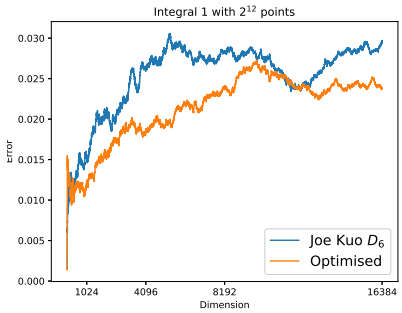

Fig. 2. Estimate the integral 1 for different dimensions using 2^{12} points

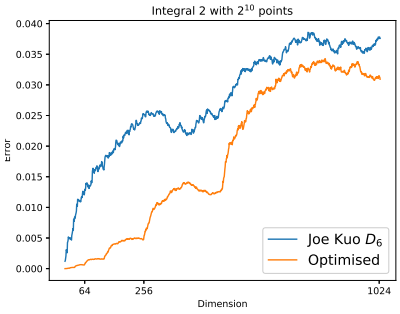

Fig. 3. Estimate the integral 2 for different dimensions using 2^{10} points

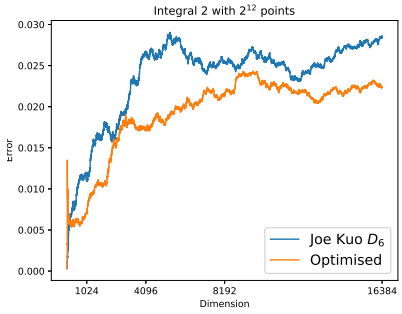

Fig. 4. Estimate the integral 2 for different dimensions using 2^{12} points

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53

The algorithm yields results relatively fast for small values of the number of points, like 2^{10} or 2^{12}. For larger values of the number of points, the number of computations should be reduced to make it run faster. This can be accomplished, e.g., by first selecting a suitable set of direction numbers for smaller number of points and then extending it, thus drastically limiting the number of possibilities to consider. Validation can also be limited or even skipped entirely if needed.

5 Conclusions and directions for future work

Our algorithm produces direction numbers that are suitable for use in algorithms with high constructive dimension, especially in situations where the number of dimensions are comparable with the number of trajectories/points. When the same computations are to be performed using different parameters, it is feasible to compute a set of direction numbers specific for the problem, especially when servers with powerful GPUs are available. We aim to make publicly available sets of direction numbers that are optimised and validated for some combinations of dimensions and number of points that are widely used. In this work we did not explore the possibility to set different weights for the different dimensions, thus allowing for, e.g., declining importance of the dimensions, but our framework is well suited to that problem and we hope that even better results will be obtained for simulations such.

References

1. Hellekalek, P., & Leeb, H.: Dyadic diaphony. Acta Arithmetica, 80, 187–196 (1997)
2. Joe, S., & Kuo, F.Y.: Constructing Sobol Sequences with Better Two-Dimensional Projections. SIAM J. Sci. Comput., 30, 2635–2654 (2008)
3. Kuipers, L. and Niederreiter, H.: Uniform Distribution of Sequences. John Wiley (reprint edition published by Dover Publications, Inc., Mineola, New York in 2006) (1974)
4. Liu, R., & Owen, A. B.: Estimating Mean Dimensionality of Analysis of Variance Decompositions. Journal of the American Statistical Association, 101(474), 712–721 (2006). http://www.jstor.org/stable/27590729
5. Matousek, J.: On the L2-Discrepancy for Anchored Boxes. J. Complex., 14, 527–556 (1998)
6. Owen, A.B.: Scrambling Sobol' and Niederreiter-Xing Points. J. Complex., 14, 466–489 (1998)
7. Sobol sequence generator, `https://web.maths.unsw.edu.au/~fkuo/sobol/`. Last accessed 15 Feb 2022
8. Sobol, I.M., Asotsky, D.I., Kreinin, A., & Kucherenko, S.S.: Construction and Comparison of High-Dimensional Sobol' Generators. Wilmott, 2011, 64–79 (2011)
9. Sobol, I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. Ussr Computational Mathematics and Mathematical Physics, 7, 86–112 (1967)
10. Zinterhof, P.: Uber einige Abschätzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden. Sitzungsber. Osterr. Akad. Wiss. Math.-Natur. Kl. II, 185, 121–132 (1976)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_53