Quasi-Monte Carlo Algorithms for Diffusion Equations in High Dimensions

G. Venkiteswaran [a,*], M. Junk [b]

[a] *Graduiertenkolleg Mathematik und Praxis, Universität Kaiserslautern*
67653 Kaiserslautern, Germany

[b] *Fachbereich Mathematik, Universität des Saarlandes*
66041 Saarbrücken, Germany

Abstract

Diffusion equation posed on a high dimensional space may occur as a sub-problem in advection-diffusion problems (see [1] for a specific application). Although the transport part can be dealt with the method of characteristics, the efficient simulation of diffusion in high dimensions is a challenging task. The traditional Monte Carlo method (MC) applied to diffusion problems converges and is $N^{-1/2}$ accurate, where N is the number of particles. It is well known that for integration, Quasi-Monte Carlo (QMC) outperforms Monte Carlo in the sense that one can achieve N^{-1} convergence, up to a logarithmic factor. This is our starting point to develop methods based on Lécot's approach [2], which are applicable in high dimensions, with a hope to achieve better speed of convergence. Through a number of numerical experiments we observe that some of the QMC methods not only generalize to high dimensions but also show faster convergence in the results and thus slightly outperform standard MC.

Key words: QMC, diffusion equation, MC

1 Introduction

In a variety of physical applications, there is a need to simulate plain diffusion problems posed on high dimensional spaces. Generally, diffusion problems are encountered as sub-problems while solving more complicated ones. Refer [1]

* Corresponding author.
Email addresses: gvenki@mathematik.uni-kl.de (G. Venkiteswaran), junk@num.uni-sb.de (M. Junk).

Preprint submitted to Elsevier Science

for one such occurance, where it is of interest to simulate plain diffusion in order to carry out the splitting method for a high-dimensional Fokker-Planck equation. By high dimensions we mean dimensions of the order of 50.

Traditional methods like finite differences etc., are not applicable in high dimension since the number of node points needed to achieve a prescribed accuracy grows exponentially with the dimension. This is referred to as the **curse of dimension** [3]. The famous approach which breaks this curse is the Monte Carlo method where the error estimate in terms of the number of nodes N goes like $1/\sqrt{N}$, independent of the dimension. Other methods which are known to avoid the curse of dimension at least in high-dimensional integration problems are quasi-Monte Carlo (QMC) methods [4] and sparse grid method [5].

Following the quasi-Monte Carlo approach, the idea is to replace Monte Carlo points by well determined sequences (quasi-random points) which are better uniformly distributed than the former. Though this change works for plain integration and order close to $1/N$ can be achieved, it cannot be simply applied to particle simulations (see [6]). This is due to the correlation among the quasi-random points. This problem was first studied by Lécot [7], for the spatially homogeneous Boltzmann equation and he gave a convergence proof when the quasi-random points were used. The idea of Lécot was to renumber the particle positions at each time step to break the correlations. Morokoff and Caflisch [6], applied this technique to solve the heat equation in one and two dimensions and they obtained significant improvement over the standard Monte Carlo approach. However, for the high dimensional case, the idea of reordering was not clear.

Lécot [2], introduced a sorting algorithm which was adaptable to higher dimensions and also shuffled the particle positions at each time step. The sorting was done with respect to each coordinate of the particle position and convergence was proved for any dimension s. For the simple diffusion problem, there was some improvement achieved over the standard Monte Carlo method *only in one and two dimensions*. The method however had a drawback, namely, the particle numbers were drastically increasing. For a problem in s dimensions, a Faure generator of base b, the least prime $\geq 2s$, was taken. The minimal particle number was then b^s if sorting is to be done in each coordinate. To be concrete, for the case $s = 10$, the base b is 23 and the minimal particle number is of the order $23^{10} (\approx 10^{13})$ which leads to an enormous memory requirement.

Thus there is a quest for a QMC method which would be applicable in real high dimensions and hopefully to have a better convergence order than MC. In this article, we present modifications of the algorithm presented in [2], which work with low particle numbers and thus can be applied in high dimensions. One such algorithm is to reorder the particle position only with respect to the first coordinate which allows us to work with an s dimensional

sequence instead of an $2s$ dimensional one and get a low minimum particle number. In this case, the possible particle numbers that can be considered for the case $s = 10$ are $11, 11^2, 11^3, \ldots$. Even for $s = 100$, the possible particle numbers $101, 101^2, 101^3, \ldots$ are still reasonable. Apart from the drastically reduced memory requirement, the modified ordering has the advantage of increased speed and in total the proposed algorithm slightly outperforms standard Monte Carlo for the diffusion equation.

The paper is organized as follows. Section 2 outlines the various methods in detail followed by section 3 on numerical results.

2 The methods

We are interested in the numerical simulation of

$$\frac{\partial \psi}{\partial t} = \Delta \psi, \quad \psi(\boldsymbol{Q}, 0) = \psi_0(\boldsymbol{Q}) \tag{1}$$

Since (1) is posed on a very high dimensional space, we are using particle methods for the numerical approximation. The basic idea in this approach is to relate the unknown function ψ to the measure $\psi d\boldsymbol{Q}$ and to approximate $\psi d\boldsymbol{Q}$ by a discrete measure,

$$\psi d\boldsymbol{Q} \sim \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{Q}^{(i)}}. \tag{2}$$

The quality of such an approximation is measured using a quantity called the discrepancy (refer [4] for details). The idea is to compare both measures on a sufficiently large collection of sets and to take the maximal difference to quantify the approximation error. Specifically, if $\boldsymbol{X}$ is a point set consisting of $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(N)}$, as in (2), then its discrepancy with respect to a set family $\mathcal{B}$ is

$$D_{\mathcal{B}}(\boldsymbol{X}, \psi) = \sup_{B \in \mathcal{B}} \left| \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{Q}^{(i)}}(B) - \int_B \psi(\boldsymbol{Q}) d\boldsymbol{Q} \right|. \tag{3}$$

An example for $\mathcal{B}$ which is used later is

$$\mathcal{B} = \left\{ (-\boldsymbol{\infty}, \boldsymbol{\omega}) = \prod_{i=1}^{s} (-\infty, \omega_i) \;\middle|\; \boldsymbol{\omega} \in \mathbb{R}^s \right\}. \tag{4}$$

This is the family of left open boxes in $\mathbb{R}^s$ and in this case the discrepancy is called the star discrepancy denoted by $D^*(\boldsymbol{X}, \psi)$. For $\boldsymbol{Q}^{(i)} \in [0,1]^s =: I^s$, we set $D^*(\boldsymbol{X}) := D^*(\boldsymbol{X}, \chi)$, where χ is the characteristic function of I^s. Since QMC sequences are constructed to minimize the discrepancy, they are also

known as low discrepancy sequences. The term "approximating by point measure" is synonymous with *sampling*, usually encountered in statistical applications.

Once we adopt this measure theoretic approximation of ψ, we have to translate the dynamics described by (1) in the framework of the particles $\boldsymbol{Q}^{(i)}$ which fully determines our approximation. The details are given in the following subsections.

2.1 Approximation of the initial value: inversion method

The initialization step of every particle simulation is the particle approximation of the initial value. For our diffusion equation (1), we shall later employ the Gaussian initial condition:

$$\psi_0(\boldsymbol{Q}) = (2\pi)^{-s/2} \exp(-|\boldsymbol{Q}|^2/2) \tag{5}$$

We now describe one of the commonly used techniques, the inversion method [8], to generate this point measure. According to this method, let $\frac{1}{N}\sum_{i=1}^{N}\delta_{y_i}$ be a measure approximation of the characteristic function on $(0,1)$. Utilizing the points y_i, one can generate a measure approximation $\frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}$, of a function $\psi \geq 0$ with $\int \psi = 1$, using the cumulative distribution function

$$F(x) = \int_{-\infty}^{x} \psi(\gamma) d\gamma,$$

by solving the equation $F(x_i) = y_i$. Since $F(x)$ is a monotonically increasing function, the existence of an inverse is assured. Evaluating the approximating measure on general intervals $(-\infty, x)$, $x \in \mathbb{R}$, we find,

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}(-\infty,x) &= \frac{1}{N}\sum_{i=1}^{N}\delta_{F^{-1}(y_i)}(-\infty,x) \\ &= \frac{1}{N}\sum_{i=1}^{N}\delta_{y_i}(0,F(x)) \approx \int_0^{F(x)} 1 d\gamma = \int_{-\infty}^{x}\psi(\gamma)d\gamma. \end{aligned} \tag{6}$$

Since intervals $(-\infty, x)$ generate the Borel σ-algebra, (6) shows the approximation property.

Following the inversion technique, we construct a particle approximation of the Gaussian initial value by setting,

$$Q_k^{(i)} = H^{-1}(y_k^{(i)}), \quad i = 1, \ldots, N, \quad k = 1, \ldots, s$$

where H is the cumulative distribution function of the scalar Gaussian

$$H(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \quad \text{where} \quad \operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt \tag{7}$$

and $\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{y}^{(i)}}$ is a measure approximation of the indicator function of the s-dimensional unit cube I^s. This yields,

$$Q_k^{(i)} = \sqrt{2}\,\operatorname{erf}^{-1}(2y_k^{(i)} - 1).$$

As before, one can check the approximation property by evaluating the measure on a general multidimensional interval $(-\boldsymbol{\infty}, \boldsymbol{x}) = (-\infty, x_1) \times \cdots \times (-\infty, x_s)$.

Although, the inversion method is not economical for sampling Gaussian initial values, it has its advantages over the traditional Box-Mueller method, especially for the simulation of diffusion (refer [6] pp. 1569 - 1571 for details).

2.2 Diffusion

Let us suppose that we have approximated the initial value ψ_0 by $\tilde{\psi}_0 = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{Q}_0^{(i)}}$. Knowing that the solution of (1) at time Δt is given by the convolution of the initial value $\psi_0(\boldsymbol{Q})$ with the fundamental solution

$$G_{\Delta t}(\boldsymbol{Q}) = \frac{1}{(4\pi\Delta t)^{s/2}}\exp\left(-\frac{|\boldsymbol{Q}|^2}{4\Delta t}\right),$$

it is natural to use the convolution with $\tilde{\psi}_0$, leading to

$$\psi_1(\boldsymbol{Q}) = \int_{\mathbb{R}^s} G_{\Delta t}(\boldsymbol{Q} - \boldsymbol{X})\, d\tilde{\psi}_0(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N} G_{\Delta t}(\boldsymbol{Q} - \boldsymbol{Q}_0^{(i)}).$$

Since ψ_1 is a continuous function, an additional measure approximation

$$\psi_1 d\boldsymbol{Q} \approx \tilde{\psi}_1 = \frac{1}{M}\sum_{i=1}^{M}\delta_{\boldsymbol{Q}_1^{(i)}} \tag{8}$$

is needed to come back to a particle formulation. Since the quality of such a measure approximation is assessed in terms of the discrepancy, we have to find points $\boldsymbol{Q}_1^{(i)}$ such that

$$\int_B \psi_1(\boldsymbol{Q})\, d\boldsymbol{Q} \approx \frac{1}{M}\sum_{i=1}^{M}\delta_{\boldsymbol{Q}_1^{(i)}}(B)$$

for all sets B from a sufficiently large class. Introducing the characteristic function of the set B as a test function φ, we thus consider

$$\int_{\mathbb{R}^s} \varphi(\boldsymbol{Q})\psi_1(\boldsymbol{Q})d\boldsymbol{Q} = \int_{\mathbb{R}^s}\int_{\mathbb{R}^s} \varphi(\boldsymbol{Q}+\boldsymbol{X})G_{\Delta t}(\boldsymbol{Q})d\boldsymbol{Q}\, d\tilde{\psi}_0(\boldsymbol{X}).$$

In order to simplify the measure $G_{\Delta t}(\boldsymbol{Q})d\boldsymbol{Q}$, we use the transformation

$$\boldsymbol{Y} = \boldsymbol{H}\left(\boldsymbol{Q}/\sqrt{2\Delta t}\right), \text{where } \boldsymbol{H}(\boldsymbol{z}) = (H(z_1),\ldots,H(z_s))$$

with H given by (7), which has Jacobian $G_{\Delta t}(\boldsymbol{Q})$. Finally, we get

$$\int_{\mathbb{R}^s} \varphi(\boldsymbol{Q})\psi_1(\boldsymbol{Q})d\boldsymbol{Q} = \int_{I^s}\int_{\mathbb{R}^s} \varphi\left(\sqrt{2\Delta t}\,\boldsymbol{H}^{-1}(\boldsymbol{Y})+\boldsymbol{X}\right)\, d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y}. \qquad (9)$$

If $(\boldsymbol{X}_k, \boldsymbol{Y}_k)$, $k=1,\ldots,M$ defines a reasonable point-measure approximation of the product measure $d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y}$, then

$$\int_{\mathbb{R}^s} \varphi(\boldsymbol{Q})\psi_1(\boldsymbol{Q})d\boldsymbol{Q} \approx \frac{1}{M}\sum_{k=1}^{M} \varphi\left(\boldsymbol{X}_k + \sqrt{2\Delta t}\,\boldsymbol{H}^{-1}(\boldsymbol{Y}_k)\right)$$

which motivates us to define

$$\tilde{\psi}_1 = \frac{1}{M}\sum_{k=1}^{M} \delta_{\boldsymbol{Q}_1^{(k)}}, \qquad \boldsymbol{Q}_1^{(k)} = \boldsymbol{X}_k + \sqrt{2\Delta t}\,\boldsymbol{H}^{-1}(\boldsymbol{Y}_k). \qquad (10)$$

Consequently, the problem to set up a particle method for the diffusion equation reduces to finding a good point-measure approximation for the product $d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y}$ between a point measure $\tilde{\psi}_0$ and the Lebesgue measure on the s-dimensional unit cube.

2.2.1 Direct approximation of the product measure

Unfortunately, the most natural approach to the approximation of the product measure does not lead to a practicable program. The idea is to use a measure approximation of the characteristic function on I^s which yields, for example, N' points $\boldsymbol{Y}_j \in I^s$. Combining these points with the points $\boldsymbol{Q}_0^{(i)}$ that make up the measure $\tilde{\psi}_0$, we obtain $M = NN'$ pairs $(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}_j)$. The discrepancy of this construction is easily determined from the discrepancies of the participating point-measure approximations. However, we end up with $M = NN'$ particles after the first time step. So after k time steps the particle number is $N(N')^k$, which is enormously large even for reasonable values of k, leading to an impracticable algorithm.

2.2.2 *Monte Carlo approach*

If we assume that $\boldsymbol{Q}_0^{(i)}$ are obtained as independent realizations of a random variable with distribution ψ_0 then we can construct a reasonable product measure by taking N independent realizations $\boldsymbol{Y}_i$ of an s-dimensional uniform distribution in I^s and forming the pairs $(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}_i)$, $i = 1, \ldots, N$. These pairs can then be regarded as realizations of the product measure $\psi_0 d\boldsymbol{Q}\, d\boldsymbol{Y}$ and, consequently, can be used as point approximation. Note that the particle number stays constant in this approach. We want to stress that the independence of the participating random variables is the most important ingredient here. While this can be achieved with pseudo random points, it is generally *not* available with QMC points.

To exemplify this situation, we consider problem (1) with initial condition (5) in a single space dimension. According to our algorithm, we sample a set of N particles according to the initial condition. The update rule (10) then implies that the particle positions are incremented each time step by $\mathcal{N}(0, \Delta t)$ distributed random numbers (according to the inversion method, $\sqrt{2\Delta t}\, \boldsymbol{H}^{-1}(\boldsymbol{Y}_k)$ produces such points if $\boldsymbol{Y}_k$ is uniformly distributed in (0,1)). Figure 1 shows the result of the simulation using 1000 MC points at time $T = 1.0$ doing 50 time steps. If we simply replace MC points by QMC points,

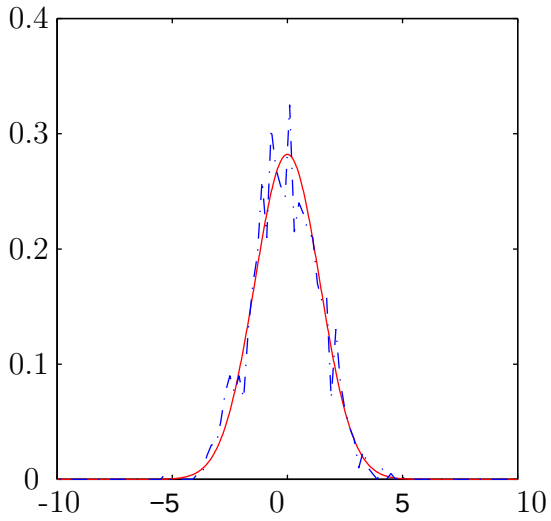

Fig. 1. Solution of diffusion problem (solid) and MC approximation (dashed dot).

we do not obtain the expected result as shown in figure 2. This is because of the non-independence (correlation) among the QMC points which spoils the convergence and can be explained with the following argument, taken from [6]. Assume we had taken N as a power of two and used the van der Corput sequence,

$$(\boldsymbol{x}_n) = \left(\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \ldots\right)$$

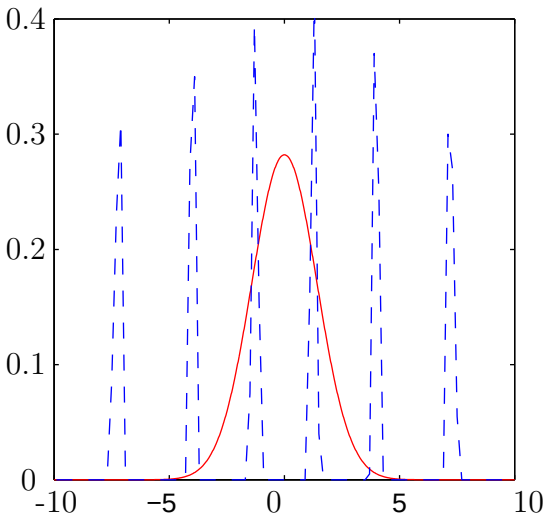

Fig. 2. MC approach with QMC points.

Then all the odd numbered particles would *always* get a positive increment and all the even numbered particles would get a negative increment, thereby the particles drift away deterministically.

2.2.3 A QMC approach: quasi-random mixing

Since the concept of independence is not available in connection with the deterministic QMC sequences, a different approach is required in order to stay with a fixed number of particles. In the following, we generalize the idea in [2]. To derive a particle approximation of the product measure, let us take a test function $\phi : \mathbb{R}^s \times I^s \to \mathbb{R}$ on the product space and consider

$$\int_{I^s} \int_{\mathbb{R}^s} \phi(\boldsymbol{X}, \boldsymbol{Y})\, d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y} = \sum_{i=1}^{N} \int_{I^s} \frac{1}{N} \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y})\, d\boldsymbol{Y}.$$

If we tessellate the r-dimensional unit cube I^r into N disjoint sets of equal volume $1/N$, and if we denote the characteristic function of set i by χ_i, we can write

$$\int_{I^r} \chi_i(\boldsymbol{\lambda}) d\boldsymbol{\lambda} = \frac{1}{N},$$

respectively

$$\int_{I^s} \int_{\mathbb{R}^s} \phi(\boldsymbol{X}, \boldsymbol{Y})\, d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y} = \sum_{i=1}^{N} \int_{I^s} \int_{I^r} \chi_i(\boldsymbol{\lambda}) \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y})\, d\boldsymbol{\lambda}\, d\boldsymbol{Y}.$$

Hence, if $(\boldsymbol{\lambda}_k, \boldsymbol{Y}_k)$, $k = 1, \ldots, N$ is a low discrepancy approximation of the characteristic function on I^{r+s}, then

$$\sum_{i=1}^{N} \int_{I^s} \int_{I^r} \chi_i(\boldsymbol{\lambda}) \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y})\, d\boldsymbol{\lambda}\, d\boldsymbol{Y} \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} \chi_i(\boldsymbol{\lambda}_k) \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}_k). \tag{11}$$

Observe that each $\boldsymbol{\lambda}_k$ is located in exactly one set of the tessellation, so that the double sum in (11) actually involves at most N non-zero terms. Introducing the mapping $\sigma : \{1, \ldots, N\} \to \{1, \ldots, N\}$ which assigns to each k the index i for which $\chi_i(\boldsymbol{\lambda}_k) = 1$, we can write

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} \chi_i(\boldsymbol{\lambda}_k) \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}_k) = \frac{1}{N} \sum_{k=1}^{N} \phi(\boldsymbol{Q}_0^{(\sigma(k))}, \boldsymbol{Y}_k).$$

Since σ is generated by the quasi-random vector $\boldsymbol{\lambda}_k$, it acts like a *quasi-random mixing* of the N points $\boldsymbol{Q}_0^{(i)}$ and the N points $\boldsymbol{Y}_k$. The mapping σ is even invertible (i.e. a permutation) if we assume that each set of the tessellation contains *exactly* one point $\boldsymbol{\lambda}_k$. In this case, the particle approximation of the product measure is given by

$$(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}_{\sigma^{-1}(i)}), \qquad i = 1, \ldots, N.$$

The idea of having in each set of the tessellation exactly one $\boldsymbol{\lambda}_k$ can be achieved using the concept of $(0, m, s+r)$-nets defined below. See [4] for details.

Definition 1 *A $(0, m, s+r)$-net in base b is a point set P consisting of b^m points in I^{s+r} such that every elementary interval E of the form*

$$\prod_{i=1}^{s+r} \left[a_i b^{-d_i}, (a_i + 1) b^{-d_i} \right), \quad a_i, d_i \in \mathbb{N}_0,\ a_i < b^{d_i}, \sum_{i=1}^{s+r} d_i = m$$

with volume b^{-m} contains exactly one point.

Using this notation, we choose r-dimensional intervals for our tessellation of I^r

$$I_{\boldsymbol{a}} = \left[a_1 b^{-d_1}, (a_1 + 1) b^{-d_1} \right) \times \cdots \times \left[a_r b^{-d_r}, (a_r + 1) b^{-d_r} \right),$$

where $\boldsymbol{a} = (a_1, \ldots, a_r)$. If $N = b^m$ and $d_1 + \cdots + d_r = m$, the sets are clearly of volume $1/N$. By the property of $(0, m, s+r)$ nets, the elementary intervals $I_{\boldsymbol{a}} \times I^s$ contain exactly a single point $(\boldsymbol{\lambda}_k, \boldsymbol{Y}_k)$, which implies that in each $I_{\boldsymbol{a}}$ there will be exactly one point $\boldsymbol{\lambda}_k$.

2.2.4 The QMC approach of Lécot

While the quasi-random mixing introduced in the previous section may break some of the correlations among the points $\boldsymbol{Q}_0^{(i)}$ and $\boldsymbol{Y}_i$, it generally does not

guarantee a sufficient approximation quality for the product measure. Observe that the approximation of the product measure appears in (11) where the integral over I^{r+s} is replaced by a sum using the points $(\boldsymbol{\lambda}_k, \boldsymbol{Y}_k)$. In principle, the numerical error in this approximation can be estimated by the Koksma-Hlawka inequality [4], according to which, this error is bounded by the product of the discrepancy of the point set used and the variation of the function which is integrated, i.e.

$$(\boldsymbol{\lambda}, \boldsymbol{Y}) \mapsto F_N(\boldsymbol{\lambda}, \boldsymbol{Y}) = \sum_{i=1}^{N} \chi_i(\boldsymbol{\lambda}) \phi(\boldsymbol{Q}_0^{(i)}, \boldsymbol{Y}).$$

While the discrepancy of a set $(\boldsymbol{Y}_k, \lambda_k), k = 1, \ldots, N$ can be chosen essentially of order $O(1/N)$, the function F_N turns out to be quite rough with a variation of $O(N)$. Hence, a direct application of the Koksma-Hlawka inequality gives no information about the convergence of (11) for large N. In [2], the approximation (11) is therefore split into *two* approximations. Before using the discrepancy estimate, an auxiliary function $\tilde{F}_N$ is constructed which has less variation ($O(N^\beta)$, $\beta < 1$) than F_N. If this function is used instead of F_N, one finds two error contributions

$$\int_{I^{r+s}} F_N \, d\boldsymbol{\lambda} \, d\boldsymbol{Y} \approx \int_{I^{r+s}} F_N - \tilde{F}_N \, d\boldsymbol{\lambda} \, d\boldsymbol{Y} + \frac{1}{N} \sum_{k=1}^{N} \tilde{F}_N(\boldsymbol{\lambda}_k, \boldsymbol{Y}_k).$$

While the error in the measure approximation can now be made essentially of order $N^{\beta-1}$, the integral error gives rise to $N^{-\alpha}$ depending on how well $\tilde{F}_N$ approximates F_N in $\mathbb{L}^1$. Balancing the two errors leads to an overall convergence proof but the order is clearly less than in a pure QMC integral approximation. In the proof presented in [2], the construction of $\tilde{F}_N$ requires a *multidimensional sorting* of the points $\boldsymbol{Q}_0^{(i)}$ which is combined with an s-dimensional mixing (i.e. $r = s$ in the previous section).

The sorting is based on the numbers $d_1, \ldots, d_s$ (where $N = b^{\sum_i d_i}$) which are also used to describe the number b^{d_i} of subintervals into which the coordinate direction i is divided to form the tessellation of I^s. An s-dimensional sorting of the points $\boldsymbol{Q}_0^{(i)}$ is then achieved by first sorting with respect to the first coordinates of the points. In the resulting sorted list, consecutive groups of $b^{d_2+\cdots+d_s}$ particles are formed in which sorting is performed with respect to the second coordinate. After that, the list is split into smaller groups of length $b^{d_3+\cdots+d_s}$ which are sorted with respect to the third coordinate and so on. In this way, the particle which used to have number i gets the new number $\tau(i)$ where τ is a permutation of $\{1, \ldots, N\}$. Afterwards, the permutation τ due to sorting is combined with the one due to mixing which yields the pairs

$$(\boldsymbol{Q}_0^{(\tau^{-1}(i))}, \boldsymbol{Y}_{\sigma^{-1}(i)}), \qquad i = 1, \ldots, N.$$

Using this approximation of the product measure, the convergence of the dif-

fusion algorithm can be proved (see [2]). If $\boldsymbol{X}^{(r)}$ is the point set consisting of $\boldsymbol{Q}_r^{(1)}, \ldots, \boldsymbol{Q}_r^{(N)}$, which are the positions of the particles at time $T = r\Delta t$, and ψ_r is the exact solution of the Cauchy problem for pure diffusion at time $T = r\Delta t$, then the discrepancy error can be estimated as

$$D^*(\boldsymbol{X}^{(r)}; \psi_r) \leq D^*(\boldsymbol{X}^{(0)}; \psi_0) + b^{d_1+\cdots+d_{s-1}+\lfloor d_s/2 \rfloor} \sum_{i=0}^{r-1} D^*(\boldsymbol{Y}_{(i)}) + r\left(\frac{1}{b^{d_1}} + \cdots + \frac{1}{b^{d_{s-1}}} + \frac{1}{b^{\lfloor d_s/2 \rfloor}}\right) \quad (12)$$

where, $D^*(\boldsymbol{Y}_{(i)})$, $i = 0, \ldots, r-1$ is the star discrepancy of the $(0, m, 2s)$-net $\boldsymbol{Y}_{(i)}$ in base b, the least prime $\geq 2s$.

In order to conclude that the error vanishes for $N \to \infty$, we see from the last term on the right that all the d_is must increase as we make N larger. Taking this fact into account, and since we are dealing with the isotropic diffusion problem, let us set, for example, all the d_is equal. If we assign $d_i = 2k, i = 1, \ldots, s$, then the term $\frac{1}{b^{\lfloor d_s/2 \rfloor}}$ essentially dictates the order of convergence of the method. Now, $N = b^{d_1+\cdots+d_s} = b^{2ks}$, so that, $b^k = N^{1/2s}$. Hence the estimate only guarantees a very low order of convergence like $N^{-1/2s}$. However, as we will demonstrate in the numerics section, the scheme behaves much better in practice and actually outperforms the basic Monte Carlo approach. The only drawback is the required minimal particle number which is quite large as has been explained already in the introduction.

It is also evident from (12) that the estimate fails on setting one or more of the d_is to zero. To be concrete, if we set $d_1 = 0$, then from (12), it is clear that the last term on the right has a leading term r, and this does not go to zero as $N \to \infty$. Nevertheless, the scheme works remarkably well even in the case where we set $d_2 = \cdots = d_s = 0$, with the additional advantage that much smaller minimal particle numbers are possible.

2.2.5 A QMC approach: sorting

As we have seen, both mixing and sorting introduce permutations and the question arises whether the two permutations are equally important for the approximation of the product measure. We therefore also consider the case in which *only* sorting is used. For the 1d case, Morokoff [6] sorted the particle positions at every time step and obtained significant improvement over standard MC. As a generalization of the previous case, we allow for r-dimensional sorting for which we choose r out of the s coordinate directions and sort in the same manner as described above based on the numbers $e_1, \ldots, e_r$ with the property that $b^{e_1+\cdots+e_r} = N$. In the extreme case $r = 1$, this means that we simply sort with respect to a single coordinate of the points $\boldsymbol{Q}_0^{(i)}$. The results

of the simulation in this case for the one dimensional example of Section 2.2 are better than plain MC as depicted below in figure 3.

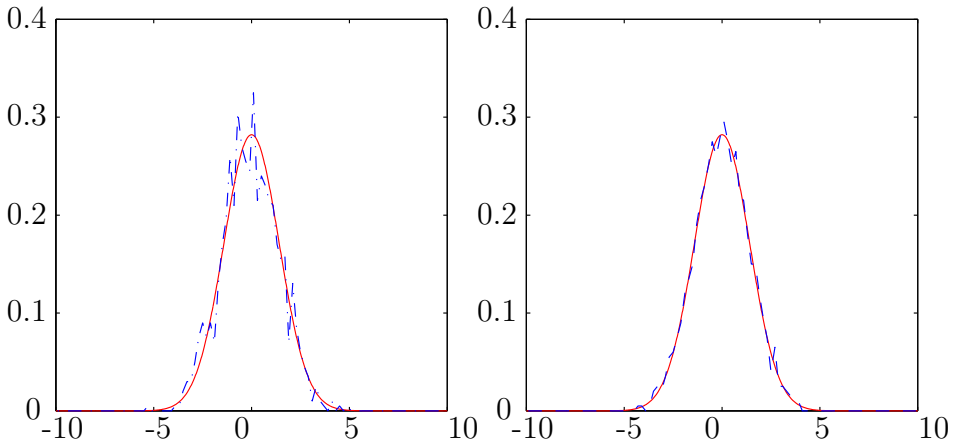

Fig. 3. Solution of the diffusion problem (solid). MC approximation (left figure with dashed dot curve) and QMC approximation with sorting (right figure with dashed curve).

Our motivation for the sorting idea is based on the following result, which shows, at least in the two dimensional case, that sorting is a reasonable approach for the construction of product measure.

Lemma 2 *Let S be an arbitrary sequence of points $x_0, x_1, \ldots$ in I. For $N \geq 1$, let P be the point set consisting of $(x_n, y_n)_{n=0}^{N-1}$ in I^2, where $Y = (y_n)_{n=0}^{N-1}$ is a $(0, m, 1)$-net in base b sorted in ascending order. Then*

$$ND_N^*(P) \leq \max_{0 \leq M \leq N-1} MD_M^*(S) + 1 \tag{13}$$

Remark 3 *Observe that if S is a low discrepancy sequence of points, then $D_M^*(S)$ essentially behaves like $1/M$, up to a logarithmic factor and hence the right hand side of (13) is $O(1)$. Dividing (13) by N, it then follows that $D_N^*(P)$ is $O(1/N)$ up to a logarithmic factor.*

PROOF. Let $J = [0, u_1) \times [0, u_2) := J_1 \times J_2$. If $A(J; P)$ is the counting function that indicates the number of n with $0 \leq n \leq N-1$ for which $x_n \in J$, then we need to estimate

$$|A(J; P) - N\lambda_2(J)|,$$

where λ_2 is the two dimensional Lebesgue measure.

Observe that $(x_n, y_n) \in J$ if and only if $x_n \in J_1$ and $y_n \in J_2$. Since Y is a

$(0, m, 1)$-net in base b, every elementary interval

$$E_i = [i/N, (i+1)/N), \quad i = 0, \ldots, N-1$$

contains exactly one point. Since $u_2 \in [0, 1)$, there exists $M \in \mathbb{Z}_+$ such that

$$\frac{M}{N} \leq u_2 \leq \frac{M+1}{N} \tag{14}$$

Depending upon whether $u_2 < y_M$ or $u_2 \geq y_M$, $A([0, u_2), Y)$ is either M or $M+1$ respectively. We consider the two cases separately.

Let $u_2 < y_M$, so that $A([0, u_2), Y) = M$. Then

$$|A(J; P) - N\lambda_2(J)| = |A([0, u_1); S_M) - N\lambda_2(J)|$$

where S_M is the point set $x_0, \ldots, x_{M-1}$. On using triangle inequality we find,

$$|A(J; P) - N\lambda_2(J)| \leq |A(J_1; S_M) - M\lambda_1(J_1)| + |M\lambda_1(J_1) - N\lambda_2(J)|$$

and hence on taking the supremum over u_1 and u_2 we get

$$ND_N^*(P) \leq \max_{0 \leq M \leq N-1} MD_M^*(S) + |Mu_1 - Nu_1u_2| \tag{15}$$

Multiplying (14) by Nu_1, we can estimate

$$0 \leq Nu_1u_2 - Mu_1 \leq u_1 < 1$$

and (13) follows.

For the other case $u_2 \geq y_M$, we have $A([0, u_2), Y) = M + 1$. So

$$|A(J; P) - N\lambda_2(J)| \leq |A(J_1; S_{M+1}) - (M+1)\lambda_1(J_1)| + |(M+1)\lambda_1(J_1) - N\lambda_2(J)|$$

where S_{M+1} is the point set $\{x_0, \ldots, x_{M+1}\}$. Hence, as before,

$$ND_N^*(P) \leq \max_{0 \leq M \leq N-1} MD_M^*(S) + |(M+1)u_1 - Nu_1u_2| \tag{16}$$

Again from (14), we have $|(M+1)u_1 - Nu_1u_2| \leq u_1$ and thus (13) follows. □

The requirement that Y be a $(0, m, 1)$-net in base b can be replaced by the assumption that it is a general sorted point set. Given a $u_2 \in [0, 1)$, there exists an index M such that $y_M \leq u_2$ and $y_{M+1} > u_2$. Then, if $J_2 = [0, u_2)$, we have $A(J_2; Y) = M$ points of Y in J_2 and

$$|M - Nu_2| = |A(J_2; Y) - N\lambda_1(J_2)| \leq ND_N^*(Y).$$

For any $u_1 \in [0,1)$, we conclude

$$|Mu_1 - Nu_1u_2| \le u_1 N D_N^*(Y) \le N D_N^*(Y)$$

which can be used in (15). Thus we have proved the following.

Theorem 4 *Let S be an arbitrary low discrepancy sequence of points $x_0, x_1, \ldots$ in I. For $N \ge 1$, let P be the point set consisting of $(x_n, y_n)_{n=0}^{N-1}$ in I^2, where $Y = (y_n)_{n=0}^{N-1}$ is a point set sorted in ascending order. Then*

$$N D_N^*(P) \le \max_{0 \le M \le N-1} M D_M^*(S) + N D_N^*(Y)$$

Again by arguments of remark 3, we find that P inherits the discrepancy order of S and Y.

2.2.6 General structure of the diffusion algorithm

Obviously, all the practicable algorithms presented above can be summarized as follows: to construct a product measure approximation of $d\tilde{\psi}_0(\boldsymbol{X})\, d\boldsymbol{Y}$, the N pairs

$$(\boldsymbol{Q}_0^{(\tau^{-1}(i))}, \boldsymbol{Y}_{\sigma^{-1}(i)}), \qquad i = 1, \ldots, N \tag{17}$$

are used. In case of MC, both permutations are given by the identity and the particle approximations are based on independent (pseudo) random numbers. Otherwise, for the QMC generated points, the permutation τ is obtained by a sorting of the particle positions and σ by a quasi-random mixing. If either of the steps is dropped, the corresponding permutation can be taken as the identity. Referring back to (10), the approximation (17) leads to the following algorithm for the movement of the particles in the first time step

$$\boldsymbol{Q}_1^{(i)} = \boldsymbol{Q}_0^{(i)} + \sqrt{2\Delta t}\, \boldsymbol{H}^{-1}(\boldsymbol{Y}_{\sigma^{-1}(\tau(i))}), \quad i = 1, \ldots, N. \tag{18}$$

Note that this update rule is quite similar to an explicit Euler discretization with time step Δt, of the stochastic differential equation, (refer [9])

$$d\boldsymbol{Q} = \sqrt{2}\, d\boldsymbol{W}_t \tag{19}$$

where $\boldsymbol{W}_t$ is a standard Wiener process, because $\boldsymbol{H}^{-1}(\boldsymbol{Y}_{\sigma^{-1}(\tau(i))})$ is a measure approximation in sampling of the standard normal distribution.

2.3 The algorithms

In the previous subsections, we have seen how to interpret the dynamics of equation (1) in the framework of particles. In every time step, the particles

reach an intermediate stage $\boldsymbol{Q}^{(i)}_{n'+1/2}, i = 1, \ldots, N$ (due to sorting), which is then followed by diffusion which completes one cycle (mixing of the particles may give rise to a second intermediate stage).

Our notation for the different type of algorithms will be as follows: the algorithm with $\sigma = \tau = id$ and particle approximations based on independent (pseudo) random numbers is called MC. The name QMC$(r, 0)$ with $1 \leq r \leq s$ stands for the algorithm with r-dimensional mixing and no sorting. Similarly, QMC$(0, r)$ indicates the algorithm with no mixing but r-dimensional sorting and QMC(l, r) represents the algorithm where both mixing and sorting is performed in l and r of the s dimensions respectively. For example, QMC(s, s) is the algorithm proposed in [2] where both sorting and mixing is done in all the coordinates. QMC$(0, 1)$ would be an algorithm where only sorting is performed along one coordinate. The algorithm QMC$(1, 1)$ has been studied in [10]. It involves sorting and mixing along a single coordinate.

We remark that a full description of the algorithms QMC$(0, r)$, QMC$(r, 0)$, and QMC(l, r) actually requires the specification of the numbers $d_1, \ldots, d_l$ and $e_1, \ldots, e_r$ (see Sections 2.2.3 and 2.2.5). Among other things, this choice determines the possible particle numbers N with which the algorithm can operate. Given a base b (in our examples this will be the least prime $\geq s + l$ for the algorithm QMC(l, r) in dimension s), the particle numbers have to be of the form b^m with $m \geq \max(l, r)$. Note that the choice of the d_is and e_is are not unique.

Example 5 *As an example, consider the algorithm QMC*$(2, 3)$ *in dimension* $s = 5$. *In this case, the base* b *is the least prime* $\geq 5 + 2$ *and hence* $b = 7$. *As a possible way to define a sequence of the parameters* d_1, d_2 *and* e_1, e_2, e_3 *to get an increasing sequence of particle numbers, we mention*

e_1	e_2	e_3	$N = b^{\sum e_i}$
1	*1*	*1*	*343*
2	*1*	*1*	*2401*
2	*2*	*1*	*16807*
2	*2*	*2*	*117649*
3	*2*	*2*	*823543*

d_1	d_2
2	*1*
2	*2*
3	*2*
3	*3*
4	*3*

The complete algorithm for the diffusion equation can be summarized as follows.

ALGORITHM QMC(l,r)

(1) **Input** s, $1 \leq m \in \mathbb{Z}$, Δt and T

(2) **Choose** b, the least prime $\geq s + \max(l,r)$ and **set** $N = b^m$

(3) **Approximate the initial value**, $\psi_0 d\boldsymbol{Q} \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{Q}_0^{(i)}}$

(4) **Initialize** $n' \leftarrow 0$

(5) WHILE $(n'\Delta t < T)$

{

(a) Construct the permutation τ (by sorting if $r > 0$). Set up
$\boldsymbol{Q}_{n'+1/2}^{(i)} = \boldsymbol{Q}_{n'}^{(\tau^{-1}(i))}$

(b) **Diffusion**

Construct the permutation σ (by mixing if $l > 0$) using $\boldsymbol{\lambda}_k$ from a $(0, m, s+l)$-net $(\boldsymbol{\lambda}_k, \boldsymbol{Y}_k)$, $k = 1, \ldots, N$

FOR $k = 1, N$

$$\boldsymbol{Q}_{n'+1}^{(\sigma(k))} = \boldsymbol{Q}_{n'+1/2}^{(\sigma(k))} + \sqrt{2\Delta t}\, \boldsymbol{H}^{-1}(\boldsymbol{Y}_k)$$

END

(c) $n' \leftarrow n' + 1$

}

2.4 Applicability of the QMC(l,r) algorithm

In general, the applicability of the QMC(l,r) algorithm in dimension s is restricted by memory limitations and this forces the choice of l and r to be as small as possible for large values of s. Following algorithm QMC(l,r), the possible particle numbers are $b^m, b^{m+1}, b^{m+2}, \ldots$, where b is the least prime $\geq s + l$ and $m = \max(l,r)$. In order to find the order of convergence of the method, a minimum of three data sets relating particle numbers and absolute errors are required and this forces us to work with b^{m+2} particles. To store the b^{m+2} particle positions (in double precision) in dimension s, a total memory of $b^{m+2} \cdot s \cdot \text{sizeof(double)}$ is required. Bounding this quantity by the total available memory T_{mem} yields the condition

$$b^{m+2} \cdot s \cdot \text{sizeof(double)} \leq T_{mem}.$$

Table 1 summarizes the maximum applicable dimension s for various values of l and r, assuming the available memory to be 1.5GB.

	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=7$
$r=0$		112	41	20	9	6	1	0
$r=1$	113	112	41	20	9	6	1	0
$r=5$	10	10	9	8	7	6	1	0
$r=10$	3	2	1	0	0	0	0	0
$r=15$	2	1	1	0	0	0	0	0
$r=20$	2	1	0	0	0	0	0	0
$r=25$	1	1	0	0	0	0	0	0
$r=26$	0	0	0	0	0	0	0	0

Table 1
Maximum applicable dimension s for a given value of l and r, given T_{mem}= 1.5GB

3 Numerical results

In this section we summarize the results of various numerical simulations that have been carried out using different approaches. We mainly compare the computations based on straightforward MC algorithm (MC without variance reduction etc.,) with the algorithms outlined in the previous section. All the computations are done on a AMD Athlon 1400 MHz machine with 1.5GB memory running Debian Linux 3.0. The CPU time we shall refer to is as measured on this machine. The complete implementation is done in ANSI C language.

In our computations we take as $(0, m, s+l)$-net in base b the Faure sequence [11]. The fast Faure generator implementation is due to Eric Thiémard [12] based on the idea presented in [13], which requires only $O(ms)$ time compared to $O(m^2 s)$ proposed in [4,11,14]. For the Monte Carlo simulation, we use the Unix inbuilt random number generator function *drand*48. For sorting, we use the *quicksort* algorithm proposed by Hoare [15].

This section is organized as follows. In the first part we review the better accuracy of QMC over MC in sampling initial values. The second part deals with the problem of plain diffusion in high dimensions.

3.1 Initial sampling

As a first task, we compare the time taken by the Faure generator and the random generator to sample particles distributed according to standard normal distribution in various dimensions. We use the inversion technique outlined

in subsection 2.1 for sampling. It is clear from figure 4 that there is a small

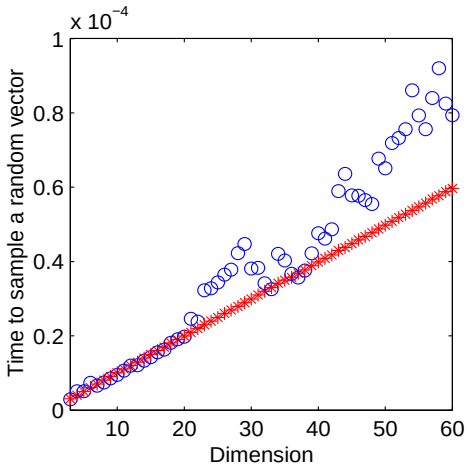

Fig. 4. Time taken to sample particles distributed according to standard normal for dimensions s=3 to s=60. The stars correspond to MC whereas the circles correspond to QMC.

overhead involved in generating Faure points especially in high dimensions.

Having seen that it is slightly slower to generate and use QMC points, we now show that they approximate the sampled function better than MC points. Consistent with later use, we consider sampling from the standard normal distribution ψ_0. In order to calculate numerically the discrepancy introduced in (3), we take as $\mathcal{B}$ in (4), a set $\mathcal{B}_l$ of 10000 boxes $(-\boldsymbol{\infty}, \boldsymbol{\omega})$, where the components of $\boldsymbol{\omega}$ are normally distributed with mean 3 and variance 1. If $\boldsymbol{X}^{(0)}$ is the point set consisting of $\boldsymbol{Q}^{(1)}, \ldots, \boldsymbol{Q}^{(N)}$, sampled according to standard normal, a parameter α is fit in such a way that $D_{\mathcal{B}_l}(\boldsymbol{X}^{(0)}, \psi_0) = CN^{\alpha}$, in the sense of least squares. We see from figure 5 that the order of convergence of QMC is significantly better than MC.

Remark 6 *It is to be observed that the rate of convergence we get by the least squares fit is dependent heavily on the data under consideration. Especially for the strongly fluctuating results obtained with MC and QMC, an extra data point can either improve or worsen the order of convergence considerably.*

For the data set presented in figure 6 for example, the order of convergence is reduced by including the error value corresponding to the largest particle number. Since the particle numbers cannot be freely chosen in our QMC algorithm for diffusion problems, the size of the data set used for fitting is eventually restricted by memory limitations. Thus the estimated convergence order is not of high precision and it should just give an indication of the general behavior of the algorithm.

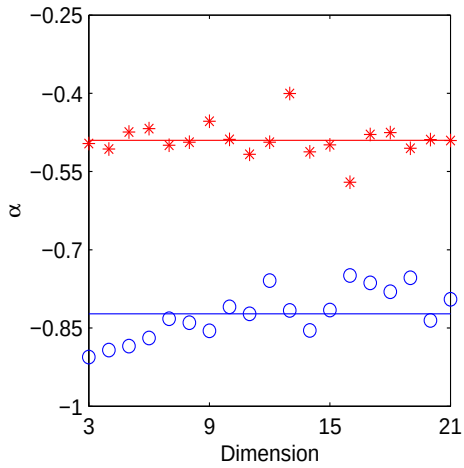

Fig. 5. Initial sampling error. Value of α in MC (stars) and QMC (circles) simulations for dimensions s=3 to s=18, restricting to a maximum of 10^6 particles. The line indicates the average order of convergence over dimension.

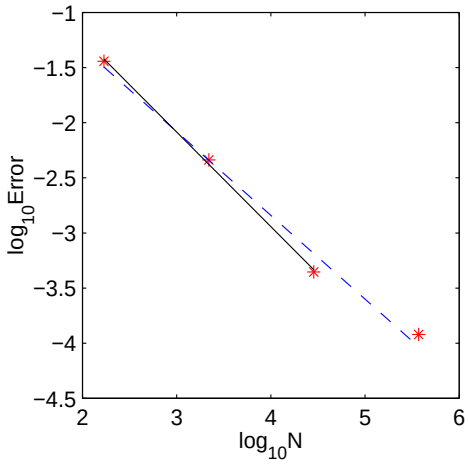

Fig. 6. The data set is represented by stars. The least square fit of the data is shown by the dashed line and the same for the data without the last point included is shown by the solid line. The data corresponds to the case $s = 12$ in figure 5 for QMC.

3.2 Plain diffusion

We now consider the Cauchy problem (1) with $\psi_0(\boldsymbol{Q}) = \pi^{-s/2}\exp(-|\boldsymbol{Q}|^2)$. The exact solution for this problem can be written down as the convolution of the Gauss kernel $G_t(\boldsymbol{Q})$ with the initial value $\psi_0(\boldsymbol{Q})$. Thus

$$\psi(\boldsymbol{Q},t) = \frac{1}{(\pi(1+4t))^{s/2}}\exp\left(-\frac{|\boldsymbol{Q}|^2}{1+4t}\right) \tag{20}$$

We now compare the various QMC methods outlined in the previous section with Monte Carlo. The first aspect in this regard would be to check the accuracy of the methods. With the same notion of discrepancy used in subsection 3.1, we calculate α for different dimensions taking 100 time steps of 0.0001 each.

Method	$s = 3$	$s = 6$	$s = 9$	$s = 12$	$s = 50$
MC	-0.2698	-0.2538	-0.2862	-0.2765	-0.2946
QMC$(1, 1)$	-0.4517	-0.4162	-0.5299	-0.4757	-0.6106
QMC$(0, 1)$	-0.5662	-0.5407	-0.5783	-0.5542	-0.6070
QMC$(1, 0)$	-0.00002	-0.0002	-0.0001	-0.0001	-0.00008
QMC(s, s)	-0.4555	NA	NA	NA	NA
QMC$(0, s)$	-0.2149	NA	NA	NA	NA
QMC$(s, 0)$	-0.1715	NA	NA	NA	NA
QMC$(3, 2)$	-0.1699	-0.4100	-0.2268	-0.4627	NA

Table 2
Order of convergence α for the various methods. NA refers to non applicability due to memory restrictions.

From table 2, we conclude that algorithm QMC$(0, 1)$ outperforms the others. One can also observe that algorithm QMC$(1, 1)$ performs well with the only disadvantage of extra mixing time. Algorithm QMC$(1, 0)$ does not converge at all implying that sorting is essential for convergence and is in accordance with [6].

The anomaly in performance of QMC$(3, 2)$ in dimension $s = 3$ and $s = 9$ is due to the fact that the error corresponding to large particle numbers does not drop considerably. This is depicted in figure 7. In view of remark 6, we stress the fact that, more than the order of convergence, the absolute error should be taken into consideration. A closer look into figure 7 reveals that even though the order of convergence of QMC$(s, 0)$ and QMC$(3, 2)$ algorithms in dimension $s = 3$ are poor, the absolute error is still quite less compared to MC. A similar conclusion can be drawn for the case $s = 9$.

Looking at table 2, we conclude that the only applicable algorithms in higher dimensions are MC, QMC$(1, 1)$ and QMC$(0, 1)$. A comparison of the absolute error for these methods is given in figure 8. Again, the QMC algorithms show significantly less error compared to the corresponding MC result.

Regarding the computational time we start with a comparison of MC and QMC$(0, 1)$ again doing 100 steps of 0.0001 each. It is clear from figure 9 that

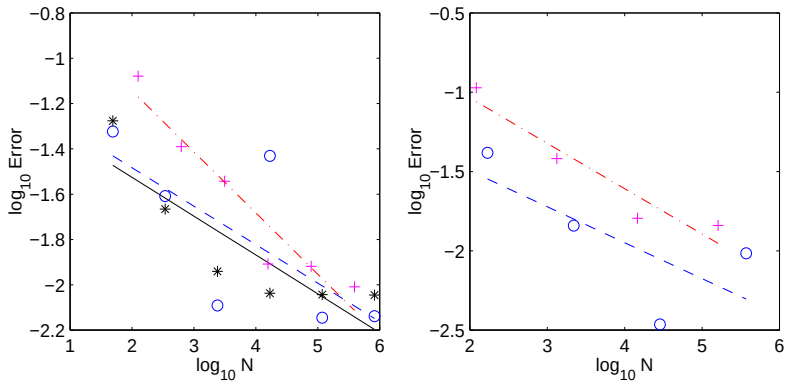

Fig. 7. QMC$(s,0)$ data (stars), QMC$(3,2)$ data (circles) and MC data (plus) for dimensions $s=3$ (left) and $s=9$ (right). The least square fit is depicted by line for QMC$(s,0)$, by dashed line for QMC$(3,2)$ and MC by dashed dot line respectively. It may be observed that QMC$(s,0)$ results does not seem to converge.

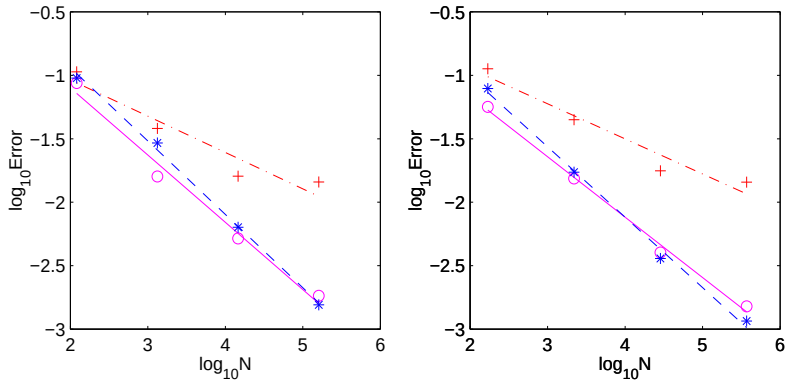

Fig. 8. Absolute error for MC data (plus), QMC$(1,1)$ data (circles) and QMC$(0,1)$ data (stars) for dimensions $s=9$ (left) and $s=12$ (right). The least square fit is depicted by dashed dot line for MC, solid line for QMC$(1,1)$, and dashed line for QMC$(0,1)$.

MC is faster compared to QMC$(0,1)$ especially in high dimensions. A time comparison between the various QMC algorithms is not so straightforward, because the methods work on completely different particle numbers. For example, for a problem in s dimension, QMC$(0,1)$ works on an s dimensional Faure sequence in base b, the least prime number $\geq s$, whereas for QMC(s,s), Faure sequence of dimension $2s$ in base b, the least prime $\geq 2s$ is considered. So we cannot estimate the time required to carry on the computation with a fixed number of particles. However, in order to have an idea about the computational time, we compare the time taken for the sorting step.

From figure 9 (right), it is clear that multi-index sorting, which is applicable only up to space dimension s=6 due to memory restrictions, takes considerably much more time compared to sorting only along one dimension. The percentage of sorting time in QMC$(0,1)$ can be estimated by comparing the

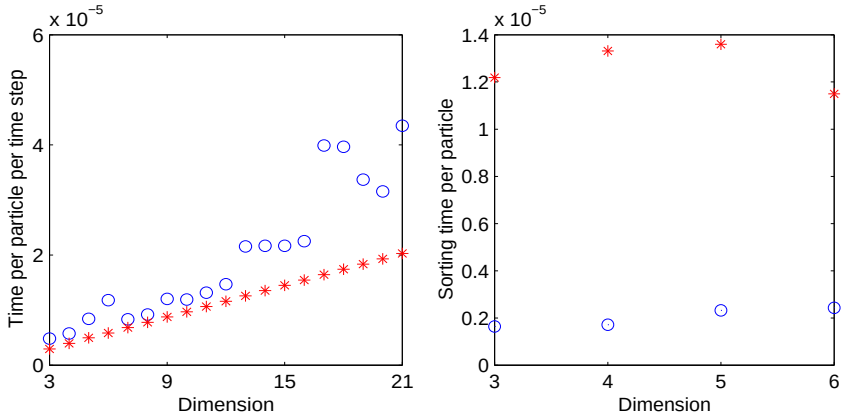

Fig. 9. (Left)CPU time required by MC (stars) and QMC$(0,1)$ (circles) in carrying out a single step of plain diffusion for dimensions s=3 to s=21, excluding the generation time. (Right) Sorting time for QMC$(0,1)$ (circles) and QMC(s,s) (stars) algorithms.

left and right pictures in figure 9 (around 6%). Since the mixing part takes very less time compared to sorting, the computational time for the QMC$(0,1)$ and QMC$(1,1)$ algorithms are almost the same.

4 Conclusion

The superiority of QMC over MC in plain integration is well known and is again verified in section 3.1. The better order of convergence (eventually $1/N$ versus $1/\sqrt{N}$) however, does not carry over to plain diffusion problems due to the correlation among QMC points.

It can in general be observed from the numerical experiments, that QMC$(0,1)$ and QMC$(1,1)$ algorithms have better order of convergence compared to MC algorithm. Although the straightforward MC algorithm takes less time compared to QMC$(0,1)$ (and QMC$(1,1)$ since mixing does not add considerably to the total CPU time), since considerably more particles have to be considered in order to obtain a prescribed accuracy, we conclude that there is some improvement achieved over the simple MC method.

Finally, the purpose of this paper is the development of a QMC algorithm which is applicable in high dimensions and its comparison with the simplest MC algorithm. Currently efforts are on in proving the convergence of the QMC$(0,1)$ algorithm.

References

[1] G. Venkiteswaran, M. Junk, A QMC approach for high dimensional Fokker-Planck equations modelling polymeric liquids, Mathematics and Computers in Simulation .

[2] C. Lécot, F. E. Khettabi, Quasi-Monte Carlo simulation of diffusion, Journal of Complexity 15 (1999) 342–359.

[3] E. Novak, K. Ritter, The curse of dimension and a universal method for numerical integration, in: G. Nüneberger, J. Schmidt, G. Walz (Eds.), Multivariate approximation and splines, ISNM, 1997, pp. 177–188.

[4] H. Niederreiter, Random number generation and quasi-Monte Carlo methods, Vol. 6, Society for Industrial and Applied Mathematics, 1992.

[5] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical algorithms 18 (1998) 209–232.

[6] W. J. Morokoff, R. E. Caflisch, A quasi-Monte Carlo approach to particle simulation of the heat equation, Siam J. Numer. Anal. 30 (6) (1993) 1558–1573.

[7] C. Lécot, Low discrepancy sequences for solving the Boltzmann equation, Journal of Comp. and Applied Math. 25 (1989) 237–249.

[8] L. Devroye, Non-uniform random variate generation, Springer, New York, 1986.

[9] T. Mikosch, Elementary stochastic calculus with finance in view, World Scientific, Singapore, 1998.

[10] G. Venkiteswaran, A particle method for fokker-planck equations in high dimensions, PhD dissertation, University of Kaiserslautern, Department of Mathematics (Jan. 2003).

[11] H. Faure, Discrépance de suites associées à un systéme de numération (en dimension s), Acta Arith. 42 (1982) 337–351.

[12] E. Thiémard, Economic generation of low-discrepancy sequences with a b-ary gray code, EPFL-DMA-ROSO, RO981201 (1998).

[13] S. Tezuka, Uniform random numbers: Theory and practice, Kluwer Academic Publishers, Boston, 1995.

[14] I. M. Sobol, On the distribution of points in a cube and the approximate evaluation of integrals, USSR Comput. Math. Math. Phys .

[15] R. Sedgewick, Algorithms in C, Addison-Wesley, Reading, Massachusetts, 1990.