Randomized quasi-Monte Carlo simulation of fast-ion thermalization

L. J. Höök, T. Johnson and T. Hellsten
School of Electrical Engineering, Div Fusion Plasma Physics, Royal Institute of Technology (KTH), Teknikringen 31 SE-100 44 Stockholm, Sweden
E-mail: joh@kth.se

Abstract. This work investigates the applicability of the randomized quasi-Monte Carlo method for simulation of fast-ion thermalization processes in fusion plasmas, e.g. for simulation of neutral beam injection and RF-heating. In contrast to the standard Monte Carlo method, the quasi-Monte Carlo method use deterministic numbers instead of pseudo-random numbers and have a statistical weak convergence close to $O(N^{-1})$, where $N$ is the number of markers. We have compared different quasi-Monte Carlo methods for a neutral beam injection scenario, which is solved by many realizations of the associated stochastic differential equation, discretized with the Euler-Maruyama scheme. The statistical convergence of the methods is measured for time steps up to sixteen thousand.

AMS classification scheme numbers: 65C30, 65C10, 65C05

Submitted to: Computational Science & Discovery
Introduction

The quasi-Monte Carlo method is a well established method for improving the statistical convergence for problems solved with the standard Monte Carlo method. It is commonly used for improving high-dimensional integration, but there is a growing research on the applicability of the method for simulation of diffusion processes. Simulation of diffusion is achieved by moving particles in the Lagrangian framework according to the associated stochastic differential equation (SDE). Examples in fusion plasma physics are simulation of neutral beam injection [1, 2] and RF-heating [3]. In the simulation, two sources of error appear; the time discretization error from the numerical scheme of the SDE and the statistical error from the finite number of particles describing the density. The standard Monte Carlo method gives a statistical convergence of $\alpha N^\beta$ with $\beta = -1/2$. To improve this value one often tries to reduce the value of $\alpha$ with some sort of variance reduction method. Common methods are the importance sampling and the control-variate method. In the plasma community a variance reduction concept known as the $\delta f$-method is commonly used, which is a family of different methods based on combinations of the importance sampling method and the control-variate method [4, 5, 6, 7, 8, 9]. Instead of reducing $\alpha$, the noise can be reduced by improving the order of convergence, $\beta$. This can be achieved by replacing the pseudo-random numbers in the Monte Carlo method with deterministic numbers with the low-discrepancy property. An early application of low-discrepancy sequences is the quiet start method in particle-in-cell simulations, [10, 11] where quasi-random points are used for sampling the initial particle distribution.

The asymptotic convergence of the quasi-Monte Carlo method is,

$$O(N^{-1} \log(N)^s),$$

where $s$ is the number of dimensions. For modest dimensions the $\log(N)^s$ term can be ignored. The convergence is dependent on the number of dimensions, forcing a greater number of particles to be used as the number of dimensions increase. For integration problems over the unit hypercube we require at least $N \geq \exp(s)$ number of particles, (from $\partial_N N^{-1} \log(N)^s = 0$), for (1) to converge. The number of particles will blow up as the number of dimensions increase. This suggest that the quasi-Monte Carlo method is only efficient for small dimensions and suffer from the curse of dimensionality. Fortunately it has been shown in [12] and reference therein, that the quasi-Monte carlo method have better convergence in practice than theory predicts. This can partly be explained by the fact that a large class of integrands have a so called low effective dimension; the function to be integrated has most of its variation located in few dimensions.

The purpose of this paper is to test the applicability of the quasi-Monte Carlo method for simulation of the fast-ion thermalization process for a neutral beam injection scenario. In this paper the method of Brownian bridges and the method in [13], here called 'the sorting and mixing method', is tested for the unscrambled and scrambled
Faure sequence, which is a quasi-random sequence in the family of \((t, s)\)-sequences. In section 1 we will briefly introduce the Fokker-Planck equation and its connection to stochastic differential equations. In section 2 the quasi-Monte Carlo method is introduced where the concept of low-discrepancy sequence, randomization and effective dimensions are briefly touched upon. The sorting and mixing method and the method of Brownian bridges are presented in section 2.5 and 2.6. In section 3 we derive the stochastic differential equation for the fast-ion thermalization process, which is followed by the simulation results in section 4 and ending with conclusions.

1. The Fokker-Planck equation

Consider the Fokker-Planck equation for particle interactions in \(s\) coordinates \(x = x_1, \ldots, x_s\),

\[
\frac{\partial f}{\partial \tau} = L(f), \quad f : (y, \tau) \in \mathbb{R}^s \times [0, T] \mapsto \mathbb{R}
\]

\[
L = -\sum_i \frac{\partial}{\partial y_i} A_i(y) + \frac{1}{2} \sum_{i,j} \frac{\partial^2}{\partial y_i \partial y_j} B_{ij}(y)
\]

with initial condition \(f(y, 0) = \delta(x - y)\) where \(A_i\) and \(B_{ij} = \sum_{l=1}^s \sigma_{il} \sigma_{jl}\) are the drift vector and diffusion tensor respectively both independent of time. The 'characteristics' of (2) is described by an Itô stochastic differential equation (SDE)

\[
dX(t) = A(X(t))dt + \sigma(X(t))dW(t), \quad X_0 = x
\]

where \(dW(t)\) is the Wiener process also known as Brownian motion, with normal distributed components having zero mean and \(dt\) variance. Many realizations of the SDE give a distribution of particles, which is the solution of (2). A common time-discretization method of the above SDE is the Euler-Maruyama scheme [14]. Let \(i = \{1, \ldots, I\}\) and define the end time by \(T = I\Delta t\) and assume \(X \in \mathbb{R}\); then the Euler-Maruyama scheme of (3) is,

\[
X_{i+1} = X_i + A(X_i)(t_{i+1} - t_i) + \sigma(X_i)(W_{i+1} - W_i)
\]

where \(\Delta W_i = W_{i+1} - W_i\) are normally distributed pseudo-random numbers with zero mean and \((t_{i+1} - t_i)\)-variance. In the following section we will discuss how the pseudo-random numbers can be replaced with quasi-random sequences.

2. The quasi-Monte Carlo method

The naive approach of replacing the pseudo-random numbers with quasi-random numbers will not work for simulation of SDEs since the quasi-random numbers are deterministic and the correlation will introduce artificial drift of the the particles. However, it is possible to use the quasi-random points, if combined with methods
that break the correlation. When using the quasi-Monte Carlo method for SDEs the term *dimensions* has a different meaning than *the physical number of dimensions*. The number of dimensions of a process, $X(t)$, which solves a certain SDE is the number of time steps times the number of physical dimensions. From here onward we let $s$ denote the physical number of dimensions. This is most easily explained by the following example.

**Example**

Consider a one-dimensional process $X(t)$ satisfying the scalar SDE $dX = \sigma(X)\,dW$ with initial condition $X_0 = x$. We would like to simulate this SDE for three time steps using the discretization of Euler-Maruyama. Unrolling the SDE we obtain,

$$
X_0 = x \\
X_1 = X_0 + \sigma(X_0)\sqrt{\Delta t}\xi_1 = X_1(X_0, \xi_1) \\
X_2 = X_1(X_0, \xi_1) + \sigma(X_1(X_0, \xi_1))\sqrt{\Delta t}\xi_2 = X_2(X_0, \xi_1, \xi_2) \\
X_3 = X_2(X_0, \xi_1, \xi_2) + \sigma(X_2(X_0, \xi_1, \xi_2))\sqrt{\Delta t}\xi_3 = X_3(X_0, \xi_1, \xi_2, \xi_3)
$$

where $\xi_i \in \mathcal{N}(0, 1)$, are normally distributed random numbers with zero mean and unit variance. Clearly after three time steps the process $X$ is dependent on three random-numbers. Now assume that we are interested in calculating an expected value of the process $\langle g(X(t_3)) \rangle$ for a known function $g(\cdot)$. We know that the normally distributed random numbers $\xi_i$ is related to uniformly distributed numbers $z_i \in U(0, 1)$ over the unit interval $[0, 1)$ by the inverse cumulative Normal distribution $\xi = \Phi^{-1}(z)$. Using this in the above sequence we obtain,

$$
X_0 = x \\
X_1 = X_0 + \sigma(X_0)\sqrt{\Delta t}\Phi^{-1}(z_1) = X_1(X_0, z_1) \\
X_2 = X_1(X_0, z_1) + \sigma(X_1(X_0, z_1))\sqrt{\Delta t}\Phi^{-1}(z_2) = X_2(X_0, z_1, z_2) \\
X_3 = X_2(X_0, z_1, z_2) + \sigma(X_2(X_0, z_1, z_2))\sqrt{\Delta t}\Phi^{-1}(z_3) = X_3(X_0, z_1, z_2, z_3)
$$

The expected value of $g(X_3)$ is now a four dimensional integration,

$$
\langle g(X_3) \rangle = \int_{\mathbb{R}} \int_{[0,1]^3} \delta(x - y)g(x_3(z_1, z_2, z_3; y))dz_1dz_2dz_3dy.
$$

(5)

From this example we clearly see how the number of dimensions is dependent on the number of physical dimensions and the number of time steps. An estimate of the expected value is obtained by sampling the hypercube with a finite number of particles,

$$
\langle g(X_3) \rangle \approx \frac{1}{N} \sum_{j=1}^{N} g(x_3(z_1^j, z_2^j, z_3^j)).
$$

(6)
If the hypercube is sampled with quasi-random points, we obtain a more accurate estimate of the expected value than if we would use pseudo-random points since quasi-random points are more uniformly distributed than pseudo-random numbers and do not form clusters, which is a common effect for pseudo-random numbers.

2.1. Low discrepancy sequences

There exist three major classes of low-discrepancy sequences. These are the (digital) nets, sequences and lattice rules. In this paper we will use the Faure sequence, which is a \((0, s)\)-sequence in base \(b\), where \(b\) is the smallest prime number greater than the number of dimensions \(s\). Similar to pseudo-random numbers the law of large numbers must hold for these sequences. A deterministic version of the law of large numbers is provided by the Koksma-Hlawka inequality, which give a bound on the quadrature error,

\[
\left| \int_{[0,1]^s} g(z) dz_1 dz_2 \ldots dz_s - \frac{1}{N} \sum_{j=1}^{N} g(Z^j) \right| \leq D^*_N V_{HK}(g) \tag{7}
\]

where \(V_{HK}(g)\) is bounded variation of \(g\) in the sense of Hardy and Krause. The exact definition of the Hardy and Krause variation contains higher order derivatives of \(g\). Therefore the smoothness of \(g\) determines the size of \(V_{HK}\) and is small for smooth functions. \(D^*_N\) is called the discrete star discrepancy and it measures how uniformly the samples are distributed over the \(s\)-dimensional unit cube. A sequence with a small \(D^*_N\) is called a low-discrepancy sequence. The discrete star discrepancy is defined as,

\[
D^*_N(Z_1, \ldots, Z_N) := \sup \left| \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{0 \leq Z_j < p} - \text{vol}([0, p]) \right| \tag{8}
\]

where \(\mathbb{1}\) is the indicator function and is 1 at \(Z_j\), inside the box \([0, p]^s\) and zero otherwise. The second term, \(\text{vol}\) measures the volume of the \(s\)-dimensional box \([0, p]^s\). An illustration in 2D is given in figure 1.

2.2. Randomization

Scrambling is a randomization method for obtaining new uncorrelated sequences generated from a common mother \((t, s)\)-sequence and can partly break the correlation between points from the same sequence. There are many different randomization methods, the simplest method is the so called linear scrambling also known as generalized Faure [15, 16]. In short, the \(n\)th \(s\)-dimensional quasi-random point is generated by,

\[
X_n = [\Phi_b(L^{(1)}P^0n), \Phi_b(L^{(2)}P^1n), \ldots, \Phi_b(L^{(s)}P^{s-1}n)]
\]

where \(\Phi_b(n) = \frac{n_0}{b} + \frac{n_1}{b^2} + \ldots + \frac{n_m}{b^m}\), is the so called radical inverse. The vector \(n = [n_m, \ldots, n_0]\) is the b-adic representation of the scalar \(n\), \((n = n_mb^m + \ldots + n_1b + n_0)\). The matrix \(P^s\) is the \(s\)th power of the Pascal matrix modulo \(b\) where the \((k, l)\)-element of
Figure 1. Illustration of the discrepancy for the first 50 points of the two-dimensional Faure sequence in base 3. The area of the smallest box, plotted in the figure, is $0.3^2 = 0.09$ and the number of points in this box is 5, which gives an estimate of $1/N \sum I_{box} = 5/50 = 0.1$. The discrepancy for this box is $|0.1 - 0.09| = 0.01$. Similarly the area of the second smallest box is $0.4^2 = 0.16$ and the number of points is 8, which gives $8/50 = 0.16$ and a discrepancy of 0. The star discrepancy measures the maximum difference between the number of points and the volume of the box over all possible boxes with one corner in the origin.

$P$ is equal to $\binom{l-1}{k-1} \mod b$. The matrices, $L^{(s)}$ are independent lower triangular matrices with diagonal elements selected randomly from $\{1, \ldots, b-1\}$ and the other elements chosen randomly from $\{0, \ldots, b-1\}$. For a more in depth treatment on randomized digital-nets we refer to [17, 18, 19, 20].

Figure 2. Two-dimensional projection plot (dim 37, dim 38) of the first hundred points from the 38-dimensional Faure sequence before (left) and after scrambling (middle) and a plot of hundred pseudo-random points (right).

2.3. Effective dimension

High dimensional problems often have low-effective dimensions, they depend on few very important dimensions. This property was first analyzed in [21, 22] where the
authors introduced a definition of effective dimension by the ANOVA decomposition of the integrand. For a \( s \)-dimensional function \( f \) the ANOVA decomposition is given by a sum of projections on to lower-dimensional hyperplanes in the \( s \)-dimensional unit cube. The measure of effective dimension is defined as,

\[
\sum_{p \subseteq \{1,2,\ldots,s\}} \text{Var}(f_p) \geq 0.99 \text{Var}(f) \quad (9)
\]

where \( p \) is a subset of the dimension indices and the function \( f_p \) is nonzero on the dimension indices in \( p \). The integrand is said to have \( s' = \max p \) effective dimensions if it can be well approximated by a linear combination of length(\( p \)) functions such that 99\% percent of the variance is resolved. For a more in-depth treatment on the reduction of effective dimension for SDEs we refer to [23].

2.4. QMC for diffusion

The quasi-Monte Carlo method for simulation of diffusion can be used in two ways, not to different in spirit to the Klimontovich and Liouville representation of plasmas with respect to dimensions. In the first concept the Brownian motion is simulated using an \( s \times I \)-dimensional sequence. Here \( I \) is the number of time steps. The other alternative is to use a \( 2s \)-dimensional sequence. Successive realizations of the Brownian motion is obtained by shifting the index pointer to the sequence with \( N \), e.g. use the \( N \) first points for the first time-step and use the \([N + 1, 2N] \) points for the second time-step and so forth. What is important for both methods is that the realizations should be uncorrelated. In the beginning of this section we mentioned the importance of breaking the correlation and pointed out that scrambling can partly break the correlation. This is only true in high-dimensions. The difference in correlation is illustrated in figure 3(a) and in figure 3(b) where we have plotted the auto-correlation between elements of 50 one-dimensional scrambled Faure points and the auto-correlation between elements of one 50-dimensional point. The figures show that correlation is very strong between points in the one-dimensional case but not very strong in the 50-dimensional case, which is of the same order as for a standard pseudo-random generator. Scrambling is apparently not sufficient for breaking correlation in the low-dimensional case without using more advanced techniques. In [13], a method for resolving this issue has been suggested. It is based on sorting and mixing the particles in a clever way such that decorrelation is ensured, which is the topic of the next section.

2.5. The sorting and mixing method

The idea of the sorting and mixing method is to break the correlation by sorting the particles along each dimension. We will give a simplified presentation of the method. For an in-depth treatment of the concept we refer to [13]. The foremost drawback of this method is that it requires at least \( b^s \) number of particles being simulated where \( b \) is
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

Figure 3. Difference in auto-correlation between 50 one-dimensional scrambled Faure points and one 50-dimensional scrambled Faure point.

the least prime greater than $2s$. In six dimensions this would give $13^6 \approx 4 \times 10^6$ number of particles. For higher dimensions this value will blow up.

The first step of the sorting and mixing method is to treat the drift and diffusion part of (3) separately,

\begin{align*}
X_{i+1/2} &= X_i + A(X_i)(t_{i+1} - t_i) \\
X_{i+1} &= X_{i+1/2} + \sigma(X_i)(W_{i+1} - W_i)
\end{align*}

The drift (10) is evaluated with the standard forward Euler method. For the diffusion (11) part we need to generate quasi-random numbers for the Wiener process. Let $N = b^m$ be the number of particles to be simulated. Here $b$ is the least prime greater than $2s$, e.g. for $s = 2 \Rightarrow b = 5$, and let $d_1, \ldots, d_s$ be integers greater than zero such that $m = d_1 + \ldots + d_s$. For a two-dimensional diffusion process a minimum of $N = 5^2$ particles are needed.

- **Sorting**: First sort the particles in ascending order given the magnitude of the first coordinate of the particles into $b^{d_1}$ groups. Secondly sort the particles in each group in ascending order given the magnitude of the second coordinate into $b^{d_2}$ subgroups. For a three dimensional problem sort the particles in each $b^{d_2}$ subgroup, in ascending order given the magnitude of the third coordinate of the particles into $b^{d_3}$ sub-subgroups. Continue sorting if dimensions are greater than three. This is a computationally demanding operation requiring $O(N \log N)$ operations (with quick-sort) for each sort.

- **Mixing**: Consider a vector $y = (y_1, \ldots, y_{2s})$ from a $(0, 2s)$-net in base $b$ and define two selection functions $P'$ and $P''$ by $P'y := (y_1, \ldots, y_s)$ and $P''y := (y_{s+1}, \ldots, y_{2s})$. Use the first $s$ dimension of the $(0, 2s)$-net for mixing. Let $a(y) := ([b^{d_1}y_1], \ldots, [b^{d_s}y_s])$ where the floor function, $\lfloor \cdot \rfloor$, returns the greatest
integer less than or equal to the argument. Define the one-to-one map between array indices using the first \( s \) variables of the \((0,2s)\)-net, \(j \to a(P'y_{iN+j})\). The map is from \([0,N]\) into \([0,b^{d_1}) \times \ldots \times [0,b^{d_s})\), e.g. in two dimension we map the vector \([0,N]\) to the matrix \([0,b^{d_1}) \times [0,b^{d_2})\). Particle with array index \(a(P'y_{iN+j})\) is pushed by the quasi-random points, \(P''y_{iN+j}\).

The sorting and mixing procedures are illustrated in figures 4(a)-4(b) for \(s = 2\). It was proved in [13] that this construction have the following bound,

\[
D_N^*(X_j^i) \leq D_N^*(X_0^j) + b^{d_1+\ldots+d_{s-1}+[d_s/2]} \sum_{k=0}^{i-1} D_N^*(Y^k) + i \left( \frac{1}{b^{d_1}} + \ldots + \frac{1}{b^{d_{s-1}}} + \frac{1}{b^{[d_s/2]}} \right)
\]  

\(12\)
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

where $0 \leq j < N$ and $D_N(Y^k)$ is the star discrepancy of the $(0,2s)$-sequence, $Y^k$ in base $b$. Since $N$ is defined as the product of $b^d$, the second term converges in the limit of infinite number of particles. Note that for a finite $N$ the bound increases with the number of time steps, $i$. To keep the error fixed, more particles are needed as the simulation time increases.

2.6. The method of Brownian bridges

In this section we will turn the attention to the $s \times I$-dimensional case. If many dimensions are used the number of effective dimensions become important and it has been argued in [12, 21], that the midpoint Brownian bridge method effectively reduces the effective dimensions. The general Brownian bridge formula is defined as

$$W_{t_j} = (1 - \lambda)w_{t_i} + \lambda w_{t_k} + (\lambda(1 - \lambda)(k - i)\Delta t)^{1/2} Z$$

(13)

where $\lambda = (j - i)/(k - i)$ for $t_i < t_j < t_k$ and $Z$ is a normally distributed random number. The variables $w_{t_i}, w_{t_k}$ are previous realization of the Wiener process. The midpoint Brownian bridge formula is obtained from the special case where the time axis is populated by successive midpoint splits. First generate the start-point $w_0$ and end-point $w_T$. Then draw the midpoint and continue splitting, see figure 5. The generated order is:

$w_0, w_T, w_{T/2}, w_{T/4}, w_{3T/4}, \ldots, w_{T-1}$. One interesting note is that the midpoint Brownian bridge construction reduces the variance by a factor of 2 for each split. This can be seen from (13),

$$\text{Var}[W_{k/2}] = 1/4k\Delta t, \quad t_0 < t_{k/2} < t_k$$

$$\text{Var}[W_{k/4}] = 1/4k/2\Delta t, \quad t_0 < t_{k/4} < t_{k/2}$$

$$\vdots$$

The success of the Brownian bridge method can be explained by the way it utilize low-discrepancy sequences. One important property of low-discrepancy sequences is that
they are typically more uniformly distributed in lower dimensions than in higher. This is efficiently used by the Brownian bridge construction since the first few quasi-random points describe the complete Brownian path on a coarse level. It should be noted that the Brownian bridge method does not always perform better than the standard Monte Carlo method. For example in [24] it was shown that the performance of the Brownian bridge construction is not consistent and can actually perform worse than the standard Monte Carlo method for certain types of integrands. Thus the Brownian bridge method is problem dependent and have the potential of generating many papers.

3. Derivation of the SDE for fast-ion thermalization

In this section we derive a SDE for a simple model of fast-ion thermalization in fusion plasmas given by the Fokker-Planck equation with the Coulomb collision operator of Spitzer, [25],

\[
\frac{\partial f}{\partial t} = -\frac{1}{v^2} \frac{\partial}{\partial v} \left[ v^2 \langle \Delta v \rangle + \frac{v}{2} \langle (\Delta v^\perp)^2 \rangle \right] f + \frac{1}{2v^2} \frac{\partial^2}{\partial v^2} \left[ v^2 \langle (\Delta v^\parallel)^2 \rangle \right] f + \frac{1}{4v^2} \frac{\partial}{\partial \xi} (1 - \xi^2) \frac{\partial}{\partial \xi} \left[ \langle (\Delta v^\perp)^2 \rangle \right] f. \tag{14}
\]

The above equation is not on the form of (2) due to the Jacobian \(v^2\) therefore the drift \(A\) and diffusion coefficients \(\sigma\) cannot be read off directly from the Feynman-Káč formula. Instead the coefficients is obtained by calculating moments of the the Coulomb operator,

\[
C(\cdot) = -\frac{1}{v^2} \frac{\partial}{\partial v} \left[ v^2 \langle \Delta v \rangle + \frac{v}{2} \langle (\Delta v^\perp)^2 \rangle \right] + \frac{1}{2v^2} \frac{\partial^2}{\partial v^2} \left[ v^2 \langle (\Delta v^\parallel)^2 \rangle \right] + \frac{1}{4v^2} \frac{\partial}{\partial \xi} (1 - \xi^2) \frac{\partial}{\partial \xi} \left[ \langle (\Delta v^\perp)^2 \rangle \right].
\]

The first two moments of the above operator is given by,

\[
\frac{d}{dt} \langle v \rangle = \int \int v C(f) f dv d\xi = \alpha(v) \tag{15}
\]

\[
\frac{d}{dt} \langle \xi \rangle = \int \int \xi C(f) f dv d\xi = -\frac{1}{2v^2} \xi \gamma(v) \tag{16}
\]

\[
\frac{d}{dt} \langle v^2 \rangle = \int \int v^2 C(f) f dv d\xi = 2v \alpha(v) + \beta(v) \tag{17}
\]

\[
\frac{d}{dt} \langle \xi^2 \rangle = \int \int \xi^2 C(f) f dv d\xi = \frac{1}{2v^2} (1 - 3\xi^2) \gamma(v) \tag{18}
\]

\[
\frac{d}{dt} \langle v \xi \rangle = \int \int v \xi C(f) f dv d\xi = \xi \langle \Delta v \rangle, \tag{19}
\]

where we have introduced the variables, \(\alpha(v) = \langle \Delta v \rangle + \frac{1}{2v^2} \langle (\Delta v^\perp)^2 \rangle\), \(\beta(v) = \langle (\Delta v^\parallel)^2 \rangle\), \(\gamma(v) = \langle (\Delta v^\perp)^2 \rangle\). Armed with the equations above, the covariance matrix is calculated from the following general formula,

\[
\sigma_{ij}^2 = \frac{d}{dt} \langle X_i X_j \rangle - X_i \frac{d}{dt} \langle X_j \rangle - X_j \frac{d}{dt} \langle X_i \rangle. \tag{20}
\]
Inserting (15)-(19) in the equation above we obtain the following,
\[
\begin{align*}
\sigma^2_{11} &= 2v\langle (\Delta v_{||})^2 \rangle + \langle (\Delta v_{\perp})^2 \rangle + \langle (\Delta v_{||})^2 \rangle \\
&= \beta(v) \\
\sigma^2_{12} &= \sigma^2_{21} = 0 \\
\sigma^2_{22} &= \frac{\gamma(v)}{2\sigma^2}(1 - \xi^2).
\end{align*}
\]

Since the nonzero elements of the covariance matrix is on the matrix diagonal, the diffusion coefficient \(\sigma\) is immediately obtained. The elements of the drift vector are given by,
\[
\begin{align*}
A_1 &= \alpha(v) \\
A_2 &= -\frac{1}{2\sigma^2} \xi \gamma(v).
\end{align*}
\]

The derived drift and diffusion coefficients is in \(v, \xi\) coordinates. Using the Itô formula
\[
dg = \left( A(V) \partial_v g(V) + \frac{\sigma(v)^T \sigma(V)}{2} \partial_{vv} g(V) \right) dt + \sigma(V) \partial_V g(V) dW
\]
and after simplification,
\[
dE = \left( A(V)mV + 0.5m \sigma^2(V) \right) dt + \sigma(V)mV dW.
\]

Inserting the derived moments give,
\[
dE = \left( \alpha m^{1/2}(2E)^{1/2} + m/2\beta \right) dt + (2mE\beta)^{1/2} dW
\]
where the argument of \(\alpha, \beta\) is \(\alpha = \alpha \left( \sqrt{2E/m} \right), \beta = \beta \left( \sqrt{2E/m} \right)\). Together with the pitch-angle scattering process we have,
\[
\begin{align*}
dE &= \left( \alpha m^{1/2}(2E)^{1/2} + m/2\beta \right) dt + (2mE\beta)^{1/2} dW \\
d\xi &= -\frac{m}{4E} \xi dt + \left( \frac{m\gamma}{4E} (1 - \xi^2) \right)^{1/2} dW
\end{align*}
\]
with \(\gamma = \gamma \left( \sqrt{2E/m} \right)\). This system is solved with the Euler-Maruyama scheme,
\[
\begin{align*}
E_{n+1} &= E_n + (\alpha(E_n)m^{1/2}(2E_n)^{1/2} + m/2\beta(E_n)) dt \\
&\quad + (2mE_n\beta)^{1/2} \sqrt{\Delta t} Z_1 \\
\xi_{n+1} &= \xi_n - \frac{m\gamma(E_n)}{4E_n} \xi_n dt + \left( \frac{m\gamma(E_n)}{4E_n} (1 - (\xi_n)^2) \right)^{1/2} \sqrt{\Delta t} Z_2
\end{align*}
\]
where \(Z_1, Z_2\) are normal distributed random numbers with zero mean and unit variance.

4. Application to simulation of neutral beam injection

We have simulated the fast-ion thermalization process in energy and pitch-angle (26), for a neutral beam scenario with a continuous injection of particles. The simulation was run to an approximate stationary state of the distribution function. For stationary
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

distributions, the neutral beam source can be modeled by either injecting particles at the source energy each time-step and remove particles that are considered thermalized or start with a fixed number of particles and save the complete path of the particles. In this work we have used the second approach. Four methods have been tested, the standard Monte Carlo, the sorting and mixing method, the Brownian bridge method and a naive method where the pseudo-random numbers are replaced with Faure numbers, scrambled and unscrambled without any extra modification. Simulation parameters are: neutral beam injection source at 100 keV for a 4 keV plasma with initial pitch-angle at, $\xi = -0.8$. The simulation was considered stationary at $T = 2^{-1}s$ and the time increment was set to $\Delta t = 2^{-8}$, which gives $2^7$ time steps. The parameters for the sorting and mixing method are, $d_1 = \{1, 2, 2, 2, 2\}$, $d_2 = \{1, 1, 2, 3, 4\}$ and $b = 5$ for $N = b^{(d_1+d_2)}$. The Faure sequence was scrambled to a depth of 30-digits. The tests was conducted in Matlab$^\text{TM}$ version R2010b on a twelve core Intel machine @ 2.67 GHz with 62 gigabyte of RAM. Since the entire particle trajectory was saved, a massive amount of memory was required with a maximum of $5^9 \times 2^{14} = 32 \times 10^9$ particles. The exact solution for this test case is unknown. Therefore we have measured the convergence in terms of sample variance of the sample mean for a batch of 20 runs of each simulation with different initial seed of the randomly scrambled Faure sequence. Two different sample-means were considered, the parallel velocity $\langle g(E, \xi) \rangle = \langle v_\parallel \rangle = \langle v\xi \rangle = \langle \xi \sqrt{2E/m} \rangle$, which is a two-dimensional quantity and the energy squared $\langle g(E) \rangle = \langle E^2 \rangle$. The sample mean is calculated for each $j = 1 \ldots M = 20$ by,

$$\hat{\mu}^j = \frac{1}{N} \sum_{i=1}^{N} g(E_i^j, \xi_i^j)$$

with $N$ particles. The batch-mean value is calculated from,

$$\hat{\mu} = \frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} g(E_i^j, \xi_i^j)$$

and the variance of the batch-mean is given by,

$$\hat{\sigma}^2 = \frac{1}{M-1} \sum_{j=1}^{M} (\hat{\mu}_j - \hat{\mu})^2.$$  

From these expressions we have calculated the 99% confidence interval for the batch-mean from the Student t-distribution with $M - 1$ degrees of freedom,

$$\hat{\mu} \pm t_{99\%, M-1} \sqrt{\hat{\sigma}^2/M}$$

where $t_{99\%, M-1} = 2.86$. Simulation results of the parallel velocity for the test cases is given in table 1 and for the energy squared test case in table 2.
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

<table>
<thead>
<tr>
<th>( N )</th>
<th>MC</th>
<th>Brownian bridge</th>
<th>Sorting &amp; mixing</th>
<th>Scrambled Faure</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 5^2 )</td>
<td>-2.5862 ± 0.0801</td>
<td>-2.5922 ± 0.1319</td>
<td>-2.6189 ± 0.0073</td>
<td>-2.5853 ± 0.1008</td>
</tr>
<tr>
<td>( 5^3 )</td>
<td>-2.5761 ± 0.0337</td>
<td>-2.5875 ± 0.0356</td>
<td>-2.5923 ± 0.0042</td>
<td>-2.5701 ± 0.0467</td>
</tr>
<tr>
<td>( 5^4 )</td>
<td>-2.5783 ± 0.0192</td>
<td>-2.5799 ± 0.0080</td>
<td>-2.5832 ± 0.0021</td>
<td>-2.5748 ± 0.0068</td>
</tr>
<tr>
<td>( 5^5 )</td>
<td>-2.5807 ± 0.0068</td>
<td>-2.5797 ± 0.0020</td>
<td>-2.5806 ± 0.0007</td>
<td>-2.5808 ± 0.0014</td>
</tr>
<tr>
<td>( 5^6 )</td>
<td>-2.5804 ± 0.0033</td>
<td>-2.5806 ± 0.0003</td>
<td>-2.5804 ± 0.0002</td>
<td>-2.5819 ± 0.0004</td>
</tr>
<tr>
<td>( 5^9 )</td>
<td>-2.5805 ± 0.0002</td>
<td>-2.5804 ± 0.0008</td>
<td>-2.5804 ± 0.0008</td>
<td>-2.5804 ± 0.0008</td>
</tr>
</tbody>
</table>

Table 1. Computational results for \( \langle v_\xi \rangle \) with the different methods. Scrambled Faure sequences have been used for all tested methods in the table and the values of the parallel velocity have been normalized by \( 10^6 \). The result \( 5^9 \) in the last entry of the first column have been obtained with \( M = 180 \), as a reference value.

<table>
<thead>
<tr>
<th>( N )</th>
<th>MC</th>
<th>Brownian bridge</th>
<th>Sorting &amp; mixing</th>
<th>Scrambled Faure</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 5^2 )</td>
<td>5.1802 ± 0.2026</td>
<td>5.2769 ± 0.4364</td>
<td>5.07933 ± 0.01355</td>
<td>5.1869 ± 0.3130</td>
</tr>
<tr>
<td>( 5^3 )</td>
<td>5.0672 ± 0.0923</td>
<td>5.1500 ± 0.1631</td>
<td>5.08620 ± 0.00717</td>
<td>5.2554 ± 0.1334</td>
</tr>
<tr>
<td>( 5^4 )</td>
<td>5.0673 ± 0.0354</td>
<td>5.1096 ± 0.0365</td>
<td>5.09003 ± 0.00124</td>
<td>5.1792 ± 0.0206</td>
</tr>
<tr>
<td>( 5^5 )</td>
<td>5.0929 ± 0.0154</td>
<td>5.0923 ± 0.0060</td>
<td>5.09075 ± 0.00013</td>
<td>5.1010 ± 0.0047</td>
</tr>
<tr>
<td>( 5^6 )</td>
<td>5.0902 ± 0.0085</td>
<td>5.0913 ± 0.0007</td>
<td>5.09058 ± 0.00003</td>
<td>5.0912 ± 0.0011</td>
</tr>
<tr>
<td>( 5^9 )</td>
<td>5.0908 ± 0.0007</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( 5^9 )</td>
<td>5.09059 ± 0.00026</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2. Computational results for \( \langle E^2 \rangle \) with the different methods. Scrambled Faure sequences have been used for all tested methods in the table and the values have been normalized by \( 10^3 \). The result \( 5^9 \) in the last entry of the first column have been obtained with \( M = 180 \), as a reference value.

![Figure 6](image_url)

(a) Parallel velocity, \( \langle v_\xi \rangle \)  
(b) Energy squared, \( \langle E^2 \rangle \)

Figure 6. Normalized root mean square convergence of the expected value of the parallel velocity and the energy squared.
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

The normalized root mean square error, n.r.m.s = $\sqrt{\sigma^2/(\hat{\mu}\sqrt{M})}$ for the two test cases is plotted in figures 6(a) and 6(b). In these figures, $(BB)$ correspond to the method of Brownian bridges and $(SM)$ is the method of sorting and mixing. The histogram of the stationary distribution for the different methods are plotted in the figures 7(a)-10(b). A reference distribution, obtained from a standard Monte Carlo simulation, is plotted in figure 7(a). In figure 7(b), the distribution of the naive method with unscrambled Faure points is plotted. The method does not converge as expected since the correlation in the Faure sequence introduce spurious drifts of the particles. The result is slightly improved if combined with the Brownian bridge method, see figure 8(a) but the distribution does not agree with the reference figure 7(a). The sorting and mixing method is the only method with a theoretical proof of the convergence for unscrambled Faure points. The distribution for the sorting and mixing method is plotted in figure 8(b) and agrees with the standard Monte Carlo distribution. The distribution from a simulation with scrambled Faure sequences is plotted in figure 9. From this figure we can see that scrambling reduces the correlation. Even though the distribution of the naive scrambled Faure method is in agreement with the MC reference distribution, the moments are not in agreement for the case $\langle v\xi \rangle$ as seen in table 1. The distribution for the scrambled Faure with the Brownian bridge method is plotted in figure 10(a) and it is in agreement with the MC reference. The distribution for the sorting and mixing method is plotted in figure 10(b) and agrees with the MC reference. The sorting and mixing method and the Brownian bridge method perform much better than standard Monte Carlo for $2^7$ time steps.

![Graphs](image.png)

(a) The standard Monte Carlo   (b) The naive method with unscrambled Faure points

**Figure 7.** Histogram plot of the distribution function obtained with the standard Monte Carlo (left) and with the naive method (right)

4.1. Very long-time simulation

In order to test the applicability for long-time particle simulations we have estimated the slope of the convergence with a least-squares fit for different number of time steps. The
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

Figure 8. Histogram plot of the distribution function obtained with the Brownian bridge method with unscrambled Faure points (left) and with the sorting and mixing method with unscrambled Faure points (right)

Figure 9. Histogram plot of the distribution function obtained with the naive method with scrambled Faure points

result is plotted in figure 11. The results indicate that the sorting and mixing method perform worse than the standard Monte Carlo above 400 time steps when measuring parallel velocity but remain very low for $\langle E^2 \rangle$ for all tested values. The convergence of the Brownian bridge method degrade almost linearly for both measured quantities but the convergence is better than the standard Monte Carlo even for thousand time steps. A rough extrapolation of the slope for the Brownian bridge method indicate that it will perform worse than the standard Monte Carlo at about 1700 time steps.

Finally we have tested the methods for a very large number of time steps, $I = 2^{14} = 16384$. The computational results are given in tables 3 and 4. For this value the number of dimensions of the hypercube is $2^{15}$ for the Brownian bridge method. The theoretical number of particles required to uniformly cover the hypercube is astronomical, therefore we cannot expect the method to perform well. The sorting and mixing method cannot
Perform well either because the bound of the discrepancy (12) increases with the number of time steps, the sum in (12) is of the order of \( b^{d_1+\lfloor d_2/2 \rfloor} \times I \times D^*(Y^0) \approx 10^7 \times D^*(Y^0) \). The Brownian bridge method only give an acceptable estimate of \( \langle E^2 \rangle \) for \( N = 5^6 \) but have the same confidence interval as the standard Monte Carlo, table 4. The performance is slightly better for \( \langle v_\xi \rangle \) where the value is estimated correctly except for \( N = 5^2 \) but the confidence interval is smaller than the standard Monte Carlo at \( N = 5^6 \), table 3. The sorting and mixing method does not converge for the parallel velocity case, seen in table 3 and it gives inconsistent estimate for the energy squared case, table 4.
Randomized quasi-Monte Carlo simulation of fast-ion thermalization

5. Conclusion

The method of sorting and mixing and the Brownian bridge method is much better than the standard Monte Carlo method with a faster convergence and more accurate estimate up to thousand time steps for simulation of fast-ion thermalization. When very long simulations are required these methods can fail to give accurate estimate and they may not converge. For modest number of time steps ($2^{7}$) the Brownian bridge method give a convergence close to $O(N^{-1})$ and the sorting and mixing method about $O(N^{-0.6})$ for $\langle v_|| \rangle$ and approximately $O(N^{-1})$ for both methods for $\langle E^2 \rangle$. The sorting and mixing method is more accurate than the Brownian bridge method but is also more computationally demanding due to the sorting stage and require simultaneous simulation of an ensemble of particles. The sorting and mixing method is therefore not suitable for higher dimensional problems $s > 3$. The Brownian bridge method has similar convergence as the sorting and mixing method and can be used for evolution of single particles, but the method can converge to the wrong value for a large number of time steps with few particles and depends strongly on the measured quantity. The Brownian bridge method and the sorting and mixing method combined with the scrambled Faure sequence are very efficient within their working domain.

6. References


Randomized quasi-Monte Carlo simulation of fast-ion thermalization


