Parameterization of Zinterhof Sequences for GRID-based QMC Integration

Heinz Hofbauer* and Andreas Uhl† and Peter Zinterhof‡

Abstract. *We have already investigated blocked and leaped substreams of the Zinterhof sequence for the use in the GRID. When comparing this sequence to others they have either the drawback of being computationally expensive (e.g. Niederreiter/Xing sequence) or have some kind of problematic defect (e.g. Sobol' sequence). In contrast, the Zinterhof sequence behaves well and is computationally inexpensive. In this paper we will demonstrate the use of parameterized Zinterhof sequences, where each processing element uses a different point generating sequence. In this context, the Zinterhof sequence is used on all processing elements but with a shift in dimension to generate low correlated sequences. Additionally the benefits of using parameterization in the GRID are discussed.*

1. Introduction

High dimensional numerical integration problems may require a significant amount of computation power. Therefore, substantial effort has been invested in finding techniques for performing these computations on all kinds of parallel architectures (see [5, 9, 10, 13] for an exhaustive overview). In order to minimize the communication amount within a parallel system, each processing element (PE) requires its own source of integration nodes. Therefore, the aim is to investigate techniques for using separately initialized and disjoint sets of integration nodes on a single PE.

Currently, the most efficient numerical techniques for evaluating high-dimensional integrals are based on Monte Carlo and quasi-Monte Carlo techniques [7]. Whereas in the Monte Carlo (MC) case the integration nodes are produced by a random number generator (RNG), low-discrepancy point sets and sequences (e.g. (t,m,s)-nets or (t,s)-sequences [11]) are employed in quasi-Monte Carlo (QMC) algorithms. QMC techniques improve the probabilistic error bounds of MC techniques especially in higher dimensions. Nevertheless, these techniques are related [6] since a full period random number sequence may be seen as a low-discrepancy point set (e.g. a rank-1 lattice rule in the case of a linear congruential generator) as well.

GRID environments exhibit a potentially high heterogeneity in terms of network capacity (i.e. bandwidth and latency) and computing speed (memory capacity, cache sizes, processor speed). In addition to that, these environments are error prone with respect to broken network links or failing PEs. Moreover, additional resources may become available during an ongoing simulation which should be used to optimize resource consumption. As a consequence, the following requirements should be met by a QMC technique employed in a GRID environment:

*Department or Computer Sciences, University of Salzburg, Salzburg, Austria, email: hhofbaue@cosy.sbg.ac.at
†Department or Computer Sciences, University of Salzburg, Salzburg, Austria, email: uhl@cosy.sbg.ac.at
‡Department or Computer Sciences, University of Salzburg, Salzburg, Austria, email: peter.zinterhof@sbg.ac.at

- Variety in computing speed requires dynamic load balancing capability.
- Variety in network capacity requires load balancing strategies without central organization and a minimal number of control messages exchanged among the computing nodes.
- Failure in hardware resources requires tolerance to lost partial results.
- Additional resources becoming available require a possibility to assign workload to these resources (i.e. by redistributing or redefining workload).

In addition to that, error bounds and computation results should preferably carry over from sequential execution. If the QMC point sets differ between sequential and parallel execution, the quality of the results needs to be investigated thoroughly. Reproducibility is as well an important issue to be considered.

So far, two entirely different strategies have been discussed in literature to employ QMC sequences in parallel and distributed environments.

1. Splitting a given QMC sequence into separately initialized and disjoint parts which are then used independently on the PEs. This strategy comes in two flavors:
   - **Blocking**: p disjoint contiguous blocks of maximal length l of the original sequence are used on the PEs. This is achieved by simply using a different starting point on each PE (e.g., PE$_i$, $i = 0, \ldots, p-1$, generates the vectors $x_{il}, x_{il+1}, x_{il+2}, \ldots, x_{il+l-1}$). In case a large number of smaller blocks is used index j is assigned dynamically to PE$_i$ which generates the vectors $x_j, x_{j+1}, \ldots, x_{j+l-1}$ (where j is incremented in steps of size l to avoid overlap).
   - **Leaping**: interleaved streams of the original sequence are used on the PEs. Each PE skips those points consumed by other PEs (*leap-frogging*) (e. g. employing p PEs, PE$_i$, $i = 0, \ldots, p-1$, generates the vectors $x_i, x_{i+p}, x_{i+2p}, \ldots$).
2. Using inherently independent sequences on the different PEs (denoted as "parameterization" which can be realized for example by randomization of given QMC sequences). The most important difference (and also disadvantage) of parameterization as compared to blocking and leaping is that the QMC point set used in parallel or distributed computation does not correspond to a single (sequentially used) point set. Therefore, the investigation of the results' quality when using this technique is of great importance since it is not clear a priori how results from different point sets will interact in the final result. The findings so far indicate a good quality of the results based on theoretical estimates and empirical tests. Due to the use of defacto independent QMC point sets on the PEs load balancing at low cost comes for free (each PE generates as much points as it requires locally) and the same is true for reacting to changes with respect to available resources. Therefore, parameterization is a well suited approach for GRID environments provided the quality of the results can be guaranteed. A disadvantage is that knowledge about the reliability of the results is restricted so far to scrambled Halton sequences ([12, 14]), (Korobov) lattice and Richtmeyr rules ([2, 4, 3, 1]). An advantage of using randomized QMC techniques in general is that error control techniques like variance reduction are integral parts of this approach. This also holds true for their use in GRID environments.

While parameterization exhibits many traits which are desirable there are also disadvantages concerning computational effort. All approaches so far use scrambled (randomized) sequences, which generate another overhead on the PEs. Since we trouble ourself with the circumstances of the GRID to gain more computational power for the integration it seems wasteful to loose some to the parallelization. What we propose is a parameterization approach which utilizes the Zinterhof sequence and is computationally efficient.

In Section 2. the parameterization approach is given along with a mathematical analysis of the required properties for the parameterization approach. To further study the results, numerical experiments are given in Section 3. and Section 4. gives a conclusion and summary of the findings.

2. Parameterization of Zinterhof Sequences

Zinterhof sequences ([15]) are a special case of Weyl sequences. Weyl sequences are defined by

$$x_n = (\{n\theta_1\}, \{n\theta_2\}, \ldots, \{n\theta_s\}) \; n = 1, 2, 3, \ldots$$

where s is the dimension and $\{x\}$ is the fractional part of x. It is well known that a Weyl sequence is uniformly distributed if and only if θ_i are independent irrational numbers. An important issue with respect to their quality in terms of uniformity of distribution is the amount or degree of irrationality of the employed starting vector $\Theta = (\theta_1, \ldots, \theta_s)$. For the Zinterhof sequence we set $\theta_i = e^{1/i}$ and consequently:

$$x_n = (\{ne^{1/1}\}, \ldots, \{ne^{1/s}\}) \; n = 1, 2, 3, \ldots. \tag{1}$$

The Zinterhof sequence has been shown to exhibit excellent distribution behavior and they excel by their ease of construction and implementation even for non-specialists.

For a parameterization approach however we need P s-dimensional sequences which are independent. This can be achieved by splitting a Ps-dimensional Zinterhof sequence into P s-dimensional sequences by splitting the generator $\vec{\theta}$. Let $\vec{\theta} = (e^{1/1}, \ldots, e^{1/s}, \ldots, e^{1/(P-1)s+1}, \ldots, e^{1/Ps})$ be the generator of the Ps-dimensional sequence than we can obtain the P s-dimensional generators $\vec{\theta_1}, \ldots, \vec{\theta_P}$ as follows

$$\vec{\theta_1} = (e^{1/1}, \ldots, e^{1/s}),$$

$$\vdots$$

$$\vec{\theta_P} = (e^{1/(P-1)s+1}, \ldots, e^{1/Ps}).$$

What we now need is assurance that the $\vec{\theta_1}, \ldots, \vec{\theta_P}$ are independent and that the independence is enough to grant a higher convergence rate that just using a single s-dimensional sequence.

Concerning the convergence rate, Ökten, et.al. [12] has given the following. For P PEs the estimation of the integral is

$$I'_N(f) = \frac{1}{N}\sum_{n=1}^{N} f(x_n) = \sum_{i=1}^{P} \frac{c_i}{N}\frac{1}{c_i}\sum_{n=1}^{c_i} f(x_n^{(i)}) = \sum_{i=1}^{P}\frac{c_i}{N} I'^{(i)}(f),$$

where $N = c_1 + \cdots + c_P$ and the ith sequence contributes the points $x_1^{(i)}, \ldots, x_{c_i}^{(i)}$ to the overall sequence $x_i, \ldots, x_N$. Now if the sequences are independent $I'_N(f)$ becomes an unbiased estimator for the integral

$I(f)$ with

$$\mathrm{Var}\left[I'_N(f)-I(f)\right]=E\left|I'_N(f)-I(f)\right|^2=\sum_{i=1}^{P}\left(\frac{c_i}{N}\right)^2 E\left|I'^{(i)}(f)-I(f)\right|^2\leq(V(f))^2\sum_{i=1}^{P}\left(\frac{c_i}{N}\right)^2(D^*_{c_i})^2,$$

which for equal speeds simplifies to

$$\mathrm{Var}\left[I'_N(f)-I(f)\right]=(D^*_{N/P})^2(V(f))^2/P. \tag{2}$$

The only thing left to do is to assure the independence of this parameterization approach.

It is well known [16] that the Weyl sequence

$$n(\theta_1,\ldots,\theta_s,\theta_{s+1},\ldots,\theta_{2s},\ldots,\theta_{(P-1)s+1},\ldots,\theta_{Ps})$$

fulfills the sequential test $D_N=\mathrm{O}(1/N^{1-\varepsilon})$ and a diaphony test of $F_N=\mathrm{O}(1/N^{1-\varepsilon})$ for almost all choices $\vec{\theta}=(\theta_1,\ldots,\theta_{Ps})$ and especially for $\vec{\theta}$ with θ_i of the form $\theta_i=e^{r_i}$ where $r_i\in\mathbb{Q}$,$r_i\neq r_j\neq 0$,$1\leq i,j\leq Ps$.

Independence of events is usually defined as follows: Let (Ω,α,P) be a probability space over the set Ω with the σ-algebra α of events with probability P. The events $E,F\in\alpha$ are independent if

$$P(E\cap F)=P(E)\cdot P(F).$$

Now let $E\sqsubseteq[0,1)^{2s}$,$F\sqsubseteq[0,1)^{2s}$ such that

$$E=\{(x_1,\ldots,x_{2s})|\alpha_i\leq x_i<\beta_i,\,i=1,\ldots,s,\,0\leq x_j<1,\,j=s+1,\ldots,2s\},$$

$$F=\{(x_1,\ldots,x_{2s})|0\leq x_i<1,\,i=1,\ldots,s,\,\alpha_j\leq x_j<\beta_j,\,j=s+1,\ldots,2s\}$$

and

$$E\cap F=\{(x_1,\ldots,x_{2s})|\alpha_i\leq x_i<\beta_i,\,i=1,\ldots,s,\,\alpha_j\leq x_j<\beta_j,\,j=s+1,\ldots,2s\}.$$

Thus, for the Lebesgue measure $P=\lambda$ being the natural probability on $I^{2s}:=[0,1)^{2s}$,

$$P(E\cap F)=P(E)\cdot P(F)$$

holds. So the events E and F are independent in the sense of the Lebesgue measure. On the other hand holds for almost all Weyl sequences $n\vec{\theta}$ and especially for those Weyl sequences with non zero different rational logarithms

$$D_N=\sup_J\left|\frac{A_N(J)}{N}-\lambda(I)\right|=\mathrm{O}(1/N^{1-\varepsilon}).$$

So we get

$$P(E) = \frac{A_N(E)}{N} + \mathrm{O}(N^{\varepsilon-1}),$$

$$P(F) = \frac{A_N(F)}{N} + \mathrm{O}(N^{\varepsilon-1})$$

and

$$P(E \cap F) = \frac{A_N(E \cap F)}{N} + \mathrm{O}(N^{\varepsilon-1}),$$

with generally different O-constants and $\varepsilon > 0$.

Consequently, we get

$$\frac{A_N(E)}{N} \cdot \frac{A_N(F)}{N} = \frac{A_N(E \cap F)}{N} + \mathrm{O}(N^{\varepsilon-1}),$$

which means that the arbitrary intervals $E, F \in I^s$ are independent up to an error $\mathrm{O}(N^{\varepsilon-1})$ with relative frequencies $A_N(.)/N$.

It is well known that the independence properties of events are hard to prove, thus we will turn to another measure of independence, covariance and correlation. The covariance and correlation for two (pseudo-) random variables x, y are given as follows

$$\mathrm{cov}(x,y) = E(X - E(x)) \cdot (Y - E(y))$$

and

$$\mathrm{cor}(x,y) = \frac{\mathrm{cov}\, x,y}{(\mathrm{cov}\, x,x \cdot \mathrm{cov}\, y,y)^{1/2}}.$$

Now for two (pseudo-) random sequences $(x_n)_{n\geq 1}, (y_n)_{n\geq 1}$, $x_n, y_n \in I^s$ we consider the empirical correlation and covariance as

$$\mathrm{cov}((x_n)_{n=1}^N, (y_n)_{n=1}^N) := \frac{1}{N}\sum_{i=1}^N x_i \cdot y_i - \left(\frac{1}{N}\sum_{i=1}^N x_i\right) \cdot \left(\frac{1}{N}\sum_{i=1}^N y_i\right),$$

$$\mathrm{cor}((x_n)_{n=1}^N, (y_n)_{n=1}^N) := \frac{\mathrm{cov}((x_n)_{n=1}^N, (y_n)_{n=1}^N)}{[\mathrm{cov}((x_n)_{n=1}^N, (x_n)_{n=1}^N) \cdot \mathrm{cov}((y_n)_{n=1}^N, (y_n)_{n=1}^N)]^{1/2}},$$

where $x_n \cdot y_n$, denotes in a natural way the scalar product of the vectors x_n and y_n.

We considered only the pairwise independence of events E, F and of two Weyl sequences respectively. We give now some theoretical results concerning the covariance and correlation coefficient (and the correlation matrix respectively) of multidimensional sequences of nodes.

Let us start with a $2s$-dimensional sequence $x_k := (x_1^k, \ldots, x_s^k, x_{x+1}^k, \ldots, x_{2s}^k)$, $k = 1, \ldots, N$. We split this $2s$-dimensional sequence into two s-dimensional sequences $x_k' := (x_1^k, \ldots, x_s^k)$ and $x_k'' := (x_{x+1}^k, \ldots, x_{2s}^k)$, $k = 1, \ldots, N$. We are interested in the covariance and correlation behaviour of these sequences. The mathematical expectation of a function $f(x) \mapsto y$, $I^{2s} \to \mathbb{R}, \mathbb{C}$ is

$$E(f) = \int_{I^{2s}} f(x)dx = \int_0^1 \cdots \int_0^1 f(x_1, \ldots, x_2 s)dx_1 \ldots dx_{2s}.$$

If f,g are independent functions then $E(f\cdot g)=E(f)\cdot E(g)$ holds and clearly $\operatorname{cov}(f,g)=E(f\cdot g)-(E(f)\cdot E(g))=0$.

Let

$$\sigma^2(f)=E(|f|^2)-|E(f)|^2,$$
$$\sigma^2(g)=E(|g|^2)-|E(g)|^2,$$

then for $\sigma^2(f)\neq\sigma^2(g)\neq 0$ the correlation coefficient of f and g can be given as

$$\operatorname{cor}(f,g)=\frac{\operatorname{cov}(f,g)}{\sigma(f)\cdot\sigma(g)}.$$

Given a finite collection of functions $f_i,\ldots,f_P$, the (hermitian) covariance and correlation matrices are defined by the entries $c_ij=\operatorname{cov}(f_i,f_j)$ and $r_ij=\operatorname{cor}(f_i,f_j)$ respectively.

Now we consider the random vector $(x_1,\ldots,x_s,x_{s+1},\ldots,x_{2s})$ and split it up into $x'=(x_1,\ldots,x_s)$ and $x''=(x_{s+1},\ldots,x_{2s})$. So we get the important special case

$$\operatorname{cov}x',x''=E(x_1x_{s+1}+\cdots x_sx_{2s})-E(x')\cdot E(x'').$$

We can easily calculate these values as

$$E(x_1x_{s+1}+\cdots x_sx_{2s})=\int_0^1\cdots\int_0^1(x_1x_{s+1}+\cdots x_sx_{2s})dx_1\ldots x_{2s}=\frac{s}{4},$$

$$E(x_1,\ldots,x_s)=E(x_{s+1},\ldots,x_{2s})=(\frac{1}{2},\ldots,\frac{1}{2})$$

and consequently

$$\operatorname{cov}(x',x'')=\frac{s}{4}-(\frac{1}{2},\ldots,\frac{1}{2})\cdot(\frac{1}{2},\ldots,\frac{1}{2})=0.$$

For the variances $\sigma^2(x')$ and $\sigma^2(x'')$ we get

$$\sigma^2(x')=E(x_1^2+\cdots+x_s^2)-E(x_1,\ldots,x_s)^2=\frac{s}{3}-\frac{s}{4}=\frac{s}{12},$$

and likewise $\sigma^2(x'')=s/12$. So the random points x' and x'' are non constant uncorrelated random variables because

$$\operatorname{cor}(x',x'')=\frac{\operatorname{cov}(x',x''}{\sigma(x')\cdot\sigma(x'')}=0.$$

We consider now the covariance and the correlation of sequences $(x_1^k,\ldots,x_s^k,x_{s+1}^k,\ldots,x_{2s}^k)$, $k=1,\ldots,N$ where $x'=(x_1^k,\ldots,x_s^k)\in I^s$ and $x''=(x_{s+1}^k,\ldots,x_{2s}^k)\in I^s$. When we replace the expectation E by the mean of the N values we get

$$\operatorname{cov}(x',x'')=\frac{1}{N}\sum_{k=1}^{N}(x_1^kx_{x+1}^k+\cdots+x_s^kx_{2s}^k)-\left(\frac{1}{N}\sum_{k=1}^{N}(x_1^k,\ldots,x_s^k)\right)\cdot\left(\frac{1}{N}\sum_{k=1}^{N}(x_{s+1}^k,\ldots,x_{2s}^k)\right).$$

We continue by applying number theoretical methods to this problem. It is clear that the covariance contains sums of the types

$$\frac{1}{N}\sum_{k=1}^{N} x_i^k x_{s+i}^k$$

and

$$\frac{1}{N}\sum_{k=1}^{N} x_i^k.$$

The variation in the sense of Hardy and Krause of the functions xy and x itself is easily computed as $V_2(x\cdot y)=1$, $V_1(x)=1$. The expectations $E(xy)$ and $E(x)$ are easily computed as

$$E(xy)=\int_0^1\int_0^1 xydxdy=\frac{1}{4}$$

and

$$E(x)=\frac{1}{2}.$$

By the Koksma-Hlawka inequality we get

$$\frac{1}{N}\sum_{k=1}^{N} x_i^k x_{s+i}^k=\frac{1}{4}+R_N(i,2),\ i=1,\ldots,s$$

and

$$\frac{1}{N}\sum_{k=1}^{N} x_i^k=\frac{1}{4}+R_N(i,1),\ i=1,\ldots,s$$

where for the errors $R_N(i,1)$ and $R_N(i,2)$ the approximation

$$|R_N(i,1)|=\mathrm{O}(D_N)$$

and

$$|R_N(i,2)|=\mathrm{O}(D_N)$$

holds.

Consequently, for the covariance we get

$$\begin{aligned}\operatorname{cov}(x',x'') &= \frac{s}{4}+\sum_{i=1}^{s}R_N(i,2)-\left[\left(\frac{1}{2},\ldots\frac{1}{2}\right)+(R_N(1,1),\ldots,R_N(s,1))\right]\cdot\\ &\quad\cdot\left[\left(\frac{1}{2},\ldots\frac{1}{2}\right)+(R_N(s+1,1),\ldots,R_N(2s,1))\right]=\\ &= \frac{s}{4}+\sum_{i=1}^{s}R_N(i,2)-\frac{s}{4}+\frac{1}{2}\left(\sum_{i=1}^{s}(R_N(i,1)+R_N(s+i,1))\right)+\sum_{i=1}^{s}R_N(i,1)\cdot R_N(s+i,1).\end{aligned}$$

And finally we get the estimation

$$|\operatorname{cov}(x',x'')|=\mathrm{O}(D_N+D_N^2),$$

where the O-constant is depending only on the dimension s of the sequences x',x'', $k=1,\ldots,N$. So in essence the covariance of the two sequences only depends on the distribution of the sequences and on

the dimension. Consequently, the better distributed the $2s$-dimensional sequence is the less correlated the two resulting s-dimensional sequences will be.

Since we have now an estimation of the covariance we can proceed to the correlation

$$\mathrm{cor}(x',x'') = \frac{\mathrm{cov}(x',x'')}{\sigma(x')\cdot\sigma(x'')},$$

it remains only to compute the variances σ.

We will only look at $\sigma^2(x')$, since the same results will hold for $\sigma^2(x'')$,

$$\sigma^2(x') = \frac{1}{N}\sum_{k=1}^{N}((x_1^k)^2+\cdots+(x_s^k)^2) - \left(\frac{1}{N}\sum_{k=1}^{N}(x_i^k,\ldots,x_s^k)\right)^2.$$

Because of $E(x^2) = 1/3$ and $E(x) = 1/2$ we get

$$\frac{1}{N}\sum_{k=1}^{N}(x_i^k)^2 = \frac{1}{3}+S(i),\ i=1,\ldots,s$$

and

$$\frac{1}{N}\sum_{k=1}^{N}x_i^k = \frac{1}{2}+T(i),\ i=1,\ldots,s.$$

The Koksma-Hlawka inequality leads to the estimations $|S(i)| \le \mathrm{O}(D_N)$ and $|T(i)| \le \mathrm{O}(D_N)$ and thus to the result

$$\sigma^2(x') = \frac{s}{12}\sum_{i=1}^{s}(S(i)-T(i)-T(i)^2) = \frac{s}{12}+\mathrm{O}(D_N),$$

where for both $\sigma^2(x')$ and $\sigma^2(x'')$ the O-constant only depends on the dimension s. Combining these results we get

$$\mathrm{cor}(x',x'') = \mathrm{O}(D_N),$$

where the O-constant depends only on the dimension s.

Skipping the other calculations for pairwise comparisons for $P > 2$ (since they consist only of an almost word for word repetition of the previous arguments) we can state the following.

Let $f_1 = (\{n\theta_1\},\ldots,\{n\theta_s\})$, …, $f_P = (\{n\theta_{P-1}s+1\},\ldots,\{n\theta_{Ps}\})$, $n = 1,\ldots,N$ and let I_P be the $P\times P$ unit matrix with entries $e_i i = 1$ and $e_j k = 0$ for $j \ne k$ and $i,j,k = 1,\ldots,P$. Let D_N^{Ps} denote the discrepancy for the Ps-dimensional sequence $(\{n\theta_1\},\ldots,\{n\theta_s\},\ldots,\{n\theta_{P-1}s+1\},\ldots,\{n\theta_{Ps}\})$, $n = 1,\ldots,N$. Then the covariance and correlation matrix for the sequences $f_1,\ldots,f_P$ can be given as

$$\mathrm{cov}(f_1,\ldots,f_P) = \frac{s}{12}\cdot I_P + \mathrm{O}(D_N^{Ps})$$

$$\mathrm{cor}(f_1,\ldots,f_P) = I_P + \mathrm{O}(D_N^{Ps}).$$

And for almost all Weyl sequences we can write

$$\mathrm{cov}(f_1,\ldots,f_P) = \frac{s}{12}\cdot I_P + \mathrm{O}(N^{\varepsilon-1}),$$

and

$$\mathrm{cor}(f_1,\ldots,f_P) = I_P + \mathrm{O}(N^{\varepsilon-1}), \tag{3}$$

where the estimation holds especially for generators of the type $\vec{\theta} = (e^{r_1},\ldots,e^{r_{Ps}})$ where $r_i \in \mathbb{Q}$ and $r_j \neq r_k$ for $j \leq k$ with $i,j,k = 1,\ldots,Ps$.

We can clearly see that the Zinterhof sequences fulfill the later requirements and thus the estimation holds.

3. Experimental Results for Parameterization

Two things are apparently of interest, one this the handling of GRID specific problems, e.g. failing machines. The other is the effect on the integration if a higher number of PEs is used. This second problem stems from equation (2) where on the one hand the variance is reduced when the number P of PEs rises, and on the other hand the discrepancy, upon which the variance also depends, is increased since the discrepancy is a function of N and an increasing P means a decreasing number of points per PE. Furthermore, we can see from equation (3) that for an increasing number of PEs the correlation raises, where low correlation means independant sequences.

To conduct these experiments we use the following functions, all given for dimension s, unless noted otherwise $s = 10$:

$$\begin{aligned}
fc1: \quad f(\vec{x}) &= \prod_{i=1}^{s} \frac{1}{x_i^{0.5}},\\
fc2: \quad f(\vec{x}) &= \prod_{i=1}^{s} \ln(\frac{1}{x_i})^{0.5-1},\\
mult: \quad f(\vec{x}) &= x_1 x_2 \cdots x_s,\\
gus: \quad f(\vec{x}) &= \sqrt{\frac{45}{4s}}\left(\sum_{i=1}^{s} x_i^2 - \frac{s}{3}\right),\\
hus: \quad f(\vec{x}) &= \prod_{i=i}^{s}\left(x_i^3 - \frac{1}{4} + 1\right).
\end{aligned}$$

The integration method we used for the unbounded functions fc1 and fc2 are described in detail in [8].
We use the parameterization approach as described above, i.e. PE_n uses $x_n = (\{ke^{1/(n-1)s+1}\},\ldots,\{ke^{1/ns}\})$, $k = 1,2,\ldots$.

In Fig. 1 and Fig. 2 we study the influence of failing nodes under parameterization. As reference we use the baseline of sequential Zinterhof sequence of dimension 10. The parameterization as such has no baseline so it's useful to compare it to a similar baseline to estimate the overall performance. The *parameterization* plot is the normal uninfluenced parameterization in a heterogeneous environment. The case of *one-fast* means that one PE did nearly all the work, this can happen if a lot of PEs fail along the way of if the come available only later during the calculation. The case *one-slow* is the

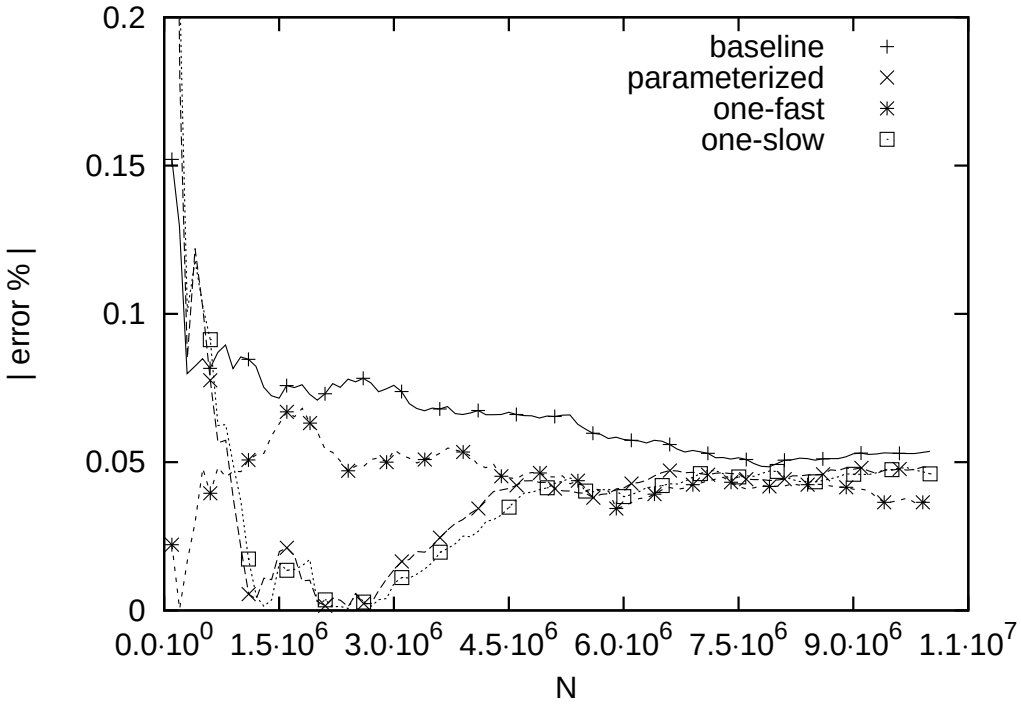


Figure 1. Comparison of function fc1 under parameterization in dimension $s = 10$.

other end of the spectrum, here one machine fails or becomes available only close to the end of the calculation. Overall we used 12 PEs for both tests with dimension 10, the number of points used is given at the x-axis and the percentage of the error in relation to the exact value of the integral is given at the y-axis.

What can clearly be seen is that the parameterization performs comparable to regular sequential integration, sometimes worse (Fig. 2) and sometimes better (Fig. 1), but in any case it generates an error of the same magnitude. In the plots it can also be seen that the influence of failing nodes, be it one or many, is negligible. While for a low N it could become problematic, e.g. *one-fast* in Fig. 1. The same results show for the bounded functions mult (Fig.3), gus (Fig.4) and hus (Fig.5) for dimensions 10, 15 and 20 respectively. All tests show that the usually problematic case of failing machines has little influence in the overall result.

However, the convergence rate can be reduced under parameterization, Fig. 4 is an extreme example. While it is clear that for higher dimensions the convergence rate is reduced, the difference between the baseline and the parameterization results in this case is quite huge.

We also want to look at the behaviour of the integration when we use more PEs. For this test we used PEs with equal speeds so that we can concentrate on the effects of the changing number of PEs rather than the heterogeneity of the network. Even though we used PEs with similar speed we did not synchronize so a slight discrepancy of points calculated per PE is given, however the deviation was small. We kept the number of points N set to 5000000 and 10000000 and dimension $s = 10$ for functions fc1 and fc2 (Fig. 6) and mult, hus and gus (Fig. 7). Please note that in Fig. 7 the value of mult is a realtive error rather than an absolute error, it was still included in this figure for reasons of ordinate scale. The number of machines used was in the range of 12–72, one machine was used as

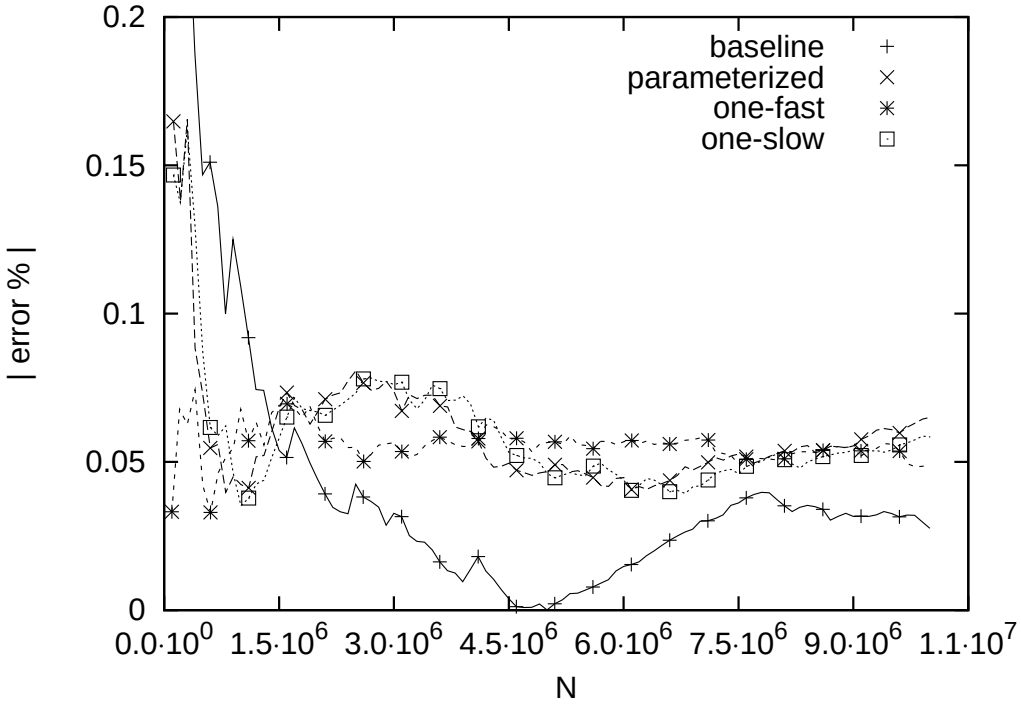


Figure 2. Comparison of function fc2 under parameterization in dimension $s = 10$.

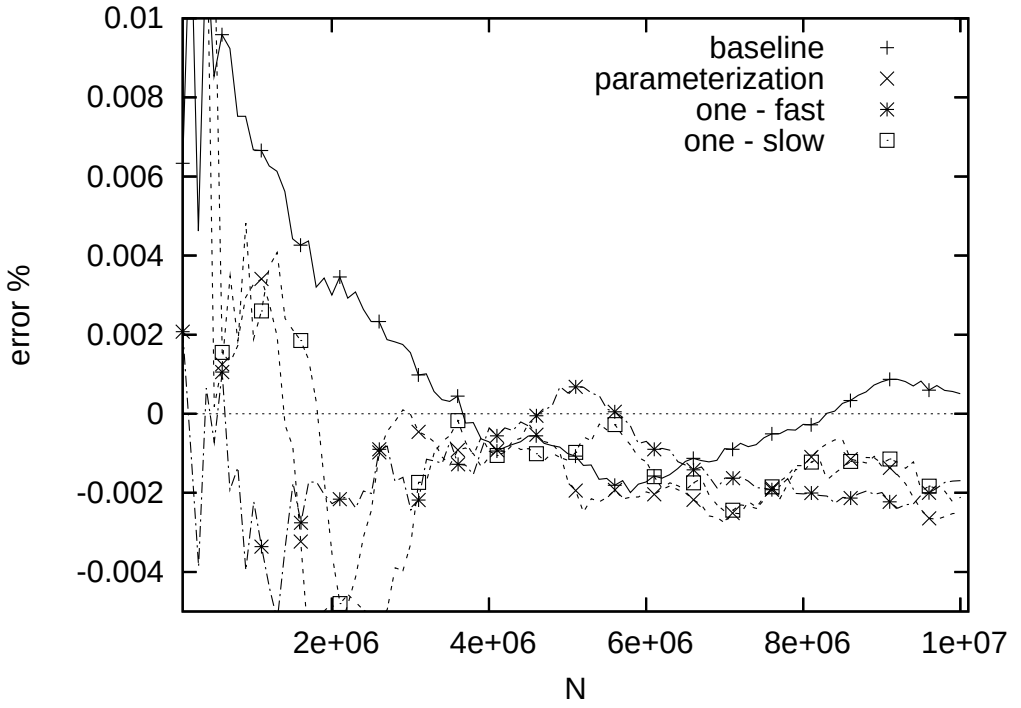


Figure 3. Comparison of function mult under parameterization in dimension $s = 10$.

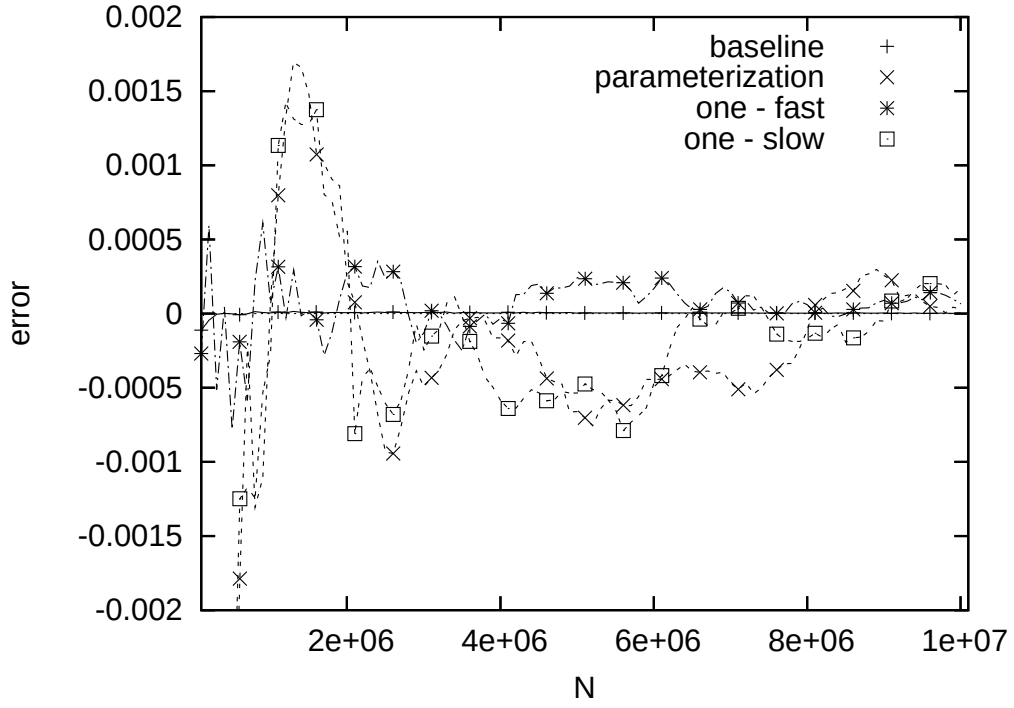


Figure 4. Comparison of function gus under parameterization in dimension $s = 15$.

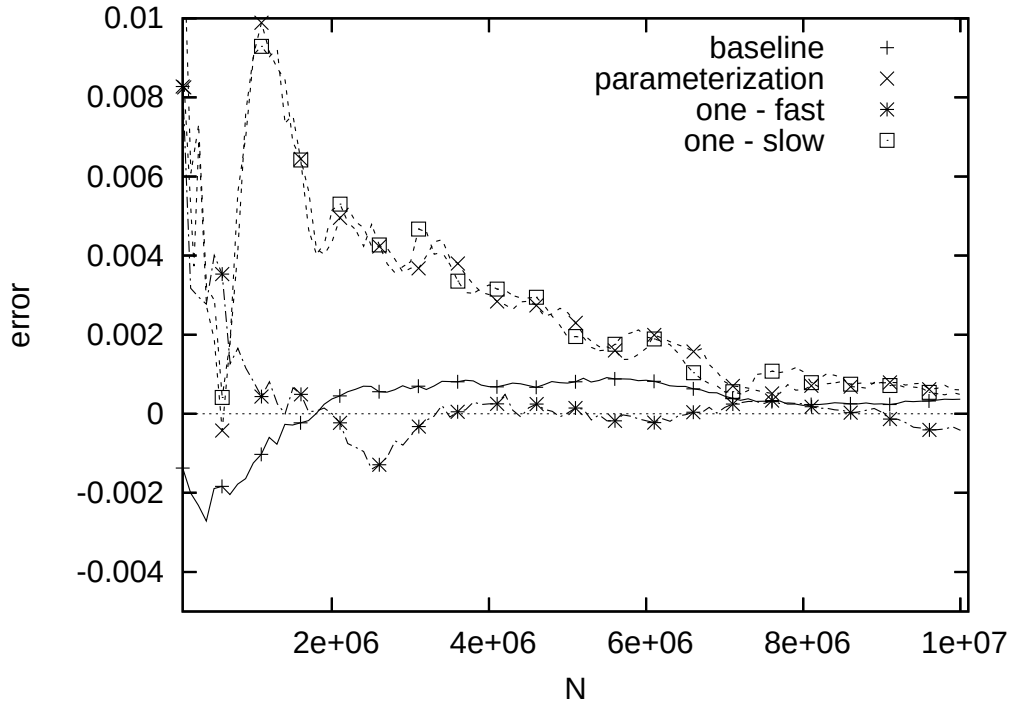
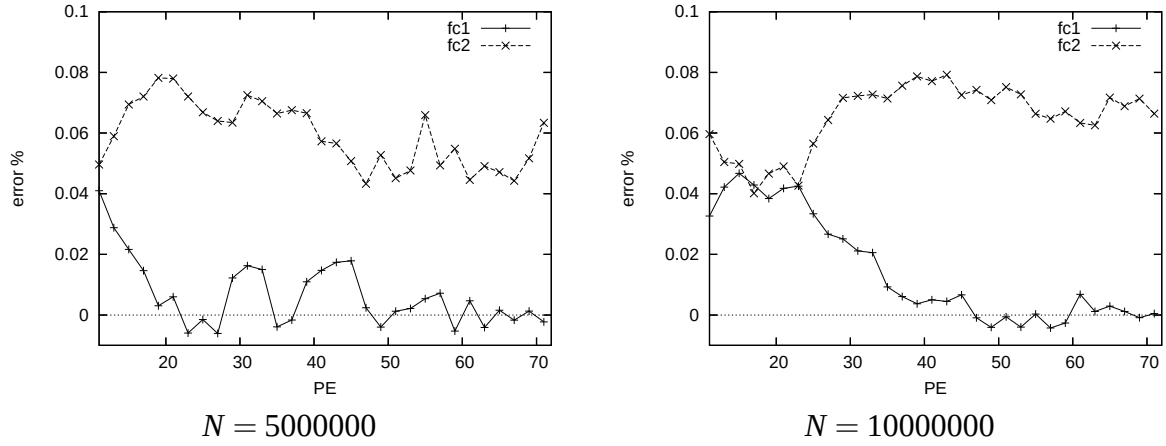
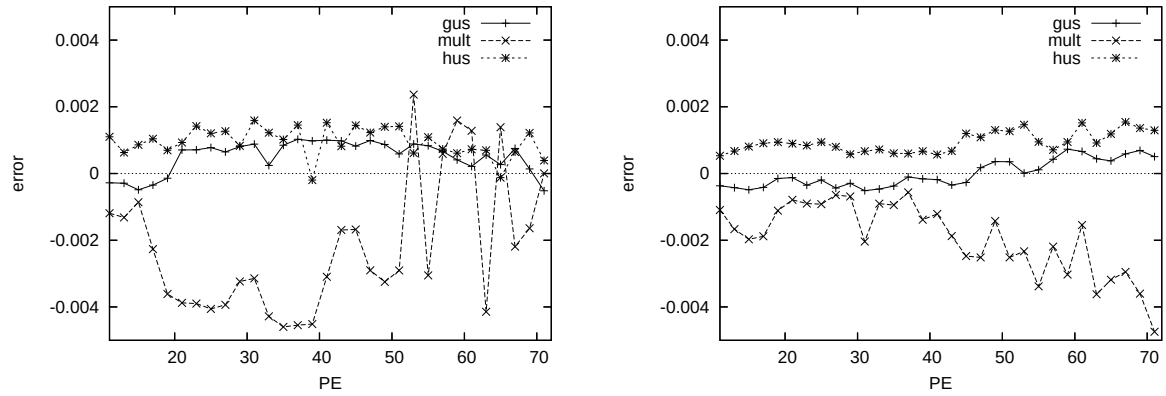


Figure 5. Comparison of function hus under parameterization in dimension $s = 20$.



$N = 5000000$ $N = 10000000$

Figure 6. Comparing the integration error of functions fc1 and fc2 when the number of PEs is increased under parameterization for a fixed N.



$N = 5000000$ $N = 10000000$

Figure 7. Comparing the integration error of functions hus, gus and mult when the number of PEs is increased under parameterization for a fixed N.

a root element, so the actual number of PEs went from 11 to 71, which is given on the abscissa. On the ordinate the error is displayed, for the functions gus and hus this is the absolute error, since the integral of these functions is 0 and we can not give a relative error. For the functions mult, fc1 and fc2 the errors are given as a percentage of the integral value of the function.

Overall what we see is that the error either stays roughly the same in most cases. However, in the case of *fc1* it even decreases and in the case of *mult* for N=10000000 it increases. Furthermore, a higher number of PEs can possibly degrade the result, and since we can not be predicted for which functions this will happen this is a severe drawback.

4. Conclusion

We discussed the properties of parameterization with regard to the GRID, and made a point that of all parallelization schemes it can handle the GRID environment best. Additionally, since the Zinterhof sequence is computationally efficient (and extraordinarily simple to implement) we are able to utilize

the GRID to it's maximum extent for a given task without further overhead. We showed, theoretically and experimentally, that the proposed parameterization scheme for the Zinterhof sequence is sound. However, the experiments and the error estimation, i.e. a probabilistic error bound instead of a discrete, show that the convergence rate of the numerical problem can decrease. Additionally, an increase in the number of PEs can lead to a increase in the integration error. This is a serious drawback since otherwise it would be excelently fit for use in the GRID.

For future work it might be interesting to find a better error bound than variance reduction in order to study the convergence rate in more detail. Also the peculiar effect which continually decreases the error for function *fc1* with a rising number of PEs is worth mentioning. Even though the probabilistic error bound does not hint at this it is well worth contemplating what leads to this behaviour.

Acknowledgments

The work described in this paper is partially supported by the Austrian Grid Project, funded by the Austrian BMBWK (Federal Ministry for Education, Science and Culture) under contract GZ 4003/2-VI/4c/2004.

References

[1] L. Cucos and E. deDoncker. Distributed QMC algorithms: new strategies for and performance evaluation. In *Proceedings of the High Performance Computing Symposium 2002 (HPC'02)/Advanced Simulation Techniques Conference*, pages 155–159, 2002.

[2] E. deDoncker, A. Genz, and M. Ciobanu. Parallel compuation of multivariate normal probabilities. *Computing Science and Statistics*, 30, 1999.

[3] E. deDoncker, R. Zanny, M. Ciobanu, and Y. Guan. Asynchronous quasi-Monte Carlo methods. In *Proceedings of the High Performance Computing Symposium 2000 (HPC'00)*, pages 130–135, 2000.

[4] E. deDoncker, R. Zanny, M. Ciobanu, and Y. Guan. Distributed quasi-Monte Carlo methods in a heterogeneous environment. In *Proceedings of the Heterogeneous Computing Workshop 2000 (HCW'2000)*, pages 200–206. IEEE Computer Society Press, 2000.

[5] E. deDoncker, R. Zanny, and K. Kaugars. Distributed numerical integration algorithms and applications. In *Proceedings of the 4th World Multiconference on Systemics, Cybernetics, and Informatics (SCI'00)*, pages 244–249, 2000.

[6] K. Entacher, P. Hellekalek, and P. L'Ecuyer. Quasi-Monte Carlo node sets from linear congruential generators. In H. Niederreiter and J. Spanier, editors, *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 188–198. Springer, 2000.

[7] G. Evans. *Practical numerical integration*. Wiley, Chichester, 1993.

[8] H. Hofbauer, A. Uhl, and P. Zinterhof. Quasi Monte Carlo Integration in GRID Environments: Further Leaping Effects. *Parallel Processing Letters*, 16(3):285–311, 2006.

[9] A.R. Krommer and C.W. Überhuber. *Numerical Integration on Advanced Computer Systems*, volume 848 of *Lecture Notes in Computer Science*. Springer, Berlin, 1994.

[10] A.R. Krommer and C.W. Überhuber. *Computational integration*. SIAM, Philadelphia, 1998.

[11] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Number 63 in CBMS-NSF Series in Applied Mathematics. SIAM, Philadelphia, 1992.

[12] G. Ökten and A. Srivivasan. Parallel quasi-Monte Carlo methods on a heterogeneous cluster. In K. T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 406–421. Springer-Verlag, 2002.

[13] R. Schürer and A. Uhl. An evaluation of adaptive numerical integration algorithms on parallel systems. *Parallel Algorithms and Applications*, 18(1–2):13–26, 2003.

[14] A. Srinivasan. Parallel and distributed computing issues in pricing financial derivatives through quasi-Monte Carlo. In *Proceedings of the International Parallel & Distributed Processing Symposium 2002 (IPDPS'02)*, Fort Lauderdale, FL, USA, April 2002. IEEE Computer Society Press.

[15] Peter Zinterhof. Einige zahlentheoretische methoden zur numerischen Quadratur und Interpolation. *Sitzungsberichte der Österreichischen Akademie der Wissenschaften, math.-nat.wiss. Klasse Abt. II*, 177:51–77, 1969.

[16] Peter Zinterhof. Über einige Abschätzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden. *Sitzungsber. Österr. Akad. Wiss. Math.-Natur. Kl. II*, **185**:121–132, 1976.