Zinterhof Sequences in GRID-based Numerical Integration

Heinz Hofbauer[1], Andreas Uhl[1], and Peter Zinterhof[2]

[1] Salzburg University, Department of Computer Sciences
{hhofbaue,uhl}@cosy.sbg.ac.at
[2] Salzburg University, Department of Computer Sciences
peter.zinterhof@sbg.ac.at

Summary. The appropriateness of Zinterhof sequences to be used in GRID-based QMC integration is discussed. Theoretical considerations as well as experimental investigations are conducted comparing and assessing different strategies for an efficient and reliable usage. The high robustness and ease of construction exhibited by those sequences qualifies them as excellent QMC point set candidates for heterogeneous environments like the GRID.

1 Introduction

High dimensional numerical integration problems may require a significant amount of computational effort. Therefore, substantial effort has been invested in finding techniques for performing these calculations on all kinds of high performance computing platforms (see e.g. [KÜ94, SU03]). GRID environments are highly beneficial but exhibit specifically challenging properties for numerical integration techniques. This class of computing facilities show extreme heterogeneity in terms of computing speed (caused by different memory capacity, cache sizes, and processor speed of the involved compute nodes) and network connections, moreover the available computing resources may change over time even during ongoing computations. These hardware properties require the employed integration routines to exhibit certain features:

- Variety in computing speed requires dynamic load balancing capability.
- Variety in network bandwidth and latency 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).

Additionally, error bounds and numerical results should preferable carry over from sequential execution, also reproducibility is considered an important issue.

Quasi-Monte Carlo (QMC) algorithms are among the most efficient techniques for evaluating high-dimensional integrals. Consequently, recent work has been devoted to apply this numerical integration approach in GRID environments [LKd05, LM05, HUZ06], however, many QMC techniques investigated for heterogeneous distributed systems may be used in the GRID context as well (e.g. [ÖS02, SU01, dZCG00]).

In this work we investigate a special type of QMC sequences, so-called Zinterhof sequences, for their applicability in GRID environments. In Section 2, we discuss the use of Zinterhof sequences in the general (sequential) QMC setting. Section 3 reviews strategies for using QMC techniques on parallel or distributed architectures. The main contribution of this work is presented in Section 4 where we give theoretical as well as experimental results on the use of Zinterhof sequences in GRID-type environments. Section 5 concludes the paper.

2 QMC Integration using Zinterhof Sequences

The basic concept of any QMC method for numerical integration is to approximate the integral by a finite sum, such that

$$I(f) := \int_{I^s} f(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} f(x_n) =: I'_N(f)$$

where x_n are suitably chosen and I^s is the unit interval. To identify suitable, i.e. uniformly distributed, points x_n with low star discrepancy are selected in order to exploit the Koksma-Hlawka inequality [Nie92]:

$$E_N(f) \le V(f)D^*_N(f),$$

where $E_N(f) := |I(f) - I'_N(f)|$ is the integration error.

2.1 Zinterhof Sequences

Zinterhof sequences [Zin69] are a special case of Weyl sequences. Weyl sequences are defined by

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

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 $\Theta = (\theta_1, \dots, \theta_s)$. See [KY81][Theorem 4.15] for an estimation of discrepancy for this type of sequences. For the Zinterhof sequence we set $\theta_i = e^{1/i}$ and consequently:

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

Note that due to their simplicity these sequences are extremely easy to generate and may be used by non-experts in a straightforward way without specific knowledge (which is not the case for all types of QMC sequences).

2.2 Numerical Integration with Zinterhof Sequences

Consider for dimension s the Fourier series expansion of the function $f(\mathbf{x})$ to be numerically integrated

$$f(\mathbf{x}) = \sum_{m_1,\ldots,m_s=-\infty}^{\infty} C(\mathbf{m}) e^{2\pi i(m_1x_1+\cdots+m_sx_s)} \tag{2}$$

with the integration error

$$E_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} f(n\boldsymbol{\theta}) - \int_0^1 \cdots \int_0^1 f(\mathbf{x})dx_1 \ldots dx_s \tag{3}$$

where $\mathbf{x} = (x_1, \ldots, x_s)$, $\mathbf{m} = (m_1, \ldots, m_s)$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)$.

For absolute convergent Fourier series the error is

$$E_N(\boldsymbol{\theta}) = \sum_{\substack{m_1,\ldots,m_s=-\infty \\ \mathbf{m}\neq 0}}^{\infty} C(\mathbf{m}) \frac{1}{N} \sum_{n=1}^{N} e^{2\pi i(\theta_1 m_1+\cdots+\theta_s m_s)n} = \sum_{\mathbf{m}\neq\mathbf{0}} C(\mathbf{m}) S_N(\boldsymbol{\theta}) \;\; .$$

Thus to determine the quality of the integration method we have to estimate $S_N(\boldsymbol{\theta})$. Clearly $\theta_1, \ldots, \theta_s$ must be rational independent unless $\theta_1 m_1 + \cdots + \theta_s m_s \in \mathbb{Z}$ and thus $S_N(\boldsymbol{\theta}) = 1$. Furthermore, by using Weyl's criterion, we know that for independent irrational numbers $\boldsymbol{\theta}$ it holds that

$$\lim_{N\to\infty} S_N(\boldsymbol{\theta}) \to 0 \quad \forall \mathbf{m} \in \mathbb{Z}^s \backslash \{0\}.$$

Since $S_N(\boldsymbol{\theta})$ is a geometric series we can write

$$S_N(\boldsymbol{\theta}) = \frac{1}{N} e^{2\pi i(m_1\theta_1+\cdots+m_s)} \frac{1 - e^{2\pi i(m_1\theta_1+\cdots+m_s)N}}{1 - e^{2\pi i(m_1\theta_1+\cdots+m_s)}}.$$

For the rational independent $\theta_1, \ldots, \theta_s$ with the equality $e^{ix} = \cos(x) + i\sin(x)$ and the basic approximation $|\sin(\pi x)| \geq 2 \ll x \gg$, where $\ll x \gg$ is the distance of x to the nearest integer, we can approximate

$$|S_N(\boldsymbol{\theta})| \leq 1/N \frac{1}{2 \ll m_1\theta_1 + \cdots + m_s\theta_s \gg} \;\; .$$

Consider for $\alpha > 1$ the class $E_\alpha^s(C) = \{f(\mathbf{x}) : |C(\mathbf{m})| \leq \frac{C}{||\mathbf{m}||^\alpha}\}$ then

$$|E_N(\boldsymbol{\theta})| \leq 1/N \sum_{\mathbf{m}\neq\mathbf{0}} \frac{C}{||\mathbf{m}||^{\alpha}} \frac{1}{2 \ll m_1\theta_1 + \cdots + m_s\theta_s \gg}$$

where $||\mathbf{m}|| = \prod_{i=1}^{s} \max(1, |m_i|)$.

Now for $\theta_1 = e^{r_1}, \ldots, \theta_s = e^{r_s}$, $r_i \neq r_j$ for $i \neq j$, $r_i \in \mathbb{Q}$ the subsequent result follows from an approximation by A. Baker (c.f. [KY81]):

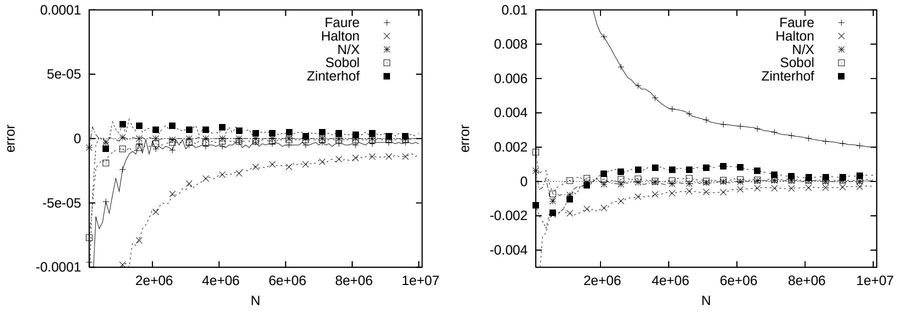
$$\ll m_1\theta_1 + \ldots + m_s\theta_s \gg \geq \frac{C(\boldsymbol{\theta})}{||\mathbf{m}||\psi(\mathbf{m})} \ ,$$

where $\psi(\mathbf{m})$ weakly converges towards ∞ for $||\mathbf{m}|| \to \infty$.

Since there is no irrational vector $\boldsymbol{\theta}$ such that for all $\mathbf{m} \ll m_1\theta_1 + \cdots + m_s\theta_s \gg \geq \frac{C(\boldsymbol{\theta})}{||\mathbf{m}||}$ holds, we obtain the final error approximation for $\alpha > 2$ (Zinterhof provides the same error magnitude even for $\alpha > 1$ [Zin69])

$$|E_N(\boldsymbol{\theta})| \leq 1/N \sum_{\mathbf{m}\neq\mathbf{0}} \frac{C||\mathbf{m}||}{||\mathbf{m}||^{\alpha}} \frac{\psi(\mathbf{m})}{2C(\boldsymbol{\theta})} \ .$$

To give an illustration of the excellent actual integration performance, Fig. 1 shows a comparison of numerical integration accuracy among several QMC sequences for two of the test functions used in Section 4 (we plot the integration error versus sample size).



$g(\mathbf{x})$, $s = 15$ $h(\mathbf{x})$, $s = 20$

Fig. 1. Comparison of Zinterhof, Halton, Sobol, Niederreiter/Xing (N/X) and Faure sequences.

It can be clearly seen that for each of the two test scenarios there is a single QMC sequence which shows very poor integration results, the Halton sequence in the first case and the Faure sequence in the second case. While being the top performing sequence considered for some test functions (compare also [HUZ06] and Fig. 5), Zinterhof sequences are at least always competitive to the best sequences available and lead to consistently low integration errors.

This fact taken together with the available error estimates and the simplicity of their construction and generation makes these sequences attractive candidates for practical QMC integration applications.

3 QMC Techniques in GRID Environments

The generation of the points of a QMC sequence on a single machine and and the subsequent distribution of the generated points generates a significant bottleneck for the integration application. When considering GRID properties, the constraint of the unknown network capacity can become a problem, as such a fast processing element (PE) behind a slow link would be wasted. Likewise, if the point generating PE is behind a slow network link all other PEs are penalized when they have to wait for new points. Thus, rather than distributing the points, the generation of the points itself is distributed in such a fashion that each PE can generate the points nearly independently of other PEs.

So far, two entirely different strategies have been discussed in literature to employ QMC sequences in parallel and distributed environments (see [HUZ06] for an exhaustive literature review and a detailed assessment of the effectiveness of the different strategies in GRID 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 (assuming availability of p PEs):

   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 $\mathbf{x}_{il}, \mathbf{x}_{il+1}, \mathbf{x}_{il+2}, \ldots, \mathbf{x}_{il+l-1}$) ("big blocks" scenario). In case a large number of smaller blocks is used index j is assigned dynamically to PE$_i$ which generates the vectors $\mathbf{x}_j, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+l-1}$ (where j is incremented in steps of size l to avoid overlap – "small blocks" scenario). See [LM05, SU01] for investigations and applications with respect to the blocking approach.

   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 $\mathbf{x}_i, \mathbf{x}_{i+p}, \mathbf{x}_{i+2p}, \ldots$). Usually a QMC point set is partitioned into p interleaved substreams if p PEs are available. However, if more PEs become available during the computation, there is no additional substream available in this scenario. A way to handle this situation is to partition a given QMC point set into $I > p$ substreams in case of p PEs are available. The $I-p$ substreams are not used by default but kept as additional work share in case additional PEs become available. See [Bro96, SU01, ESSU03] for investigations and applications with respect to the leaping approach.
2. Using inherently independent sequences on the different PEs (denoted as "parameterization" which can be realized for example by randomizations of a 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. See [Cd02, ÖS02] for investigations and applications with respect to the parameterization approach.

4 Zinterhof Sequences in GRID Environments

In this section we investigate whether Zinterhof sequences are sensible candidates for use in GRID environments. We present theoretical as well as experimental results with respect to the three approaches for distributed generation of QMC point sets as discussed in the previous section.

4.1 Theoretical Results

Leaping

For estimating the integration error resulting from using leaped Zinterhof sequences, we replace $\boldsymbol{\theta}$ in equation (3) by $L\theta_1, \dots, L\theta_s$ for leap size $L \in \mathbb{N}$. Then instead of $S_N(\boldsymbol{\theta})$ we have

$$S_N(L\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} e^{2\pi i (Lm_1\theta_1 + \cdots + Lm_s\theta_s)n}.$$

By analogy to the general case we can approximate the integration error, however this approximation is worse since instead of $\mathbf{m}$ we now have $L\mathbf{m}$ in all formulas. Thus with $||L\mathbf{m}|| = \prod_{i=1}^{s} \max(1, |Lm_i|) \leq L^s ||\mathbf{m}||$ and $\psi(L\mathbf{m})$ instead of $\psi(\mathbf{m})$ we get

$$|E_N(L\boldsymbol{\theta})| \leq 1/N \sum_{\mathbf{m} \neq \mathbf{0}} \frac{L^s C ||\mathbf{m}||}{||\mathbf{m}||^\alpha} \frac{\psi(L\mathbf{m})}{2C(\boldsymbol{\theta})} \quad .$$

Considering that $\psi(\mathbf{m})$ grows only logarithmically for $\boldsymbol{\theta} = (\theta_1^{r_1}, \dots, \theta_s^{r_s})$ and likewise for $\psi(L\mathbf{m})$ the difference of $\psi(\mathbf{m})$ to $\psi(L\mathbf{m})$ plays hardly any role. Thus the error approximation for leaping with leap size L is worse by the factor L^s than the error approximation for the unleaped sequence. This indicates a potentially significant deterioration of the results independent of the specific leap value (note that contrasting to this result we have derived poor discrepancy estimates only for 2^n type leaped (t,s)-sequence substreams in earlier work [SU01]).

Blocking

Again, consider the Fourier series given in Equation (2) and the error given in Equation (3) with the same parameters.

Then we have for $f \in E_\alpha^s(C)$ with $\alpha > 3/2$ and $x_1, \ldots, x_N \in I^s := [0,1]^s$ the approximation ([DT97, Theorem 1.35])

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_{I^s} f(\mathbf{x}) dx_1, \ldots, dx_s \right| \leq C \left(\frac{4\alpha - 4}{2\alpha - 3} \right)^{s/2} F_N(x_n) \qquad (4)$$

where $F_N(x_n)$ is the diaphony of $x_1, \ldots, x_N$.

It is known [Zin76] that for the Zinterhof sequence the estimation of the diaphony

$$F_N(n\boldsymbol{\theta}) = \mathrm{O}(1/N^{1-\epsilon}), \qquad (5)$$

for $\epsilon > 0$ holds, since $\boldsymbol{\theta}$ is of the form $\theta_1 = e^{r_1}, \ldots, \theta_s = e^{r_s}$ where the $r_i \in \mathbb{Q}\, \forall i = 1, \ldots, s$ are rationally independent.

The definition of the diaphony F_N for a general s-dimensional sequence $\mathbf{x_1}, \ldots, \mathbf{x_N}$ is

$$F_N^2(\mathbf{x_n}) = \frac{1}{N} \sum_{i,j=1}^{N} H_2(\mathbf{x_i} - \mathbf{x_j}), \qquad (6)$$

with

$$H_2(\mathbf{x}) = \prod_{i=1}^{s} h_2(x_i) - 1,$$

and h_2 being the normed Bernoulli polynomial of degree 2,

$$h_2 = 1 + 2\pi^2 \left(\{x\}^2 - \{x\} + \frac{1}{6} \right),$$

where $\{x\}$ is the fractional part of x.

The diaphony F_N of the sequence $x_1, \ldots, x_N$ is translation invariant, which follows directly from Equation (6) where we get $H_2((a + \mathbf{x_i}) - (a + \mathbf{x_j})) = H_2(\mathbf{x_i} - \mathbf{x_j})$, thus for any $a = (a_1, \ldots, a_s)$

$$F_N(x_n) = F_N(a + x_n)$$

holds.

For the Zinterhof sequence we can choose $a = x_B = (B\theta_1, \ldots, B\theta_s)$ such that we obtain

$$F_N(x_n) = F_N(x_B + x_n) = F_N(x_{B+n})$$

where $n = 1, \ldots, N$.

Thus when using the error approximation (4) we see that we can use an arbitrary block of length N instead of the first N points without deterioration of the integration error. Note that this corresponds well to an earlier result on (t,s)-sequences where we showed that discrepancy estimates of arbitrary blocks do not degrade as compared to estimates of entire (t,s)-sequences [SU01].

Now, similar to [ÖS02], let us consider the general case of blocking with block size b where new blocks are handed out as requested ("small blocks"). The classical blocking scheme, which we call "big blocks", is essentially a subset of this general case. When using p PEs we have always p continuous subsets, each subset of points ends where a block is still unfinished. So we have p sequences each generating an approximation of the integral I

$$I'^i = \frac{1}{c_i} \sum_{\lambda(c_i)} f(x_i)$$

where $i = 1, \ldots, p$, c_i is the number of vectors in sequence i and $\lambda(c_i)$ is the set of indices of vectors of the original Zinterhof sequence which generates sequence i and the numbering be such that $c_p \leq c_{p-1} \leq \cdots \leq c_1$ holds. Figure 2 illustrates this for three PEs, when an PE finishes with block 3 the former c_1 and c_2 collapse to form the new c_1. Also when one block is finished another block is assigned and an PE starts to work on it, this forms a new sequence c_3.

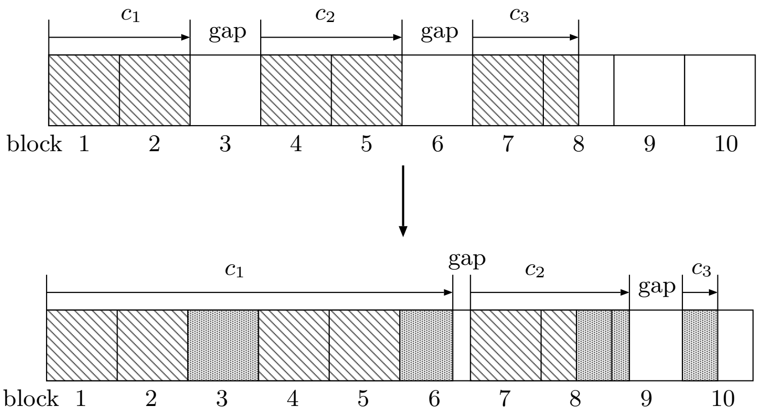


Fig. 2. Growth of subsequences with small blocks.

Now with N the total number of points, we get

$$I'_N = \frac{1}{N} \sum_{\lambda(N)} f(x_i) = \sum_{i=1}^{p} \frac{c_i}{N} \frac{1}{c_i} \sum_{\lambda(c_i)} f(x_i) = \sum_{i=1}^{p} \frac{c_i}{N} I'^i,$$

which gives us the overall estimate from the estimates of the individual sequences.

We can consider the error

$$E_N(f) = |I'_N - I(f)| \leq \sum_{i=1}^{p} \frac{c_i}{N} |I'^i - I(f)| \leq \sum_{i=1}^{p} \frac{c_i}{N} D^*_{c_i} V(f).$$

When looking at blocking with a small block size, i.e. not the big block scenario, it is clear that the first sequence grows continuously as more intermediate blocks are finished and likewise new sequences are introduced at the end with a very small c_p. From the above error estimate we see that the weighted average of the discrepancies is used, but since c_1 continually grows for $N \to \infty$ we get $c_i/c_1 \to 0$ for $1 < i \leq p$. Since $c_p \leq c_i \leq c_2$ for $i = 3, \ldots, p-1$, we get

$$E_N(f) \leq V(f)(\frac{c_1}{N} D^*_{c_1} + \frac{(p-1)c_2}{N} D^*_{c_p}).$$

For very big N the error estimation thus becomes approximately

$$E_N(f) \leq V(f) D^*_{c_1}$$

where $c_1 \approx N$.

Clearly the smaller the blocks are the faster they become insignificant and the faster the first sequence grows. For big blocks we have the same problem as with parameterization since unlike normal blocking no sequence becomes insignificant and for the error we can only get the general error estimate. Given a homogenous environment where $c_1 = \cdots = c_p$ we get only

$$E_N(f) \leq V(f) D^*_{N/p}$$

which shows no advantage over using a single machine.

Parameterization

A result with respect to a possible parameterization of Zinterhof sequences may be found in [HUZ07], which provides a set of almost uncorrelated sequences.

For almost all collections of P specimen of s-dimensional Weyl sequences $f_1^{(k)} = (\{k\theta_1\}, \ldots, \{k\theta_s\}), \ldots, f_i^{(k)} = (\{k\theta_{(i-1)s+1}\}, \ldots, \{k\theta_{is}\}), \ldots, f_P^{(k)} = (\{k\theta_{(P-1)s+1}\}, \ldots, \{k\theta_{Ps}\}),\ k = 1, 2, \ldots, N, \ldots$ the estimations for covariance and correlation

$$\mathrm{cov}_N(f_1, \ldots, f_P) = \frac{s}{12} I^P + \mathrm{O}(N^{\epsilon-1})$$

and

$$\mathrm{cor}_N(f_1, \ldots, f_P) = I^P + \mathrm{O}(N^{\epsilon-1})$$

hold, where I^P is the $P \times P$ unit matrix with entries $e_{ii} = 1$ and $e_{jk} = 0$ for $j \neq k$ and $i, j, k = 1, \ldots, P$. The estimations hold especially for the collections of P sequences of the Weyl type having generators $\theta_1, \ldots, \theta_s, \theta_{s+1}, \ldots, \theta_{Ps}$ of the form $\theta_u = \exp(r_u), r_u \in \mathbb{Q}, r_u \neq r_v \neq 0, 1 \leq u, v \leq Ps$.

Since the Zinterhof sequences are of the form given above, we have well distributed s-dimensional point generating sequences which are nearly uncorrelated. Essentially this allows us to use Zinterhof sequences for a parameterization approach where PE_n uses $x_n = (\{ke^{1/(n-1)s+1}\}, \ldots, \{ke^{1/ns}\}),\ k = 1, 2, \ldots$.

4.2 Numerical Experiments

Settings

For the Zinterhof sequences, we use our own custom implementation. In order to be able to assess the accuracy of our results, we also employ different QMC sequences. Overall we used Zinterhof, Sobol', Halton, Faure and Niederreiter/Xing sequences, due to page limitation only the more interesting results are shown, however, we will always comment on the remaining sequences. For generating the Sobol', Halton, Faure and Niederreiter-Xing sequences we use the implementation of the "High-dimensional Integration Library" HIntLib[3]. The HIntLib uses an implementation of construction 6, 8 and 18 from [MMN95] for the Sobol', Faure and Niederreiter/Xing sequences respectively. For Halton it uses the construction which was introduced in [Hal60]. For more information on Sobol', Faure, and Niederreiter-Xing sequences see [SS], and for Halton sequences see [Hal60].

The numerical experiments have been conducted by integrating the following test functions:

$$f(\mathbf{x}) = \prod_{i=1}^{s} \frac{1}{x_i^{0.5}}, \tag{7}$$

$$g(\mathbf{x}) = \sqrt{\frac{45}{4s}} \left(\sum_{i=1}^{s} x_i^2 - \frac{s}{3} \right), \tag{8}$$

$$h(\mathbf{x}) = \prod_{i=i}^{s} \left(x_i^3 - \frac{1}{4} + 1 \right). \tag{9}$$

All three functions have been employed extensively in experimental evaluations, e.g. in own earlier work function (7) in [HUZ06] and functions (8) and (9) in [SU01]. The function $f(\mathbf{x})$ is unbounded due to the singularity in 0, the value of the integral is 2^s (we use the identical integration routine as outlined in [HUZ06]). The integral for $g(\mathbf{x})$ and $h(\mathbf{x})$ is 0 in both cases, therefore we display an absolute integration error on the ordinate of the plots instead of a relative error as for $f(\mathbf{x})$ (the abscissa shows the number of points used in numerical integration).

For all experiments we used a randomly chosen mixture of machines using AMD CPUs with 1200, 1600, and 2000 MHz interconnected by 100Mbit ethernet [HUZ06]. The actual number of machines used is given for each experiment.

Results

For leaping, the experimental results (fortunately) do not confirm the poor error estimate. Figure 3 shows integration results when single leaps (with different leap sizes) are used in sequential execution instead of the unleaped

[3] Available at: http://www.cosy.sbg.ac.at/˜rschuer/hintlib/

sequence. Surprisingly it turns out that the leaped substreams actually improve the integration result and lead to significantly faster convergence as compared to the baseline case. A similar behavior (except for leap 65) has been observed for $f(\mathbf{x})$ (see also [HUZ06]), $g(\mathbf{x})$ is very easy to handle and therefore almost no differences show up between the original and leaped versions of the Zinterhof sequence.

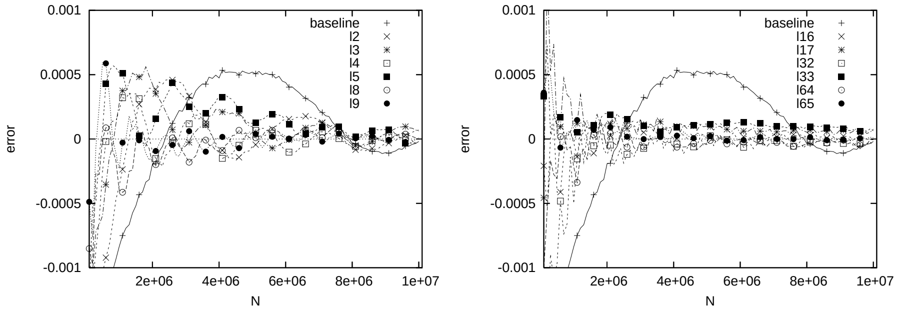


Fig. 3. Comparison of different leap sizes for the Zinterhof sequence, function $h(\mathbf{x})$, $s = 10$.

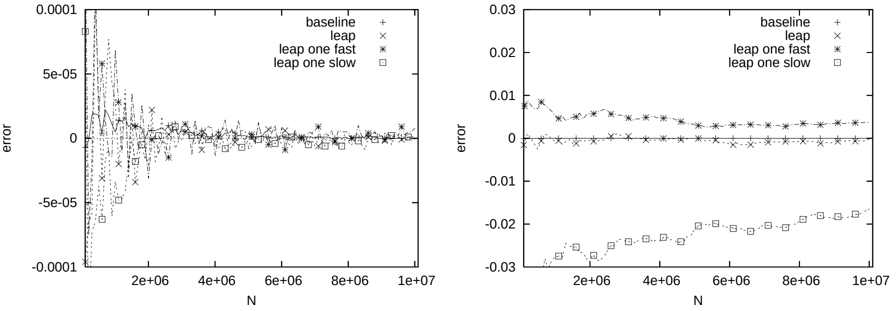
Of the remaining sequences only the Niederreiter-Xing sequences show comparable (and even better) stability with respect to splitting. All other sequences show significant result degradation for one or more leaps.

In order to relate these results to an execution in a GRID like environment, we simulate failure of PEs in the following manner: "client-server" is numerically identical to the sequential result (but executed in the distributed environment), "leap" is the standard case where one stream is assigned each PE, and "one-fast" and "one slow" are scenarios where one PE is speed up or slowed down by a factor of 10^3: consequently, one-slow simulates the case of one failing PE, whereas one-fast simulates the rather unlikely case that all but one PE fail after an initial start.

Figure 4 gives the results for leap 11 (on 11 PEs) and compares them to the result employing the Sobol' sequence with leap $2^3 + 1 = 9$ which is of the form $2^n + 1$ (which is known to have a good star discrepancy estimate [SU01]).

It is clearly visible, that the Zinterhof sequence is very robust against PE failure and that the integration errors are significantly smaller as compared to the Sobol' sequence. However, the Sobol' case shows severely degraded results, especially in the highly probable one-slow case. In this scenario we have a systematic missing of equally spaced points which can not be grasped by a discrepancy estimate of a single leaped substream and obviously leads to significant problems in this type of point sets. Comparable results are found for $h(\mathbf{x})$.

For the other sequences, except for the Niederreiter/Xing sequence, we find problems comparable to those seen with the Sobol' sequence. Overall, a



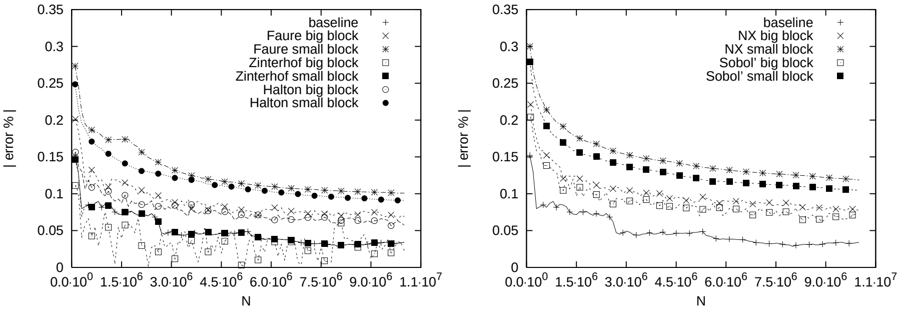
Zinterhof (leap 11) Sobol' (leap 9)

Fig. 4. Comparison of Zinterhof and Sobol' sequences, $g(x)$, $s = 10$.

high robustness in the case of employing leaped substreams can be stated for Zinterhof sequences.

Contrasting to the leaping case, robust behavior is to be expected due to the theoretical result when using contiguous blocks of Zinterhof sequences in distributed integration.

Figure 5 compares the results of different QMC sequences for blocking using 11 machines. For small blocks we use block size 500 and for big blocks the relatively bad case of block size N is chosen, which results in gaps roughly nine times the size of the actually used blocks. As a baseline for comparison we show integration with the Zinterhof sequence using all N points in consecutive manner.

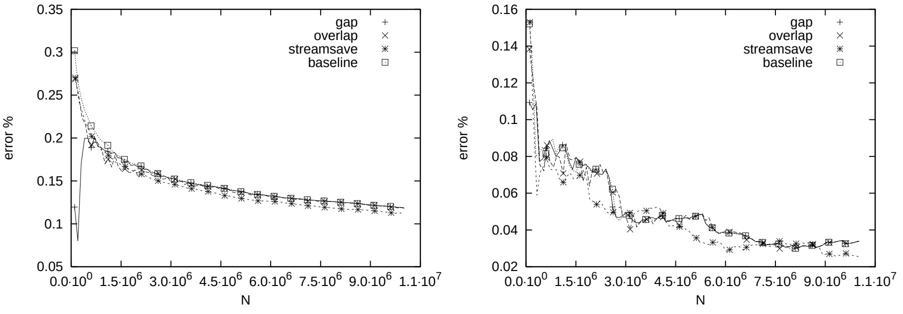


Faure, Zinterhof, Halton Niederreiter/Xing, Sobol'

Fig. 5. Comparison of blocking (big and small blocks) using Zinterhof, Halton, Faure, Niederreiter/Xing and Sobol' sequences, $f(\mathbf{x})$, $s = 10$.

The first things to note is that the three results corresponding to the Zinterhof sequence are the best ones in terms of error magnitude. The small blocks' result is in fact identical to the baseline version (which holds true for all sequences and was to be expected since the error estimate for small blocks is practically independent of block discrepancy for high N and thus valid for all sequences) whereas the big blocks' result shows lower error but a higher degree of result fluctuations in the Zinterhof sequence case.

For all cases (except the Zinterhof sequence between 8×10^6 and 9×10^6 integration points) the big blocks case is better than the small blocks case, and thus the baseline. While this isn't generally the case (see Fig. 6) it seems that the integration error estimation gained from our approach for big blocks is not the best possible.



Niederreiter/Xing

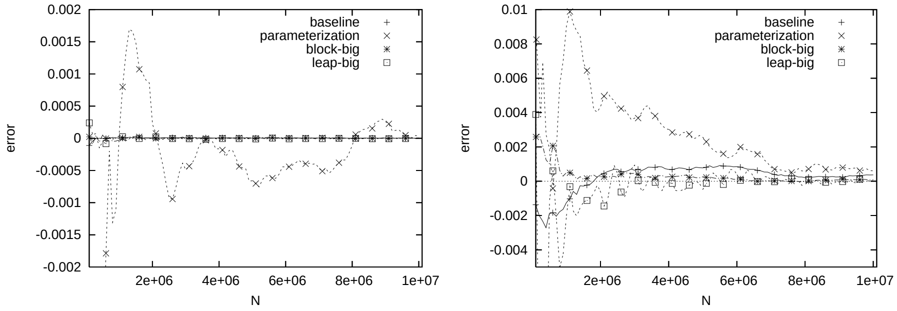
Zinterhof

Fig. 6. Comparison Niederreiter/Xing and Zinterhof sequences regarding overlap, gaps and unused streams for $f(\mathbf{x})$, $s = 10$.

Figure 6 shows results corresponding to the more realistic case that we employ big blocks with a gap between blocks that cover 20% of the size of a block and big blocks resulting in an overlap where about 30% of one block overlaps the following block. Additionally, we investigate the "Streamsave" scenario, where we use small blocks with a block size of 75 and between the blocks there is a 25 point gap. This simulates the synchronized use of substreams with leap 100 where the last 25 out of 100 streams are reserved for PEs which become available during the computation (but are not used in the experiment). 71 PEs are used for these computations.

The result shows that both considered sequences can cope well with gaps, overlap, and the "streamsave" scenario, no degradation of the result is observed.

To relate parameterization behavior to blocking and leaping effects, we compare all three approaches in the following. Results with respect to functions $f(\mathbf{x})$ and $h(\mathbf{x})$ for dimension $s = 10$ (not shown) raise doubts about the reliability of parameterization since the error seems to decrease slower for increasing N as compared to other techniques. In order to facilitate a fair comparison we increase the dimension and employ 10 PEs and less favorable conditions for leaping and blocking: For blocking the block size is set to N and for leaping we use leap 100 resulting in 90% gaps for blocking and likewise to 90% unused streams for leaping (note that the gaps are distributed differently in both variants).



$g(\mathbf{x})$, $s = 15$ $h(\mathbf{x})$, $s = 20$

Fig. 7. Comparison of leaping (with large leapsize), blocking (large gaps) and parameterization for Zinterhof sequences.

As observed in Fig. 7, for higher dimension the approximation becomes worse, which is to be expected since we use the same number of points independent of dimension (compare e.g. Figs. 3 and 4). The results of blocking and leaping are almost identical to the baseline version, although the conditions are much more difficult in the current setting. On the other hand, parameterization shows a larger error and a slower convergence towards the correct solution.

5 Conclusion

Overall, we have shown that Zinterhof sequences are well suited for numerical integration in GRID environments. Whereas the error estimation for leaped substreams suggests worse integration errors as compared to sequential usage, we have found no experimental evidence corresponding to this result. In contrary, leaping turns out to behave very reliable and robust even to hardware failures and may be used in a flexible way.

For the case of using contiguous blocks for integration the theoretical prediction suggesting behavior equal to the sequential case is supported by experimental results. Similar to leaping, high robustness against gaps between blocks and against overlap has been observed.

While the suggested parameterization scheme works in principle, the results show clearly slower convergence as compared to the leaping or blocking strategies, respectively. Parameterization (at least in the proposed manner) should only be used if this effect is acceptable.

Concluding we may state that Zinterhof sequences have been shown to exhibit excellent behavior when using separately initialized and disjoint substreams for distributed numerical integration and they excel by their ease of construction and implementation even for non-specialists.

References

[Bro96] B.C. Bromley, *Quasirandom number generators for parallel Monte Carlo algorithms*, Journal of Parallel and Distributed Computing **38** (1996), 101–104.

[Cd02] L. Cucos and E. deDoncker, *Distributed QMC algorithms: new strategies for and performance evaluation*, Proceedings of the High Performance Computing Symposium 2002 (HPC'02)/Advanced Simulation Techniques Conference, 2002, pp. 155–159.

[DT97] M. Drmota and R.F. Tichy, *Sequences, discrepancies and applications*, Lect. Notes in Math., vol. 1651, Springer-Verlag, 1997.

[dZCG00] E. deDoncker, R. Zanny, M. Ciobanu, and Y. Guan, *Asynchronous quasi-Monte Carlo methods*, Proceedings of the High Performance Computing Symposium 2000 (HPC'00), 2000, pp. 130–135.

[ESSU03] K. Entacher, T. Schell, W. Ch. Schmid, and A. Uhl, *Defects in parallel Monte Carlo and quasi-Monte Carlo integration using the leap-frog technique*, Parallel Algorithms and Applications **18** (2003), no. 1–2, 27–47.

[Hal60] J. H. Halton, *On the efficiency of certain quasi-random sequences of points in evaluating multi-dimension integrals*, Numer. Math. **2** (1960), 84–90, Berichtigung, ibid., (1960), p. 196.

[HUZ06] H. Hofbauer, A. Uhl, and P. Zinterhof, *Quasi Monte Carlo Integration in GRID Environments: Further Leaping Effects*, Parallel Processing Letters **16** (2006), no. 3, 285–311.

[HUZ07] ______, *Parameterization of Zinterhof Sequences for GRID-based QMC Integration*, Proceedings of the 2nd Austrian Grid Symposium (Innsbruck, Austria) (J. Volkert, T. Fahringer, D. Kranzlmüller, and W. Schreiner, eds.), books@ocg.at, Austrian Computer Society, 2007, To appear.

[KÜ94] A.R. Krommer and C.W. Überhuber, *Numerical integration on advanced computer systems*, Lecture Notes in Computer Science, vol. 848, Springer, Berlin, 1994.

[KY81] Hua Loo Keng and Wang Yuan, *Applications of Number Theory to Numerical Analysis*, Springer Verlag, Science Press, 1981.

[LKd05] S. Li, K. Kaugars, and E. deDoncker, *Grid-based numerical integration and visualization*, Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05), IEEE Computer Society Press, 2005, pp. 260–265.

[LM05] Yaohang Li and Michael Mascagni, *Grid-based Quasi-Monte Carlo applications*, Monte Carlo Methods and Appl. **11** (2005), no. 1, 39–55.

[MMN95] Gary L. Mullen, Arijit Mahalanabis, and Harald Niederreiter, *Tables of (t, m, s)-net and (t, s)-sequence parameters*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (Harald Niederreiter and P. J.-S. Shiue, eds.), 106, vol. Lecture Notes in Statistics, Springer-Verlag, 1995, pp. 58–86.

[Nie92] Harald Niederreiter, *Random number generation and quasi-monte carlo methods*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 62, Society for Industrial and Applied Mathematics (SIAM), 1992.

[ÖS02] G. Ökten and A. Srinivasan, *Parallel quasi-Monte Carlo methods on a heterogeneous cluster*, Monte Carlo and Quasi-Monte Carlo Methods 2000 (K. T. Fang, F. J. Hickernell, and H. Niederreiter, eds.), pub-springer, 2002, pp. 406–421.

[SS] Rudol Schürer and Wolfgang Ch. Schmid, *Mint - the database of optimal (t, m, s)-net parameters*, online: `http://mint.sbg.ac.at/`.

[SU01] W. Ch. Schmid and A. Uhl, *Techniques for parallel quasi-Monte Carlo integration with digital sequences and associated problems*, Mathematics and Computers in Simulation **55** (2001), 249–257.

[SU03] R. Schürer and A. Uhl, *An evaluation of adaptive numerical integration algorithms on parallel systems*, Parallel Algorithms and Applications **18** (2003), no. 1–2, 13–26.

[Zin69] Peter Zinterhof, *Einige zahlentheoretische Methoden zur numerischen Quadratur und Interpolation*, Sitzungsberichte der Österreichischen Akademie der Wissenschaften, math.-nat.wiss. Klasse Abt. II **177** (1969), 51–77.

[Zin76] P. Zinterhof, *Über einige Abschätzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden*, Sitzungsber. Österr. Akad. Wiss. Math.-Natur. Kl. II **185** (1976), 121–132.