High order of convergence using lattice sequences for numerical integration

Dirk Nuyens$^{a,b}$

Joint work with Fred J. Hickernell$^c$, Peter Kritzer$^a$ and Frances Y. Kuo$^a$

$^a$School of Mathematics & Statistics, University of NSW, Australia
$^b$Department of Computer Science, K.U.Leuven, Belgium
$^c$Department of Applied Mathematics, Illinois Institute of Technology, USA

7th Australia New Zealand Mathematics Convention
ANZMC2008, University of Canterbury, Christchurch, New Zealand 8–12 December 2008
Outline

1. Quasi-Monte Carlo integration
2. A modified classical result
3. Higher order convergence
4. Numerical examples
5. Conclusion
<table>
<thead>
<tr>
<th>QMC integration</th>
<th>A modified classical result</th>
<th>Higher order convergence</th>
<th>Numerical examples</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

1. **Quasi-Monte Carlo integration**
   - Introduction
   - Lattice rules and sequences
   - Discrepancy
   - Reproducing kernel Hilbert spaces

2. **A modified classical result**

3. **Higher order convergence**

4. **Numerical examples**

5. **Conclusion**

High order of convergence using lattice sequences for numerical integration

Dirk Nuyens
Quasi-Monte Carlo integration

We try to approximate the $s$-dimensional integral

$$I(f) := \int_{[0,1)^s} f(x) \, dx$$

by an $N$-point quasi-Monte Carlo rule

$$Q(f) := \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}).$$

The point set $\{x^{(0)}, x^{(1)}, \ldots, x^{(N-1)}\}$ could be taken a lattice rule where

$$x^{(k)} = \frac{kg \mod N}{N} = \frac{kg}{N} \mod 1, \quad k = 0, 1, \ldots, N - 1,$$

with generating vector $g \in (\mathbb{Z}_N^s) \times \mathbb{N}$; or a digital net/sequence; or ...
Lattice rules and sequences

Lattice sequences

Besides fixed lattice rules, there exist lattice sequences in base $b$.

That is: A generating vector $g \in \mathbb{Z}^s$ which is “good” for all values of $N$ of the form $b^m$.

This gives a sequence of embedded lattice rules

$$P_{b^0} \subset P_{b^1} \subset P_{b^2} \subset \cdots \subset P_{b^{10}} \subset \cdots \subset P_{b^{20}} \subset \cdots$$

We could stop anywhere using points in a permuted order:

$$x^{(k)} = \frac{\varphi_{b,m}(k) g \mod b^m}{b^m} = \varphi_b(k) g \mod 1,$$

$m$ sufficiently large. Where $\varphi_{b,m}$ gives a permutation on the integers $0, \ldots, b^m - 1$ which keeps the embedding.

E.g., $\varphi_b$ could be van der Corput’s radical inverse function in base $b$.

The question for this talk: what happens at intermediate $N$?
Lattice rules and lattice sequences

(a) rank-1 rule  (b) Fibonacci lattice  (c) rank-2 copy rule
(d) $3^3$ sequence points  (e) 64 sequence points  (f) $3^4$ sequence points
Uniform distribution and low-discrepancy point sets

A sequence \( \{x^{(0)}, x^{(1)}, \ldots \} \) is *uniformly distributed* iff for every Riemann integrable function \( f \) on \([0, 1)^s\)

\[
\lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}) = \int_{[0,1)^s} f(x) \, dx.
\]

The *discrepancy* of a point set \( P \) with \( N \) points is measured as the deviation from the ideal uniform distribution

\[
D_N(P) = \sup_{J \in \mathcal{B}} \left| \frac{|P \cap J|}{N} - \text{vol}_s(J) \right|,
\]

with \( \mathcal{B} \) a set of boxes of a particular form in \([0, 1)^s\).

The *star discrepancy* takes boxes of the form \( \prod_{j=1}^s [0, x_j) \).

Dirk Nuyens
Quasi-Monte Carlo uses low-discrepancy points

Following Niederreiter: If the discrepancy of a point set has

\[ D_N(P) = O\left(\frac{(\log N)^s}{N}\right), \]

then it is called a \textit{low-discrepancy point set}.
For fixed \( N \)-point rules actually \((\log N)^{s-1}\), you pay a log factor for not fixing \( N \).

The Koksma-Hlawka inequality now gives a bound on the error for \( Q \) using \( P \)

\[ |I(f) - Q(f)| \leq D^*_N(P) V(f), \]

where \( V(f) \) is the variation of \( f \) (in the sense of Hardy and Krause).
Towards reproducing kernel Hilbert spaces

Different norms for measuring the discrepancy are of course possible.

E.g., take the classical star discrepancy

\[
D_N^*(P) := \sup_{x \in [0,1)^s} \left| \frac{|P \cap [0,x]|}{N} - \prod_{j=1}^{s} x_j \right|
\]

and use the \(L_2\)-norm instead of the \(L_\infty\) one:

\[
T_N^*(P) := \left( \frac{1}{3^s} - \frac{2}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left( 1 - \frac{(x_j^{(k)})^2}{2} \right) + \frac{1}{N^2} \sum_{k,\ell=0}^{N-1} \prod_{j=1}^{s} (1 - \max(x_j^{(k)}, x_j^{(\ell)})) \right)^{1/2}
\]

this is the \textit{worst-case error} in a reproducing kernel Hilbert space.
The worst-case error

Define the *worst-case error* for a given cubature formula $Q$ to approximate integration $I$ of functions in the unit ball of a Banach space $\mathcal{F}$ as

$$e(Q, \mathcal{F}) := \sup_{f \in \mathcal{F}, \|f\|_{\mathcal{F}} \leq 1} |I(f) - Q(f)|.$$ 

If $\mathcal{H}(K)$ is a reproducing kernel Hilbert space with error representer $\xi_{K,Q} = I(K) - Q(K)$ for a given cubature rule $Q$, then the error for functions $f \in \mathcal{H}(K)$ is bounded by a “Koksma-Hlawka inequality”

$$|I(f) - Q(f)| \leq \|f\|_{\mathcal{H}} \|\xi_{K,Q}\|_{\mathcal{H}},$$

where now the worst-case error

$$e(Q, \mathcal{H}(K)) = \|\xi_{K,Q}\|_{\mathcal{H}}$$

can be interpreted as a kind of discrepancy.
Quasi-Monte Carlo integration

A modified classical result
- Worst-case error for compound point sets
- What can be obtained by this?
- But we want more...

Higher order convergence

Numerical examples

Conclusion
A classical result from Kuipers and Niederreiter (1974)

**Theorem:** Suppose you have $M$ point sets $P_i$ with $N_i$ points, which have discrepancy $D_{N_i}^*(P_i)$ each, then the combined discrepancy satisfies

$$D_N^*(P) \leq \sum_{i=1}^{M} \frac{N_i}{N} D_{N_i}^*(P_i),$$

with $P = \bigcup_i P_i$ and $N = \sum_i N_i$.

This is a remarkable result. There are no relational conditions on the different point sets! Of course, you want to keep $M$ small.
The previous result is easily modified for the worst-case error:

**Theorem:** Given $M$ equal-weight rules $Q_i$ using point sets $P_i$, which have worst-case error $e_{N_i}(P_i, K)$, in a reproducing kernel Hilbert space $\mathcal{H}(K)$, then the worst-case error for the equal-weight rule using all the points satisfies

$$e_N(P, K) \leq \sum_{i=1}^{M} \frac{N_i}{N} e_{N_i}(P_i, K),$$

with $P = \bigcup_i P_i$ and $N = \sum_i N_i$. 
What can be obtained by this?

Given a “good” lattice sequence in base $b$ up to $N$ points, where

$$N = \sum_{\ell=0}^{L-1} n_{\ell} b^{\ell}, \quad L = \lfloor \log_b N \rfloor + 1,$$

and chop it up in blocks of powers of $b$:

<table>
<thead>
<tr>
<th>nb of blocks</th>
<th>of size</th>
<th>with worst-case error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\ell}$</td>
<td>$b^{\ell}$</td>
<td>$e_{\ell} = O(b^{-\ell} (\log b^{\ell})^{s-1})$</td>
</tr>
</tbody>
</table>

Then, if each block has $e_{\ell} = O(b^{-\ell} (\log b^{\ell})^{s-1})$, the total worst-case error behaves like

$$e_N(P, K) = O\left(\frac{(b - 1)(\log N)^s}{N}\right).$$
What can be obtained by this?

\[ O(N^{-1}) \] anywhere for lattice sequences

A lattice sequence constructed in a *shift-invariant* reproducing kernel Hilbert space has exactly the right properties.

E.g., for the first 22 points of a sequence in base 3:

\[
\begin{align*}
S_{22} & \rightarrow S_{21} \rightarrow S_{18} \rightarrow S_9 \\
\{x_0, x_1, x_2, x_3, \ldots, x_8\} & \rightarrow \{x_9, x_{10}, x_{11}, \ldots, x_{17}\} \rightarrow \{x_{18}, x_{19}, x_{20}, x_{21}\}
\end{align*}
\]

the worst-case error of \( P_{2,2} \) is exactly that of the first 9 points, the worst-case error of \( P_{1,1} \) is exactly that of the first three points, etc...
But we want more…

The previous result showed that one can *stop anywhere* with a good lattice sequence. The error bound of $O(N^{-1+\epsilon})$ automatically holds for all $N$, not only powers of the base.

However for some function spaces the worst-case error bounds are

$$e_N(S_N, K) = O(N^{-\alpha+\epsilon}), \quad N = b^m, \quad m = 0, 1, \ldots,$$

with $\alpha > 1$, and $\epsilon > 0$ hides the log $N$ term.

But using

$$e_N(P, K) \leq \sum_{i=1}^{M} \frac{N_i}{N} e_{N_i}(P_i, K),$$

can at best only give $O(N^{-1+\epsilon})$. We need something else…
1. Quasi-Monte Carlo integration

2. A modified classical result

3. Higher order convergence
   - Equal-weight rules are stuck with $O(N^{-1})$
   - Weights to the rescue

4. Numerical examples

5. Conclusion
Theorem: Given a quasi-Monte Carlo rule

\[ Q(f) = \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}), \]

then we have for the error for \( N + 1 \) points and \( N \) points:

\[ E_{N+1}(f) = I(f) - \frac{1}{N+1} \sum_{k=0}^{N} f(x^{(k)}), \]

\[ E_{N}(f) = I(f) - \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}), \]

such that using their difference gives

\[ |N E_N(f)| + |(N + 1) E_{N+1}(f)| \geq |I(f) - f(x^{(N)})| = O(1). \]

Suppose the error is \( O(N^{-\beta}) \), then it follows that \( \beta \leq 1 \).
Equal-weight rules are stuck with $O(N^{-1})$

Theorem: Given a quasi-Monte Carlo rule

$$Q(f) = \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}),$$

then we have for the error for $N + 1$ points and $N$ points:

$$(N + 1) E_{N+1}(f) = (N + 1) I(f) - \sum_{k=0}^{N} f(x^{(k)}),$$

$$N E_N(f) = N I(f) - \sum_{k=0}^{N-1} f(x^{(k)}),$$

such that using their difference gives

$$|N E_N(f)| + |(N + 1) E_{N+1}(f)| \geq |I(f) - f(x^{(N)})| = O(1).$$

Suppose the error is $O(N^{-\beta})$, then it follows that $\beta \leq 1$. 
Equal-weight rules are stuck with $O(N^{-1})$.

**Theorem:** Given a quasi-Monte Carlo rule

$$Q(f) = \frac{1}{N} \sum_{k=0}^{N-1} f(x^{(k)}) ,$$

then we have for the error for $N + 1$ points and $N$ points:

$$(N + 1) E_{N+1}(f) = (N + 1) I(f) - \sum_{k=0}^{N} f(x^{(k)}) ,$$

$$N E_N(f) = N I(f) - \sum_{k=0}^{N-1} f(x^{(k)}) ,$$

such that using their difference gives

$$|N E_N(f)| + |(N + 1) E_{N+1}(f)| \geq |I(f) - f(x^{(N)})| = O(1) .$$

Suppose the error is $O(N^{-\beta})$, then it follows that $\beta \leq 1$. 
Weights to the rescue

A modification of the compound point sets theorem:

**Theorem**: Consider $M$ arbitrary rules $Q_i$ using $N_i$ function evaluations and define the compound rule

$$Q(f) := \sum_{i=1}^{M} w_i Q_i(f), \quad \sum_{i=1}^{M} w_i = 1.$$ 

Let $e_{N_i}(Q_i, K)$ be the worst-case error of $Q_i$ then

$$e_N(Q, K) \leq \sum_{i=1}^{M} |w_i| e_{N_i}(Q_i, K).$$

Quasi-Monte Carlo, i.e., the equal-weight rule, uses $w_i = N_i/N$. 
A clever choice of weights

**Theorem:** If the weights are chosen as in

\[
Q(f) := \sum_{i=1}^{M} w_i Q_i(f), \quad w_i := \frac{N_i^a}{N_1^a + \cdots + N_M^a}, \quad a > 0,
\]

and the worst-case error for \( Q_i \) in \( H(K) \) satisfies

\[
e_{N_i}(Q_i, K) \leq C N_i^{-\alpha},
\]

then the worst-case error for the compound rule \( Q \) satisfies

\[
e_N(Q, K) \leq C \frac{M^\max(\min(\alpha, a), 1)}{N^{\min(\alpha, a)}}.
\]

If \( a \geq \alpha \) and \( \alpha > 1 \) then we have \( O(M^\alpha/N^\alpha) \).
An $O(N^{-\alpha})$ algorithm for lattice sequences

Applying the previous theorem to a “good” lattice sequence, with $O(N^{-\alpha})$ convergence for base powers, now gives the desired result.

**Theorem:** For $N = \sum_{\ell=0}^{L-1} n_{\ell} b^\ell$, $L := \lfloor \log_b N \rfloor + 1$, define for each $\ell = 0, \ldots, L-1$ and $j = 1, \ldots, n_{\ell}$:

$$Q_{\ell,j}(f) := \frac{1}{b^\ell} \sum_{k=T_{\ell,j}}^{T_{\ell,j}+b^\ell-1} f(x_k), \quad T_{\ell,j} := \sum_{r=\ell+1}^{L-1} n_r b^r + (j-1)b^\ell,$$

$$Q(f) := \sum_{\ell=0}^{L-1} w_{\ell} \sum_{j=1}^{n_{\ell}} Q_{\ell,j}(f), \quad w_{\ell} := \frac{(b^\ell)^a}{\sum_{\ell'=0}^{L-1} n_{\ell'} (b^{\ell'})^a},$$

then for any $a \geq \alpha$

$$e_N(Q, K) \leq C \left(\left\lfloor \log_b N \right\rfloor + 1)(b - 1)\right)^{\max(\alpha, 1)} \frac{\left(\left\lfloor \log_b N \right\rfloor + 1)(b - 1)\right)^{\max(\alpha, 1)}}{N^\alpha}.$$
Actually easy to implement...  

Although the formulas on the previous slide look complicated, they are actually very easy to implement. For $N = \sum_{\ell=0}^{L-1} n_{\ell} b^{\ell}$:

<table>
<thead>
<tr>
<th>nb of blocks</th>
<th>of size</th>
<th>with weight</th>
<th>and accumulated approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\ell}$</td>
<td>$b^{\ell}$</td>
<td>$w_{\ell} = (b^{\ell})^a / W(N)$</td>
<td>$Q_{\ell} = \sum_{j=1}^{n_{\ell}} Q_{\ell,j}(f)$</td>
</tr>
</tbody>
</table>

Where $W(N) := \sum_{\ell=0}^{L-1} n_{\ell} (b^{\ell})^a$. Then

$$Q(f) = \sum_{\ell=0}^{L-1} w_{\ell} Q_{\ell}.$$  

- When $N$ changes, the values of $Q_{\ell}$ ripple through the table.  
- The table is only of size $\log_b N_{\text{max}}$.  
- The computation can be done for multiple $a$’s at once.
<table>
<thead>
<tr>
<th></th>
<th>QMC integration</th>
<th>A modified classical result</th>
<th>Higher order convergence</th>
<th>Numerical examples</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Quasi-Monte Carlo integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A modified classical result</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Higher order convergence</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Numerical examples</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technology used for the tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functions with infinite smoothness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functions with limited smoothness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For the numerical tests an embedded lattice rule was constructed using the fast component-by-component algorithm from Cools, Kuo and Nuyens (2006).

The rule was constructed for a Korobov space with smoothness parameter 6, i.e., we expect order of convergence 3 for sufficiently smooth functions.

Two generating vectors were used.

For a rule with $2^0 \leq N \leq 2^{15} = 32768$:

$$z = (1, 5277, 4315, 6171, 6413, 14777, 13967, 13215, 2733, 9871),$$

and for a rule with $2^0 \leq N \leq 2^{20} = 1048576$:

$$z = (1, 364981, 245389, 97823, 488939, 62609, 400749, 385317, 21281, 223487).$$
A function with infinite smooth mixed partial derivatives

As a first test function consider:

$$f_\infty(x) := \prod_{j=1}^{s} (1 + \sin(2\pi x_j)),$$

where for all $s \geq 1$

$$\int_{[0,1)^s} f(x) \, dx = 1.$$

From the theory it follows that we have an equal-weight rule for pure powers of the base for any choice of $a$.

At powers of 2 in low dimensions the lattice rule will be exact.
Functions with infinite smoothness

Figure: Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$
### Functions with infinite smoothness

<table>
<thead>
<tr>
<th>QMC integration</th>
<th>A modified classical result</th>
<th>Higher order convergence</th>
<th>Numerical examples</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Figure:** Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$

High order of convergence using lattice sequences for numerical integration

Dirk Nuyens
Figure: Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$
A function up to smoothness 3

As a second test function consider:

\[ f_3(x) := \prod_{j=1}^{s} \left( 1 + (x_j^3 - \frac{3}{2}x_j^2 + \frac{1}{2}x_j) \right), \]

with \( x^3 - \frac{3}{2}x^2 + \frac{1}{2}x \) the Bernoulli polynomial of degree 3, where for all \( s \geq 1 \)

\[ \int_{[0,1)^s} f(x) \, dx = 1. \]

Again, at powers of 2 the results for any \( a \) are all the same.
Figure: Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$
Figure: Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$
A function up to smoothness 6

As a third test function consider:

\[ f_6(x) := \prod_{j=1}^{s} (1 + B_6(x_j)), \]

with \( B_6(x) = x^6 - 2x^5 + \frac{5}{2}x^4 - \frac{1}{2}x^2 + \frac{1}{42} \) the Bernoulli polynomial of degree 6, where for all \( s \geq 1 \)

\[ \int_{[0,1]^s} f(x) \, dx = 1. \]

Again, at powers of 2 the results for any \( a \) are all the same.
Figure: Blue: $a = 1$ (standard QMC); Green: $a = 2$; Red: $a = 3$; Cyanish: $a = 4$; Magentaish: $a = 5$; Yellowish: $a = 6$
Good lattice sequences automatically have $O(N^{-1})$ for any $N$, not just powers of the base.

Using a quasi-Monte Carlo rule, i.e., equal-weight rule, one can do no better than $O(N^{-1})$.

Using weights one can recover the good rate of convergence, as long as the number of point sets is relatively small, e.g., $M \approx \log N$.

Using a good lattice sequences, with high order of convergence for base powers, and the new weighted algorithm, one retains the $O(N^{-\alpha})$ rate of convergence for any $N$.

Note: All big-Oh notations on this slide are modulo powers of $\log N$ and $(\log N)^\alpha$. 