Journal of Computational and Applied Mathematics 233 (2010) 2302–2311

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Theoretical and empirical convergence results for additive congruential random number generators

Roy S. Wikramaratna *

RPS Group plc, A31 Winfrith Technology Centre, Dorchester, Dorset DT2 8DH, United Kingdom

ARTICLE INFO

Article history:
Received 24 July 2008
Received in revised form 6 August 2009

MSC:
65C10
68U20
11K36
11K45

Keywords:
Pseudo-random number generator
Algorithm
Implementation
Theoretical analysis
Well-distributed sequences
Empirical test

ABSTRACT

Additive Congruential Random Number (ACORN) generators represent an approach to generating uniformly distributed pseudo-random numbers that is straightforward to implement efficiently for arbitrarily large order and modulus; if it is implemented using integer arithmetic, it becomes possible to generate identical sequences on any machine.

This paper briefly reviews existing results concerning ACORN generators and relevant theory concerning sequences that are well distributed mod 1 in k dimensions. It then demonstrates some new theoretical results for ACORN generators implemented in integer arithmetic with modulus $M = 2^{\mu}$ showing that they are a family of generators that converge (in a sense that is defined in the paper) to being well distributed mod 1 in k dimensions, as $\mu = \log_2 M$ tends to infinity. By increasing k, it is possible to increase without limit the number of dimensions in which the resulting sequences approximate to well distributed.

The paper concludes by applying the standard TestU01 test suite to ACORN generators for selected values of the modulus (between 2^{60} and 2^{150}), the order (between 4 and 30) and various odd seed values. On the basis of these and earlier results, it is recommended that an order of at least 9 be used together with an odd seed and modulus equal to 2^{30p}, for a small integer value of p. While a choice of $p = 2$ should be adequate for most typical applications, increasing p to 3 or 4 gives a sequence that will consistently pass all the tests in the TestU01 test suite, giving additional confidence in more demanding applications.

The results demonstrate that the ACORN generators are a reliable source of uniformly distributed pseudo-random numbers, and that in practice (as suggested by the theoretical convergence results) the quality of the ACORN sequences increases with increasing modulus and order.

1. Introduction

There are many mathematical and numerical problems whose solution requires a reliable source of uniformly distributed pseudo-random numbers. Monte Carlo methods provide one example of such a solution method, with applications including numerical optimisation, numerical integration, Bayesian inference, geostatistical simulation, statistical physics and other statistical applications.

The term 'Monte Carlo method' has been widely used, and various definitions have been adopted by different authors, see for example [1–4]. In this article, following Wikramaratna [5], the term is used in its most general sense.

In theory the accuracy of a properly formulated Monte Carlo method increases with the number of realisations; in practice this continuing improvement is dependent on the quality of a pseudo-random number generator. The question of precisely

* Corresponding author. Tel.: +44 1305 217440.
E-mail addresses: roy@wikramaratna.fsnet.co.uk, WikramaratnaR@rpsgroup.com.

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.10.015

what conditions must be satisfied by the random sequence used in a Monte Carlo calculation to ensure convergence of the simulation results has been the subject of much debate (see, for example, [6]). Knuth considers various alternative definitions of what is meant by a random sequence, and introduces the concepts of a k-distributed sequence and of an infinite-distributed or completely uniformly distributed sequence (which is the limiting case of a k-distributed sequence as k tends to infinity). We note that, although it is possible to define infinite-distributed sequences deterministically (Franklin [7]), there is no computationally practical algorithm for evaluating these sequences accurately on a computer with finite word-length and working in finite-precision arithmetic. In fact, it can be argued that the concept of an infinite-distributed sequence is meaningful only in relation to infinite sequences (rather than for the finite-length sequences that are used in practice in computations); as a result we believe that k-distributedness (for any specified value of k) is a more useful property to consider in practice.

The ACORN generators were first proposed in [8] in 1989. Subsequent papers over an extended period [9,5,10] have suggested that the ACORN approach compares favourably with some other commonly used approaches, in particular the linear congruential generators.

New results, documented in this paper, show that ACORN generators merit consideration for use in the most demanding applications. We demonstrate that the ACORN generators are a family of generators that allow us to approximate arbitrarily closely to a k-distributed sequence for any specified finite value of k. We go on to provide a practical demonstration of the convergence results by applying a standard suite of empirical tests (the TestU01 test suite – see L'Ecuyer and Simard [11]) to a whole series of ACORN sequences of different order and modulus and initialised with a range of different seeds.

2. The ACORN generator

The kth-order Additive Congruential Random Number (ACORN) generator is defined in [8,9] from an integer modulus M, an integer seed Y_0^0 satisfying $0 < Y_0^0 < M$ and an arbitrary set of k integer initial values Y_0^m, $m = 1, \ldots, k$, each satisfying $0 \leq Y_0^m < M$, through the equations

$$Y_n^0 = Y_{n-1}^0 \quad n \geq 1 \tag{1}$$

$$Y_n^m = (Y_n^{m-1} + Y_{n-1}^m)_{\bmod M} \quad n \geq 1, m = 1, \ldots, k \tag{2}$$

where by $(Y)_{\bmod M}$ we mean the remainder on dividing Y by M. The numbers Y_n^m can be normalised to the unit interval by dividing by M:

$$X_n^m = Y_n^m / M \quad n \geq 1. \tag{3}$$

The original implementation proposed in [8] used real arithmetic modulo 1, calculating the X_n^m directly. Owing to the effects of rounding errors in real arithmetic the sequences were not reproducible on different machines or with different compilers, although the sequences still exhibited similar statistical behaviour; consequently, although period lengths were large, they could not be predicted or determined with any certainty; finally, it was not possible to make a clear and unambiguous statement of how best to initialise the generator. Use of an integer implementation based on Eqs. (1) to (3) overcomes these limitations and the original implementation is therefore considered to be superseded by this newer approach [9,10]. In the present paper we give theoretical results that illustrate the close relationship between the two different representations, and that demonstrate the value of considering the algorithm defined by Eqs. (1) to (3) as a special case of the more general concept that will be defined below by Eqs. (5) and (6).

Empirical testing, including that documented in [8] and more recently (making use of the Diehard statistical test suite, Marsaglia [12]) in [10,13], has demonstrated that the numbers X_n^k approximate to being uniformly distributed in the half-open unit interval [0, 1) and satisfy a wide range of statistical tests of randomness. Results from these tests suggested that increasing the order of the generator improves the randomness and also that increasing the modulus improves the randomness of the generator. These tentative conclusions are strongly supported by the more detailed testing that has now been completed (making use of the TestU01 test suite, L'Ecuyer and Simard [11]) — the results of these empirical tests, which were applied to several hundreds of different ACORN generators, are documented later in the present paper.

We observe that the ACORN generator is defined on the half-open interval [0, 1), and that it is possible for identically zero values to occur, albeit rarely. When the resulting variates are converted to real values on the unit interval, it is possible in addition to have values that get rounded to either 0 or 1 given the available precision. With some applications, the occurrence of the values 0 or 1 can cause computational problems; if this is the case then either the application or the pseudo-random number generator needs to be modified to address this. We note that the computational issue arises primarily because of the limited precision available in real arithmetic rather than because of the generator being defined on the half-open interval (for example, for an ACORN generator defined modulo 2^{60}, we can in principal redefine it on the open interval (0, 1) simply by adding 2^{-61} to each variate; however this would make no difference whatsoever to the issues that may arise from variates getting rounded to 0 or 1).

It is worth briefly touching on the computational performance of the ACORN algorithm, compared with other standard algorithms for pseudo-random number generation. L'Ecuyer and Simard [11] quote figures of 4.3 s for the Mersenne Twister algorithm MT19337 (see [14], for a description of this algorithm) to generate 10^8 random variates on a 32-bit processor with

2.8 GHz clock speed; in general they obtained times that ranged from 2 s up to more than 100 s for algorithms that were successful on the TestU01 tests. Wikramaratna [10] tabulated the computational performance of one implementation of the ACORN algorithm for a range of values of order and modulus, giving performance of between 100 and 1000 s to generate 10^8 variates on a 32-bit processor with 600 MHz clock speed. We have undertaken further testing of the ACORN algorithm and have been able to get an overall speed-up by a factor of around 50 by a combination of faster 32-bit processor (2.55 GHz clock speed), full compiler optimisation using a Digital Visual Fortran compiler (v5.0.A) and some improvements to the way in which we have implemented the algorithm and in particular the integer addition modulo 2^{30}. This overall speedup can be broken down into a factor of 5 due to the faster cpu, a factor of 2 as a result of the compiler optimisation, and a factor of 5 from the algorithmic improvements. For a tenth-order ACORN generator, modulus 2^{60}, the time for generating 10^8 variates was reduced to around 5 s (compared with 287 s for generating the same number of variates based on Table 6 of [10]). If the modulus is increased from 2^{60} to 2^{90}, this increases the time by about 40%. These timings are still for a completely general version of ACORN, which can be called with any value of the order and for any modulus of the form 2^{30p}, where p is any positive integer. Further performance improvements might be obtained by restricting the routine to a single specified order and/or a single specified value of p. On the basis of these results we believe the computational performance is on a par with the best alternative pseudo-random number generation algorithms. A more precise evaluation of computational performance would require direct comparison of specific implementations of each algorithm run on the same hardware and under similar conditions. However, we note that in practice for any real Monte Carlo calculation the generation of random variates is likely to consume only a small proportion of the total computational effort, so any more detailed comparison of timings is likely to be of mainly academic interest. From a practical point of view the quality of the pseudo-random sequences and the impact of this quality on the convergence of the Monte Carlo estimates is much more significant than the computational performance.

Wikramaratna [9] showed that the numbers Y_n^m are of the form

$$Y_n^m = \left(\sum_{i=0}^{m} Y_0^i Z_n^{m-i}\right)_{\bmod M} = \left(\sum_{i=0}^{m} Y_0^i \left(\frac{(n+m-i-1)!}{(n-1)!(m-i)!}\right)\right)_{\bmod M}. \tag{4}$$

The analysis went on to show that the kth-order ACORN generator approximates to being uniformly distributed in all dimensions up to k. In this paper we define more precisely the sense in which this statement holds true, and specify a few simple constraints on the parameter values that need to be satisfied in order to achieve this.

3. Background theory

3.1. Basic definitions

We recap briefly on some notation and basic definitions used in [9]. Let $\mathcal{R}$ be the real line, let $\mathcal{I}$ be the restriction of the real line to the unit interval, and let $\mathcal{Z}$ be the set of integers. Let $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ be a vector with real components; thus $\mathbf{x}\varepsilon\mathcal{R}^k$. The *fractional part* of $\mathbf{x}$ is $\{\mathbf{x}\} = (\{x_1\}, \{x_2\}, \ldots, \{x_k\})$. The *k-dimensional unit cube* $\mathcal{I}^k$ is the interval $[\mathbf{0}, \mathbf{1})$. The set of points $\mathbf{x}\varepsilon\mathcal{I}^k$ such that $a_j \le x_j < b_j$ for $j = 1, 2, \ldots, k$ will be denoted by $[\mathbf{a}, \mathbf{b})$. Let $(\mathbf{x}_n)$, $n = 1, 2, \ldots$, be a sequence of vectors in $\mathcal{R}^k$. For a subset $[\mathbf{a}, \mathbf{b})$ of $\mathcal{I}^k$ and for any given integer p let the counting function $A([\mathbf{a}, \mathbf{b}); N, p)$ denote the number of points $\{\mathbf{x}_{p+n}\}$, $1 \le n \le N$, that lie in $[\mathbf{a}, \mathbf{b})$.

Definition. The sequence of vectors $(\mathbf{x}_n)$, $n = 1, 2, \ldots$, is said to be *u.d. (uniformly distributed) mod 1 in* $\mathcal{R}^k$ if, for all half-open intervals $[\mathbf{a}, \mathbf{b})$ contained in or equal to $\mathcal{I}^k$, and for each fixed p greater than or equal to zero we have $\lim_{N\to\infty} (A([\mathbf{a}, \mathbf{b}); N, p)/N) = \prod_{j=1}^{k}(b_j - a_j)$. If in addition the convergence to this limit is uniform in p, then the sequence $(\mathbf{x}_n)$, $n = 1, 2, \ldots$, is said to be *w.d. (well distributed) mod 1 in* $\mathcal{R}^k$. A sequence of real numbers (r_n), $n = 1, 2, \ldots$, is said to be u.d. (respectively w.d.) mod 1 in $\mathcal{R}^k$ if the corresponding sequence of vectors $(\mathbf{x}_n) = (r_n, r_{n+1}, \ldots, r_{n+k-1})$, $n = 1, 2, \ldots$ is u.d. (respectively w.d.) mod 1 in $\mathcal{R}^k$. □

The formal definition of u.d. mod 1 was due originally to Weyl [15,16], who proved the Weyl Criterion giving a necessary and sufficient condition for a sequence to be u.d. mod 1. The special case known as w.d. mod 1 was defined by [17,18], who also gave the corresponding Weyl Criterion for a sequence to be w.d. mod 1. These definitions and the corresponding versions of the Weyl Criterion are also discussed in [19]. It follows from the definitions that a sequence that is w.d. mod 1 in $\mathcal{R}^k$ must necessarily be u.d. mod 1 in $\mathcal{R}^k$; on the other hand it is straightforward to find counter-examples to demonstrate that the converse is not true.

3.2. The key existing result concerning ACORN sequences

Eqs. (1) to (3) can be rewritten in the form

$$X_n^0 = X_{n-1}^0 \quad n \ge 1 \tag{5}$$

$$X_n^m = (X_n^{m-1} + X_{n-1}^m)_{\bmod 1} \quad n \ge 1, m = 1, \ldots, k \tag{6}$$

where X_n^0 satisfies $0 < X_n^0 < 1$ and X_n^m $(m \geq 1)$ satisfy $0 \leq X_n^m < 1$. We observe that if the X_n^m are restricted to being rational fractions, then this is exactly equivalent to (1) to (3) for a suitably chosen value of M; however it is also useful from a theoretical point of view to generalise Eqs. (5) and (6) to allow the X_n^m to take any real values in the appropriate ranges. We observe that the X_n^m can be written in the following form, which is equivalent to Eq. (4):

$$X_n^m = \left(\sum_{i=0}^{m} X_0^i Z_n^{m-i} \right)_{\text{mod } 1} = \left(\sum_{i=0}^{m} X_0^i \left(\frac{(n+m-i-1)!}{(n-1)!(m-i)!} \right) \right)_{\text{mod } 1}. \tag{7}$$

Wikramaratna [9] proved a number of theorems concerning well-distributed sequences, leading to the key result that the kth-order ACORN sequence X_n^k, $n = 0, 1, 2, \ldots$, defined by Eq. (7) with $m = k$ is w.d. mod 1 in $\mathcal{R}^k$ provided only that the seed X_0^0 takes an irrational value.

It is worth noting as an aside that Knuth [6] has shown that if a sequence of real numbers (r_n), with u_n equal to the fractional part of r_n, is u.d. mod 1 in $\mathcal{R}^k$ and if $f(x_1, x_2, \ldots, x_k)$ is a Riemann-integrable function of k variables, then the limit $\left(\lim_{n\to\infty} \frac{1}{n} \sum_{1\leq j\leq n} f(u_j, u_{j+1,\ldots,} u_{j+k-1})\right)$ exists and is equal to $\int_0^1 \cdots \int_0^1 f(x_1, x_2, \ldots, x_k)\, \mathrm{d}x_1 \ldots \mathrm{d}x_k$. Knuth uses this result to prove that various standard statistical tests (in particular, the permutation test of order k and the serial correlation test for pairs of terms with a separation of $k-1$) will necessarily be satisfied for any sequence that is u.d. mod 1 in $\mathcal{R}^k$. Since w.d. mod 1 in $\mathcal{R}^k$ is a stronger condition than u.d. mod 1 in $\mathcal{R}^k$, it is clear that these same results would apply to ACORN sequences defined by (7) and having an irrational seed.

It remains an open question whether, and if so under what precise conditions, the properties of u.d. (or w.d.) mod 1 in $\mathcal{R}^k$ might be sufficient to guarantee convergence of particular examples or classes of Monte Carlo simulations.

4. Convergence results for ACORN sequences

An interesting feature of the ACORN approach concerns the existence of two different ways of specifying the ACORN sequences. The first uses Eqs. (1) to (3) and exact integer arithmetic (leading to practical implementations, reproducibility of sequences as well as to theoretical results concerning periodicity that in practice define a maximum useable length within which the sequence is guaranteed not to repeat itself). The second approach uses Eqs. (5) and (6), which are exactly equivalent to (1) to (3) for certain specific choices of (rational) seed and initial values, but which also go further, allowing some powerful number theoretic results to be proved when the method is generalised to allow irrational values to be chosen for the seed (and also, if desired, for the initial values). The two theorems that are proved in the following section help to clarify the relationship between the two approaches, and give us practical criteria under which we can calculate the terms in the more general sequences to arbitrary accuracy (limited only by the precision available for the representation of real numbers with the particular combination of hardware and compiler) as the limit of a sequence of variates that can be evaluated to any desired accuracy using the more restrictive approach.

We show first, in Theorem 1, that every kth-order ACORN sequence with modulus $M = 2^\mu$ is equivalent (to $\beta < \mu$ binary digits precision) to the first M terms in a sequence that is w.d. mod 1 in $\mathcal{R}^k$ (in fact, we go further than this, showing that there are actually an infinite number of such equivalent sequences). This result demonstrates that every kth-order ACORN sequence is an approximation to a sequence that is w.d. mod 1 in $\mathcal{R}^k$, and that increasing the modulus used in defining the ACORN sequence leads to a better approximation to w.d. mod 1 in $\mathcal{R}^k$ (in the sense that this increases both the maximum number of terms M that can be generated and also the available precision β).

Second, in Theorem 2, we show that given a specified ACORN sequence that is w.d. mod 1 in $\mathcal{R}^k$ (in particular, using the result proved in [9], this requires only that the order should be k and the seed should be irrational) a number N of terms required and a number β of binary digits precision required then we can define an ACORN sequence and evaluate it on a finite-precision computer such that the first N terms of the ACORN sequence are equal (to β binary digits precision) to the first N terms of the specified k-distributed sequence. We also provide a lower bound on the modulus and a specification of the corresponding seed and initial values that will allow these terms to be calculated to the desired accuracy. This shows that it is possible to generate any specified number of terms from certain particular sequences that are w.d. mod 1 in $\mathcal{R}^k$ to arbitrary accuracy, limited only by the available precision for storing real numbers.

Theorem 2 shows that in the event of there being certain particular choices of seed that gave a particularly rapid convergence, then we would have a practical method of generating those specific sequences to arbitrary accuracy and for arbitrary numbers of terms on any computer. In practice we have found that for any given choice of modulus $M = 2^\mu$, the ACORN method gives remarkably consistent statistical behaviour over the whole spectrum of choices of seed and initial values, provided only that the seed takes an odd value (which in turn guarantees the period length of the resulting sequence). Thus in our experience we do not usually need to fall back on Theorem 2 in practical applications; however the existence of the convergence proofs tells us about the limiting behaviour, and would allow us for example to test the adequacy of the results of a Monte Carlo calculation by repeating the calculation using an ACORN generator with successively larger values of the modulus and order until we were able to verify that the results were sufficiently well converged.

Theorem 1. *Given an arbitrary kth-order ACORN sequence specified by Eqs. (1) to (3) together with a modulus $M = 2^\mu$ and an appropriate set of initial conditions (including an odd value for the seed), together with a required precision $\beta \leq \mu$, then the first*

M terms of the sequence are equal (to β binary digits precision) to the first M terms of an infinite sequence that is w.d. mod 1 *in* $\mathcal{R}^k$.

Proof. The proof is by construction of a sequence that is w.d. mod 1 in $\mathcal{R}^k$ and satisfies the requirements of the theorem (in fact, we show that there are an infinite number of such sequences).

Consider an arbitrary kth-order ACORN sequence, having seed Y_0^0 and initial values Y_0^m, $m = 1, \ldots, k$.

By definition, the kth-order ACORN sequence defined by Eqs. (5) and (6) with seed $X_0^0 = Y_0^0/M$ and initial values $X_0^m = Y_0^m/M$ is identical to the specified sequence.

Now perturb this ACORN sequence by adding $\varepsilon/2^s$ to the seed, for an arbitrary irrational ε, $0.5 < \varepsilon < 1$, and some integer $s \geq 1$ (to be specified), and keeping the initial values unchanged. We note that restricting the choice of ε and s in this way ensures that each allowable combination gives rise to a unique value of $\varepsilon/2^s$ and consequently to a unique perturbed sequence. For each choice of ε and s, the perturbed sequence is an ACORN sequence with an irrational seed, equal to $(X_0^0 + \varepsilon/2^s)$; therefore using the result from [9] the perturbed sequence is w.d. mod 1 in $\mathcal{R}^k$ for every choice of s.

The magnitude of the perturbation, equal to the difference between the Mth term in the original sequence and the Mth term in the perturbed sequence, is itself an ACORN sequence with seed $\varepsilon/2^s$ and all the initial values set to zero; substituting these values into Eq. (7), we obtain the following equation for the magnitude of the perturbation:

$$\begin{aligned}\delta_n^k &= \left(\frac{\varepsilon}{2^s} Z_n^k\right)_{\text{mod 1}} = \left(\frac{\varepsilon}{2^s}\frac{(n+k-1)!}{(n-1)!k!}\right)_{\text{mod 1}} \\ &= \frac{\varepsilon}{2^s} n^k \left(\frac{1\left(1+\frac{1}{n}\right)\left(1+\frac{2}{n}\right)\ldots\left(1+\frac{k-1}{n}\right)}{k!}\right) \leq \frac{\varepsilon}{2^s} n^k \leq \frac{\varepsilon}{2^s} M^k \end{aligned} \tag{8}$$

provided that $1 \leq n \leq M$. Now choose

$$s > s_0 = 1 + \beta + k\mu. \tag{9}$$

Then the magnitude of the perturbation will satisfy

$$\delta_n^k \leq \delta_M^k < \frac{\varepsilon}{2^{1+\beta+k\mu} M^{-k}} = \frac{\varepsilon}{2^{1+\beta+k\mu}\,(2^\mu)^{-k}} = \frac{\varepsilon}{2^{1+\beta}} < \frac{1}{2^{1+\beta}}, \quad n = 1, , 2, \ldots, M. \tag{10}$$

Thus corresponding terms in the two sequences will be equal to an accuracy of at least β binary digits. □

Theorem 2. *Given any kth-order ACORN sequence specified by Eqs. (5) and (6), together with an appropriate set of initial conditions (in particular, with an irrational seed χ_0^0 – which ensures that the sequence is w.d. mod 1 in $\mathcal{R}^k$ – and an arbitrary set of k initial values χ_0^m, $m = 1, \ldots, k$), then we can calculate the first $N = 2^v$ terms of the sequence to β binary digits accuracy from an ACORN sequence specified by Eqs. (1) to (3) with appropriate values of the modulus, seed and initial values.*

In particular, given β and $N = 2^v$, we can choose the modulus equal to $M = 2^\mu$ where μ is the least integer such that

$$\begin{aligned}\mu &> 2 + \beta + kv + \log_2\left(\frac{(1+\frac{k-1}{2N})^k}{(k-1)!}\right) \\ &> 2 + \beta + kv + \log_2\left(\frac{1(1+\frac{1}{N})\ldots(1+\frac{k-1}{N})}{(k-1)!}\right) = 2 + \beta + \log_2\left(\frac{(N+k-1)!}{(N-1)!(k-1)!}\right)\end{aligned} \tag{11}$$

with the seed and initial values chosen to be equal to

$$Y_0^m = 1 + 2\left[\frac{M}{2}\chi_0^m\right] \quad m = 0, \ldots, k \tag{12}$$

where the notation $[\chi]$ means the integer part of χ.

Proof. Let $M = 2^\mu$, where μ is as specified in (11) and consider a kth-order ACORN generator, defined by Eqs. (1) to (3) with modulus equal to M and seed and initial values as specified in (12). Note that choosing this form for the Y_0^m ensures in particular that the seed takes an odd value; Wikramaratna [9] has shown that the period length of an ACORN sequence with modulus equal to a power of 2 will be an integer multiple of the modulus, provided only that the seed is chosen to be odd, so the adopting of (12) guarantees a period length of at least M. By definition, this generator is identical to the kth-order ACORN generator defined by Eqs. (5) to (7) with rational seed and initial values equal to

$$X_0^m = \frac{Y_0^m}{M} \quad m = 0, \ldots, k. \tag{13}$$

Table 1
Values of the function $f(k, v)$, Eq. (19), for various values of k and v.

v	k				
	8	10	12	15	20
5	29.9	34.4	38.5	43.9	51.7
10	68.7	82.6	95.8	114.8	144.5
15	108.7	132.5	155.8	189.7	244.3
20	148.7	182.5	215.7	264.7	344.2
25	188.7	232.5	275.7	339.7	444.2
30	225.7	279.2	332.2	410.7	539.9

We can consider this sequence to be a perturbation of the original sequence; the magnitude of the perturbation is itself a kth-order ACORN sequence defined by Eqs. (5) to (7) with seed and initial values given by

$$\delta_0^m = \chi_0^m - X_0^m < \frac{1}{M} \quad m = 0, \ldots, k. \tag{14}$$

By definition, since χ_0^0 is irrational, the seed δ_0^0 must be non-zero.

Substituting the seed and initial values for the perturbation into Eq. (7), then we can use the inequality in (14) to show that, as long as the magnitude of the perturbation remains less than unity,

$$\begin{aligned}\delta_n^k &= \sum_{i=0}^{k} \delta_0^i Z_n^{k-i} = \sum_{i=0}^{k} \delta_0^i \left(\frac{(n+k-i-1)!}{(n-1)!(k-i)!}\right) \\ &< \frac{(k+1)}{M}\left(\frac{(n+k-1)!}{(n-1)!k!}\right).\end{aligned} \tag{15}$$

Rearranging Eq. (11), and making use of the fact that $\frac{(k+1)}{k} < 2$

$$M = 2^{\mu} > 2^{1+\beta}(k+1)\left(\frac{(N+k-1)!}{(N-1)!k!}\right) \tag{16}$$

$$\frac{(k+1)}{M}\left(\frac{(N+k-1)!}{(N-1)!k!}\right) < \frac{1}{2^{1+\beta}}. \tag{17}$$

Then, making use of the fact that the δ_n^k in (15) is a monotonic increasing function of n, we can substitute from Eqs. (17) into (15) to give

$$\delta_n^k \le \delta_N^k < \frac{(k+1)}{M}\left(\frac{(N+k-1)!}{(N-1)!k!}\right) < \frac{1}{2^{1+\beta}} \quad n = 1, 2, \ldots, N \tag{18}$$

so the first N terms in the perturbed sequence and in the original sequence are identical to an accuracy of β binary digits, as required by the theorem. □

We observe that in Theorem 2 we can choose any irrational value lying between 0 and 1 for the seed, provided that we are able to calculate it to any required accuracy. Thus suitable choices would for example include values such as $(\pi/4), (\pi - 3)$, $(\sqrt{2}/2), (\sqrt{3} - 1), (\sqrt{3} - \sqrt{2})$.

If we define the function

$$f(k, v) = 1 + kv + \log_2\left(\frac{(1 + \frac{k-1}{2N})^k}{(k-1)!}\right) \tag{19}$$

(where it must be remembered that $N = 2^v$ is itself a function of v) then the first line of Eq. (11) becomes

$$\mu > 1 + \beta + f(k, v). \tag{20}$$

For any given values of k and v, it is possible to estimate the corresponding value of $f(k, v)$ from Eq. (19). Table 1 shows values of $f(k, v)$ tabulated for a range of values of k and v. Suppose that we are working in double-precision real arithmetic (for which the typical precision would be around 43 binary digits). We can use Eq. (20) to estimate a lower bound on the modulus to generate sequences of length 2^v that can be guaranteed well distributed to the full available precision of 43 binary digits to be $M = 2^{\mu}$, where $\mu > 1 + 43 + f(k, v)$. Thus for example taking $k = 10$ and $v = 20$, we would obtain $f(k, v) = 182.5$ and $\mu > 227$. Use of the resulting sequence would be computationally practicable, taking some three times as long, per variate, compared with a tenth-order ACORN generator with $\mu = 90$.

Table 2
Results of applying TestU01 BigCrush test battery to ACORN generators with modulus 2^{60}.

Order	Period	Seed = 54739173		Seed = 1		Seed = 123456789		Seed = 987654321		Seed = 12101955		Seed = 55910121		Average	
		Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect
4	2^{62}	34	3	33	2	34	4	29	2	33	3	32	2	32.5	2.7
5	2^{62}	19	0	19	0	19	2	18	1	19	0	19	0	18.8	0.5
6	2^{62}	8	0	9	0	9	0	9	0	8	0	9	0	8.7	0.0
7	2^{62}	6	1	6	1	5	2	7	0	6	1	7	0	6.2	0.8
8	2^{63}	2	0	2	0	2	0	2	1	2	0	2	0	2.0	0.2
9	2^{63}	1	0	1	0	1	0	1	0	1	0	0	1	0.8	0.2
10	2^{63}	1	0	1	0	1	0	1	0	1	0	0	1	0.8	0.2
11	2^{63}	1	0	1	0	1	0	1	0	1	0	1	0	1.0	0.0
12	2^{63}	1	0	1	0	1	0	0	1	0	1	1	0	0.7	0.3
13	2^{63}	0	1	1	0	1	0	1	0	0	1	1	0	0.7	0.3
14	2^{63}	1	1	1	0	1	0	1	0	1	0	1	0	1.0	0.2
15	2^{63}	1	0	1	0	0	0	1	0	0	0	1	0	0.7	0.0
16	2^{64}	1	0	1	1	1	0	1	1	1	0	1	2	1.0	0.7
17	2^{64}	0	1	0	0	0	0	0	0	0	0	0	1	0.0	0.3
18	2^{64}	0	1	0	0	0	0	0	0	0	0	1	0	0.2	0.2
19	2^{64}	0	0	0	2	0	0	0	1	0	0	1	0	0.2	0.5
20	2^{64}	0	0	0	1	0	0	0	0	0	0	0	0	0.0	0.2
25	2^{64}	0	1	0	0	0	0	0	0	0	0	0	1	0.0	0.3
30	2^{64}	0	1	0	0	0	0	0	0	0	0	0	0	0.0	0.2

5. Empirical testing using TESTU01

L'Ecuyer and Simard [11] have considered the application of empirical tests of uniformity and randomness for sequences generated by a wide range of algorithms. They have developed a comprehensive set of empirical tests that are designed to detect undesirable characteristics in such sequences. L'Ecuyer and Simard describe the TestU01 package in some detail and present results of applying it to a large number of different sequences, identifying generators that pass all of the tests (collectively called the BigCrush test battery), as well as identifying many generators (including some that are widely used) that have serious deficiencies as regards certain specific tests.

In their paper, L'Ecuyer and Simard [11] tested a large number of different pseudo-random number generation algorithms (about 90 different algorithms in total, that were either widely used or recently proposed) and identified a subset of these generators that passed all of their tests, according to a specific set of criteria. L'Ecuyer and Simard identify the generators that survived their testing as the long-period multiplicative recursive generators with good structure, the multiplicative lagged-Fibonacci generators, some non-linear generators designed for cryptology and some combined generators having components from different families; they suggest that the combined generators should be given more attention because theoretical guarantees about their uniformity can be proved, their period can easily be made very long, and splitting their period into disjoint sub-sequences is easy (assuming that it can be done for their components).

The ACORN generators were not included among those that were specifically considered by L'Ecuyer and Simard, although all ACORN generators have previously been shown [10] to be equivalent to particular examples of multiple recursive generators that are straightforward to initialise and to compute and have good structure. In the present paper, we have applied the same set of criteria (as used by L'Ecuyer and Simard) to systematically test more than three hundred ACORN generators with different moduli (2^{60}, 2^{90}, 2^{120}, and 2^{150}) and orders (ranging between 4 and 30) and for a number of initialisations (selecting different odd values for the seed in each case). The tests were run in 2008 using Version 0.6.1 of the TestU01 package, using the BigCrush battery of tests (which calculates 90 different test statistics for each sub-sequence that is tested, generating some 2^{38} pseudo-random numbers from each sub-sequence) together with an implementation of the ACORN algorithm that was being developed and tested by Numerical Algorithms Group Ltd for inclusion in Mark 22 of their Software Libraries [20], which has recently been released. We have followed L'Ecuyer and Simard in defining a "failure" to be a p-value outside the range $[10^{-10}, 1-10^{-10}]$ with a "suspect" value falling in one of the ranges $[10^{-10}, 10^{-4}]$ or $[1-10^{-4}, 1-10^{-10}]$. The choice of the six seed values used for testing was in effect arbitrary, with the only restriction being to choose odd values. Results are presented in Tables 2–5 (for modulus 2^{60}, 2^{90}, 2^{120}, and 2^{150} respectively). The tables show, in each case, the specification of the ACORN sequence (order and resulting period length) and then for each choice of seed a tabulation of the number of tests failed and the number of suspect values. The final columns of each table give the average number of failures and of suspect values over the whole range of initialisations that were tested.

The results obtained with the ACORN sequences show that for each of the initialisations that were tested the ACORN generators with order 9 or more passed all of the tests, provided the modulus was 2^{90} or greater. Close inspection of the small number of "suspect" values for generators that passed all of the tests shows that there is no systematic pattern in the particular tests that threw up suspect values, suggesting that they do indeed represent statistical 'noise' rather than any systematic problem with the generators.

Where the modulus was limited to 2^{60}, the reduction in precision led to a slight degradation in the TestU01 results – even so, each ACORN sequence of order 9 or more failed on at most one of the tests (with the failure occurring on one

Table 3
Results of applying TestU01 BigCrush test battery to ACORN generators with modulus 2^{90}.

Order	Period	Seed = 54739173		Seed = 1		Seed = 123456789		Seed = 987654321		Seed = 12101955		Seed = 55910121		Average	
		Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect
4	2^{92}	25	1	28	1	25	2	24	5	36	2	28	2	27.7	2.2
5	2^{92}	17	1	18	1	19	0	18	2	19	0	18	1	18.2	0.8
6	2^{92}	9	0	9	0	9	0	9	0	9	0	9	0	9.0	0.0
7	2^{92}	5	2	5	1	7	0	7	0	5	2	7	0	6.0	0.8
8	2^{93}	2	0	2	0	2	0	2	0	2	0	2	0	2.0	0.0
9	2^{93}	0	0	0	0	0	0	0	1	0	1	0	0	0.0	0.3
10	2^{93}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
11	2^{93}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
12	2^{93}	0	1	0	0	0	0	0	0	0	0	0	0	0.0	0.2
13	2^{93}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
14	2^{93}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
15	2^{93}	0	0	0	0	0	0	0	1	0	0	0	0	0.0	0.2
16	2^{94}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
17	2^{94}	0	0	0	0	0	0	0	1	0	1	0	0	0.0	0.3
18	2^{94}	0	0	0	0	0	0	0	0	0	0	0	1	0.0	0.2
19	2^{94}	0	0	0	0	0	1	0	0	0	0	0	0	0.0	0.2
20	2^{94}	0	0	0	0	0	0	0	0	0	2	0	0	0.0	0.3
25	2^{94}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
30	2^{94}	0	0	0	0	0	0	0	0	0	1	0	0	0.0	0.2

Table 4
Results of applying TestU01 BigCrush test battery to ACORN generators with modulus 2^{120}.

Order	Period	Seed = 54739173		Seed = 1		Seed = 123456789		Seed = 987654321		Seed = 12101955		Seed = 55910121		Average	
		Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect
6	2^{122}	9	0	8	0	8	1	9	0	7	1	8	1	8.2	0.5
7	2^{122}	6	1	6	2	5	3	5	2	6	2	6	0	5.7	1.7
8	2^{123}	2	0	2	1	2	0	2	0	2	0	2	0	2.0	0.2
9	2^{123}	0	0	0	0	0	0	0	0	0	0	0	1	0.0	0.2
10	2^{123}	0	0	0	1	0	0	0	0	0	0	0	0	0.0	0.2
12	2^{123}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
15	2^{123}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
20	2^{124}	0	0	0	0	0	0	0	0	0	1	0	0	0.0	0.2

Table 5
Results of applying TestU01 BigCrush test battery to ACORN generators with modulus 2^{150}.

Order	Period	Seed = 54739173		Seed = 1		Seed = 123456789		Seed = 987654321		Seed = 12101955		Seed = 55910121		Average	
		Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect	Failures	Suspect
6	2^{152}	9	0	9	0	9	0	9	0	9	0	9	0	9.0	0.0
7	2^{152}	5	2	7	1	6	2	6	1	6	1	7	1	6.2	1.3
8	2^{153}	2	0	2	0	2	0	2	0	2	0	2	0	2.0	0.0
9	2^{153}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
10	2^{153}	0	0	0	0	0	2	0	0	0	0	0	0	0.0	0.3
12	2^{153}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
15	2^{153}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0
20	2^{154}	0	0	0	0	0	0	0	0	0	0	0	0	0.0	0.0

specific example of the eight-dimensional Birthday Spacings test in each such case), while for generators with order 20 or more, each of the ACORN generators that was tested passed all of the tests even with modulus 2^{60}. The TestU01 tests also demonstrated the limitation of ACORN generators with lower order: for all of the ACORN generators with order 8 (or less) that were tested, the generators failed on at least two of the tests irrespective of the modulus, with the number of failures progressively increasing as the order was further reduced.

Our latest results suggest that all ACORN generators with order 9 or more having sufficient modulus (a power of 2, at least 2^{90}) and initialised with an odd seed value are likely to be included in the list of generators that survive the TestU01 tests (whereas ACORN generators with order 8 or less should certainly be excluded from this list). We have also demonstrated that these ACORN generators share the attractive characteristics (namely, that theoretical guarantees about their uniformity can be proved, their period can be made very long, and splitting their period into disjoint sub-sequences is straightforward) that were attributed by L'Ecuyer and Simard to combined generators and in addition they have the property that their statistical performance can be further improved as required by increasing the order and then choosing a sufficiently large power of 2

as the modulus and an arbitrary odd value for the seed. We believe that this makes the ACORN generators attractive for use in simulation, either on their own or as one component in a combined generator.

It is worth observing that L'Ecuyer and Simard [11] made use of a more recent version of their TestU01 test suite, which includes some additional test statistics that were not included in the version 0.6.1 that was used for the work described in this paper (some 106 different test statistics, compared with 90 in version 0.6.1). Both versions generate and test some 2^{38} pseudo-random numbers from each generator that is considered. Without re-running all of our test cases using a more recent version of TestU01 (which would require many thousands of hours of cpu time), it is not possible to say for certain what the outcome would be for the additional test statistics in any particular case. However, even if we were to undertake these additional tests (or indeed any further tests that might be devised in the future), there is always a possibility that a specific generator that has passed all tests to date will fail on a new test that exposes a particular weakness of that generator. For the ACORN algorithm we believe, on the basis of the theoretical analysis that led to Theorems 1 and 2 above, that any such 'weaknesses' are likely to be resolved by sufficiently increasing both the order and the modulus of the generator. As an example, suppose that a new test were to be devised that required uniformity in 12 dimensions; any ACORN generator with order 11 or less would be expected to fail such a test, regardless of the modulus used. However, by increasing the order of the generator to 12 or more, and using a sufficiently large modulus we would then expect the ACORN generator to consistently pass such a test.

From a theoretical viewpoint a generator that passes all the tests would be expected to have far smaller likelihood of a 'failure' than of a 'suspect value' arising by chance. It might seem surprising that in certain cases (in particular in all cases with order less than or equal to 8 in Tables 2–5) the number of failures is greater than the number of suspect values. The reason for this behaviour is that all these lower order ACORN generators have certain weaknesses which are being detected by the consistent failure on specific tests — exactly the same behaviour is observed in the results presented by L'Ecuyer and Simard for generators that fail on one or more of their tests. Our objective in this paper has been to investigate the performance of ACORN generators for different values of order and modulus, and all the tables include results for some generators which show inadequacies because either the order or the modulus is too small. The performance of the ACORN generators consistently improves with increasing order and modulus, and with sufficiently large order and modulus the likelihood of 'failure' is indeed significantly less than the likelihood of a 'suspect values'. This is clear from an inspection of the final two columns on each of the Tables 2–5, which show the average number of failures and suspect values for each order and modulus of ACORN generator that was tested.

6. Conclusions

In this paper, we have considered two different definitions of the kth-order ACORN random number generator.

The first of these, Eqs. (1) to (3), is defined in exact integer arithmetic modulo a large integer power of 2 (with the resulting variates finally rescaled to the unit interval by dividing by the modulus). This leads to an efficient implementation, theoretically provable periodicity with arbitrarily long period length and sequences that are reproducible on any hardware using virtually any high-level computing language.

The second, Eqs. (5) and (6), is defined in real arithmetic modulo 1, and (provided the seed could be chosen to be irrational) gives rise to an infinite sequence of variates that can be proved to be w.d. mod 1 in $\mathcal{R}^k$. Although it cannot be used directly for evaluating the sequences (since this would require infinite-precision real arithmetic to guarantee identical sequences on different machines) it has proved useful as a starting point for theoretical analysis of the ACORN algorithm.

We have given proofs of two theorems concerning the relationship between these different definitions of ACORN random number generators:

(i) Any kth-order ACORN sequence defined and evaluated using the integer formulation modulo $M = 2^\mu$ is equivalent (to a precision of $\beta < \mu$ binary digits) to the first M terms of a sequence that is w.d. mod 1 in $\mathcal{R}^k$.
(ii) Any kth-order ACORN sequence defined using the real formulation, modulo 1, and having an irrational seed (which we have previously shown to be well distributed in k dimensions) can be approximated to any specified accuracy (β binary digits, for any β) over an arbitrary number $N = 2^\nu$ of terms. The sub-sequence can be evaluated using the integer formulation, choosing a modulus M that depends on N and β, together with an appropriate initialisation.

Empirical testing, carried out using the TestU01 test suite, has provided a practical demonstration of the theoretical convergence results, giving consistent results over the range of initialisations that were tested. The results presented in this paper provide support for the recommendation (made by the author in 2000 [5]) that an order of at least 10 be used together with an odd seed and modulus equal to 2^{30p}, for a small integer value of p. The latest results suggest that while a choice of $p = 2$ might be adequate for most typical applications, increasing p to 3 or more should guarantee a sequence that will consistently pass all of the tests from version 0.6.1 of the TestU01 BigCrush test battery, giving increased confidence in more demanding applications. As far as the restriction on order is concerned, the new results show (at least as regards the TestU01 tests) that a similar quality of performance might also be obtained for order 9, but not for orders of 8 or less.

On the basis of the results obtained to date, combined with the theoretical analysis, we might therefore reasonably expect all the TestU01 tests to be passed by almost all ACORN generators of order 9 or more having modulus 2^{30p} (where p is at least 3), an odd seed value and arbitrary initial values. Since each choice of seed gives rise to a completely different sequence, the ACORN algorithm (even when restricted to modulus 2^{90} and order 10) gives us a choice of at least 2^{89} ($\sim 6 \times 10^{26}$)

different sequences, each having period length in excess of 2^{93} ($\sim 10^{28}$), that we might reasonably expect to pass each of the TestU01 tests. Every different choice of order larger than 9 would give a similar number of sequences to choose from, while choosing the modulus to be a larger power of 2 would give access to even larger numbers of sequences with even longer period length.

If the range of tests were to be extended so as to demand uniformity for a greater number of digits precision or uniformity in higher dimensions, then we might need to increase the modulus and/or order of the ACORN generators in order to pass the new test. However, on the basis of the theoretical analysis of the algorithm we would still expect to see a similar pattern in the results of empirical testing — so all ACORN generators with a sufficiently large order, a sufficient modulus and an odd seed might still be expected to pass the new test.

The ACORN algorithm is simple both conceptually and to program, requiring a few tens of lines of code in virtually any high-level computing language (see [10] for an example implementation). Thus using the kth-order ACORN generator has been shown to be a computationally practical method of defining sequences that are w.d. mod 1 in $\mathcal{R}^k$, for any k, and of evaluating the terms in arbitrarily long sub-sequences of those sequences to any specified accuracy. It remains to be seen whether analogous theoretical convergence results can be derived for any of the existing alternative approaches to pseudo-random number generation. However, it is worth noting that this would probably require an approach different to that adopted here, since it is the unique recursive nature of the ACORN algorithm that leads to simplicity and efficiency in implementation (which works for arbitrary modulus and order), as well as to the theoretical convergence results that have been demonstrated in this paper.

Acknowledgements

I would like to thank my good friend and colleague, Chris L. Farmer (OCCAM, Mathematical Institute, University of Oxford, UK; formerly Schlumberger Abingdon Technology Centre, UK), whose enthusiasm and constructive criticism over a period of more than twenty years has encouraged me to persevere with my attempts to analyse, understand and explain the theory underlying the ACORN algorithm. Thanks are due to Numerical Algorithms Group Ltd for providing me with an executable version of TestU01, including their test implementation of the ACORN algorithm; also to my employer, RPS Group plc, for permitting me to make use of spare capacity on their 8-cpu linux cluster over a period of several months to carry out the empirical testing.

References

[1] J.H. Halton, A retrospective and prospective survey of the Monte-Carlo method, SIAM Rev. 12 (1970) 1–63.
[2] F. James, Monte Carlo theory and practice, Rep. Progr. Phys. 43 (1980) 1145–1189.
[3] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, PA, 1992.
[4] J.E. Gentle, Random Number Generation and Monte-Carlo Methods, 2nd ed., Springer, New York, 2003.
[5] R.S. Wikramaratna, Pseudo-random number generation for parallel processing — A splitting approach, SIAM News 33 (9) (2000).
[6] D. Knuth, The Art of Computer Programming, 2nd ed., in: Seminumerical Algorithms, vol. 2, Addison-Wesley Publishing Company, 1998.
[7] J.N. Franklin, Deterministic simulation of random processes, Math. Comp. 17 (1963) 28–59.
[8] R.S. Wikramaratna, ACORN - A new method for generating sequences of uniformly distributed pseudo-random numbers, J. Comput. Phys. 83 (1989) 16–31.
[9] R.S. Wikramaratna, Theoretical Background for the ACORN Random Number Generator, Report AEA-APS-0244, AEA Technology, Winfrith, Dorset, UK, 1992.
[10] R.S. Wikramaratna, The additive congruential random number generator — A special case of a multiple recursive generator, J. Comput. Appl. Math. 261 (2008) 371–387. doi:10.1016/j.cam.2007.05.018.
[11] P. L'Ecuyer, R. Simard, TestU01: A C library for empirical testing of random number generators, ACM Trans. on Math. Software 33 (4) (2007) Article 22, 40 pages.
[12] G. Marsaglia, The Marsaglia random number cdrom including the Diehard battery of tests of randomness, Florida State University, Florida, USA, 1995. http://stat.fsu.edu/pub/diehard.
[13] R.S. Wikramaratna, Empirical testing of the additive congruential random number generator, J. Comput. Appl. Math., 2008 (submitted for publication).
[14] M. Matsumoto, T. Nishimura, Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Trans. Modeling Comput. Simulation 8 (1998) 3–30.
[15] H. Weyl, Uber ein Problem aus dem Gebeite der diophantischen Approximationen, Nachr. Ges. Wiss. Gottingen, Math.-Phys. Kl. (1914) 234–244. Also in Gesammelte Abhandlungen, Band I, Springer-Verlag, Berlin, Heidelberg, New York, 1968, pp. 487–497.
[16] H. Weyl, Uber die Gleichverteilung von Zahlen mod. Eins, Math. Ann. 77 (1916) 313–352. also in Gesammelte Abhandlungen, Band I, Springer-Verlag, Berlin, Heidelberg, New York, 1968, pp. 563–599; and in Selecta Hermann Weyl, Birkhauser Verlag, Basel, Stuttgart, 1956, pp. 111–147.
[17] E. Hlawka, Zur formalin Theorie der Gleichverteilung in kompakten Gruppen, Rend. Circ. Mat. Palermo (2) (4) (1955) 33–47.
[18] G.M. Peterson, Almost convergence and uniformly distributed sequences, Q. J. Math. 7 (1956) 188–191.
[19] L. Kuipers, H. Niederreiter, Uniform Distribution of Sequences, John Wiley and Sons, New York, 1974.
[20] NAG, Numerical Algorithms Group (NAG) Fortran Library Mark 22, Numerical Algorithms Group Ltd., Oxford, UK, 2009. www.nag.co.uk.