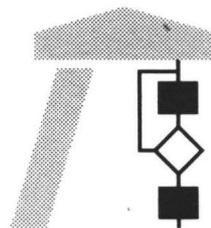


# Interner Bericht

## Trajectory Splitting by Restricted Replication

A. Keller

316/01



FACHBEREICH  
INFORMATIK



UNIVERSITÄT  
KAISERSLAUTERN

Postfach 3049 · D-67653 Kaiserslautern

# Trajectory Splitting by Restricted Replication

A. Keller

316/01

Universität Kaiserslautern  
AG Numerische Algorithmen  
Postfach 30 49  
67653 Kaiserslautern  
Germany

November 2001

Herausgeber: AG Numerische Algorithmen  
Leiter: Professor Dr. S. Heinrich

# Trajectory Splitting by Restricted Replication

Alexander Keller

*Universität Kaiserslautern, D-67653 Kaiserslautern, Germany*

---

## Abstract

New algorithms for efficient trajectory splitting are presented. By derandomizing these techniques that are derived from randomized quasi-Monte Carlo integration, trajectory splitting for the quasi-Monte Carlo method becomes available.

*Key words:* Monte Carlo integration, quasi-Monte Carlo integration.

---

## 1 Introduction and Summary

Whenever high-dimensional integrals have to be evaluated numerically, the Monte Carlo method provides a simple estimate of the integral and the variance of the estimate by using random numbers. The quasi-Monte Carlo method applies deterministic points that expose better uniformity properties than random numbers can obtain resulting in a much faster convergence on certain function classes. Opposite to the Monte Carlo method there is no cheap way of computing the approximation error. However, by randomizing quasi-Monte Carlo methods a practically negligible fraction of convergence is sacrificed, but the cheap Monte Carlo error estimate becomes available again. In addition theory is available for the large class of square integrable functions instead of only the very restricted class of functions of bounded variation in the sense of Hardy and Krause [Nie92].

We first briefly resume randomized replication techniques underlying randomized quasi-Monte Carlo integration. Restricting replication to only some of the dimensions of an integrand allows one to efficiently realize trajectory splitting for randomized quasi-Monte Carlo and (deterministic) quasi-Monte Carlo integration.

---

*URL:* <http://www.uni-kl.de/AG-Heinrich/Alex.html> (Alexander Keller).

## 2 Random Replicates

Given a quadrature rule on the  $s$ -dimensional unit cube  $[0, 1]^s$  that consists of  $n \in \mathbb{N}$  deterministic points  $P_n := \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\} \subset [0, 1]^s$  with equal weights  $\frac{1}{n}$ ,  $r$  independent, random mappings  $\mathbf{R}_i : [0, 1]^s \rightarrow [0, 1]^s$  are realized such that the expectation of the estimate

$$\int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x} \approx I_{r,n} f := \frac{1}{r} \sum_{i=1}^r \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{R}_i(\mathbf{u}_j)) \quad (1)$$

is the integral value. The inner deterministic average over  $j$  is supposed to reduce the variance by smoothing the integrand  $f$ . If the replicated points

$$\mathbf{x}_{i,j} := \mathbf{R}_i(\mathbf{u}_j)$$

are independent and uniformly distributed on  $[0, 1]^s$  for fixed  $j$ , the variance of the approximation (1) very efficiently can be estimated by

$$\sigma^2(I_{r,n} f) \approx \frac{1}{r(r-1)} \sum_{i=1}^r \left( \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{x}_{i,j}) - I_{r,n} f \right)^2. \quad (2)$$

Note that under the above condition of independence both estimates are unbiased for any deterministic point set  $P_n$ , however, large variance reduction is obtained if for fixed  $i$  the replicated points  $\mathbf{x}_{i,j}$  are of low discrepancy. Randomized quasi-Monte Carlo integration can be regarded as the tensor product of Monte Carlo and quasi-Monte Carlo integration and in fact itself is Monte Carlo integration. "Random quadrature formulae" already are mentioned in [Shr66]. Here we summarize the most important randomization algorithms  $\mathbf{R}_i$  for quasi-Monte Carlo integration. Further details are found in the surveys of Owen [Owe98] and Lémieux and L'Ecuyer [LL02, Ch. 5].

### 2.1 Cranley Patterson Rotations

Cranley and Patterson [CP76] were the first to randomize so-called lattice points. However, their method can be applied for randomizing every point set in a very efficient way. A random replicate of the point  $\mathbf{u}_j = (u_j^{(1)}, \dots, u_j^{(s)})$  is obtained by just adding a random vector  $\xi \in [0, 1]^s$  yielding the replicated components

$$x_{i,j}^{(k)} = u_j^{(k)} + \xi_i^{(k)} \bmod 1.$$

## 2.2 Random Scrambling

Many low discrepancy point sets are constructed using radical inversion in an integer base  $b$ . A randomly scrambled component [Owe95] is represented by

$$x_{i,j}^{(k)} = \sum_{l=0}^{M-1} \pi_{i,a_0(j),\dots,a_{l-1}(j)}^{(k)} \left( a_l^{(k)}(j) \right) b^{-l-1}, \quad (3)$$

where  $u_j^{(k)} = \sum_{l=0}^{M-1} a_l^{(k)}(j) b^{-l-1}$  and we partially omitted the superscript  $(k)$  of the coefficients  $a_l^{(k)}(j)$  for clarity of presentation. The random permutations  $\pi_{i,a_0(j),\dots,a_{l-1}(j)}^{(k)}$  on  $\{0, \dots, b-1\}$  are independently, uniformly realized, where the permutation of the  $l$ -th digit  $a_l^{(k)}(j)$  is selected by the leading digits  $a_0^{(k)}(j), \dots, a_{l-1}^{(k)}(j)$ . Spoken visually this procedure corresponds to starting out with the unit interval, which is divided into  $b$  equally sized intervals that become randomly permuted and recursively repeating this procedure for the subintervals ad infinitum. The algorithm, however, becomes finite due to the finite precision of computation, i.e. using only  $M \in \mathbb{N}$  digits in base  $b$ .

Each component of a  $(t, m, s)$ -net or a  $(t, s)$ -sequence (for details see e.g. [Nie92]) is constructed by a radical inversion in base  $b$ . Morohosi and Fushimi [MF00] found that the randomization of these points by Cranley Patterson rotations yields a noticeable loss in convergence, since the quality parameter  $t$  can be affected [Tuf96]. In contrast the above scrambling procedure does not change the value of  $t$ . Efficient scrambling algorithms along with an implementation are found in [FK01].

Halton and Hammersley points (for details see e.g. [Nie92]) including improved constructions as e.g. by Faure [Fau92] use  $s$  relatively prime bases  $b_k$  for the components instead of only one base. In a straightforward way random scrambling is realized by independently applying the one dimensional scheme for component  $k$  in base  $b_k$ . This implicitly generalizes the concept of  $b$ -adic elementary intervals [Nie92] to  $(b_1, \dots, b_s)$ -adic elementary intervals and already has been investigated by Matoušek [Mat98]. We briefly sketch some useful restrictions of random scrambling (see also [Mat98]).

### 2.2.1 Random Digit-Scrambling

The restriction of random scrambling

$$x_{i,j}^{(k)} = \sum_{l=0}^{M-1} \pi_{i,l}^{(k)} \left( a_l^{(k)}(j) \right) b^{-l-1}, \quad (4)$$

where instead of the full permutation tree for each level  $l$  only one random permutation  $\pi_{i,l}$  is used, still produces independent replicates of the points in  $P_n$  for use with (1) and (2). However, the implementation is much simpler, since only one path of permutations is used instead of the full permutation tree.

### 2.2.2 Random Digital $b$ -ary Shift

It is even possible to restrict the random permutations used in (4) to addition modulo  $b$ . Then the random replicate becomes

$$x_{i,j}^{(k)} = \sum_{l=0}^{M-1} (a_l^{(k)}(j) + d_{i,l}^{(k)} \bmod b) b^{-l-1}. \quad (5)$$

The integer shifts  $d_{i,l}^{(k)}$  can be found by representing a random number  $\xi_i^{(k)} =: \sum_{l=0}^{M-1} d_{i,l}^{(k)} b^{-l-1}$  in base  $b$ . For an implementation the shifts directly can be extracted from the finite state machine of the pseudo-random number generator thus avoiding floating point conversions. Halton [HW64] already remarked that any point can be used as starting point for the incremental Halton sequence generation. This is especially true for a random starting point  $\zeta \in [0, 1)^s$  as analyzed by Wang and Hickernell [WH00] (for applications in radiation transport see also [Kel00]) and corresponds to extracting the shifts by representing the components of  $\zeta$  in base  $b$  and subtracting the contribution of  $\mathbf{u}_0$ . The explicit extraction becomes necessary whenever incremental algorithms for point generation are not available as for e.g. the improved Halton and Hammersley points by Faure [Fau92].

An extremely efficient vectorized algorithm [Fri98] is available in base  $b = 2$ , where the random digital  $b$ -ary shift reduces to bit-wise addition without carry. This just means XORing the components with an integer random number before floating point conversion. This method of scrambling then is as efficient as Cranley Patterson rotations and valid for the estimates (1) and (2) with any point set  $P_n$ .

### 2.2.3 Random Linear Scrambling

Another subset of random scrambling has been introduced by Hong and Hickernell [HH01], which however does not fit into the above hierarchy of restrictions. A random non-singular  $M \times M$  lower triangular matrix  $T_i^{(k)} = (t_{i,l,m}^{(k)})$  with entries from  $\{0, \dots, b-1\}$  is used in combination with a random digital

$b$ -ary shift  $\mathbf{d}_{i,l}$  to scramble the points  $\mathbf{u}_j \in P_n$  by

$$x_{i,j}^{(k)} = \sum_{l=0}^{M-1} \left( \left( \sum_{m=0}^{M-1} t_{i,l,m}^{(k)} a_m^{(k)}(j) \right) + d_{i,l}^{(k)} \bmod b \right) b^{-l-1}.$$

The method efficiently can be implemented by once multiplying the matrices  $T_i^{(k)}$  with the generator matrices of a particular  $(t, m, s)$ -net or  $(t, s)$ -sequence. The special case of  $T_i^{(k)}$  being a diagonal matrix is called *random linear digit-scrambling*.

### 3 High-Dimensional Samples by Replication

So far randomized replications of low discrepancy points have been discussed. However, often low discrepancy points do not perform very well in high dimensions. Therefore Spanier [Spa95] used a hybrid scheme, where high dimensional samples were padded by low dimensional low discrepancy points completed by random samples. We generalize this idea.

#### 3.1 Padded Replications Sampling

For the estimates (1) and (2) to work only the independence of the replicates of one point  $\mathbf{u}_j$  is required. Consequently padding independent random replicates of low dimensional low discrepancy point sets yields unbiased estimators for high-dimensional integrals. In order to illustrate the principle two low discrepancy point sets  $P_n \subset [0, 1]^{s_1}$  and  $Q_n \subset [0, 1]^{s_2}$  of  $n$  points each are considered. Using the replicates  $\mathbf{x}_{i,j}$  of the points in  $P_n$  and  $\mathbf{y}_{i,j}$  of the points in  $Q_n$ , respectively, the  $s = s_1 + s_2$  dimensional integral

$$\int_{[0,1]^s} f(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \approx \frac{1}{r} \sum_{i=1}^r \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{x}_{i,j}, \mathbf{y}_{i,j})$$

is estimated in an unbiased way. Although the assembled samples are not of low discrepancy in general, their low dimensional low discrepancy properties easily can be designed to perfectly match the structure of integrals from e.g. transport problems. The efficiency of this approach has been investigated profoundly in [KK01]. Padded replications sampling can be combined with Latin supercube sampling [Owe98] in order to resolve correlations between the padded replicates. This is especially useful for replicates from Cranley Patterson rotations [KK01].

### 3.2 Trajectory Splitting by Restricted Replication

Some dimensions of an integrand may expose much more smoothness than other dimensions. In consequence an efficient algorithm should sample these dimensions much less than others. This is known as the technique of trajectory splitting from the domain of particle transport simulation. Instead of tracing only one trajectory a particle is split into  $n$  particles in an interesting region [Sob94]. This is achieved by sampling the smooth dimensions by only  $r$  random points  $\mathbf{x}_i \in [0, 1)^{s_1}$  and padding  $r$  random replicates of a deterministic set  $Q_n \subset [0, 1)^{s_2}$  of  $n$  points yielding

$$\int_{[0,1)^s} f(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \approx \frac{1}{r} \sum_{i=1}^r \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{x}_i, \mathbf{y}_{i,j}). \quad (6)$$

Thus replication is restricted to the complicated dimensions of the integrand only. Compared to the uncorrelated trajectory splitting [Sob94] a variance reduction is obtained by replicating a low discrepancy point set for trajectory splitting. This concept yields very efficient algorithms for volume rendering in software [PKK00] as well as in hardware and for motion blur simulation [KH01]. Many Monte Carlo algorithms from the field of computer graphics as e.g. the illumination by area light sources benefit from the variance reduction by replacing jittered samples [CPC84] by random replicates of a low discrepancy point set.

Abandoning the availability of an unbiased variance estimate (2), instead of the random independent points  $\mathbf{x}_i$  one random replicate of a low discrepancy point set  $P_n \subset [0, 1)^{s_1}$  can be used to increase the speed of convergence. Using the deterministic point set  $P_n$  directly yields a consistent hybrid algorithm that follows the original padding idea of Spanier [Spa95].

### 3.3 Trajectory Splitting for Quasi-Monte Carlo Integration

Spanier and Maize [SM94] began to transfer Monte Carlo techniques to the quasi-Monte Carlo method. By derandomizing the randomized quasi-Monte Carlo method (6) we contribute trajectory splitting for quasi-Monte Carlo integration. For the example of Cranley Patterson rotations a deterministic low discrepancy point set  $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^r \subset [0, 1)^s$  is selected as the global quadrature

$$\int_{[0,1)^s} f(\mathbf{x}, \mathbf{y}) d\mathbf{y} d\mathbf{x} \approx \frac{1}{r} \sum_{i=1}^r \frac{1}{n} \sum_{j=0}^{n-1} f(\mathbf{x}_i, \mathbf{y}_i + \mathbf{z}_j \bmod [0, 1)^{s_2}), \quad (7)$$

while the integrand is locally smoothed in the difficult dimensions using deterministic low discrepancy points  $(\mathbf{z}_j)_{j=0}^{n-1} \subset [0, 1]^{s_2}$ . Care has to be taken that the components  $(\mathbf{y}_i)_{i=1}^r$  are not correlated with  $(\mathbf{z}_j)_{j=0}^{n-1}$ . This can be achieved by choosing e.g. Hammersley points for both point sets so that the bases of the components in the dimensions  $s_2$  are relatively prime. This is used by the professional renderer mental ray [Kel00] for radiance transport simulation in global illumination computations and obtains superior performance as compared to randomized algorithms. Instead of Cranley Patterson rotations, computing deterministic digital  $b$ -ary shifts (5) from the components of the points  $\mathbf{y}_i$  preserves the structure of the  $(\mathbf{z}_j)_{j=0}^{n-1}$  if they are constructed by radical inversion. For the remaining scrambling techniques the number of random parameters is too large to be efficiently determined by the  $r$  points  $\mathbf{y}_i$ .

#### 4 Minimal Randomization

Randomized quasi-Monte Carlo integration sacrifices some convergence of quasi-Monte Carlo in order to make a cheap error estimate available. The interesting question arises how little randomness is sufficient to make the estimates (1) and (2) work but introducing only the least necessary amount of noise at the same time. In this context randomization explicitly is not targeted to improve upon the discrepancy of some low discrepancy constructions; we therefore assume to have deterministic point sets at hand that already have optimal discrepancy.

Although the summarized random replication techniques were designed to exploit the structure of the underlying low discrepancy points, they nevertheless can be applied to any point set and still yield unbiased estimators, which however can be of worse performance [MF00]. This indicates that there exist more specific randomization schemes that only work for the class of points they are designed for but require less randomness for replication. In this context Owen's postulate of uniformity [Owe95] is too general: It is sufficient that every point in  $[0, 1]^s$  can be sampled, however, it is not necessary that the replicates of one point  $\mathbf{u}_j$  completely sample the  $s$ -dimensional unit cube.

We illustrate this by using an example in  $s = 1$ , where the best discrepancy is obtained by equidistantly placed points. By random scrambling (3) these points degenerate to stratified sampling, i.e. the discrepancy becomes worse since the equidistant spacing is destroyed. Using much less randomness for the random digital  $b$ -ary shifts (5) the equidistant spacing is preserved. In consequence the digits beyond the first  $M$  digits of a digital net should not be chosen independently random, but one identical random shift  $\frac{\xi}{b^M}$  should be used for all points in order not to destroy the discrepancy of low dimensional projections.

On the other hand random scrambling applied to a digital  $(0, M, 1)$ -net in base  $b$  only affects the digits beyond the  $M$  digits defined by the net, i.e. the random scrambling has no effect on the first  $M$  digits and such could be omitted.

Similarly we can observe that instead of a Cranley Patterson rotation a random shift inside the basis cell of the lattice is sufficient for randomizing rank-1 lattices. The reduced shift turns the quadrature into stratified correlated sampling, where the same samples are taken in the strata given by the lattice cells. In one dimension this observation coincides with the random digital  $b$ -ary shift of a net, i.e. a random shift of  $\frac{\xi}{n}$ . While this subtle difference in randomization has no noticeable effect for random shifts, it makes a huge difference for structured shifts as used in e.g. (7), because opposite to the Cranley Patterson rotation the reduced shift preserves the structure in the strata.

## Acknowledgements

The author is especially thankful to Thomas Kollig for his comments, and to Peter Schröder and Rolf Herken for supporting this research.

## References

- [CP76] R. Cranley and T. Patterson, *Randomization of number theoretic methods for multiple integration*, SIAM Journal on Numerical Analysis **13** (1976), 904–914.
- [CPC84] R. Cook, T. Porter, and L. Carpenter, *Distributed Ray Tracing*, Computer Graphics (SIGGRAPH 84 Conference Proceedings), 1984, pp. 137–145.
- [Fau92] H. Faure, *Good Permutations for Extreme Discrepancy*, J. Number Theory (1992), no. 42, 47–56.
- [FK01] I. Friedel and A. Keller, *Fast generation of randomized low-discrepancy point sets*, Monte Carlo and Quasi-Monte Carlo Methods 2000 (H. Niederreiter, K. Fang, and F. Hickernell, eds.), Springer, 2001, p. to appear.
- [Fri98] I. Friedel, *Abtastmethoden für die Monte-Carlo und Quasi-Monte-Carlo-Integration*, Universität Kaiserslautern, 1998.
- [HH01] H. Hong and F. Hickernell, *Implementing scrambled digital sequences*, Workshop on the Complexity of Multivariate Problems, October 4–8, 1999, Hong Kong, 2001, submitted for publication.

- [HW64] J. Halton and G. Weller, *Algorithm 247: Radical-inverse quasi-random point sequence*, Comm. ACM **7** (1964), no. 12, 701–702.
- [Kel00] A. Keller, *Strictly deterministic sampling methods in computer graphics*, Tech. report, mental images, Berlin, Germany, 2000.
- [KH01] A. Keller and W. Heidrich, *Interleaved Sampling*, Rendering Techniques 2001 (Proc. 12th Eurographics Workshop on Rendering) (K. Myszkowski and S. Gortler, eds.), Springer, 2001, p. xx.
- [KK01] T. Kollig and A. Keller, *Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration*, Monte Carlo and Quasi-Monte Carlo Methods 2000 (H. Niederreiter, K. Fang, and F. Hickernell, eds.), Springer, 2001, p. to appear.
- [LL02] P. L'Ecuyer and C. Lemieux, *Recent advances in randomized quasi-Monte Carlo methods*, Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications (M. Dror, P. L'Ecuyer, and F. Szidarovszki, eds.), Kluwer Academic Publishers, 2002, p. xx.
- [Mat98] J. Matoušek, *On the  $L_2$ -discrepancy for anchored boxes*, J. of Complexity **14** (1998), no. 4, 527–556.
- [MF00] H. Morohosi and M. Fushimi, *A practical approach to the error estimation of quasi-Monte Carlo integration*, Monte Carlo and Quasi-Monte Carlo Methods 1998 (H. Niederreiter and J. Spanier, eds.), Springer, 2000, pp. 377–390.
- [Nie92] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Pennsylvania, 1992.
- [Owe95] A. Owen, *Randomly Permuted  $(t, m, s)$ -Nets and  $(t, s)$ -Sequences*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Lecture Notes in Statistics, vol. 106, Springer, 1995, pp. 299–315.
- [Owe98] ———, *Monte Carlo Extension of Quasi-Monte Carlo*, Winter Simulation Conference, IEEE Press, 1998.
- [PKK00] M. Pauly, T. Kollig, and A. Keller, *Metropolis Light Transport for Participating Media*, Rendering Techniques 2000 (Proc. 11th Eurographics Workshop on Rendering) (B. Péroche and H. Rushmeier, eds.), Springer, 2000, pp. 11–22.
- [Shr66] Y. Shreider, *The Monte Carlo Method*, Pergamon Press, 1966.
- [SM94] J. Spanier and E. Maize, *Quasi-Random Methods for Estimating Integrals using relatively small Samples*, SIAM Review **36** (1994), no. 1, 18–44.
- [Sob94] I. Sobol', *A Primer for the Monte Carlo Method*, CRC Press, 1994.

- [Spa95] J. Spanier, *Quasi-Monte Carlo Methods for Particle Transport Problems*, Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (H. Niederreiter and P. Shiue, eds.), Lecture Notes in Statistics, vol. 106, Springer, 1995, pp. 121–148.
- [Tuf96] B. Tuffin, *On the Use of Low Discrepancy Sequences in Monte Carlo Methods*, Monte Carlo Methods and Applications **2** (1996), no. 4, 295–320.
- [WH00] X. Wang and F. Hickernell, *Randomized Halton sequences*, Math. Comput. Modelling **32** (2000), 887–899.