Transforming Low-Discrepancy Sequences from a Cube to a Simplex

Tim Pillards and Ronald Cools

Report TW 371, August 2003

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Transforming Low-Discrepancy Sequences from a Cube to a Simplex

Tim Pillards and Ronald Cools

Report TW 371, August 2003

Department of Computer Science, K.U.Leuven

Abstract

Sequences of points with a low discrepancy are the basic building blocks for quasi-Monte Carlo methods. Traditionally these points are generated in a unit cube.

To develop point sets on a simplex we will transform the low-discrepancy points for the unit cube to a simplex. An advantage of this approach is that most of the known results on low discrepancy sequences can be re-used. After introducing several transformations, their efficiency as well as their quality will be evaluated. We prove a Koksma-Hlawka inequality which says that under certain conditions the order of convergence using the new point set is the same as that of the original set.

Keywords : Multi-dimensional integration, quasi-Monte Carlo method

Transforming Low-Discrepancy Sequences from a Cube to a Simplex

Tim Pillards and Ronald Cools

August 18, 2003

Abstract

Sequences of points with a low discrepancy are the basic building blocks for quasi-Monte Carlo methods. Traditionally these points are generated in a unit cube.

To develop point sets on a simplex we will transform the low-discrepancy points for the unit cube to a simplex. An advantage of this approach is that most of the known results on low discrepancy sequences can be re-used. After introducing several transformations, their efficiency as well as their quality will be evaluated. We prove a Koksma-Hlawka inequality which says that under certain conditions the order of convergence using the new point set is the same as that of the original set.

1 Introduction

Quasi-Monte Carlo (QMC) methods approximate the integral of a function f over the unit cube

$$I[f] := \int_{I_s} f(\mathbf{x})\mathsf{d}\mathbf{x}$$

$$I_s := \{\mathbf{x} = (x_1, \ldots, x_s) : 0 \leq x_i \leq 1, i = 1, \ldots, s\} \tag{1}$$

by an equal weight cubature rule of the form

$$I[f] \approx Q[f] := \frac{1}{N} \sum_{\mathbf{x_i} \in P} f(\mathbf{x_i})$$

where $P \subset I_s$ is a low discrepancy point set with $\#P = N$. Several possibilities for defining the discrepancy of a point set exist. In this article we choose the star discrepancy.

Definition 1.1 *Let Υ be the set of all rectangles containing the origin $\mathbf{o} = (0, 0, \ldots, 0)$*

$$\Upsilon := \{\prod_{i=1}^{s}[0, x_i] : x_i \in]0, 1]\}$$

then the star discrepancy is defined as

$$D^*(P) := \sup_{U \in \Upsilon} \left| \frac{A(U)}{N} - vol(U) \right|$$

with $A(U)$ the number of points of P inside U.

Using the Variation $V(f)$ of a function $f : I_s \to \mathbb{R}$ as defined in, e.g., [6], it is known that the error of the approximation is bounded by the following theorem.

Theorem 1.1 (Koksma-Hlawka) *For $f : I_s \to \mathbb{R}$ a function of bounded variation*

$$|I[f] - Q[f]| = \left| \int_{I_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} - \sum_{\mathbf{x_i} \in P} f(\mathbf{x_i}) \right| \leq D^*(P)\ V(f).$$

For a proof of this almost classical result, we refer to [6]. Consider an integral of the form

$$I[f] = \int_{T_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} \tag{2}$$

where T_s is a simplex defined as

$$T_s := \{(x_1, \ldots, x_s) \in \mathbb{R}^s : 0 \leq x_1 \leq x_2 \leq \cdots \leq x_s \leq 1\}. \tag{3}$$

From (3) we can deduce that T_3 has the points $(0,0,0)$, $(0,0,1)$, $(0,1,1)$ and $(1,1,1)$ as vertices (see Figure 1).

The integral (2) can also be written as:

$$I[f] = \int_0^1 \int_{x_1}^1 \cdots \int_{x_{s-1}}^1 f(\mathbf{x}) \mathsf{d}x_s \ldots \mathsf{d}x_1.$$

We consider a QMC approximation for this integral

$$\begin{aligned} I[f] \approx Q[f] \quad &:= \quad \frac{vol(T_s)}{N} \sum_{\mathbf{x_i} \in P} f(\mathbf{x_i}) \\ &= \quad \frac{1}{s!\ N} \sum_{\mathbf{x_i} \in P} f(\mathbf{x_i}) \end{aligned}$$

where P is a set of N points in T_s.

For all possible s-dimensional permutations $(x_{i_1}, \ldots, x_{i_s})$ of $(x_1, \ldots, x_s)$ we define

$$S_i := \{(x_1, \ldots, x_s) \in I_s | x_{i_1} \leq \cdots \leq x_{i_s}\}.$$

Each of these simplices is equal to the original simplex T_s after a rotation. Observe that these simplices do not overlap (the s-dimensional volume of their intersections is 0) and their union is equal to the unit cube (see Figure 1).

2 Transformations from cube to simplex

One road that can be traveled to approximate integral (2) with QMC is by transforming a known point set on the unit cube to the simplex.

In this section, several transformations are introduced and evaluated. We present them first in 2 dimensions to make them easier to visualize.

2.1 Transformation Drop

A straightforward way to generate a point set on any bounded domain starting with a point set on the unit cube, is by rotating, translating and scaling the cube such that it encloses the bounded domain. Now, the new point set can be generated by dropping all the points from the given set that do not fall inside the domain. The unit cube I_s already encompasses the simplex T_s. Thus this 'transformation' can be written as follows:

If $\mathbf{x} \in T_s$	then keep this point.
	else drop this point.

This transformation is fast but in higher dimensions a lot of points get lost; only 1 out of $s!$ points is kept.

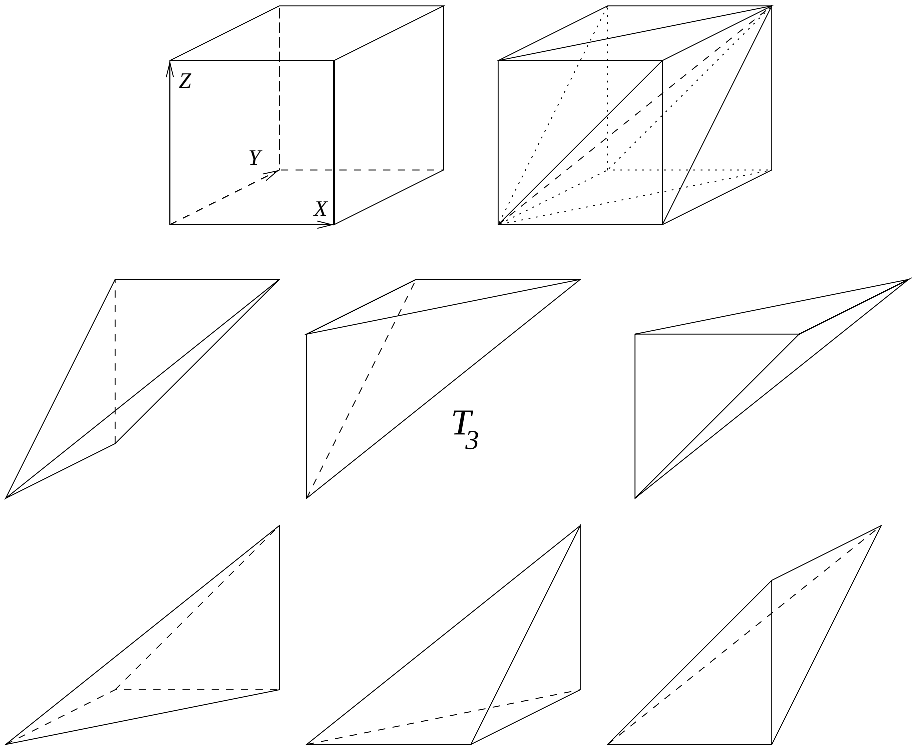


Figure 1: The unit cube I_3 divided into 6 simplices S_i

2.2 Transformation Sort

A lot of points are lost by transformation Drop. Transformation Sort will recover these. Observe that it holds for all points $\mathbf{x} = (x_1, \ldots, x_s) \in I_s$ that

$$\mathbf{x} \in T_s \Leftrightarrow x_1 \leq \cdots \leq x_s.$$

When we sort the coordinates of a point we obtain a point on the simplex T_s.

$$\boxed{T(x_1, \ldots, x_s) := \text{Sort}(x_1, \ldots, x_s).}$$

This is illustrated in Figure 2(a) for 2 dimensions. Sort is a fast continuous transformation, even for high dimensions. It is described for Monte Carlo methods in, e.g., [5].

2.3 Transformation Mirror

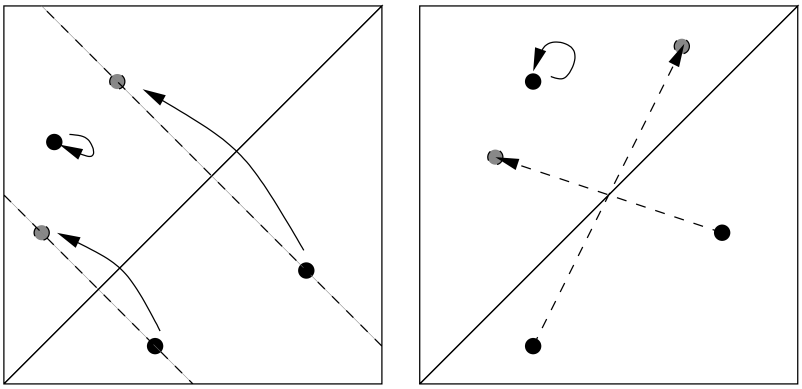
Transformation Mirror also keeps the points falling inside the simplex T_s unchanged. In 2 dimensions, the other points are reflected over the midpoint $(1/2, 1/2)$ such that they too fall inside the simplex (see Figure 2(b)). This results into the following:

If $x_1 \leq x_2$,	then	$T(x_1, x_2) := (x_1, x_2)$
	else	$T(x_1, x_2) := (1 - x_1, 1 - x_2)$

In more dimensions several consecutive reflections are needed. One possibility for 3 dimensions is:

if $x_3 \leq x_1$	then	$T(x_1, x_2, x_3) := (1 - x_1, 1 - x_2, 1 - x_3)$
if $x_3 \leq x_2$	then	$T(x_1, x_2, x_3) := (x_1, 1 - x_2 + x_1, 1 - x_3 + x_1)$
if $x_2 \leq x_1$	then	$T(x_1, x_2, x_3) := (x_3 - x_1, x_3 - x_2, x_3)$

This transformation is fast, even for high dimensions. But it is discontiuous.



(a) Transformation Sort (b) Transformation Mirror

Figure 2: Transformations Sort and Mirror in 2 dimensions

2.4 Transformation Origami

The transformation described below recursively uses the transformation Sort, beginning at small subsquares and gradually increasing the size of the squares at which Sort is used until it is used on the unit cube. A disadvantage is that Origami is no longer continuous (where Sort is).

The transformation consists of the following steps:

- Choose a base b. Choose a constant m and let $M = b^m$.
- Divide the square into M^2 squares.
- In each of these squares, use the transformation Sort, resulting in M^2 triangles.
- Now for $N = M/b, M/(b^2), \ldots, b, 1$ divide the square into N^2 squares and use transformation Sort.

For two dimensions in base 2, this transformation is due to Ph. Bekaert [1]. In Figure 3 this case is graphically illustrated. The generalisation of this algorithm to higher dimensions consists of the following steps:

- Choose a base b. Choose a constant m and let $M = b^m$.
- Divide I_s into M^s hypercubes.
- In each of these hypercubes, use the transformation Sort, resulting in M^s simplices.
- Now for $N = M/b, M/(b^2), \ldots, b, 1$ divide the square into N^s hypercubes and use transformation Sort.

This transformation becomes hard in high dimensions. If the base b is chosen to be a power of 2, then all calculations can be done in binary arithmetic, which is a practical advantage.

2.5 Transformation Root

Transformation Root is based on the inverse cumulative distribution function (cdf) and is given in [4]. Looking at the first dimension, the points are transformed such that the cdf of the new points in this dimension is the same as the cdf of uniformly distributed points on the simplex. In the second dimension, the points are then forced to fall inside the triangle.

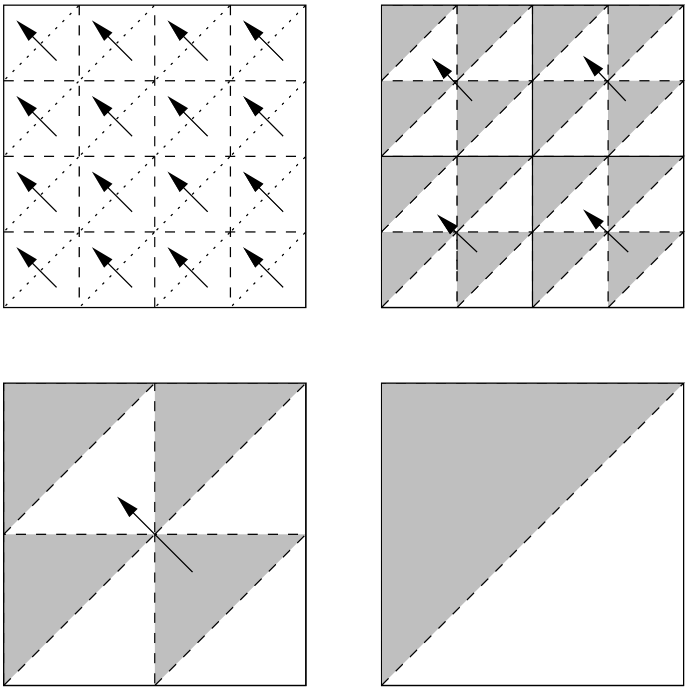


Figure 3: Transformation Origami in 2D with $b = 2$, $m = 2$ and $M = 4$

Suppose the points are uniformly distributed over the triangle T_2, then the cdf over x_2 is x_2^2. Thus to make sure we obtain the right cdf, the points must be transformed by $T(x_1, x_2) := (y_1, y_2)$ with $y_2 = \sqrt{x_2}$, and x_1 must be transformed such that the transformed points fall into the triangle. This can be done by setting $y_1 = x_1 y_2 = x_1\sqrt{x_2}$. Consequently, the transformation can be written in 2 dimensions as

$$\boxed{T(x_1, x_2) := (x_1\sqrt{x_2}, \sqrt{x_2}).}$$

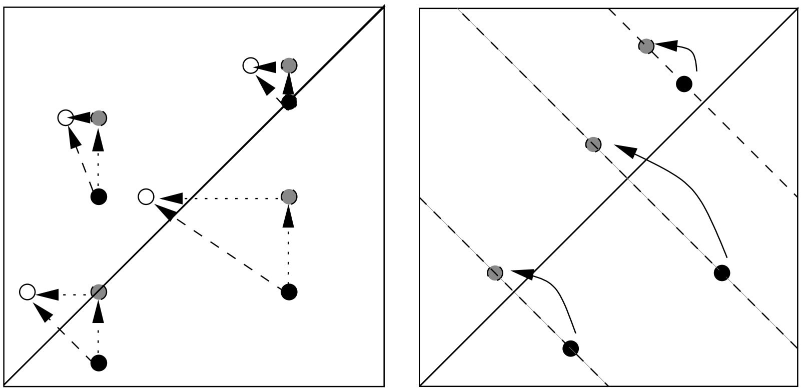
This transformation is illustrated in Figure (4(a)). Root is a continuous transformation but note that it treats both sharp corners of the simplex in an entirely different way. In more dimensions this transformation needs more and more slow calculations ('higher order roots/radicals') and thus becomes slow. The transformation for s dimensions becomes

$$\boxed{\begin{array}{l} T(x_1, \ldots, x_s) := (y_1, \ldots, y_s) \\ \text{with} \\ \left\{ \begin{array}{l} y_1 := x_1 x_2^{1/2} \ldots x_s^{1/s} \\ y_2 := x_2^{1/2} \ldots x_s^{1/s} \\ \ldots \\ y_s := x_s^{1/s} \end{array} \right. \end{array}}$$

2.6 Transformation Shift

In contrast with transformation Root we now describe another continuous transformation in 2 dimensions that treats both sharp corners in the same way.

Draw a straight line with slope -1 through the point that must be transformed. The point is moved halfway this line towards the x_1- or x_2-axis, whichever is closest (see Figure 4(b)). This results into:



(a) Transformation Root (b) Transformation Shift

Figure 4: Transformation Root and Shift in 2D

if $x_1 \geq 1 - x_2$	then	$T(x_1, x_2) := (x_1 - (1 - x_2)/2, x_2 + (1 - x_2)/2)$
	else	$T(x_1, x_2) := (x_1 - x_1/2, x_2 + x_1/2)$.

This transformation is as fast as Mirror and Sort. However, we did not find a generalisation to higher dimensions.

3 Koksma-Hlawka for transformed point sets

In this section we will first prove a Koksma-Hlawka-type inequality on T_s and we will end this section by calculating the variation of some functions in 2 variables.

3.1 A Koksma-Hlawka inequality on T_s

When solving the cubature problem with transformed points, we can produce an equivalent problem on the unit cube. Let P be the original point set on the unit cube and $T(P)$ the transformed point set. Then for a function f on T_s and $g = f\ o\ T$:

$$\frac{1}{N} \sum_{\mathbf{x_i} \in T(P)} f(\mathbf{x_i}) = \frac{1}{N} \sum_{\mathbf{x_i} \in P} g(\mathbf{x_i}).$$

Thus the numerical approximation problem on the simplex is equivalent to the numerical approximation problem on the cube for the original point set P and function $g = f\ o\ T$. This leads to a Koksma-Hlawka-type expression for the approximation on the simplex *if* the transformation satisfies

$$\int_{T_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} = \frac{1}{s!} \int_{I_s} g(\mathbf{x}) \mathsf{d}\mathbf{x}. \tag{4}$$

Indeed, under that assumption it follows from Theorem 1.1 that

$$\begin{aligned} \left| \int_{T_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} - \frac{vol(T_s)}{N} \sum_{\mathbf{x_i} \in T(P)} f(\mathbf{x_i}) \right| &= \left| \int_{T_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} - \frac{1}{s!\,N} \sum_{\mathbf{x_i} \in T(P)} f(\mathbf{x_i}) \right| & (5) \\ &= \frac{1}{s!} \left| \int_{I_s} g(\mathbf{x}) \mathsf{d}\mathbf{x} - \frac{1}{N} \sum_{\mathbf{x_i} \in P} g(\mathbf{x_i}) \right| & (6) \\ &\leq vol_s(T_s)\ D^*(P)\ V(g). & (7) \end{aligned}$$

From (5) to (6) we use the equality (4). We will show in the next section that it holds for most of the transformation we introduced in Section 2. In (7), $D^*(P)$ represents the star discrepancy of the original point set and $V(g)$ the variation of $f\ o\ T$ on the unit cube. Inequality (7) can be generalized to other transformations (not necessarily to T_s) as follows:

Lemma 3.1 *Let $T : I_s \to \Omega$ be a transformation from I_s to Ω and $f : \Omega \to \mathbb{R}$ a function on Ω such that the Variation in the sense of Hardy and Krause of $g = f\ o\ T : I_s \to \mathbb{R}$ is bounded and*

$$\int_\Omega f(\mathbf{x})\mathsf{dx} = vol_s(\Omega) \int_{I_s} g(\mathbf{x})\mathsf{dx}. \tag{8}$$

Then it holds that

$$\begin{aligned} |I[f] - Q[f]| &= \left| \int_\Omega f(\mathbf{x})\mathsf{dx} - \frac{vol_s(\Omega)}{N} \sum_{\mathbf{x_i} \in T(P)} f(\mathbf{x_i}) \right| \\ &\leq vol_s(\Omega)\ D^*(P)\ V(g). \end{aligned}$$

Consequently, for a low discrepancy point set on I_s with

$$D^*(P) = O\left(\frac{(\log N)^s}{N}\right),$$

the error of the approximation using the transformed point set $T(P)$ satisfies

$$|I[f] - Q[f]| \leq O\left(\frac{(\log N)^s}{N}\right).$$

Remark that it is sufficient that the Jacobian of T is equal to the constant $vol_s(\Omega)$ to fulfill (8). Lemma 3.1 is important as it proves that the order of convergence of a transformed point set is the same as the original point set. The constant will however depend on the transformation. Finally one should know that the result of Lemma 3.1 is not applicable to all transformation we presented in Section 2. It only makes sense if the transformation is continuous because otherwise the transformed function g is not of bounded variation.

3.2 Proving a property of the transformation

In this section we prove (4) for all transformations proposed in this article, except Drop.

3.2.1 Background

The following Lemma can be found for example in [3].

Lemma 3.2 *Let Q be an interval in $\mathbb{R}^s$, and $T : \mathbb{R}^s \to \mathbb{R}^s$ a mapping which is C^1-invertible on a neighborhood of Q. If $f : \mathbb{R}^s \to \mathbb{R}$ is an integrable function such that $f\ o\ T$ is also integrable, then*

$$\int_{T(Q)} f(\mathbf{x})\mathsf{dx} = \int_Q (f\ o\ T)(\mathbf{x})\ |J(T)|\mathsf{dx},$$

where $J(T) = \frac{\mathsf{d}T(\mathbf{x})}{\mathsf{dx}}$ is the Jacobian of T.

From this follows immediately the following.

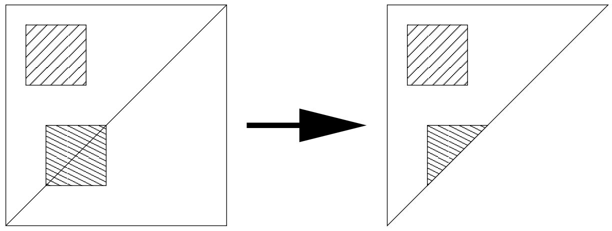


Figure 5: Squares transformed by Sort

Lemma 3.3 *Let $T : I_s \to T_s$ be a C^1-invertible transformation on a neighborhood of I_s with $|J(T)| = \frac{1}{s!}$. If $f : T_s \to \mathbb{R}$ is an integrable function such that $f\ o\ T : I_s \to \mathbb{R}$ is also integrable, then*

$$\int_{T_s} f(\mathbf{x})\mathsf{d}\mathbf{x} = \frac{1}{s!}\int_{I_s}(f\ o\ T)(\mathbf{x})\mathsf{d}\mathbf{x}.$$

Observe that for a C^1-invertible transformation T which transforms each element of volume V on to an element of volume $\frac{V}{s!}$

$$|J(T)| = \frac{1}{s!} \tag{9}$$

since $J(T)$ is the ratio of the volume of an infinitesimal region to the volume of the transformed region.

3.2.2 Root and Shift

For Root and Shift (9) can be proven directly by calculating the Jacobian. If $T(\mathbf{x}) = (y_1, \ldots, y_s)$ then the Jacobian is

$$|J(T)| = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} \cdots \frac{\partial y_1}{\partial x_s} \\ \cdots \\ \frac{\partial y_s}{\partial x_1} \cdots \frac{\partial y_s}{\partial x_s} \end{vmatrix}. \tag{10}$$

Observe that $J(T)$ is a triangular matrix for Root and Shift, therefore (10) is reduced to

$$|J(T)| = |\frac{\partial y_1}{\partial x_1}\frac{\partial y_2}{\partial x_2}\cdots\frac{\partial y_s}{\partial x_s}|.$$

It is now easy to verify that the Jacobian for both Root and Shift is equal to $\frac{1}{s!} = vol_s(T_s)$.

3.2.3 Mirror, Sort and Origami

Mirror, Sort and Origami do not transform equal volumes to equal volumes. To illustrate this in Figure 5 two rectangles with equal volume are shown with their transformed counterparts. Furthermore Mirror, Sort and Origami are not invertible since they are not injections. Therefore we cannot use Lemma 3.3.

However, because $Sort : S_i \to T_s$ is a C^1-invertible mapping which transforms each element in S_i of volume V onto an element in T_s of the same volume V (from S_i to T_s, Sort is a permutation), we know that

$$\int_{S_i}(f\ o\ T)(\mathbf{x})\mathsf{d}\mathbf{x} = \int_{T_s} f(\mathbf{x})\mathsf{d}\mathbf{x}.$$

This can be used to derived the desired result:

$$\int_{I_s} (f\ o\ T)(\mathbf{x})\mathsf{d}\mathbf{x} = \sum_{i=1}^{s!} \int_{S_i} (f\ o\ T)(\mathbf{x})\mathsf{d}\mathbf{x} = \sum_{i=1}^{s!} \int_{T_s} f(\mathbf{x})\mathsf{d}\mathbf{x} = s! \int_{T_s} f(\mathbf{x})\mathsf{d}\mathbf{x}.$$

A similar reasoning for Mirror (a reflection is a C^1-invertible mapping that preserves volumes) and Origami leads to the same result for these transformations.

3.3 The Variation of $f\ o\ T$

Reflecting a function $g(x, y)$ over the line $x+y = 1$ results into a new function $g(1-y, 1-x)$. This function has the same integral on T_s as the original function and is just as flat or steep. The only difference is a change in variables. If $g(x, y)$ varies a lot in x then $g(1 - y, 1 - x)$ will vary in y and vice versa.

Because

$$\cos(2\pi(10(1 - x) + (1 - y))) = \cos(2\pi(10x + y))),$$

$\cos(2\pi(10x + y)))$ is the reflection of $\cos(2\pi(x + 10y))$ over the line $x + y = 1$.

Table 1: Variation of a Cosinus

	$\cos(2\pi(10x + y))$	$\cos(2\pi(x + 10y))$
Sort	331.32	259.32
Shift	822.26	786.30
Root	841.19	180.10

In Table 1 we present the Variations for these 2 functions in combination with 3 transformations. This shows that the transformation *and* the ordering of the variables matter. This ordering also plays a role when an arbitrary hyperrectangle is mapped to the standard unit cube!

Observe that for the given example, both the highest and the lowest variation are obtained by transformation Root. This transformation performs well for functions with a high variation if the variation is mainly in the x-direction, and bad if the variation is mainly in the y-direction. This is caused by the peculiar way that Root treats both sharp angles of the triangle T_2.

4 Some experiments in 2 dimensions

In this section we will present the results of some numerical experiments using the transformations mentioned in Section 2. We restrict our attention to two dimensions. We used TRITST [2], a subroutine for evaluating the performance of subroutines for automatic integration over triangles.

$$\int_0^1 \int_0^{1-x} f_j(x, y)\mathsf{d}x\mathsf{d}y.$$

We first have to transform the point sets from T_2 to the triangle with vertices $(0, 0)$, $(0, 1)$ and $(1, 0)$. The transformation we used for this is $(x, y) \to (x, 1 - y)$. This is a bijection with a constant Jacobian equal to one. So this will not influence the order of convergence. Recall that this is just one of the six possible affine mappings from one triangle to the other and the choice influences the results if one looks at the details.

The results presented here use the following families:

Test families	Attributes
$f_1(x,y) = (\lvert x-\beta_1\rvert + y)^{d_1}$	singularity on x-axis
$f_2(x,y) = \begin{cases} 1 \text{ if } \sqrt{(x-\beta_1)^2+(y-\beta_2)^2} < d_2 \\ 0 \text{ otherwise} \end{cases}$	discontinuous
$f_3(x,y) = \exp(-(\alpha_1\lvert x-\beta_1\rvert + \alpha_2\lvert y-\beta_2\rvert))$	C_0 function
$f_4(x,y) = \cos(2\pi\beta_1 + \alpha_1 x + \alpha_2 y)$	oscillatory

Each family is characterised by its attribute and its difficulty parameters. We chose the following difficulty parameters:

$$d_1 = -0.9 \qquad d_2 = 0.25 \qquad d_3 = 75 \qquad d_4 = 30$$

The parameters α_1 and α_2 are first picked randomly from $[0,1]$ and then scaled according to $\alpha_1 + \alpha_2 = d_j$.

The above tests are normally used for automatic subregion-adaptive routines. Such routines are of course much more efficient for the given problems than quasi-Monte Carlo methods. We only use these tests to compare the different transformations introduced in Section 2.

Figures 6–9 show a loglog plot of the absolute error versus the number of integrand evaluations. We used 100 sample functions of each family and plotted the average error for each transformation. The original point set was a Sobol sequence.

As can be seen in Figures 6–9, there is not much difference in performance between the transformations except for transformation Drop which consistently performs a bit worse than the others. From this, and other experiments, we conclude that the speed of the transformations might be a deciding factor in practice. (In two dimensions the speed is more or less the same for all transformations.)

A problem with Lemma 3.1 is that it is not applicable for all transformations we considered. The observation that there is no significant difference between them, not even between a continuous and a discontinuous transformation, lead us to the derivation of some theory specifically for the simplex.

5 Koksma-Hlawka on a simplex

In this article we focused on the practical results we obtained. They inspired us to derive some theoretical results on variations and discrepancies defined on a simplex. These are described in [7]. Here we give a brief overview of the main results.

We defined a variation in the sense of Vitali and in the sense of Hardy and Krause on T_s, denoted by $V_\triangle(f)$, which together with a very intuitive generalization of the star discrepancy, denoted by $D^*_\triangle(P)$, leads to a Koksma-Hlawka-type inequality on T_s.

Definition 5.1 (Star Discrepancy on T_s) *Let Υ be the set of all rectangles containing the origin $\mathbf{o} = (0,0,\ldots,0)$,*

$$\Upsilon := \{\prod_{i=1}^{s}[0,x_i] : x_i \in]0,1]\},$$

then the star discrepancy on the simplex T_s can be defined as

$$D^*_\triangle(P) := \max_{U\in\Upsilon}\left|\frac{vol(U\cap T_s)}{vol(T_s)} - \frac{A(U)}{N}\right|$$

where $A(U)$ is the number of points falling inside U.

Using this definition and an appropriate definition of the variation $V_\triangle(f)$, we obtained the following result.

Legend: Drop, Sort, Sort (other ordering), Root, Shift, Origami, Mirror

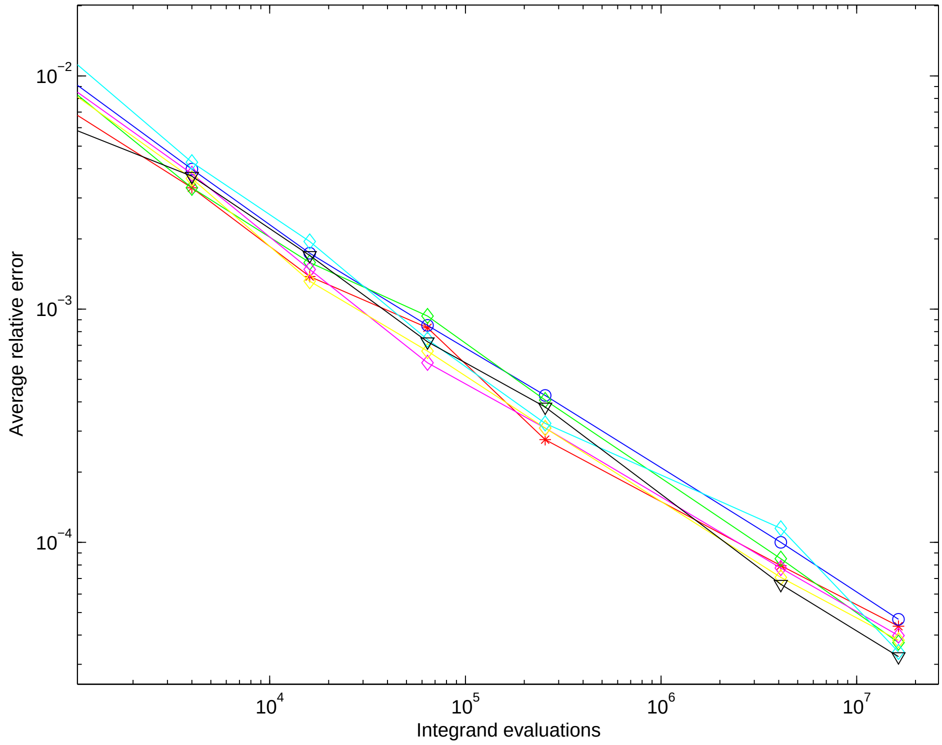


Figure 6: Family 1 in 2D for all transformations

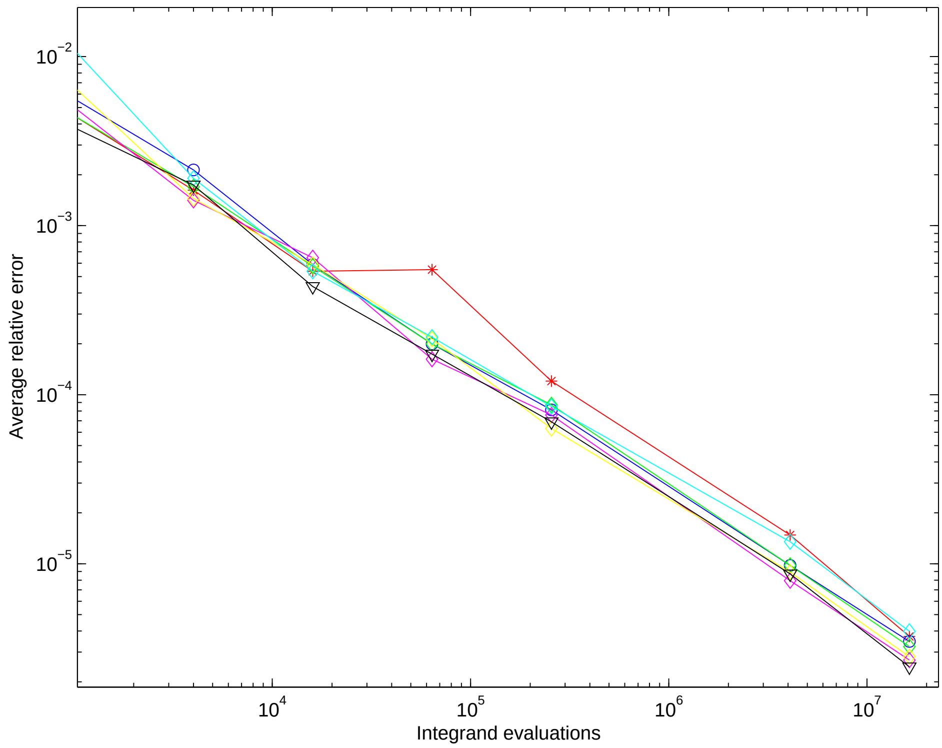


Figure 7: Family 2 in 2D for all transformations

Legend: Drop, Sort, Sort (other ordering), Root, Shift, Origami, Mirror

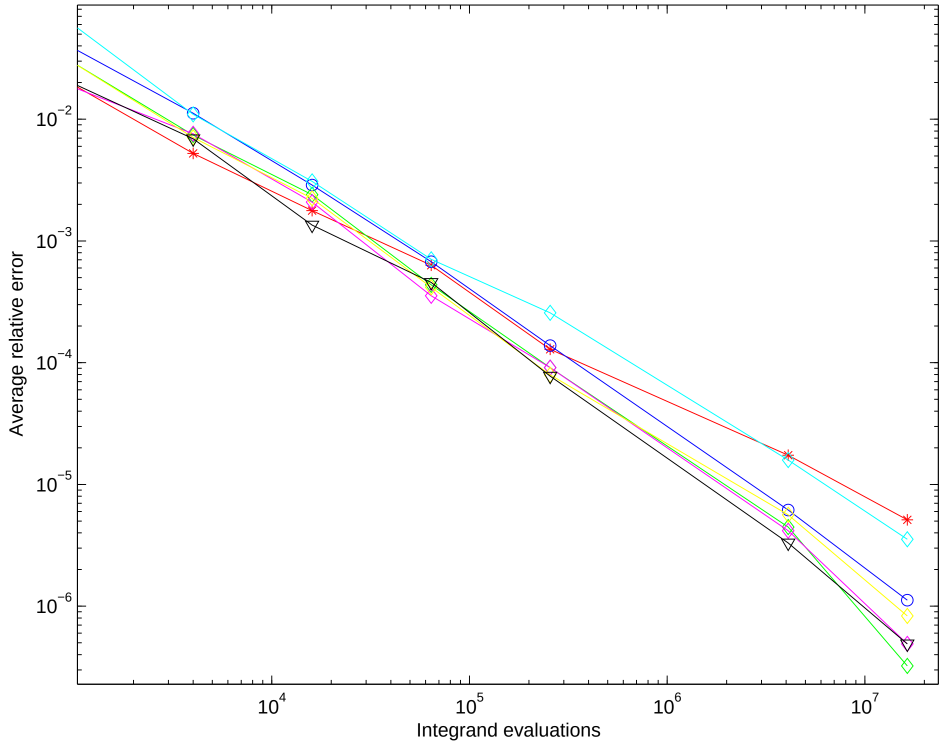


Figure 8: Family 3 in 2D for all transformations

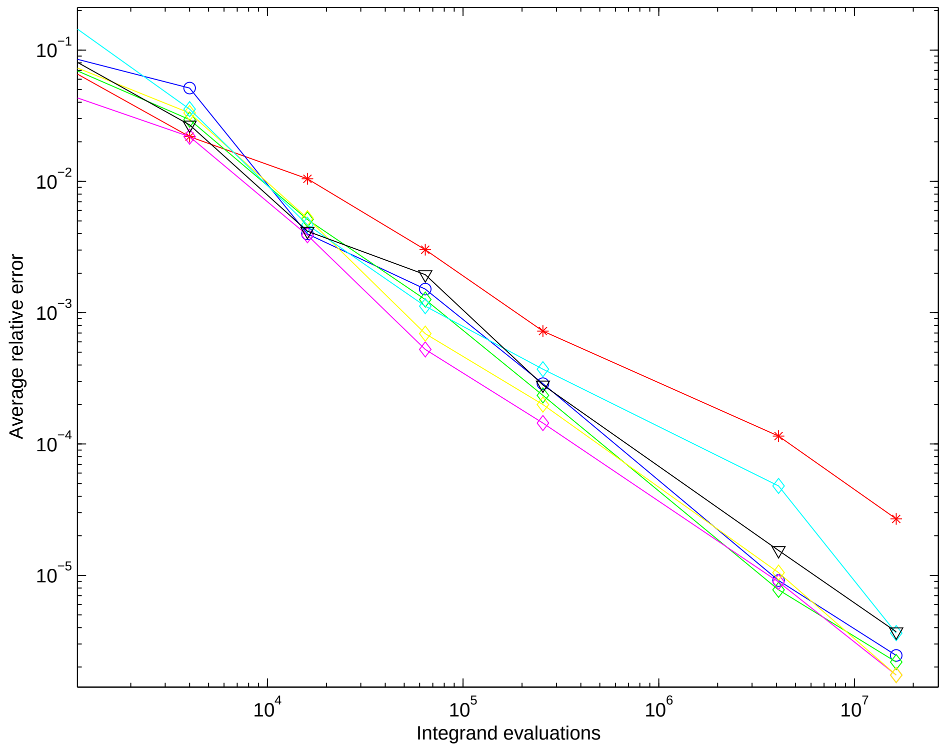


Figure 9: Family 4 in 2D for all transformations

Theorem 5.1 (Koksma-Hlawka on T_s) *For every point set $P \subset T_s$ with $\#P = N$ and every function $f : T_s \to \mathbb{R}$ with bounded variation on T_s*

$$|Q[f] - I[f]| = \left| \frac{vol_s(T_s)}{N} \sum_{\mathbf{x_i} \in P} f(\mathbf{x_i}) - \int_{T_s} f(\mathbf{x}) \mathsf{d}\mathbf{x} \right| \leq D^*_{\triangle}(P) V_{\triangle}(f).$$

It remains to be investigated what the relation is between $D^*_{\triangle}(T(P))$ and $D^*(P)$ when P is a set of low discrepancy points on the unit cube. That is beyond the scope of this paper. For more about this, we refer to [7].

References

[1] Ph. Bekaert. *Hierarchical and stochastic algorithms for radiosity.* PhD K.U.Leuven, 1999.

[2] J. Berntsen. TRITST: A Subroutine for Evaluating the Performance of Subroutines for Automatic Integration over Triangles. Reports in Informatics 34, Dept. of Informatics, University of Bergen, 1989.

[3] C. H. Edwards, Jr. *Advanced Calculus of Several Variables.* Academic Press, New York, 1973.

[4] K.-T. Fang and Y. Wang. *Number Theoretic Methods in Statistics.* Chapman and Hall, London, 1994.

[5] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*, *Springer series in operations research.* Springer, New York, 1996.

[6] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *CBMS-NSF regional conference series in applied mathematics.* SIAM, Philadelphia, 1992.

[7] T. Pillards and R. Cools *A Theoretical View on Transforming Low Discrepancy Sequences from a Cube to a Simplex.* in preparation.