The Integration Test Problems Toolbox

Michael Baudin

Version 0.1
June 2010

DRAFT

Table des matières

1 Overview

The goal of this document is to present the Integration Problems toolbox for Scilab.

2 Introduction

The goal of this toolbox is to provide integration problems in order to test integration algorithms. Each problem is provided as a function which takes a real argument $\mathbf{x} \in [0,1]^s$ in the unit hypercube and returns a real output $y = f(\mathbf{x}) \in \mathbb{R}$.

This toolbox can be used in the context of the numerical evaluation of any numerical integration algorithms. However, it was designed to serve as a base for testing Monte Carlo (MC) and Quasi Monte Carlo (QMC) algorithms.

2.1 Overview of functions

The test functions provided in the toolbox are presented in the table 1.

The functions #1 to #9 are taken from [12]. The functions #10 are taken from [14].

2.2 Designing test functions

In this section, we describe how and why most functions provided in this toolbox are with expectation zero and variance unity.

No	File	s	Name
# 1.	SUM	(10)	Sum
# 2.	SQSUM	(10)	Sum of Squares
# 3.	SUMSQROOT	(10)	Sum of Square Roots
# 4.	PRODONES	(10)	Product of Signed Ones
# 5.	PRODEXP	(10)	Product of Exponentials
# 6.	PRODCUB	(10)	Product of Cubes
# 7.	PRODX	(10)	Product of X
# 8.	SUMFIFJ	(10)	Sum of fi fj
# 9.	SUMF1FJ	(10)	Sum of f1 fj
# 10.	HELLEKALEK	(10)	Hellekalek
# 11.	ROOSARNOLD1	(10)	Roos and Arnold 1
# 12.	ROOSARNOLD2	(10)	Roos and Arnold 2
# 13.	ROOSARNOLD3	(10)	Roos and Arnold 3
# 14.	RST1	(10)	Radovic, Sobol, Tichy ($a_j = 1$)
# 15.	RST2	(10)	Radovic, Sobol, Tichy ($a_j = j$)
# 16.	RST3	(10)	Radovic, Sobol, Tichy ($a_j = j^2$)
# 17.	SOBOLPROD	(10)	Sobol Product
# 18.	OSCILL	(6)	Genz - Oscillatory
# 19.	PRPEAK	(6)	Genz - Product Peak
# 20.	CORPEAK	(6)	Genz - Corner Peak
# 21.	GAUSSIAN	(6)	Genz - Gaussian
# 22.	C0	(6)	Genz - C0
# 23.	DISCONT	(6)	Genz - Discontinuous

FIGURE 1 – Test functions provided in the toolbox.

We assume that s is a positive integer and that the point $\mathbf{x} \in [0,1]^s$. The function $f : [0,1]^s \to \mathbb{R}$ is assumed to have a finite expectation

$$E(f) = \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x}, \tag{1}$$

and finite variance

$$\sigma^2(f) = \int_{[0,1]^s} f(\mathbf{x})^2 d\mathbf{x} - E(f)^2. \tag{2}$$

Assume that we approximate the expectation of the function by the Monte-Carlo estimate [8]

$$I = \frac{1}{n} \sum_{k=1,\ldots,n} f(\mathbf{x}_k), \tag{3}$$

where the points $\mathbf{x}_k$ are independent uniformly distributed points in the hypercube $[0,1]^s$. The estimate I is an unbiased estimator of $E(f)$, and its variance is

$$\int_{[0,1]^s} (f(\mathbf{x}) - E(f))^2 d\mathbf{x} = \frac{\sigma^2(f)}{n}. \tag{4}$$

In [26], Kocis and Whiten provide analysis of test functions by using the functional ANOVA. The purpose of their work is to describe what particular functions are difficult to integrate with Monte-Carlo and Quasi-Monte-Carlo methods. They state that ”adding a large constant to the test function (assumed small) improves the relative accuracy of the integration, but clearly this is of no significance in testing an integration sequence.” Thus, ”the first step should be to remove any constant component from the test function as follows :”

$$f_0 = E(f) = \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x}, \tag{5}$$

and the remaining part of $f(\mathbf{x})$ is $f(\mathbf{x}) - f_0$.

Later in their text, they notice that, in the case of Monte-Carlo integration, the error depends on the variance of the function being integrated. Hence, the suggest that to give comparable results, the variance of the function should be normalised to a constant value, chosen, without loss of generality to be one. Let us consider the function $\overline{f}$ defined by

$$\overline{f}(\mathbf{x}) = \frac{1}{\sqrt{\sigma^2(f)}} \left(f(\mathbf{x}) - E(f) \right). \tag{6}$$

Then the expectation of the function $\overline{f}$ is

$$\begin{aligned}
E(\overline{f}(X)) &= \int_{[0,1]^s} \overline{f}(\mathbf{x}) d\mathbf{x} && (7)\\
&= \frac{1}{\sqrt{\sigma^2(f)}} \int_{[0,1]^s} \left(f(\mathbf{x}) - E(f) \right) d\mathbf{x} && (8)\\
&= \frac{1}{\sqrt{\sigma^2(f)}} \left(\int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x} - E(f) \right) && (9)\\
&= 0, && (10)
\end{aligned}$$

and the variance of the function $\overline{f}$ is

$$\begin{aligned}
\sigma^2(\overline{f}) &= \int_{[0,1]^s} \overline{f}(\mathbf{x})^2 d\mathbf{x} - E(\overline{f})^2 && (11)\\
&= \frac{1}{\sigma^2(f)} \int_{[0,1]^s} \left(f(\mathbf{x}) - E(f) \right)^2 d\mathbf{x} && (12)\\
&= \frac{1}{\sigma^2(f)} \left(\int_{[0,1]^s} f(\mathbf{x})^2 d\mathbf{x} - 2E(f) \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x} + E(f)^2 \right) && (13)\\
&= \frac{1}{\sigma^2(f)} \left(\int_{[0,1]^s} f(\mathbf{x})^2 d\mathbf{x} - 2E(f)E(f) + E(f)^2 \right) && (14)\\
&= \frac{1}{\sigma^2(f)} \left(\int_{[0,1]^s} f(\mathbf{x})^2 d\mathbf{x} - E(f)^2 \right) && (15)\\
&= \frac{1}{\sigma^2(f)} \sigma^2(f) && (16)\\
&= 1. && (17)
\end{aligned}$$

Hence the expectation of the function $\overline{f}$ is zero while its variance is one. In the bibliography, we often find functions with non-zero expectation or non-unity variance. In these cases, when both the expectation and the variance are explicitely known, we use the formula 6 and provide the function $\overline{f}$ instead of f. If the expectation only is known, we cannot normalize the variance and we provide the function $f(\mathbf{x}) - E(f)$ instead of f. When the expectation is not known analytically, we provide f directly.

3 Functions

In this section, we describe the functions provided in the toolbox. We provide the function definition and the references were the function appeared first. We also list several papers using the function and which may provide an analysis or numerical experiments using the function : this may be useful in comparison studies.

3.1 SUM

This is the function F_1 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}}(g(\mathbf{x}) - e), \tag{18}$$

where $\mathbf{x} \in [0,1]^s$ and

$$e = \frac{s}{2} \tag{19}$$

$$v = \frac{s}{12} \tag{20}$$

$$g(\mathbf{x}) = \sum_{i=1,\ldots,s} x_i. \tag{21}$$

The contours of this function in the case $s = 2$ are presented in the figure 2.

The function f is well behaved and thus satisfies conditions for proper quasi-Monte-Carlo integration. But it is not truly multi-dimensional in that it is a sum of lower-dimensional functions.

3.2 SQSUM

This is the function F_2 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}}(g(\mathbf{x}) - e), \tag{22}$$

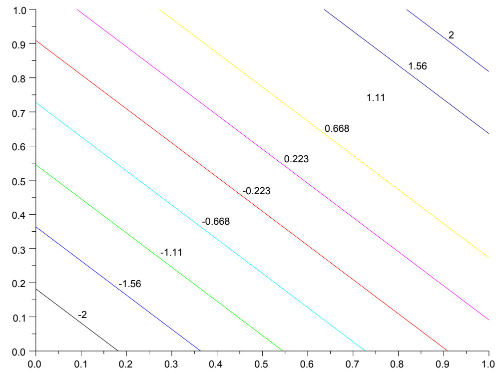

FIGURE 2 – The function SUM.

where $\mathbf{x} \in [0,1]^s$ and

$$e = \frac{s}{3} \tag{23}$$

$$v = \frac{4n}{45} \tag{24}$$

$$g(\mathbf{x}) = \sum_{i=1,\ldots,s} x_i^2. \tag{25}$$

The function f is well behaved and thus satisfies conditions for proper quasi-Monte-Carlo integration. But it is not truly multi-dimensional in that it is a sum of lower-dimensional functions.

The contours of this function in the case $s=2$ are presented in the figure 3.

3.3 SUMSQROOT

This is the function F_3 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}}(g(\mathbf{x}) - e), \tag{26}$$

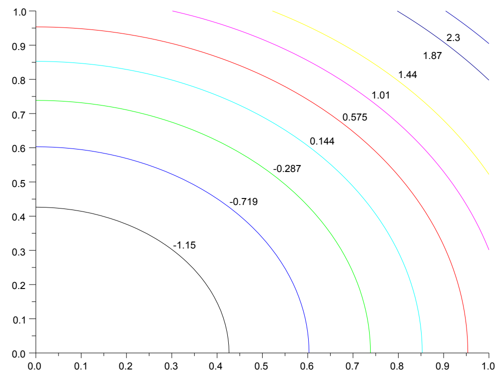

FIGURE 3 – The function SQSUM.

where $\mathbf{x} \in [0,1]^s$ and

$$e = \frac{2n}{3} \tag{27}$$

$$v = \frac{s}{18} \tag{28}$$

$$g(\mathbf{x}) = \sum_{i=1,\ldots,s} \sqrt{x_i}. \tag{29}$$

The function f is well behaved and thus satisfies conditions for proper quasi-Monte-Carlo integration. But it is not truly multi-dimensional in that it is a sum of lower-dimensional functions.

The contours of this function in the case $s = 2$ are presented in the figure 4.

3.4 PRODONES

This is the function F_4 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \prod_{i=1,\ldots,s} g(x_i), \tag{30}$$

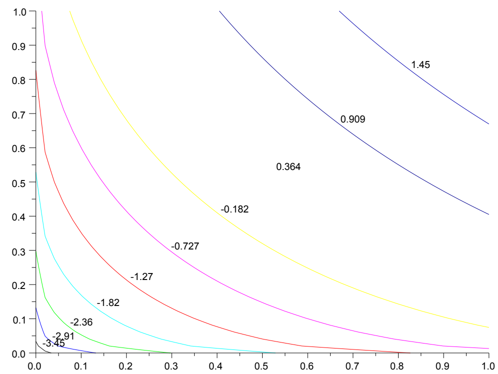

FIGURE 4 – The function SUMSQROOT.

where $\mathbf{x} \in [0,1]^s$ and

$$g(z) = \begin{cases} -1 & \text{if } z < \frac{1}{2} \\ 1 & \text{otherwise.} \end{cases} \quad (31)$$

The function f is discontinuous, but the function is either 1 or -1 (i.e. the local extremes of the function do not increase with the dimension s). The function is piecewise constant in 2^s different regions. The conditions for proper quasi-Monte Carlo are not satisfied and the rate of convergence for this function is expected to be low.

The contours of this function in the case $s = 2$ are presented in the figure 5.

3.5 PRODEXP

This is the function F_5 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = w^s \prod_{i=1,\dots,s} g(x_i), \quad (32)$$

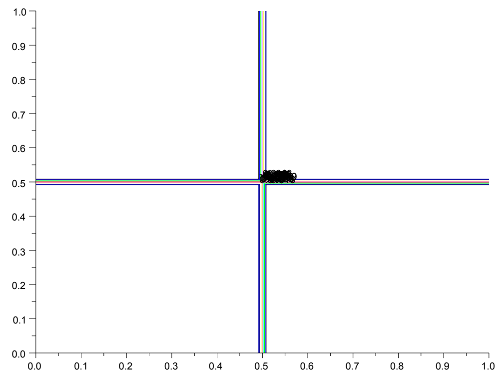

FIGURE 5 – The function PRODONES.

where $\mathbf{x} \in [0,1]^s$ and

$$w = \left(\frac{15\exp(15)+15}{13\exp(15)+17}\right)^{\frac{1}{2}} \quad (33)$$

$$g(z) = \frac{\exp(30z-15)-1}{\exp(30z-15)+1} \quad (34)$$

The function f is smooth but the number of extremes increases as 2^s. For example, the maximum of the function f in the hypercube $[0,1]^s$ is $2.69 \cdot 10^{12}$ for $s = 400$. The conditions for proper quasi-Monte Carlo are not satisfied and the rate of convergence for this function is expected to be low.

The contours of this function in the case $s = 2$ are presented in the figure 6.

3.6 PRODCUB

This is the function F_6 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \prod_{i=1,\ldots,s} g(x_i), \quad (35)$$

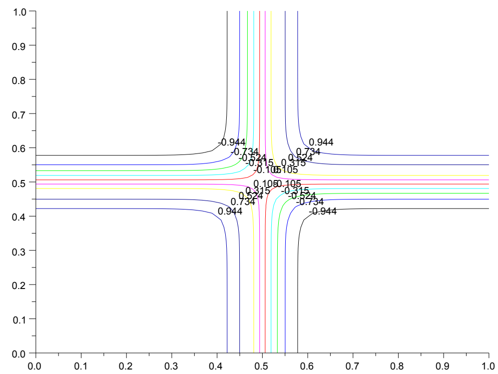

FIGURE 6 – The function PRODEXP.

where $\mathbf{x} \in [0,1]^s$ and

$$g(z) \quad = \quad -2.4\sqrt{7}\left(z - \frac{1}{2}\right) + 8\sqrt{7}\left(z - \frac{1}{2}\right)^3. \tag{36}$$

The function f is smooth but the number of extremes increases as 2^s. For example, the maximum of the function f in the hypercube $[0,1]^s$ is $4.65 \cdot 10^{50}$ for $s = 400$. The conditions for proper quasi-Monte Carlo are not satisfied and the rate of convergence for this function is expected to be low.

The contours of this function in the case $s = 2$ are presented in the figure 7.

3.7 PRODX

This is the function F_7 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = w^s \prod_{i=1,\dots,s} g(x_i), \tag{37}$$

where $\mathbf{x} \in [0,1]^s$ and

$$w \quad = \quad 2\sqrt{3} \tag{38}$$

$$g(z) \quad = \quad z - \frac{1}{2}. \tag{39}$$

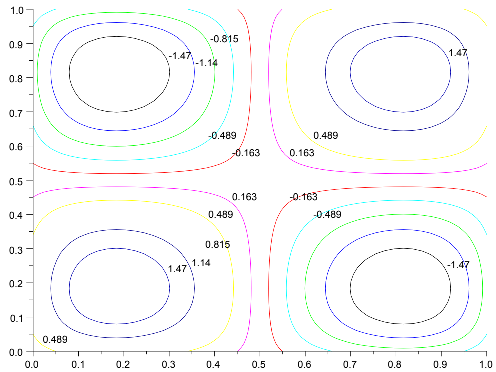

FIGURE 7 – The function PRODCUB.

The function f is smooth but the number of extremes increases as 2^s. For example, the maximum of the function f in the hypercube $[0,1]^s$ is $2.66 \cdot 10^{95}$ for $s = 400$. The conditions for proper quasi-Monte Carlo are not satisfied and the rate of convergence for this function is expected to be low.

Owen analyses this function in [14] and reports previous results from [15] and [16]. This test function is fully s dimensional and the improvement of QMC over MC sets in at $n \geq b^s$, where b is the base of the $(\lambda, 0, m, s)-$ *net*. In [16], Owen indicates that the function f is multilinear, so that f is not artificially easy for scrambled nets, at least in terms of the rates of convergence to be expected. Owen writes "there are harder s dimensional integrands, such as oscillatory or localized ones".

Owen also explains the behavior of MC and QMC algorithms when s becomes large. "In large dimensions, this integrand is difficult for MC. Indeed, most of the variation is concentrated in 2^s small corner regions. For large s and small n, the MC estimate will be usually very close to the exact zero, because non of the corner spikes will have been sampled. But, as the number of simulations n increases, a small number of spike samples will be obtained and the result will be erratic until n becomes so large that a large number of spike samples has been seen."

The contours of this function in the case $s = 2$ are presented in the figure 8.

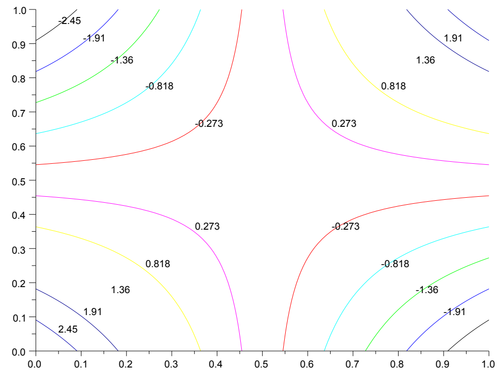

FIGURE 8 – The function PRODX.

3.8 SUMFIFJ

This is the function F_8 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}} \sum_{i=1,\dots,s} g_i(\mathbf{x}), \tag{40}$$

where $\mathbf{x} \in [0,1]^s$ and

$$g_i(\mathbf{x}) \quad = \quad g(x_i) \sum_{j=1,2,\dots,i-1} g(x_j). \tag{41}$$

The function g is defined by

$$g(z) = \begin{cases} 1 & \text{if } z < \frac{1}{6} \text{ or } z > \frac{4}{6} \\ 0 & \text{if } z = \frac{1}{6} \text{ or } z = \frac{4}{6} \\ -1 & \text{if } z > \frac{1}{6} \text{ and } z > \frac{4}{6}. \end{cases} \tag{42}$$

The function f has only two-dimensional components being constructed as sum of products of all combinations of two-one variate functions. This allows the simultaneous testing and averaging of all the possible two-dimensional functions available within the chosen dimension s. The function F_8 corresponds to the function F_5, having flat regions and discontinuities.

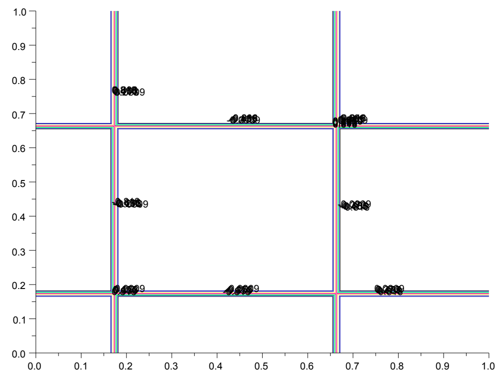

FIGURE 9 – The function SUMFIFJ.

The number of regions of F_8 is 3^s. The extremes of the function f do not increase rapidly with the number of dimensions s.

The contours of this function in the case $s = 2$ are presented in the figure 9.

3.9 SUMF1FJ

This is a modification of the function F_9 in Kocis and Whiten's [12].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{s-1}} g(x_1) \sum_{i=2,\dots,s} g(x_i), \tag{43}$$

where $\mathbf{x} \in [0,1]^s$ and

$$g(z) = 27.20917094z^3 - 36.19250850z^2 + 8.983337562z + 0.7702079855. \tag{44}$$

The function f has only two-dimensional components being constructed as sum of products of all combinations of two-one variate functions. This allows the simultaneous testing and averaging of all the possible two-dimensional functions available within the chosen dimension s. The function F_8 corresponds to the function F_6, being smooth and based on one-dimensional cubic functions. The extremes of the function f do not increase rapidly with the number of dimensions s.

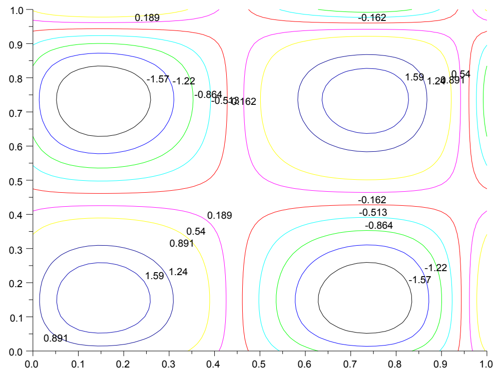

FIGURE 10 – The function SUMF1FJ.

The contours of this function in the case $s = 2$ are presented in the figure 10.

3.10 HELLEKALEK

This function is from [9] and is called "Hellekalek's example" in [14].

This function is defined by

$$f(\mathbf{x}) = \prod_{i=1,\ldots,s} g(x_i), \tag{45}$$

where $\mathbf{x} \in [0, 1]^s$ and

$$g(z) = \frac{h(z)}{\gamma_i} \tag{46}$$

$$h(z) = z^\alpha - \frac{1}{\alpha + 1}. \tag{47}$$

The parameter α ranges from 1 to 3. In the toolbox, we have chosen

$$\alpha = 1. \tag{48}$$

The variable γ_i^2 represents the variance of the function $h(x_i)$ for $i = 1, 2, \ldots, s$, which satisfies

$$\gamma_i^2 = \frac{\alpha^2}{(2\alpha + 1)(\alpha + 1)^2}. \tag{49}$$

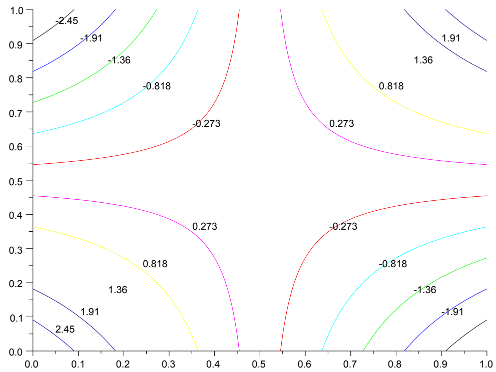

FIGURE 11 – The function HELLEKALEK.

Owen [14] prooves that this function is fully s-dimensionnal. Hence QMC does not improve on Monte-Carlo. A (t, m, s)-net does not balance any elementary intervals of dimension s, unless $n \geq b^{t+s}$.

The contours of this function in the case $s = 2$ are presented in the figure 11.

3.11 ROOSARNOLD1

This function is from Roos and Arnold [21] and is presented in Davis and Rabinowitz [3]. It is analysed by Owen [14].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}} \left(\sum_{i=1,\ldots,s} g(x_i) - e \right), \tag{50}$$

where $\mathbf{x} \in [0, 1]^s$ and

$$\begin{aligned} g(z) &= \frac{|4z - 2|}{s}, &(51)\\ e &= 1, &(52)\\ v &= \frac{1}{3s}. &(53) \end{aligned}$$

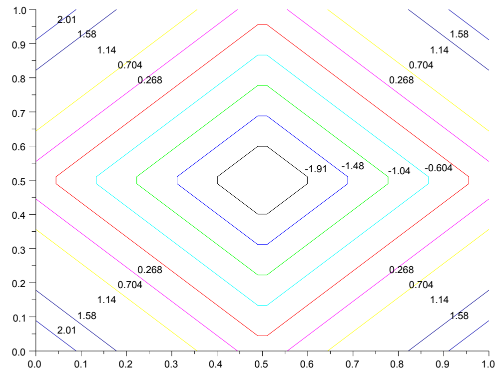

FIGURE 12 – The function ROOSARNOLD1.

Owen writes in [14] that this function is additive and purely one dimensional.

The contours of this function in the case $s = 2$ are presented in the figure 12.

3.12 ROOSARNOLD2

This function is from Roos and Arnold [21] and is presented in Davis and Rabinowitz [3]. This integrand is considered in the context of QMC by Fox in [4]. This integrand can be viewed as a particular case of Radovic, Sobol and Tichy's function [20], analysed by Owen [14].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}} \left(\prod_{i=1,\ldots,s} g(x_i) - e \right), \tag{54}$$

where $\mathbf{x} \in [0,1]^s$ and

$$g(z) = |4z - 2|, \tag{55}$$

$$e = 1, \tag{56}$$

$$v = \left(\frac{4}{3}\right)^s - 1. \tag{57}$$

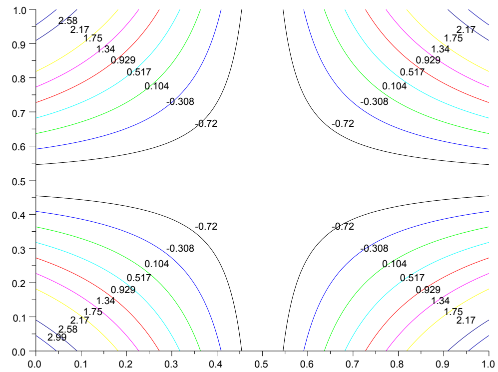

FIGURE 13 – The function ROOSARNOLD2.

All variables are important in this function.

In [4], Fox analyses this function. Consider $V(f)$, the variation of f in the sense of Hardy and Krause. It is possible to show that

$$V(f) \geq \sup\{f(\mathbf{x})|\mathbf{x} \in [0,1]^s\} - \inf\{f(\mathbf{x})|\mathbf{x} \in [0,1]^s\}. \tag{58}$$

For the current function, we have

$$V(f) \geq 2^s. \tag{59}$$

This shows that a huge error is possible in any approximation method when s is large.

The contours of this function in the case $s = 2$ are presented in the figure 13.

3.13 ROOSARNOLD3

This function is from Roos and Arnold [21] and is presented in Davis and Rabinowitz [3]. It is analysed by Owen [14].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}} \left(\prod_{i=1,\ldots,s} g(x_i) - e \right), \tag{60}$$

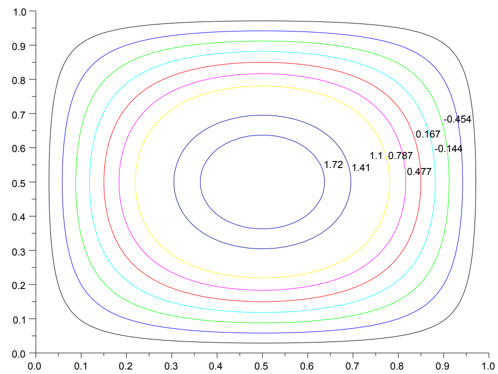

FIGURE 14 – The function ROOSARNOLD3.

where $\mathbf{x} \in [0,1]^s$ and

$$g(z) = \frac{\pi}{2}\sin(\pi z), \quad (61)$$

$$e = 1, \quad (62)$$

$$v = \left(\frac{\pi^2}{8}\right)^s - 1. \quad (63)$$

Owen writes in [14] that this function has mean dimensionnality which grows nearly linearly with s.

The contours of this function in the case $s = 2$ are presented in the figure 14.

3.14 RST1, RST2, RST3

This function is from Radovic, Sobol and Tichy [20]. It is analysed by Owen in [14], by Okten, Shah and Goncharov in [13].

This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}}\left(\prod_{i=1,\ldots,s} g_i(x_i) - e\right), \quad (64)$$

where $\mathbf{x} \in [0,1]^s$ and

$$g_i(z) = \frac{|4z-2| + a_i}{1+a_i}. \quad (65)$$

The parameters a_i for $i = 1, 2, \ldots, s$ allows to tune the importance of each variable x_i. If $a_i = 0$, then the parameter x_i is important and when $a_i \geq 0$ is larger, the variable x_i is less important. We use the following set of parameters :

- RST1 uses $a_i = 1$ for $i = 1, 2, \ldots, s$, where all variables have equal importance,
- RST2 uses $a_i = i$ for $i = 1, 2, \ldots, s$, where the importance of the variable x_i decreases as i increases,
- RST3 uses $a_i = i^2$ for $i = 1, 2, \ldots, s$, where the importance of the variable x_i decreases as i increases.

The expectation and variance of the function f can be computed based on the expectation and variance of the functions g_i. Let us define

$$\mu_i = 1, \quad (66)$$

$$\gamma_i^2 = \frac{1}{3(1+a_i)^2}, \quad (67)$$

for $i = 1, 2, \ldots, s$. Then, the expectation and variances of the function f are

$$e = \prod_{i=1,\ldots,s} \mu_i \quad (68)$$

$$v = \prod_{i=1,\ldots,s} (\mu_i^2 + \gamma_i^2) - \prod_{i=1,\ldots,s} \mu_i^2. \quad (69)$$

For numerical integration, the function RST1 may be difficult, RST2 more easy and RST3 easy.

The contours of this function in the case $s = 2$ are presented in the figures 15, 16 and 17.

3.15 SOBOLPROD

This function is designed by Sobol' in [24]. It is analysed by Owen in [14]. This function is defined by

$$f(\mathbf{x}) = \frac{1}{\sqrt{v}} \left(\prod_{i=1,\ldots,s} g_i(x_i) - e \right), \quad (70)$$

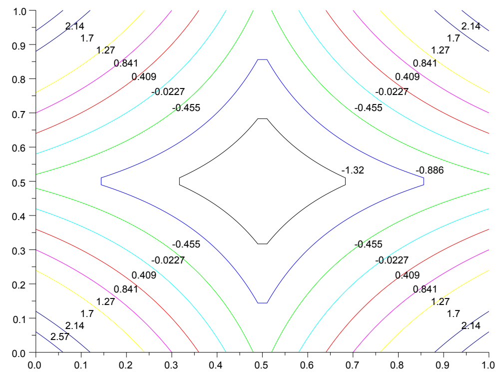

FIGURE 15 – The function RST1.

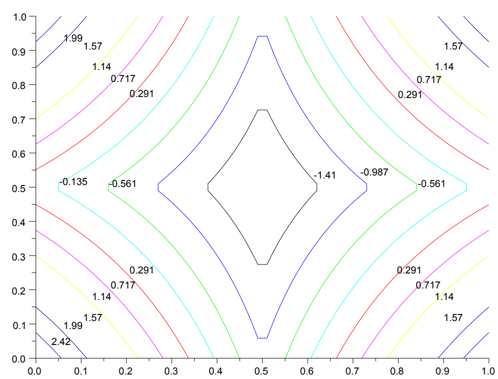

FIGURE 16 – The function RST2.

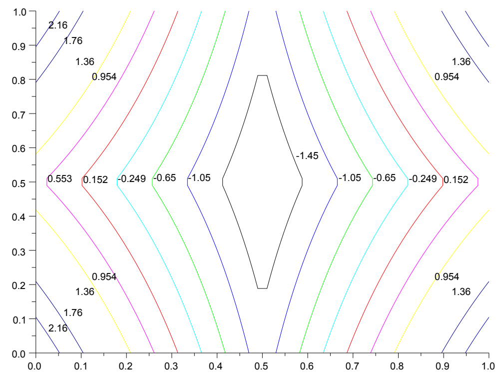

FIGURE 17 – The function RST3.

where $\mathbf{x} \in [0,1]^s$ and

$$g_i(z) = \frac{i+2z}{i+1}. \tag{71}$$

The expectation and variance of the function f can be computed based on the expectation and variance of the functions g_i. Let us define

$$\mu_i = 1, \tag{72}$$

$$\gamma_i^2 = \frac{1}{3(i+1)^2}, \tag{73}$$

for $i = 1, 2, \ldots, s$. Then, the expectation and variances of the function f are

$$e = \prod_{i=1,\ldots,s} \mu_i \tag{74}$$

$$v = \prod_{i=1,\ldots,s} (\mu_i^2 + \gamma_i^2) - \prod_{i=1,\ldots,s} \mu_i^2. \tag{75}$$

This function is an integrand which is favorable for quasi Monte Carlo. Sobol's shows in [24] a convergence of the error which has the form $1/n$ instead of the $1/\sqrt{n}$ of the Monte Carlo method.

Owen states in [14] that each variable x_{i+1} is less important than x_i. This integrand really depends on few variables. Owen writes that, for $s = 100$, the

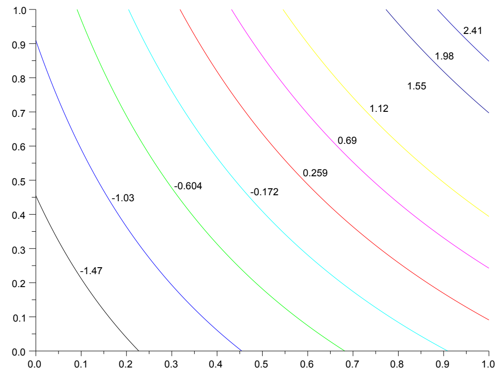

FIGURE 18 – The function SOBOLPROD.

mean dimension is 1.085. He writes that, for $s = 100$, at least 99.418% of the variation in this function is from its ANOVA components of dimension 4 and smaller.

The contours of this function in the case $s = 2$ are presented in the figure 18.

3.16 OSCILL

The next 6 functions are a part of the 6 functions designed by Genz and available in the Testpack fortran 77 package [5]. These functions have been ported to Matlab by John Burkardt in [2] and in [1].

The OSCILL function is designed by Genz in [6], [7], where it is named "Oscillatory". It is used by Joe and Sloan in [10, 11] (1993), by Shürer in [23] (2001), by Taylor and Hover in [25] (2007), by Pirsic in [18, 19].

This function is defined by :

$$f(\mathbf{x}) = \cos\left(2\pi\beta_1 + \sum_{i=1,2,\ldots,s} \alpha_i x_i\right) - e, \tag{76}$$

where $\mathbf{x} \in [0,1]^s$ and e is defined by :

$$
\begin{aligned}
e &= 2^s \cos\left(2\pi\beta_1 + \frac{1}{2}\hat{\alpha}\right) p && (77)\\
\hat{\alpha} &= \sum_{i=1}^{s} \alpha_i && (78)\\
p &= \prod_{i=1}^{s} \frac{\sin(\alpha_i/2)}{\alpha_i} && (79)
\end{aligned}
$$

For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

In this function, the parameters $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_s)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_s)$ allow to define a family of functions.

In [6], Genz distinguish two types of parameters which appear in an integration function.

1. Affective parameters are the parameters which affect the difficulty of the integration problem. This is the parameter α (denoted by $\mathbf{a}$ in [6]).
2. Unaffective parameters are the one which do not affect the difficulty of the integration problem. This is the parameter β (denoted by $\mathbf{u}$ in [6]).

The components of $\hat{\alpha}$ and β are uniformly distributed in the interval $[0,1]$. The vector β acts as a shift parameter. As detailed below, the vector α is then computed by scaling $\hat{\alpha}$.

Genz uses the uniform random number generator designed by Shrage in [22]. The generator is based on the multiplicative congruential generator

$$x = Ax(\pmod{M}) \tag{80}$$

with $A = 16807$ and $M = 2^{31} - 1 = 2147483647$. Schrage states that the generator is full cycle, with period M. In our implementation, the seed is constantly set to 123456.

The difficulty of the problem is controlled by the vector α, where increasing values of α are generating increasingly difficult problems. The scaling of α is done by the formula

$$\alpha = \frac{1}{\omega}\hat{\alpha} \tag{81}$$

where $\omega > 0$ is defined by :

$$\omega = \frac{s^e}{h} \sum_{i=1}^{s} \alpha_i. \tag{82}$$

The parameters e and h are chosen depending on the problem in order to control the difficulty of the integration. The previous equation leads to the equality $\|\alpha\|_1 = \frac{h}{s^e}$.

For this function, we have $e = 1.5$ and $h = 110$, which leads to

$$\|\alpha\|_1 = \frac{110}{s^{3/2}}. \tag{83}$$

A little modification was done to the function with respect to the Testpack library [5]. In Genz's testpack, the integration is done over an interval $[\mathbf{a}, \mathbf{b}]$, with $\mathbf{a} = (0, 0, \ldots, 0)^T$ and $\mathbf{b} = (\alpha, \alpha, \ldots, \alpha)^T$. Here, the integration is done over the unit interval $[0, 1]^s$, so that multiplying x by α is necessary to get the same problem.

According to Schurer, this integrand is very smooth, such that adaptive algorithms based on cubature performs better than Quasi-Monte-Carlo, even for a dimension as high as $s = 40$. On [18], Pirsic used the Oscillatory function in dimensions $s = 4$ to $s = 16$ with a number of experiments $n = 2^{21} \approx 10^6$. The results show a good improvement of QMC over crude Monte-Carlo, especially for low dimension, e.g. for $s \leq 10$.

The contours of this function in the case $s = 2$ are presented in the figure 19.

3.17 PRPEAK

This function is part of the 6 functions designed by Genz and available in the Testpack fortran 77 package [5]. This function has been ported to Matlab by John Burkardt in [2] and in [1].

The PRPEAK function is designed by Genz in [6], [7], where it is named "Product Peak". It is used by Joe and Sloan in [10, 11] (1993), by Shürer in [23] (2001), by Owen in [14] (2003).

This function is defined by

$$f(\mathbf{x}) = \prod_{i=1}^{s} \left(\alpha_i^{-2} + (x_i - \beta_i)^2\right)^{-1} - e, \tag{84}$$

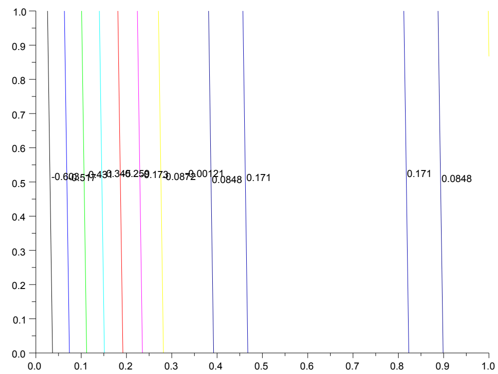

FIGURE 19 – The function OSCILL.

where $\mathbf{x} \in [0,1]^s$ and e is defined by :

$$e = \prod_{i=1}^{s} \left((s_i - t_i)\alpha_i\right), \quad (85)$$

$$s_i = \arctan((1-\beta_i)\alpha_i), \quad (86)$$

$$t_i = \arctan(-\beta_i \alpha_i). \quad (87)$$

For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

The vector α is scaled by the equations 81-82 with $e = 2$ and $h = 600$.

In [18], Pirsic used the Oscillatory function in dimensions $s = 4$ to $s = 16$ with a number of experiments $n = 2^{21} \approx 10^6$. The results show a good improvement of QMC over crude Monte-Carlo, whatever the dimension.

In [14], Owen states that the hard cases for this function are only of approximate dimension $s/2$.

The contours of this function in the case $s = 2$ are presented in the figure 20.

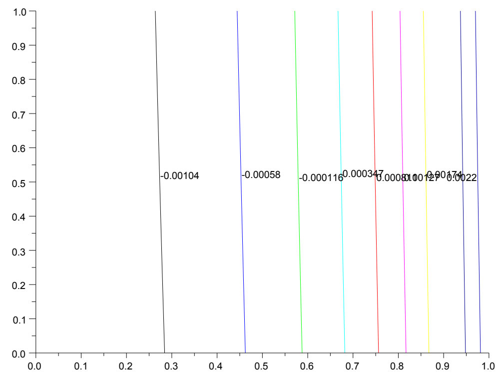

FIGURE 20 – The function PRPEAK.

3.18 CORPEAK

The CORPEAK function is designed by Genz in [6], [7], where it is named ”Corner Peak”.

It is used by Shürer in [23] (2001), by Joe and Sloan in [10, 11] (1993).

This function is defined by

$$f(\mathbf{x}) = \left(1 + \sum_{i=1}^{s} t_i\right)^{-n-1} - e, \tag{88}$$

where $\mathbf{x} \in [0,1]^s$ and t_i is defined by :

$$t_i = \begin{cases} \alpha_i x_i, & \text{if } \beta_i < 0.5 \\ \alpha_i - \alpha_i x_i, & \text{if not.} \end{cases} \tag{89}$$

The computation of e is done based on an algorithm which is not straightforward (see the source code for details). For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

The vector α is scaled by the equations 81-82 with $e = 2$ and $h = 600$.

A little modification was done to the function with respect to the Testpack library [5]. In Genz's testpack, the integration is done over an interval $[\mathbf{a}, \mathbf{b}]$, with $\mathbf{a} = (0, 0, \ldots, 0)^T$ and $\mathbf{b} = (\alpha, \alpha, \ldots, \alpha)^T$. Here, the integration is done

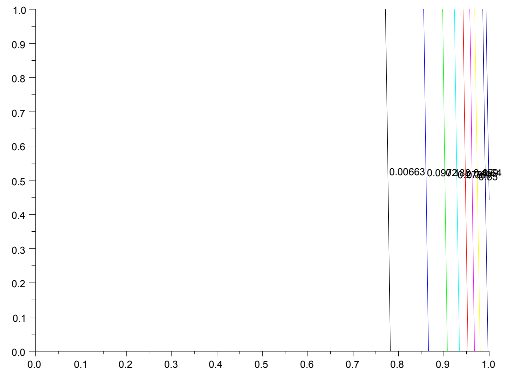

FIGURE 21 – The function CORPEAK.

over the unit interval $[0, 1]^s$, so that multiplying x by α is necessary to get the same problem.

The contours of this function in the case $s = 2$ are presented in the figure 21.

3.19 GAUSSIAN

The GAUSSIAN function is designed by Genz in [6], [7], where it is named "Gaussian". It is used by Owen in [14] (2003), by Shürer in [23] (2001), by Joe and Sloan in [10, 11] (1993).

This function is defined by

$$f(\mathbf{x}) = \exp\left(-\sum_{i=1}^{s} \alpha_i^2 (x_i - \beta_i)^2\right) - e, \tag{90}$$

where $\mathbf{x} \in [0, 1]^s$ and e is defined by :

$$e = \prod_{i=1}^{s} \frac{\sqrt{\pi}}{\alpha_i} (r_i - t_i), \tag{91}$$

$$r_i = \phi\left((1 - \beta_i)\sqrt{2}\alpha_i\right), \tag{92}$$

$$t_i = \phi\left(-\beta_i\sqrt{2}\alpha_i\right), \tag{93}$$

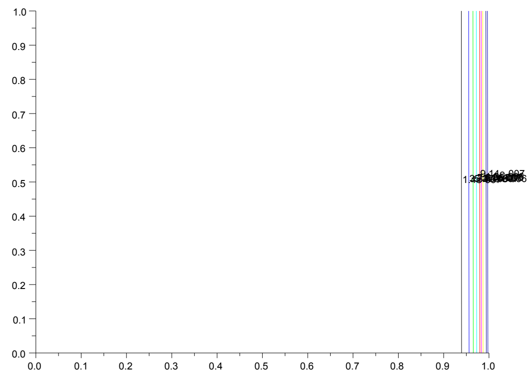

FIGURE 22 – The function GAUSSIAN.

where ϕ is the standard normal (Laplace-Gauss) cumulated distribution function, defined by

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{s^2}{2}\right) ds. \tag{94}$$

For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

Genz provides in Testpack [5] its own implementation of the function ϕ. In Scilab, we use the `cdfnor` function.

In [14], Owen states that if the parameters α_i are large enough, this function is essentially full dimension s.

The contours of this function in the case $s = 2$ are presented in the figure 22.

3.20 C0

The C0 function is designed by Genz in [6], [7], where it is named ”C0 Function” (other authors uses the term ”Continuous”).

It is used by Petras in [17] (2003), by Owen in [14] (2003), by Shürer in [23] (2001), by Joe and Sloan in [10, 11] (1993).

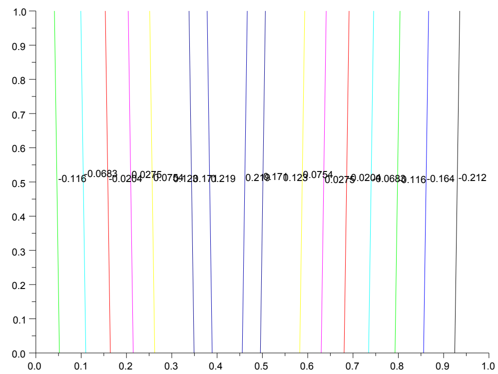

FIGURE 23 – The function C0.

This function is defined by

$$f(\mathbf{x}) = \exp\left(-\sum_{i=1}^{s} \alpha_i |x_i - \alpha_i|\right) - e, \tag{95}$$

where $\mathbf{x} \in [0,1]^s$ and e is defined by :

$$\begin{aligned} e &= \prod_{i=1}^{s} t_i, && (96) \\ t_i &= \left(2 - \exp(-\alpha_i \beta_i) - \exp(\alpha_i \beta_i - \alpha_i)\right) / \alpha_i. && (97) \end{aligned}$$

For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

In [14], Owen states that if the parameters α_i are large enough, this function is essentially full dimension s.

The contours of this function in the case $s = 2$ are presented in the figure 23.

3.21 DISCONT

The DISCONT function is designed by Genz in [6], [7], where it is named "Discontinuous". It is used by Petras in [17] (2003), by Owen in [14] (2003),

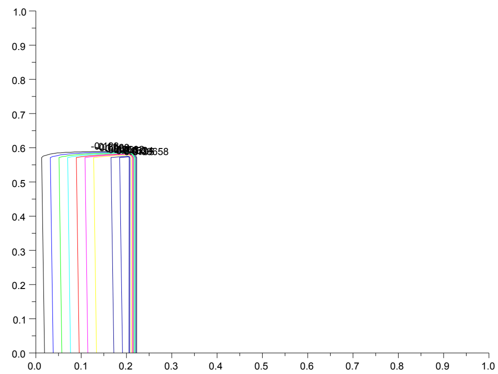

FIGURE 24 – The function DISCONT.

by Shürer in [23] (2001), by Joe and Sloan in [10, 11] (1993).

This function is defined by

$$f(\mathbf{x}) = \begin{cases} -e, \text{ if } x_1 > \beta_1, \text{ and } x_2 > \beta_2, \ldots, \text{ and } x_s > \beta_s, \\ \exp\left(\sum_{i=1}^s \alpha_i x_i\right) - e, \text{ if not.} \end{cases}, \tag{98}$$

where $\mathbf{x} \in [0,1]^s$ and e is defined by :

$$e = \prod_{i=1}^s \frac{\exp(\alpha_i \beta_i) - 1}{\alpha_i}. \tag{99}$$

For this function, the expectation is known, but not the variance. This function is not completely normalized, in the sense that its integral is zero buts its variance is not unity.

Notice that the simpler condition "$x_1 > \beta_1$ or $x_2 > \beta_2$" is used in [6].

In [14], considering the same function with the condition "$x_1 > \beta_1$ or $x_2 > \beta_2$", Owen states that if the parameters α_i are large enough, this function is essentially full dimension s.

The contours of this function in the case $s = 2$ are presented in the figure 24.

Références

[1] John Burkardt. Quadrature_test - quadrature rule applied to test integrals. http://people.sc.fsu.edu/~jburkardt/m_src/quadrature_test/quadrature_test.html.

[2] John Burkardt. Testpack - testing multidimensional integration routines. http://people.sc.fsu.edu/~jburkardt/m_src/testpack/testpack.html.

[3] J. Davis, Philip and Philip Rabinowitz. *Methods of Numerical Integration, Second Edition*. Academic Press, 1984.

[4] Bennett L. Fox. Algorithm 647 : Implementation and relative efficiency of quasirandom sequence generators. *ACM Trans. Math. Softw.*, 12(4) :362–376, 1986.

[5] Alan Genz. Multst test package with example test program and early adapt. http://www.math.wsu.edu/faculty/genz/software/fort77/testpack.f.

[6] Alan Genz. Testing multidimensional integration routines, in tools, methods, and languages for scientific and engineering computation, edited by b ford, jc rault, f thomasset. *North-Holland*, pages 81–94, 1984.

[7] Alan Genz. A package for testing multiple integration subroutines. In *in Numerical Integration : Recent Developments, Software and Applications, edited by Patrick Keast, Graeme Fairweather*, pages 337–340, 1987.

[8] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Chapman and Hall, 1964.

[9] P. Hellekalek. On the assessment of random and quasi-random point sets. *Random and Quasi-Random Point Sets of Lecture Notes in Statistics, P. Hellekalek and G. Larcher, editors*, 1998.

[10] Stephen Joe and Ian H. Sloan. Implementation of a lattice method for numerical multiple integration. *ACM Transactions on Mathematical Software*, 19(4) :523–545, December 1993.

[11] Stephen Joe and Ian H. Sloan. Corrigendum : "Implementation of a lattice method for numerical multiple integration". *ACM Transactions on Mathematical Software*, 20(2) :245–245, June 1994. See [10].

[12] Ladislav Kocis and William J. Whiten. Computational investigations of low-discrepancy sequences. *ACM Trans. Math. Softw.*, 23(2) :266–294, 1997.

[13] Giray Okten, Manan Shah, and Yevgeny Goncharov. Random and deterministic digit permutations of the halton sequence. 2009.

[14] A. B. Owen. The dimension distribution and quadrature test functions. *Statistica Sinica*, 13(1), 2003.

[15] Art B. Owen. Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25(4) :1541–1562, 1997.

[16] Art B. Owen. Scrambling sobol and niederreiter-xing points. *Journal of Complexity*, 14 :466–489, 1997.

[17] Knut Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numerische Mathematik*, 93(4), february 2003.

[18] Gottlieb Pirsic. Relative errors of numerical integration of genz test functions. http://www.ricam.oeaw.ac.at/people/page/pirsic/niedxing/netintlog.html, http://www.ricam.oeaw.ac.at/people/page/pirsic/niedxing/relerr3.html.

[19] Gottlieb Pirsic. A software implementation of niederreiter-xing sequences, 2001.

[20] Igor Radovic, Ilya M. Sobol', and Robert F. Tichy. Quasi-monte carlo methods for numerical integration : Comparison of different low discrepancy sequences. *Monte Carlo Methods and Applications*, 2 :1–14, 1996.

[21] P. Roos and L. Arnold. Numerische experimente zur mehrdimensionalen quadratur. *Random and Quasi-Random Point Sets of Lecture Notes in Statistics, P. Hellekalek and G. Larcher, editors*, pages 271–286, 1963.

[22] Linus Schrage. A more portable fortran random number generator. *ACM Trans. Math. Softw.*, 5(2) :132–138, 1979.

[23] Rudolf Schürer. Parallel high-dimensional integration : Quasi-monte carlo versus adaptive cubature rules. In *ICCS '01 : Proceedings of the International Conference on Computational Sciences-Part I*, pages 1262–1271, London, UK, 2001. Springer-Verlag.

[24] Ilya Sobol'. *A Primer for the Monte Carlo Method*. CRC Press, 1994.

[25] J. Taylor and F. Hover. High dimensional stochastic simulation and electric ship models. In *Electric Ship Technologies Symposium, 2007. ESTS '07. IEEE*, pages 402 –407, 21-23 2007.

[26] W. J. Whiten and L. Kocis. Regarding test functions for multi-dimensional integration. *SIGNUM Newsl.*, 30(1) :2–7, 1995.