

# **Construction of good polynomial lattice rules in weighted Walsh spaces by an alternative component-by-component construction**

**A. Ebert, P. Kritzer, O. Osiogun,  
T. Stepaniuk**

**RICAM-Report 2021-04**

# Construction of good polynomial lattice rules in weighted Walsh spaces by an alternative component-by-component construction

Adrian Ebert\*, Peter Kritzer, Onyekachi Osisiogu, Tetiana Stepaniuk†

January 26, 2021

## Abstract

We study the efficient construction of good polynomial lattice rules, which are special instances of quasi-Monte Carlo (QMC) methods. The integration rules obtained are of particular interest for the approximation of multivariate integrals in weighted Walsh spaces. In particular, we study a construction algorithm which assembles the components of the generating vector, which is in this case a vector of polynomials over a finite field, of the polynomial lattice rule in a component-wise fashion. We show that the constructed QMC rules achieve the almost optimal error convergence order in the function spaces under consideration and prove that the obtained error bounds can, under certain conditions on the involved weights, be made independent of the dimension. We also demonstrate that our alternative component-by-component construction, which is independent of the underlying smoothness of the function space, can be implemented relatively easily in a fast manner. Numerical experiments confirm our theoretical findings.

**Keywords:** Numerical integration; polynomial lattice points; quasi-Monte Carlo methods; weighted function spaces; component-by-component construction; fast implementation.

**2010 MSC:** 65D30, 65D32, 41A55, 41A63.

## 1 Introduction

We are interested in studying multivariate numerical integration, more precisely, we consider numerical integration of a sub-class of square-integrable functions  $f \in L^2([0, 1]^d)$ . For the approximation of the  $d$ -dimensional integrals we will use quasi-Monte Carlo (QMC) rules, which are equal-weight quadrature/cubature rules, that is,

$$I_d(f) := \int_{[0,1]^d} f(\mathbf{x}) \, d\mathbf{x} \approx Q_{N,d}(f) := \frac{1}{N} \sum_{n=0}^{N-1} f(\mathbf{x}_n),$$

where the quadrature/cubature points  $\mathbf{x}_0, \dots, \mathbf{x}_{N-1} \in [0, 1]^d$  are chosen in a deterministic way. This is in contrast to Monte Carlo rules, which are of the same form as QMC methods but are based on randomly chosen integration nodes. With QMC rules, we try to make a deliberate and sophisticated choice of the points  $\mathbf{x}_n$  with the aim of obtaining better error bounds than for Monte Carlo methods. Here, the key challenge is to systematically devise integration nodes that yield good approximation results simultaneously for a wide class of integrand functions that may depend on a large number of variables. Additionally, in order to obtain a low approximation error, we may need to construct millions of good integration nodes in very high dimensions, which presents a considerable computational challenge.

---

\*A. Ebert, P. Kritzer, and O. Osisiogu are supported by the Austrian Science Fund (FWF): Project F5506, which is part of the Special Research Program “Quasi-Monte Carlo Methods: Theory and Applications”.

†T. Stepaniuk is supported by the Alexander von Humboldt Foundation.

There are two main families of point sets for QMC methods which are commonly considered in the literature. These are, on the one hand, lattice point sets, as introduced independently by Korobov (see [9]) and Hlawka (see [8]). For more recent introductions to lattice rules, we refer to [13, 17]. The other class of commonly used QMC integration node sets is that of (digital)  $(t, m, d)$ -nets and  $(t, d)$ -sequences, as introduced by Niederreiter, building up on ideas by Sobol' and Faure (see [11, 13]). In this article, we will consider a special instance of  $(t, m, d)$ -nets, namely so-called polynomial lattice point sets. Originally introduced in [12], polynomial lattices owe their name to their construction principle which resembles that of (ordinary) lattice point sets. While the construction of lattice point sets is based on integer arithmetic, polynomial lattice point sets are based on polynomial arithmetic over finite fields  $\mathbb{F}_b$  with  $b$  elements, where  $b$  is prime. In particular, a polynomial lattice point set consists of  $b^m$  points in  $[0, 1]^d$  that are constructed by means of a modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , and a generating vector  $\mathbf{g} \in (\mathbb{F}_b[x])^d$  (we refer to Section 2.1 for the precise definition). The resulting QMC rule using the polynomial lattice point set as integration nodes is then called a polynomial lattice rule. In this article, we will use irreducible polynomials as the moduli of polynomial lattice rules.

We remark that not every choice of a generating vector  $\mathbf{g}$  yields a polynomial lattice rule with good approximation properties. In general, it is usually highly non-trivial to find good generating vectors of polynomial lattice rules, and there are (except for special cases) no explicit constructions of such good generating vectors known. Hence, one has to resort to computer search algorithms for finding generating vectors of polynomial lattice point sets with high quality. In this article, we study the worst-case setting, that is, we consider a particular normed function space and analyze the so-called worst-case error, which is the integration error of the considered QMC methods in the supremum over the unit ball of the space. The worst-case error will then serve as our quality measure for the constructed polynomial lattice rules.

It is known that polynomial lattice rules are well suited for the numerical integration of functions that can be represented by Walsh series (cf. [2, 4, 5]). Throughout this article, we will therefore consider a “weighted” function space (in the sense of Sloan and Woźniakowski (cf. [19])) which consists of elements whose Walsh coefficients decay sufficiently fast. The prescribed decay will be characterized by a smoothness parameter  $\alpha > 1$  (in some publications this parameter is also referred to as the “digital smoothness parameter” in the context of Walsh series). Indeed, the parameter  $\alpha$  is linked to the speed of decay of the Walsh coefficients of the functions in our space, but there is also a connection to the number of derivatives that exist for the elements of the space (we refer to [5] and the references therein for details). Furthermore, the varying importance of different subsets of variables will be modeled by a collection of positive reals  $\boldsymbol{\gamma} = (\gamma_u)_{u \subseteq \{1, \dots, d\}}$  which we refer to as weights. As pointed out in [19] and numerous other papers, this concept is justified by practical high-dimensional problems in which different coordinates may indeed have a very different degree of influence on the values of an integral. The weights will be incorporated in the inner product and norm of the function space in a suitable way. Using this setting, it is plausible that a nominally very high-dimensional problem may have a rather low “effective dimension”, i.e., only a certain, possibly small, part of the components has a significant influence on the integration problem and the error made by approximative algorithms.

The function space studied here is closely related to other function spaces considered in the literature, such as in [2, 4, 5], and results shown for the function space considered in the present paper immediately yield corresponding results for some of the Walsh spaces considered in these references. We refer to Section 2 below for further details.

The first efficient construction algorithm for good generating vectors of polynomial lattice point sets was introduced in [2], where a so-called component-by-component (CBC) approach was formulated and analyzed. Component-by-component constructions are greedy algorithms which construct the generating vector one component at a time. In the context of QMC integration, CBC algorithms were initially considered for (ordinary) lattice point sets, with the first examples in the literature going back to Korobov (cf. [10]), and later also Sloan and Reztsov (cf. [18]). The

CBC construction was then made widely applicable by the formulation of a fast construction algorithm. This fast CBC construction, which is due to Nuyens and Cools (see, e.g., [14–16]), makes the CBC algorithm computationally competitive and is currently the standard method to construct high-dimensional lattice point sets with good quality. It is well known (see, e.g., [2] and again [14]) that CBC constructions also work for the efficient search for generating vectors of polynomial lattice point sets; and also in this case a fast algorithm is available.

In the present paper, an alternative CBC algorithm for the efficient construction of generating vectors of polynomial lattice point sets will be formulated. Opposed to the standard CBC construction that uses the worst-case error of the considered function space as the quality measure, our algorithm is based on an alternative quality criterion, which is in particular independent of the parameter  $\alpha$ . We stress that therefore no prior knowledge of the smoothness parameter  $\alpha$  is required to construct the generating vector  $\mathbf{g}$ . The resulting generating vector will still achieve the almost optimal rate of convergence, for arbitrary values of the smoothness parameter  $\alpha > 1$ , and this result can be stated independently of the dimension  $d$ , assuming that the weights satisfy certain summability conditions. The standard CBC algorithms construct the generating vector specifically with the smoothness  $\alpha$  as an input parameter. We see the independence of  $\alpha$  in the construction algorithm presented in this paper as a big advantage of our new method.

The rest of the article is structured as follows. In Section 2, we define the concept of polynomial lattice rules, introduce the weighted function space under consideration, and analyze the corresponding worst-case error expression. Furthermore, we introduce the quality criterion on which our algorithm is based and show the existence of good polynomial lattice rules for our setting. In Section 3, a novel variant of the CBC construction algorithm for good polynomial lattice rules is proposed. We prove that the resulting polynomial lattice rules achieve the almost optimal order of convergence, with error bounds independent of the dimension assuming certain conditions on the weights are satisfied. In Section 4, we show that the introduced construction method can be implemented in a fast manner which is competitive with the state-of-the-art component-by-component algorithm. Finally, the paper is concluded in Section 5, where we illustrate our main result by numerical experiments.

For the remainder of the article we fix some basic notation. We denote the set of positive integers by  $\mathbb{N}$  and the set of non-negative integers by  $\mathbb{N}_0$ . To denote sets containing indices of components, we use fraktur font, e.g.,  $\mathbf{u} \subseteq \mathbb{N}$ , and additionally write  $\{1:d\} := \{1, \dots, d\}$  for short. For the projection of a vector  $\mathbf{x} \in [0, 1]^d$  or  $\mathbf{k} \in \mathbb{N}_0^d$  onto the components with indices in a set  $\mathbf{u} \subseteq \{1:d\}$ , we write  $\mathbf{x}_{\mathbf{u}} = (x_j)_{j \in \mathbf{u}}$  or  $\mathbf{k}_{\mathbf{u}} = (k_j)_{j \in \mathbf{u}}$ , respectively. With a slight abuse of notation, we will frequently identify elements of the finite field  $\mathbb{F}_b$  of prime cardinality  $b$  with elements of the group of integers modulo  $b$  denoted by  $\mathbb{Z}_b$ .

## 2 Preliminaries

In this section, we define polynomial lattice rules, introduce the function spaces under consideration, and then define the quality measure studied in this paper, for which we prove a first, non-constructive existence result.

### 2.1 Polynomial lattice point sets

For a prime  $b$ , let  $\mathbb{F}_b$  be the finite field with  $b$  elements and let  $\mathbb{F}_b((x^{-1}))$  be the field of formal Laurent series over  $\mathbb{F}_b$ . Elements of  $\mathbb{F}_b((x^{-1}))$  have the form

$$L = \sum_{\ell=w}^{\infty} t_{\ell} x^{-\ell},$$

where  $w$  is an arbitrary integer and all  $t_\ell \in \mathbb{F}_b$ . Furthermore, we denote by  $\mathbb{F}_b[x]$  the set of all polynomials over  $\mathbb{F}_b$ . For an integer  $m \in \mathbb{N}$  let the map  $v_m : \mathbb{F}_b((x^{-1})) \rightarrow [0, 1)$  be defined by

$$v_m \left( \sum_{\ell=w}^{\infty} t_\ell x^{-\ell} \right) = \sum_{\ell=\max(1,w)}^m t_\ell b^{-\ell}$$

and for a vector  $\mathbf{q} = (q_1, \dots, q_d) \in (\mathbb{F}_b((x^{-1})))^d$  set  $v_m(\mathbf{q}) = (v_m(q_1), \dots, v_m(q_d)) \in [0, 1)^d$ . Given an integer  $n \in \mathbb{N}_0$  with  $b$ -adic expansion  $n = n_0 + n_1 b + \dots + n_a b^a$ , we will frequently associate  $n$  with the polynomial

$$n(x) := \sum_{k=0}^a n_k x^k \in \mathbb{F}_b[x].$$

The definition of a polynomial lattice point set is as follows.

**Definition 1** (Polynomial lattice point set). Let  $b$  be prime and let  $m, d \in \mathbb{N}$  be given. Furthermore, choose  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , and let  $g_1, \dots, g_d \in \mathbb{F}_b[x]$ . Then the point set  $P(\mathbf{g}, p)$ , defined as the collection of the  $b^m$  points

$$\mathbf{x}_n := \left( v_m \left( \frac{n(x) g_1(x)}{p(x)} \right), \dots, v_m \left( \frac{n(x) g_d(x)}{p(x)} \right) \right) \in [0, 1)^d$$

for  $n \in \mathbb{F}_b[x]$  with  $\deg(n) < m$ , is called a *polynomial lattice point set* and the vector of polynomials  $\mathbf{g} = (g_1, \dots, g_d) \in (\mathbb{F}_b[x])^d$  is called the generating vector.

The point sets  $P(\mathbf{g}, p)$  which are considered in this paper are often called polynomial lattices and a QMC rule using  $P(\mathbf{g}, p)$  is referred to as a polynomial lattice rule. The polynomial  $p$  is referred to as the modulus. Without loss of generality, we can restrict the choice of the components  $g_j$  of the generating vector  $\mathbf{g}$  to the sets  $G_{b,m}$  or  $G_{b,m}^*$ , which are subsets of  $\mathbb{F}_b[x]$ , defined as

$$G_{b,m} := \{g \in \mathbb{F}_b[x] \mid \deg(g) < m\} \quad \text{and} \quad G_{b,m}^* := \{g \in \mathbb{F}_b[x] \setminus \{0\} \mid \deg(g) < m\}.$$

For two arbitrary vectors  $\mathbf{u}(x) = (u_1(x), \dots, u_d(x)), \mathbf{v}(x) = (v_1(x), \dots, v_d(x)) \in (\mathbb{F}_b[x])^d$  we define the inner product

$$\mathbf{u}(x) \cdot \mathbf{v}(x) := \sum_{j=1}^d u_j(x) v_j(x) \in \mathbb{F}_b[x],$$

and we write  $v(x) \equiv 0 \pmod{p(x)}$  if  $p(x)$  divides  $v(x)$  in  $\mathbb{F}_b[x]$ . For integer  $k \in \mathbb{N}_0$  with  $b$ -adic expansion  $k = \kappa_0 + \kappa_1 b + \dots + \kappa_{a-1} b^{a-1}$ , we define the truncation map  $\text{tr}_m : \mathbb{N}_0 \rightarrow G_{b,m}$  via

$$\text{tr}_m(k) := \kappa_0 + \kappa_1 x + \dots + \kappa_{m-1} x^{m-1}$$

with  $d$ -variate generalization  $\text{tr}_m(\mathbf{k})$  applied component-wise and additionally introduce the vector  $\tilde{\text{tr}}_m(\vec{k}) = (\kappa_0, \kappa_1, \dots, \kappa_{m-1}) \in \mathbb{F}_b^m$ . If  $a < m$ , then the  $\kappa_i$  with  $i > a$  are interpreted as being equal to zero.

Polynomial lattice point sets are special instances of so-called digital nets, which are constructed using linear algebra over finite fields or rings (see [13] for an introduction). Every digital net has a dual net, which plays a crucial role in the expression of the integration error of a QMC rule using the net. For a polynomial lattice point set with generating vector  $\mathbf{g}$  and modulus  $p$  with  $\deg(p) = m$ , its dual net (or, in this case, dual polynomial lattice)  $\mathcal{D}(\mathbf{g}, p)$  equals

$$\mathcal{D}(\mathbf{g}, p) = \{\mathbf{k} \in \mathbb{N}_0^d \mid \text{tr}_m(\mathbf{k}) \cdot \mathbf{g} \equiv 0 \pmod{p}\}. \quad (1)$$

Furthermore, for a set  $\mathbf{u} \subseteq \{1:d\}$  we introduce the notation

$$\mathcal{D}_{\mathbf{u}} = \mathcal{D}_{\mathbf{u}}(\mathbf{g}, p) = \mathcal{D}_{\mathbf{u}}(\mathbf{g}_{\mathbf{u}}) := \{\mathbf{k}_{\mathbf{u}} \in \mathbb{N}^{|\mathbf{u}|} \mid \text{tr}_m(\mathbf{k}_{\mathbf{u}}) \cdot \mathbf{g}_{\mathbf{u}} \equiv 0 \pmod{p}\}.$$

## 2.2 Walsh series representation

We consider numerical integration for square-integrable functions  $f \in L^2([0, 1]^d)$  which can be represented in terms of their Walsh series. This particular series representation of a function is based on the so-called Walsh functions which are given in the following definition.

**Definition 2.** Let  $b \geq 2$  be an integer. For a non-negative integer  $k$  with base  $b$  expansion  $k = \kappa_0 + \kappa_1 b + \dots + \kappa_{a-1} b^{a-1}$ , we define the  $k$ -th Walsh function  ${}_b \text{wal}_k : [0, 1] \rightarrow \mathbb{C}$  in base  $b$  by

$${}_b \text{wal}_k(x) := e^{2\pi i(\kappa_0 \xi_1 + \kappa_1 \xi_2 + \dots + \kappa_{a-1} \xi_a)/b},$$

where  $x \in [0, 1)$  is represented as  $x = \xi_1 b^{-1} + \xi_2 b^{-2} + \dots$  with coefficients  $\xi_i \in \{0, 1, \dots, b-1\}$  (unique in the sense that infinitely many of the  $\xi_i$  must be different from  $b-1$ ).

For  $d \in \mathbb{N}$ , an integer vector  $\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{N}_0^d$  and  $\mathbf{x} = (x_1, \dots, x_d) \in [0, 1]^d$ , we define the  $\mathbf{k}$ -th ( $d$ -variate) Walsh function  ${}_b \text{wal}_{\mathbf{k}} : [0, 1]^d \rightarrow \mathbb{C}$  in base  $b$  by

$${}_b \text{wal}_{\mathbf{k}}(\mathbf{x}) := \prod_{j=1}^d {}_b \text{wal}_{k_j}(x_j).$$

In the following, we will consider the base  $b \geq 2$  as fixed, and then simply write  $\text{wal}_k$  or  $\text{wal}_{\mathbf{k}}$  instead of  ${}_b \text{wal}_k$  or  ${}_b \text{wal}_{\mathbf{k}}$ , respectively. Note that for any function  $f \in L^2([0, 1]^d)$  the Walsh series of  $f$  is given by

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}), \quad (2)$$

where the  $\mathbf{k}$ -th Walsh coefficient of  $f$  is defined as

$$\hat{f}(\mathbf{k}) = \int_{[0,1]^d} f(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})} \, d\mathbf{x}.$$

For the numerical integration of functions given in terms of their Walsh series as in (2), it is common to consider quasi-Monte Carlo rules which are based on digital nets or their infinite counterparts, digital sequences. In this article, we will employ the special digital net type of polynomial lattice rules, as introduced in the previous section, in order to approximate integrals by QMC rules. It is well known, see, e.g., [3, Theorem 6.4], that approximating the integral  $I_d(f)$  using a QMC rule based on the polynomial lattice  $P(\mathbf{g}, p) = \{\mathbf{x}_0, \dots, \mathbf{x}_{N-1}\}$  with generating vector  $\mathbf{g} \in G_{b,m}^d$  and modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , i.e.,

$$Q_{b^m,d}(f, \mathbf{g}) = Q_{b^m,d}(f; P(\mathbf{g}, p)) := \frac{1}{b^m} \sum_{n=0}^{b^m-1} f(\mathbf{x}_n) \approx \int_{[0,1]^d} f(\mathbf{x}) \, d\mathbf{x} =: I_d(f),$$

leads to an integration error of the form

$$Q_{b^m,d}(f, \mathbf{g}) - I_d(f) = \sum_{\mathbf{0} \neq \mathbf{k} \in \mathcal{D}(\mathbf{g}, p)} \hat{f}(\mathbf{k}) \quad (3)$$

with  $\mathcal{D}(\mathbf{g}, p)$  as in (1), provided that  $f$  can be represented by a Walsh series as in (2). In particular, we will require in this paper that  $f$  lies in a function space referred to as a Walsh space, which we introduce next.

## 2.3 The weighted Walsh space $W_{d,\gamma}^\alpha$

In this section, we will introduce the function space under consideration in this paper, which consists of functions  $f$  that admit a representation as in (2) and for which the Walsh coefficients  $\hat{f}(\mathbf{k})$  decay at a prescribed rate. For this purpose, we will first define a decay function as follows.

For prime base  $b$  and a given real smoothness parameter  $\alpha > 1$ , we define the decay function  $r_\alpha : \mathbb{N}_0 \rightarrow \mathbb{R}$  by

$$r_\alpha(k) = r_\alpha(b, k) := \begin{cases} 1, & \text{if } k = 0, \\ b^{\alpha\psi_b(k)}, & \text{if } k \neq 0, \end{cases}$$

where we set  $\psi_b(k) := \lfloor \log_b(k) \rfloor$  for  $k \in \mathbb{N}$ . Additionally, we introduce the auxiliary quantity

$$\mu_b(\alpha) := \sum_{k=1}^{\infty} r_\alpha^{-1}(k) = \sum_{a=0}^{\infty} \frac{1}{b^{a\alpha}} \sum_{k=b^a}^{b^{a+1}-1} 1 = \sum_{a=0}^{\infty} \frac{(b-1)b^a}{b^{a\alpha}} = \frac{b^\alpha(b-1)}{b^\alpha - b}.$$

For an integer vector  $\mathbf{k} = (k_1, \dots, k_d) \in \mathbb{N}_0^d$  and positive weights  $\boldsymbol{\gamma} = (\gamma_u)_{u \subseteq \{1:d\}}$ , we define the (weighted) multivariate generalization of  $r_\alpha$  as

$$r_\alpha(\mathbf{k}) := \prod_{j=1}^d r_\alpha(k_j) \quad \text{and} \quad r_{\alpha, \boldsymbol{\gamma}}(\mathbf{k}) := \gamma_{\text{supp}(\mathbf{k})}^{-1} r_\alpha(\mathbf{k}) = \gamma_{\text{supp}(\mathbf{k})}^{-1} \prod_{j \in \text{supp}(\mathbf{k})} b^{\alpha\psi_b(k_j)}$$

with  $\text{supp}(\mathbf{k}) := \{j \in \{1:d\} \mid k_j \neq 0\}$ . As mentioned in the introduction, the weights  $\gamma_u$ , which are incorporated in the decay function (and thus in the definition of the function space), model the varying importance of subsets of variables  $\mathbf{x}_u = (x_j)_{j \in u}$  on the integration problem. The weights will play a crucial role in our effort to overcome an exponential dependence on the integration error of the QMC rules on the dimension  $d$ . This exponential dependence is sometimes also referred to as the curse of dimensionality.

The weighted Walsh space  $W_{d, \boldsymbol{\gamma}}^\alpha$  is then defined as the space of all functions  $f \in L^2([0, 1]^d)$  for which the norm  $\|f\|_{W_{d, \boldsymbol{\gamma}}^\alpha}$ , given by

$$\|f\|_{W_{d, \boldsymbol{\gamma}}^\alpha} := \sup_{\mathbf{k} \in \mathbb{N}_0^d} |\hat{f}(\mathbf{k})| r_{\alpha, \boldsymbol{\gamma}}(\mathbf{k})$$

is finite, that is,

$$W_{d, \boldsymbol{\gamma}}^\alpha := \{f \in L^2([0, 1]^d) \mid \|f\|_{W_{d, \boldsymbol{\gamma}}^\alpha} < \infty\},$$

where  $\alpha > 1$ , and where the weights  $\boldsymbol{\gamma} = (\gamma_u)_{u \subseteq \{1:d\}}$  are strictly positive. As pointed out in the introduction, the function space studied here is closely related to other Walsh spaces considered in the literature, such as in [2, 4, 5]. To be more precise, the spaces considered in these references have the norm given by  $\|f\|^2 = \sum_{\mathbf{k} \in \mathbb{N}_0^d} |\hat{f}(\mathbf{k})|^2 r_{\alpha, \boldsymbol{\gamma}}(\mathbf{k})$ . It can be shown easily that the worst-case error in these spaces is exactly the square root of the worst-case error in  $W_{d, \boldsymbol{\gamma}}^\alpha$ . Therefore the results shown here immediately imply results for the spaces considered in references like [2, 4, 5].

In this article, we will use the so-called worst-case error to assess the quality of integration rules. For the function space  $W_{d, \boldsymbol{\gamma}}^\alpha$ , the worst-case error of a QMC rule  $Q_{N, d}(\cdot; P)$  with underlying point set  $P \subseteq [0, 1]^d$  consisting of  $N$  points, is defined as

$$e_{N, d, \alpha, \boldsymbol{\gamma}}(P) := \sup_{\substack{f \in W_{d, \boldsymbol{\gamma}}^\alpha \\ \|f\|_{W_{d, \boldsymbol{\gamma}}^\alpha} \leq 1}} |Q_{N, d}(f; P) - I_d(f)|.$$

As was shown in [7], the worst-case error in  $W_{d, \boldsymbol{\gamma}}^\alpha$  of a polynomial lattice rule takes an explicit form.

**Theorem 1** (Worst-case error expression). *Let  $m, d \in \mathbb{N}$ , a real  $\alpha > 1$ , prime  $b$ , and positive weights  $\boldsymbol{\gamma} = (\gamma_u)_{u \subseteq \{1:d\}}$  be given. Then the worst-case error  $e_{b^m, d, \alpha, \boldsymbol{\gamma}}(P(\mathbf{g}, p))$  of a QMC rule based on the polynomial lattice  $P(\mathbf{g}, p)$  with generating vector  $\mathbf{g} \in (\mathbb{F}_b[x])^d$  and modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$  in the space  $W_{d, \boldsymbol{\gamma}}^\alpha$  satisfies*

$$e_{b^m, d, \alpha, \boldsymbol{\gamma}}(P(\mathbf{g}, p)) = \sum_{\mathbf{0} \neq \mathbf{k} \in \mathcal{D}(\mathbf{g}, p)} (r_{\alpha, \boldsymbol{\gamma}}(\mathbf{k}))^{-1},$$

where  $\mathcal{D}(\mathbf{g}, p)$  is defined as in (1).

Henceforth, we will denote the worst-case error  $e_{b^m, d, \alpha, \gamma}(P(\mathbf{g}, p))$  of a QMC rule that is based on the polynomial lattice  $P(\mathbf{g}, p)$  in the space  $W_{d, \gamma}^\alpha$  simply by  $e_{b^m, d, \alpha, \gamma}(\mathbf{g})$ .

The integration error for a single  $f \in W_{d, \gamma}^\alpha$  can then be directly related to the worst-case error of a polynomial lattice rule. Starting from the expression in (3), an application of Hölder's inequality with parameters 1 and  $\infty$  yields the estimate

$$\begin{aligned} |Q_{b^m, d}(f; P(\mathbf{g}, p)) - I_d(f)| &= \left| \sum_{\mathbf{0} \neq \mathbf{k} \in \mathcal{D}(\mathbf{g}, p)} \hat{f}(\mathbf{k}) \right| = \left| \sum_{\mathbf{0} \neq \mathbf{k} \in \mathbb{N}_0^d} \hat{f}(\mathbf{k}) r_{\alpha, \gamma}(\mathbf{k}) (r_{\alpha, \gamma}(\mathbf{k}))^{-1} \mathbf{1}_{\mathcal{D}(\mathbf{g}, p)}(\mathbf{k}) \right| \\ &\leq \left( \sup_{\mathbf{k} \in \mathbb{N}_0^d} |\hat{f}(\mathbf{k})| r_{\alpha, \gamma}(\mathbf{k}) \right) \left( \sum_{\mathbf{0} \neq \mathbf{k} \in \mathcal{D}(\mathbf{g}, p)} (r_{\alpha, \gamma}(\mathbf{k}))^{-1} \right) \\ &= \|f\|_{W_{d, \gamma}^\alpha} e_{b^m, d, \alpha, \gamma}(\mathbf{g}) \end{aligned}$$

with  $\mathbf{1}_{\mathcal{D}(\mathbf{g}, p)}$  denoting the indicator function of the dual net  $\mathcal{D}(\mathbf{g}, p)$ . Therefore, it becomes evident that we need to construct polynomial lattice rules such that the associated worst-case error  $e_{b^m, d, \alpha, \gamma}(\mathbf{g})$  is as small as possible.

## 2.4 The existence of good polynomial lattice rules

In this section, we study the existence of polynomial lattice rules with a small worst-case error which satisfies a certain asymptotic decay with respect to the number of cubature nodes. Here, we introduce an alternative quality measure which, opposed to the worst-case error expression obtained in Theorem 1, is independent of the parameter  $\alpha$ . To this end, we extend the definition of the function  $r_{\alpha, \gamma}$  to the case where we allow  $\alpha = 1$ , i.e.,

$$r_{1, \gamma}(\mathbf{k}) := \gamma_{\text{supp}(\mathbf{k})}^{-1} \prod_{j \in \text{supp}(\mathbf{k})} b^{\lfloor \log_b(k_j) \rfloor} \quad \text{for } \mathbf{k} \in \mathbb{N}_0^d.$$

For given  $\alpha \geq 1$ , given positive weights  $\gamma = (\gamma_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$ , modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , and  $\mathbf{g} \in (\mathbb{F}_b[x])^d$ , we introduce the quantities

$$T_\gamma(\mathbf{g}, p) := \sum_{\mathbf{0} \neq \mathbf{k} \in A_p(\mathbf{g})} (r_{1, \gamma}(\mathbf{k}))^{-1}, \quad \text{and} \quad T_{\alpha, \gamma}(\mathbf{g}, p) := \sum_{\mathbf{0} \neq \mathbf{k} \in A_p(\mathbf{g})} (r_{\alpha, \gamma}(\mathbf{k}))^{-1} \quad (4)$$

with index set  $A_p(\mathbf{g})$  given by

$$A_p(\mathbf{g}) := \{\mathbf{k} \in \{0, 1, \dots, b^m - 1\}^d \mid \mathbf{k} \in \mathcal{D}(\mathbf{g}, p)\}.$$

Furthermore, for a set  $\emptyset \neq \mathbf{u} \subseteq \{1:d\}$ , we introduce the sets

$$\begin{aligned} A_{\mathbf{u}} &= A_{p, \mathbf{u}}(\mathbf{g}) = A_{p, \mathbf{u}}(\mathbf{g}_{\mathbf{u}}) := \{\mathbf{k}_{\mathbf{u}} \in \{0, 1, \dots, b^m - 1\}^{|\mathbf{u}|} \mid \mathbf{k}_{\mathbf{u}} \in \mathcal{D}_{\mathbf{u}}(\mathbf{g}, p)\}, \\ A_{\mathbf{u}}^* &= A_{p, \mathbf{u}}^*(\mathbf{g}) = A_{p, \mathbf{u}}^*(\mathbf{g}_{\mathbf{u}}) := \{\mathbf{k}_{\mathbf{u}} \in \{1, \dots, b^m - 1\}^{|\mathbf{u}|} \mid \mathbf{k}_{\mathbf{u}} \in \mathcal{D}_{\mathbf{u}}(\mathbf{g}, p)\}, \end{aligned}$$

and for the modulus  $p \in \mathbb{F}_b[x]$  we define the indicator function  $\delta_p : \mathbb{F}_b[x] \rightarrow \{0, 1\}$  by

$$\delta_p(q) := \begin{cases} 1 & \text{if } q \equiv 0 \pmod{p}, \\ 0 & \text{if } q \not\equiv 0 \pmod{p}. \end{cases}$$

We can then prove the following existence result for good generating vectors  $\mathbf{g}$  with respect to the quality measure  $T_\gamma(\mathbf{g}, p)$ .



**Theorem 2** (Existence result w.r.t.  $T_\gamma$ ). *Let  $\gamma = (\gamma_u)_{u \subseteq \{1:d\}}$  be positive weights. For every irreducible  $p \in \mathbb{F}_b[x]$  of degree  $m \in \mathbb{N}$  there exists a generating vector  $\mathbf{g} \in G_{b,m}^d$  such that*

$$T_\gamma(\mathbf{g}, p) = \sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{\delta_p(\text{tr}_m(\mathbf{k}) \cdot \mathbf{g})}{r_{1,\gamma}(\mathbf{k})} \leq \frac{1}{b^m} \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u (m(b-1))^{|u|}.$$

*Proof.* Let  $\mathbf{0} \neq \mathbf{k} \in \mathbb{N}_0^d$  be such that there exists  $j \in \{1, \dots, d\}$  with  $\text{tr}_m(k_j) \neq 0$ , and assume without loss of generality that  $j = d$ . Then we find that

$$\sum_{\mathbf{g} \in G_{b,m}^d} \delta_p(\text{tr}_m(\mathbf{k}) \cdot \mathbf{g}) = \sum_{\mathbf{g}_{\{1:d-1\}} \in G_{b,m}^{d-1}} \sum_{g_d \in G_{b,m}} \delta_p(\text{tr}_m(\mathbf{k}_{\{1:d-1\}}) \cdot \mathbf{g}_{\{1:d-1\}} + \text{tr}_m(k_d) g_d) = (b^m)^{d-1}, \quad (5)$$

which follows due to the fact that

$$\sum_{g_d \in G_{b,m}} \delta_p(\text{tr}_m(\mathbf{k}_{\{1:d-1\}}) \cdot \mathbf{g}_{\{1:d-1\}} + \text{tr}_m(k_d) g_d) = 1,$$

since for  $p$  irreducible  $\mathbb{F}_b[x]/(p)$  is a finite field and thus

$$\text{tr}_m(k_1)g_1 + \dots + \text{tr}_m(k_{d-1})g_{d-1} + \text{tr}_m(k_d)g_d \equiv 0 \pmod{p}$$

has  $g_d \equiv -\text{tr}_m(k_d)^{-1}(\text{tr}_m(k_1)g_1 + \dots + \text{tr}_m(k_{d-1})g_{d-1}) \pmod{p}$  as the unique solution in  $G_{b,m}$ .

By the standard averaging argument, which implies that there is always at least one element in a set of real numbers which is as good as average, there exists a  $\bar{\mathbf{g}} \in G_{b,m}^d$  which satisfies

$$\begin{aligned} T_\gamma(\bar{\mathbf{g}}, p) &= \min_{\mathbf{g} \in G_{b,m}^d} T_\gamma(\mathbf{g}, p) \leq \frac{1}{(b^m)^d} \sum_{\mathbf{g} \in G_{b,m}^d} T_\gamma(\mathbf{g}, p) \\ &= \frac{1}{(b^m)^d} \sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{1}{r_{1,\gamma}(\mathbf{k})} \sum_{\mathbf{g} \in G_{b,m}^d} \delta_p(\text{tr}_m(\mathbf{k}) \cdot \mathbf{g}) \\ &= \frac{1}{b^m} \sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{1}{r_{1,\gamma}(\mathbf{k})}, \end{aligned} \quad (6)$$

where we used again the identity in (5). Note that we can write

$$\begin{aligned} \sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{1}{r_{1,\gamma}(\mathbf{k})} &= \sum_{\emptyset \neq u \subseteq \{1:d\}} \sum_{\mathbf{k}_u \in \{1,\dots,b^m-1\}^{|u|}} \frac{1}{r_{1,\gamma}(\mathbf{k}_u)} = \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u \sum_{\mathbf{k}_u \in \{1,\dots,b^m-1\}^{|u|}} \prod_{j \in u} b^{-\psi_b(k_j)} \\ &= \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u \prod_{j \in u} \sum_{k_j \in \{1,\dots,b^m-1\}} b^{-\psi_b(k_j)} = \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u \left( \sum_{k=1}^{b^m-1} b^{-\psi_b(k)} \right)^{|u|}. \end{aligned}$$

The inner sum of the latter term equals

$$\sum_{k=1}^{b^m-1} b^{-\psi_b(k)} = \sum_{t=0}^{m-1} \sum_{k=b^t}^{b^{t+1}-1} b^{-\lfloor \log_b k \rfloor} = \sum_{t=0}^{m-1} (b-1)b^t b^{-t} = \sum_{t=0}^{m-1} (b-1) = m(b-1),$$

such that we obtain

$$\sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{1}{r_{1,\gamma}(\mathbf{k})} = \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u (m(b-1))^{|u|}.$$

Combining this with (6) yields the existence of a good generating vector in  $G_{b,m}^d$  as claimed.  $\square$

The following proposition, which was proved in [7], gives a bound on the difference between the worst-case error  $e_{b^m, d, \alpha, \gamma}(\mathbf{g})$  and the truncated quality measure  $T_{\alpha, \gamma}(\mathbf{g}, p)$  of a polynomial lattice rule with generator  $\mathbf{g}$  and modulus  $p$ .

**Proposition 1.** *Let  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , let  $\gamma = (\gamma_u)_{u \subseteq \{1:d\}}$  be positive weights, and let  $\mathbf{g} = (g_1, \dots, g_d) \in G_{b,m}^d$  such that  $\gcd(g_j, p) = 1$  for all  $j \in \{1, \dots, d\}$ . Then, for any  $\alpha > 1$  and for  $N = b^m$ , we have*

$$e_{b^m, d, \alpha, \gamma}(\mathbf{g}) - T_{\alpha, \gamma}(\mathbf{g}, p) \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u (2\mu_b(\alpha))^{|u|}.$$

Combining the results of Theorem 2 and Proposition 1, we can immediately deduce the existence of good polynomial lattice rules with respect to the worst-case error in the space  $W_{d, \gamma}^\alpha$ . We state the corresponding result, which first appeared in [7], in the theorem below.

**Theorem 3.** *Let  $p \in \mathbb{F}_b[x]$  be an irreducible polynomial with  $\deg(p) = m$ , let  $N = b^m$ , and let  $\gamma = (\gamma_u)_{u \subseteq \{1:d\}}$  be a sequence of positive weights. Then there exists a  $\mathbf{g} \in G_{b,m}^d$  such that, for all  $\alpha > 1$ , the worst-case error  $e_{b^m, d, \alpha, \gamma}(\mathbf{g})$  satisfies*

$$e_{b^m, d, \alpha, \gamma}(\mathbf{g}) \leq \frac{1}{N^\alpha} \left( \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u (2\mu_b(\alpha))^{|u|} + \left( \sum_{\emptyset \neq u \subseteq \{1:d\}} \gamma_u^{1/\alpha} (m(b-1))^{|u|} \right)^\alpha \right).$$

While this result is interesting from a theoretical perspective, the involved argument is non-constructive. In the remainder of this article, we will devise explicit algorithms for the construction of good polynomial lattice rules.

### 3 The CBC construction method for polynomial lattice rules

In this section, we formulate and analyze a component-by-component search algorithm for the construction of good polynomial lattice rules. The advantage of the presented algorithm is that the involved quality criterion is independent of the smoothness parameter  $\alpha$ .

#### 3.1 Preliminary results

At first, we summarize a number of auxiliary statements which will be needed in the following analysis. The lemmas are taken from [6] and [7], respectively, where construction algorithms of a similar nature as the proposed method have been studied.

Consider at first the following Walsh series for  $x \in (0, 1)$ , based on the decay function  $r_1$ ,

$$\sum_{k=0}^{\infty} \frac{\text{wal}_k(x)}{r_1(k)} = 1 + \sum_{k=1}^{\infty} \frac{e^{2\pi i(\kappa_{0,k}\xi_1 + \kappa_{1,k}\xi_2 + \dots)}/b}{b^{\lfloor \log_b(k) \rfloor}},$$

which, as we shall see, is closely related to the quality measure  $T_\gamma$  in (4). Here and in the following lemma, we use similar base  $b$  expansions as introduced above, namely  $k = \kappa_{0,k} + \kappa_{1,k}b + \dots + \kappa_{a,k}b^a$  for  $k \in \mathbb{N}_0$  and  $x = \xi_1 b^{-1} + \xi_2 b^{-2} + \dots$  for  $x \in [0, 1)$ .

**Lemma 1.** *For base  $b \geq 2$ , the Walsh series of  $-(b-1)(\lfloor \log_b(x) \rfloor + 1)$  equals, pointwise for  $x \in (0, 1)$ ,*

$$-(b-1)(\lfloor \log_b(x) \rfloor + 1) = 1 + \sum_{k=1}^{\infty} \frac{e^{2\pi i(\kappa_{0,k}\xi_1 + \kappa_{1,k}\xi_2 + \dots)}/b}{b^{\lfloor \log_b(k) \rfloor}} = \sum_{k=0}^{\infty} \frac{\text{wal}_k(x)}{r_1(k)}.$$

Based on the result in Lemma 1, it was shown in [7] that the function  $-(b-1)(\lfloor \log_b(x) \rfloor + 1)$  can be written in terms of its truncated Walsh series with uniformly bounded remainder term.

**Lemma 2.** *Let  $N = b^m$  with  $m \in \mathbb{N}$  and base  $b \geq 2$ . Then for any  $x \in (0, 1)$  there exists a  $\tau(x) \in \mathbb{R}$  with  $|\tau(x)| < \frac{b}{b-1}$  such that*

$$-(b-1)(\lfloor \log_b(x) \rfloor + 1) = \sum_{k=0}^{N-1} \frac{\text{wal}_k(x)}{r_1(k)} + \frac{\tau(x)}{Nx}.$$

We will also make use of the following lemma, which was shown in [6].

**Lemma 3.** *For  $1 \leq j \leq d$ , let  $u_j, v_j$ , and  $\rho_j$  be real numbers satisfying*

$$(a) \quad u_j = v_j + \rho_j, \quad (b) \quad |u_j| \leq \bar{u}_j, \quad (c) \quad \bar{u}_j \geq 1,$$

*for all  $j \in \{1:d\}$ . Then, for any set  $\emptyset \neq \mathbf{u} \subseteq \{1:d\}$ , there exists a  $\theta_{\mathbf{u}}$  with  $|\theta_{\mathbf{u}}| \leq 1$  such that*

$$\prod_{j \in \mathbf{u}} u_j = \prod_{j \in \mathbf{u}} v_j + \theta_{\mathbf{u}} \left( \prod_{j \in \mathbf{u}} (\bar{u}_j + |\rho_j|) \right) \sum_{j \in \mathbf{u}} |\rho_j|.$$

Furthermore, we recall the character property of Walsh functions for polynomial lattice rules with prime base  $b$ . Let  $P(\mathbf{g}, p) = \{\mathbf{x}_0, \dots, \mathbf{x}_{b^m-1}\} \subset [0, 1]^d$  be a polynomial lattice with generating vector  $\mathbf{g} \in (\mathbb{F}_b[x])^d$  and modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ . Then, for a vector  $\mathbf{k} \in \mathbb{N}_0^d$ , the following identity holds:

$$\frac{1}{b^m} \sum_{n=0}^{b^m-1} \text{wal}_{\mathbf{k}}(\mathbf{x}_n) = \delta_p(\text{tr}_m(\mathbf{k}) \cdot \mathbf{g}) = \begin{cases} 1, & \text{if } \text{tr}_m(\mathbf{k}) \cdot \mathbf{g} \equiv 0 \pmod{p}, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

We remark that an analogous result to (7) also holds if we only consider projections of the polynomial lattice and the generating vector onto a non-empty subset  $\mathbf{u} \subseteq \{1:d\}$ , as also the projection of a polynomial lattice is a polynomial lattice that is generated by the corresponding projection of the generating vector.

Additionally, the following two results will be helpful for the subsequent analysis.

**Lemma 4.** *Let  $P(\mathbf{g}, p) \subset [0, 1]^d$  be a polynomial lattice with modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$  and generating vector  $\mathbf{g} = (g_1, \dots, g_d) \in (\mathbb{F}_b[x])^d$  such that  $\gcd(g_j, p) = 1$  for  $1 \leq j \leq d$ . Then each one-dimensional projection of  $P(\mathbf{g}, p)$  is the full grid*

$$\left\{ 0, \frac{1}{b^m}, \dots, \frac{b^m - 1}{b^m} \right\},$$

*and in particular the projection of the point with index 0 is always 0.*

*Proof.* The result follows from Definition 1 and [5, Remark 10.3]. □

**Lemma 5.** *Let  $P(\mathbf{g}, p) = \{\mathbf{x}_0, \dots, \mathbf{x}_{b^m-1}\} \subset [0, 1]^d$  be a polynomial lattice point set with modulus  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$  and generating vector  $\mathbf{g} \in (\mathbb{F}_b[x])^d$  such that  $\gcd(g_j, p) = 1$  for  $1 \leq j \leq d$ . Furthermore, let  $m \geq 4$ . For each point  $\mathbf{x}_n$  with  $n = 0, 1, \dots, b^m - 1$ , we denote its coordinates by  $\mathbf{x}_n = (x_{n,1}, \dots, x_{n,d})$ . Then, for any  $j \in \{1, \dots, d\}$ , it is true that*

$$\frac{1}{b^m} \sum_{n=1}^{b^m-1} \frac{1}{x_{n,j}} < 1 + m \log b \leq m(b-1).$$

### 3.2 An alternative CBC construction for polynomial lattice rules

In this section, we study a component-by-component (CBC) construction for polynomial lattice rules which is based on a quality criterion related to the quantity  $T_\gamma$ . In [6] such an algorithm was analyzed for (ordinary) lattice rules. Throughout this section, we will assume that the modulus  $p \in \mathbb{F}_b[x]$  is irreducible.

Concerning the weights, our algorithm can, as indicated in one of our main results (Corollary 1), be run with respect to the weights  $\gamma^{1/\alpha} = (\gamma_u^{1/\alpha})_{u \subseteq \{1:d\}}$  to obtain a polynomial lattice rule that yields a low worst-case error in the Walsh space  $W_{d,\gamma}^\alpha$ , or, alternatively, with respect to the weights  $\gamma$  to obtain good polynomial lattice rules in the space  $W_{d,\gamma^\alpha}^\alpha$ . In the latter case, the construction algorithm is independent of the smoothness parameter  $\alpha$  and we obtain worst-case error bounds that hold simultaneously for all  $\alpha > 1$ .

In order to avoid confusion, we will therefore denote the weights in this section by  $\boldsymbol{\eta}$  instead of  $\gamma$  and outline the algorithm based on  $\boldsymbol{\eta}$ . In Corollary 1, we will then choose  $\boldsymbol{\eta}$  equal to  $\gamma^{1/\alpha}$  or  $\gamma$ , respectively. Before we formulate the algorithm, we prove the following theorem.

**Theorem 4.** *Let  $b$  be prime, let  $m, d \in \mathbb{N}$  with  $m \geq 4$ , let  $p \in \mathbb{F}_b[x]$  be an irreducible polynomial with  $\deg(p) = m$ , and let  $\boldsymbol{\eta} = (\eta_u)_{u \subseteq \{1:d\}}$  be positive weights. Furthermore, let  $\mathbf{g} = (g_1, \dots, g_d) \in G_{b,m}^d$  be the generating vector of the polynomial lattice  $P(\mathbf{g}, p) = \{\mathbf{x}_0, \dots, \mathbf{x}_{b^m-1}\} \subset [0, 1]^d$ . Then the following estimate holds:*

$$\begin{aligned} T_\eta(\mathbf{g}, p) &\leq \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} \sum_{n=1}^{b^m-1} L_u \left( v_m \left( \frac{n(x) \mathbf{g}_u(x)}{p(x)} \right) \right) + \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} ((b-1)m)^{|u|} \\ &\quad + \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} (bm)^{|u|} \left( (b-1)m + \frac{b}{b-1} \right)^{|u|}, \end{aligned}$$

where for a subset  $\emptyset \neq u \subseteq \{1:d\}$ , we define the function  $L_u : (0, 1]^{|u|} \rightarrow \mathbb{R}$  by

$$L_u(\mathbf{x}_u) := \prod_{j \in u} ((1-b) \lfloor \log_b(x_j) \rfloor - b) = \sum_{\mathbf{k}_u \in \mathbb{N}^{|u|}} \frac{\text{wal}_{\mathbf{k}_u}(\mathbf{x}_u)}{\prod_{j \in u} b^{\lfloor \log_b(k_j) \rfloor}}.$$

*Proof.* Using the character property of Walsh functions in (7), we can rewrite  $T_\eta(\mathbf{g}, p)$  with the help of the identity in Lemma 2. First, recall that we have, for  $k \in \mathbb{N}_0$ ,

$$r_1(k) = \begin{cases} 1, & \text{if } k = 0, \\ b^{\lfloor \log_b(k) \rfloor}, & \text{if } k \neq 0. \end{cases}$$

For a point  $\mathbf{x}_n$ ,  $0 \leq n \leq b^m - 1$ , and a set  $\emptyset \neq u \subseteq \{1:d\}$ , we write  $\mathbf{x}_{n,u}$  to denote the projection of  $\mathbf{x}_n$  onto the components with indices in  $u$ . Then, we obtain that

$$\begin{aligned} T_\eta(\mathbf{g}, p) &= \sum_{\mathbf{0} \neq \mathbf{k} \in \{0,1,\dots,b^m-1\}^d} \frac{\delta_p(\text{tr}_m(\mathbf{k}) \cdot \mathbf{g})}{r_{1,\eta}(\mathbf{k})} \\ &= \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} \sum_{n=0}^{b^m-1} \left[ \sum_{\mathbf{k}_u \in \{1,\dots,b^m-1\}^{|u|}} \frac{\text{wal}_{\mathbf{k}_u}(\mathbf{x}_{n,u})}{\prod_{j \in u} b^{\lfloor \log_b(k_j) \rfloor}} \right] \\ &= \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} \left( \sum_{\mathbf{k}_u \in \{1,\dots,b^m-1\}^{|u|}} \frac{1}{\prod_{j \in u} b^{\lfloor \log_b(k_j) \rfloor}} + \sum_{n=1}^{b^m-1} \prod_{j \in u} \left[ \sum_{k=1}^{b^m-1} \frac{\text{wal}_k(x_{n,j})}{b^{\lfloor \log_b(k) \rfloor}} \right] \right) \\ &= \sum_{\emptyset \neq u \subseteq \{1:d\}} \frac{\eta_u}{b^m} \left( \sum_{\mathbf{k}_u \in \{1,\dots,b^m-1\}^{|u|}} \frac{1}{\prod_{j \in u} b^{\lfloor \log_b(k_j) \rfloor}} + \sum_{n=1}^{b^m-1} \left[ \prod_{j \in u} v_j(n) - \prod_{j \in u} u_j(n) + \prod_{j \in u} u_j(n) \right] \right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{\mathbf{k}_{\mathbf{u}} \in \{1, \dots, b^{m-1}\}^{|\mathbf{u}|}} \frac{1}{\prod_{j \in \mathbf{u}} b^{\lfloor \log_b(k_j) \rfloor}} + \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} \prod_{j \in \mathbf{u}} u_j(n) \\
&\quad - \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} \theta_{\mathbf{u}}(n) \left( \prod_{j \in \mathbf{u}} (\bar{u}_j + |\rho_j(n)|) \right) \sum_{j \in \mathbf{u}} |\rho_j(n)| \\
&= \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} L_{\mathbf{u}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{u}}(x)}{p(x)} \right) \right) + \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{\mathbf{k}_{\mathbf{u}} \in \{1, \dots, b^{m-1}\}^{|\mathbf{u}|}} \frac{1}{\prod_{j \in \mathbf{u}} b^{\lfloor \log_b(k_j) \rfloor}} \\
&\quad - \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} \theta_{\mathbf{u}}(n) \left( \prod_{j \in \mathbf{u}} (\bar{u}_j + |\rho_j(n)|) \right) \sum_{j \in \mathbf{u}} |\rho_j(n)|, \tag{8}
\end{aligned}$$

where we used that  $x_{0,j} = 0$  for all  $1 \leq j \leq d$  and Lemma 3 with

$$\begin{aligned}
u_j &= u_j(n) := -(b-1)(\lfloor \log_b(x_{n,j}) \rfloor + 1) - 1, & \bar{u}_j &= \bar{u}_j(n) := (b-1)m, \\
v_j &= v_j(n) := \sum_{k=1}^{b^m-1} \frac{\text{walk}(x_{n,j})}{b^{\lfloor \log_b(k) \rfloor}}, & \rho_j &= \rho_j(n) := \frac{\tau_j(n)}{x_{n,j} b^m},
\end{aligned}$$

with  $|\theta_{\mathbf{u}}(n)| \leq 1$  and  $\tau_j(n) = \tau_j(x_{n,j})$  with  $|\tau_j(n)| \leq \frac{b}{b-1}$  for all  $n \in \{0, 1, \dots, b^m - 1\}$ . Due to Lemma 2, Condition (a) of Lemma 3 is fulfilled. Furthermore, we recall that by Lemma 4 the one-dimensional projections of the points  $\mathbf{x}_n$  with indices  $n \geq 1$  satisfy

$$x_{n,j} = v_m \left( \frac{n(x) g_j(x)}{p(x)} \right) \geq v_m \left( \frac{1}{x^m} \right) = b^{-m}$$

for all  $1 \leq n \leq b^m - 1$  and all  $1 \leq j \leq d$  such that we obtain

$$|u_j(n)| \leq -(b-1)(\lfloor \log_b(b^{-m}) \rfloor + 1) = -(b-1)(-m+1) < (b-1)m = \bar{u}_j$$

with  $\bar{u}_j \geq 1$  and so Conditions (b) and (c) of Lemma 3 are also fulfilled.

As in the proof of Theorem 2, we can express the second sum in (8) as

$$\sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{\mathbf{k}_{\mathbf{u}} \in \{1, \dots, b^{m-1}\}^{|\mathbf{u}|}} \frac{1}{\prod_{j \in \mathbf{u}} b^{\lfloor \log_b(k_j) \rfloor}} = \frac{1}{b^m} \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|},$$

while the absolute value of the third sum in (8) can be bounded as follows,

$$\begin{aligned}
&\left| \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} \theta_{\mathbf{u}}(n) \left( \prod_{j \in \mathbf{u}} (\bar{u}_j + |\rho_j(n)|) \right) \sum_{j \in \mathbf{u}} |\rho_j(n)| \right| \\
&= \left| \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} \theta_{\mathbf{u}}(n) \left( \prod_{j \in \mathbf{u}} \left( (b-1)m + \frac{|\tau_j(n)|}{x_{n,j} b^m} \right) \right) \sum_{j \in \mathbf{u}} \frac{|\tau_j(n)|}{x_{n,j} b^m} \right| \\
&\leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} |\theta_{\mathbf{u}}(n)| \left( \prod_{j \in \mathbf{u}} \left( (b-1)m + \frac{b}{b-1} \right) \right) \sum_{j \in \mathbf{u}} \frac{|\tau_j(n)|}{x_{n,j} b^m} \\
&\leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \left( \prod_{j \in \mathbf{u}} \left( (b-1)m + \frac{b}{b-1} \right) \right) \sum_{j \in \mathbf{u}} \sum_{n=1}^{b^m-1} \frac{b}{(b-1)b^m} \frac{1}{x_{n,j}} \\
&\leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \left( \prod_{j \in \mathbf{u}} \left( (b-1)m + \frac{b}{b-1} \right) \right) \frac{b}{b-1} \sum_{j \in \mathbf{u}} m(b-1)
\end{aligned}$$

$$= \frac{1}{b^m} \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} (b^m |\mathbf{u}|) \left( (b-1)m + \frac{b}{b-1} \right)^{|\mathbf{u}|},$$

where we used Lemma 5 and the fact that  $x_{n,j} \geq b^{-m}$  for each  $j$  and all  $1 \leq n < b^m$ . Then combining the estimates yields the claimed result.  $\square$

Theorem 4 indicates that it is reasonable to search for generating vectors  $\mathbf{g}$  such that

$$\sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} \sum_{n=1}^{b^m-1} L_{\mathbf{u}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{u}}(x)}{p(x)} \right) \right)$$

is small, as then also  $T_{\boldsymbol{\eta}}(\mathbf{g}, p)$  is sufficiently small. We will now give the definition of the quality function that we try to minimize in the component-by-component algorithm.

**Definition 3** (Quality Function). For a generating vector  $\mathbf{g} = (g_1, \dots, g_d) \in (G_{b,m}^*)^d$  with prime  $b \geq 2$ , positive integers  $m, d \in \mathbb{N}$ , positive weights  $\boldsymbol{\eta} = (\eta_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$ , and an irreducible polynomial  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ , we define the quality function

$$K_{b^m, d, \boldsymbol{\eta}}(\mathbf{g}) = K_{b^m, d, \boldsymbol{\eta}}(\mathbf{g}, p) := \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} \sum_{n=1}^{b^m-1} L_{\mathbf{u}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{u}}(x)}{p(x)} \right) \right).$$

Based on this quality function, we formulate the following component-by-component construction.

---

**Algorithm 1** Component-by-component construction

---

**Input:** Prime number  $b$ , integers  $m, d \in \mathbb{N}$ , and positive weights  $\boldsymbol{\eta} = (\eta_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$ .

Choose an irreducible polynomial  $p \in \mathbb{F}_b[x]$  with  $\deg(p) = m$ .

Set  $g_1 \equiv 1$ .

**for**  $s = 2$  **to**  $d$  **do**

$$g_s = \operatorname{argmin}_{g \in G_{b,m}^*} K_{b^m, s, \boldsymbol{\eta}}(g_1, \dots, g_{s-1}, g)$$

**end for**

**Return:** Generating vector  $\mathbf{g} = (g_1, \dots, g_d) \in (G_{b,m}^*)^d$ .

---

In the following section, we analyze the worst-case error behavior of polynomial lattice rules with generating vectors constructed by Algorithm 1.

### 3.3 Error bounds for the obtained polynomial lattice rules

In the next theorem, we show an upper bound on the quantity  $K_{b^m, d, \boldsymbol{\eta}}(\mathbf{g})$  for  $\mathbf{g}$  constructed by Algorithm 1.

**Theorem 5.** *Let  $b$  be prime, let  $m, d \in \mathbb{N}$ , let  $p \in \mathbb{F}_b[x]$  be irreducible with  $\deg(p) = m$ , and let  $\boldsymbol{\eta} = (\eta_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$  be positive weights. Then the corresponding generating vector  $\mathbf{g}$  constructed by Algorithm 1 satisfies*

$$K_{b^m, s, \boldsymbol{\eta}}(\mathbf{g}) \leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|}$$

for every  $s \in \{1, \dots, d\}$ .

*Proof.* We prove the result by induction on  $s \in \{1, \dots, d\}$ . Using that  $g_1 \equiv 1$ , we obtain for  $s = 1$  that

$$\begin{aligned} K_{b^m, 1, \eta}(g_1) &= K_{b^m, 1, \eta}(1) = \eta_{\{1\}} \sum_{n=1}^{b^m-1} L_{\{1\}} \left( v_m \left( \frac{n(x)}{p(x)} \right) \right) \\ &= \eta_{\{1\}} \sum_{n=1}^{b^m-1} \left( -(b-1) \left[ \log_b \left( v_m \left( \frac{n(x)}{p(x)} \right) \right) \right] - b \right) \\ &= -\eta_{\{1\}} (b-1) \sum_{n=1}^{b^m-1} \left[ \log_b \left( v_m \left( \frac{n(x)}{p(x)} \right) \right) \right] - \eta_{\{1\}} b (b^m - 1). \end{aligned}$$

In order to deal with the sum in the latter expression, we first observe that for any fraction  $\frac{n}{b^m}$  with  $b^t \leq n < b^{t+1}$  for  $0 \leq t < m$  we have that

$$\log_b \left( \frac{b^t}{b^m} \right) \leq \log_b \left( \frac{n}{b^m} \right) < \log_b \left( \frac{b^{t+1}}{b^m} \right).$$

Therefore, we find that

$$-(m-t) \leq \log_b \left( \frac{n}{b^m} \right) < -(m-t-1),$$

and hence  $\lfloor \log_b(n/b^m) \rfloor = -(m-t)$ . This identity and the application of Lemma 4 for the case  $d = 1$  yield

$$-\sum_{n=1}^{b^m-1} \left[ \log_b \left( v_m \left( \frac{n(x)}{p(x)} \right) \right) \right] = -\sum_{n=1}^{b^m-1} \left[ \log_b \left( \frac{n}{b^m} \right) \right] = \sum_{t=0}^{m-1} (b-1) b^t (m-t) = \frac{b^{m+1} - bm + m - b}{b-1} \quad (9)$$

since there are exactly  $(b-1)b^t$  numbers  $n$  with  $b^t \leq n < b^{t+1}$ . Therefore, we obtain that

$$K_{b^m, 1, \eta}(g_1) = \eta_{\{1\}} (b^{m+1} - bm + m - b - b^{m+1} + b) = -\eta_{\{1\}} (b-1)m < \eta_{\{1\}} (b-1)m.$$

Consider then  $s \geq 2$  and assume that the claimed statement holds for  $s-1$ , that is,

$$K_{b^m, s-1, \eta}(g_1, \dots, g_{s-1}) \leq \sum_{\emptyset \neq u \subseteq \{1:s-1\}} \eta_u ((b-1)m)^{|u|}.$$

By the standard averaging argument and the induction assumption we obtain

$$\begin{aligned} K_{b^m, s, \eta}(g_1, \dots, g_{s-1}, g_s) &\leq \frac{1}{b^m - 1} \sum_{g \in G_{b, m}^*} K_{b^m, s, \eta}(g_1, \dots, g_{s-1}, g) \\ &= \frac{1}{b^m - 1} \sum_{g \in G_{b, m}^*} \left[ \sum_{\emptyset \neq u \subseteq \{1:s-1\}} \eta_u \sum_{n=1}^{b^m-1} L_u \left( v_m \left( \frac{n(x) \mathbf{g}_u(x)}{p(x)} \right) \right) \right. \\ &\quad \left. + \sum_{v \subseteq \{1:s-1\}} \eta_{v \cup \{s\}} \sum_{n=1}^{b^m-1} L_{v \cup \{s\}} \left( v_m \left( \frac{n(x) \mathbf{g}_v(x)}{p(x)} \right), v_m \left( \frac{n(x) g(x)}{p(x)} \right) \right) \right] \\ &= K_{b^m, s-1, \eta}(\mathbf{g}_{\{1:s-1\}}) + \frac{1}{b^m - 1} \sum_{v \subseteq \{1:s-1\}} \eta_{v \cup \{s\}} \sum_{n=1}^{b^m-1} L_v \left( v_m \left( \frac{n(x) \mathbf{g}_v(x)}{p(x)} \right) \right) \times \\ &\quad \times \sum_{g \in G_{b, m}^*} \left( -(b-1) \left[ \log_b \left( v_m \left( \frac{n(x) g(x)}{p(x)} \right) \right) \right] - b \right). \end{aligned}$$

Using an argument similar to that which lead to (9), we obtain

$$\sum_{g \in G_{b,m}^*} \left( -(b-1) \left\lfloor \log_b \left( v_m \left( \frac{n(x)g(x)}{p(x)} \right) \right) \right\rfloor - b \right) = b^{m+1} - m(b-1) - b - b(b^m - 1) = -m(b-1),$$

which in turn gives

$$\begin{aligned} & K_{b^m, s, \boldsymbol{\eta}}(g_1, \dots, g_{s-1}, g_s) \\ & \leq K_{b^m, s-1, \boldsymbol{\eta}}(\mathbf{g}_{\{1:s-1\}}) + \frac{1}{b^m - 1} \sum_{\mathbf{v} \subseteq \{1:s-1\}} \eta_{\mathbf{v} \cup \{s\}} \sum_{n=1}^{b^m-1} L_{\mathbf{v}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{v}}(x)}{p(x)} \right) \right) (-m(b-1)) \\ & \leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s-1\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|} + \frac{m(b-1)}{b^m - 1} \sum_{n=1}^{b^m-1} \sum_{\mathbf{v} \subseteq \{1:s-1\}} \eta_{\mathbf{v} \cup \{s\}} \left| L_{\mathbf{v}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{v}}(x)}{p(x)} \right) \right) \right|. \end{aligned}$$

Considering the term

$$\left| L_{\mathbf{v}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{v}}(x)}{p(x)} \right) \right) \right| = \prod_{j \in \mathbf{v}} \left| -(b-1) \left\lfloor \log_b \left( v_m \left( \frac{n(x) g_j(x)}{p(x)} \right) \right) \right\rfloor - b \right|,$$

we see that since  $1 - b^{-m} \geq v_m \left( \frac{n(x)g(x)}{p(x)} \right) \geq b^{-m}$  for  $g \in G_{b,m}^*$  and  $\deg(n) < m$  we have

$$b-1 \leq -(b-1) \left\lfloor \log_b \left( v_m \left( \frac{n(x)g(x)}{p(x)} \right) \right) \right\rfloor \leq (b-1)m$$

and thus

$$\left| L_{\mathbf{v}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{v}}(x)}{p(x)} \right) \right) \right| \leq (m(b-1))^{|\mathbf{v}|}.$$

Hence, the estimation yields

$$\begin{aligned} K_{b^m, s, \boldsymbol{\eta}}(g_1, \dots, g_{s-1}, g_s) & \leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s-1\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|} + \frac{m(b-1)}{b^m - 1} \sum_{n=1}^{b^m-1} \sum_{\mathbf{v} \subseteq \{1:s-1\}} \eta_{\mathbf{v} \cup \{s\}} (m(b-1))^{|\mathbf{v}|} \\ & = \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s-1\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|} + \sum_{\mathbf{v} \subseteq \{1:s-1\}} \eta_{\mathbf{v} \cup \{s\}} (m(b-1))^{|\mathbf{v}|+1} \\ & = \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:s\}} \eta_{\mathbf{u}} ((b-1)m)^{|\mathbf{u}|}, \end{aligned}$$

which is the claimed result.  $\square$

Theorem 5 allows us to prove the main result regarding the construction in Algorithm 1.

**Theorem 6.** *Let  $b$  be prime, let  $m, d \in \mathbb{N}$  with  $m \geq 4$ , let  $p \in \mathbb{F}_b[x]$  be irreducible with  $\deg(p) = m$ , and let  $\boldsymbol{\eta} = (\eta_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$  be positive weights. Furthermore, let  $\mathbf{g} \in (G_{b,m}^*)^d$  be the generating vector constructed by Algorithm 1. Then the following estimate holds:*

$$T_{\boldsymbol{\eta}}(\mathbf{g}, p) \leq \frac{1}{b^m} \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} \left( 2((b-1)m)^{|\mathbf{u}|} + b m \left( 2(b-1)m + \frac{2b}{b-1} \right)^{|\mathbf{u}|} \right).$$

Moreover, if the weights  $\boldsymbol{\eta}$  satisfy

$$\sum_{j \geq 1} \max_{\mathbf{v} \subseteq \{1:j-1\}} \frac{\eta_{\mathbf{v} \cup \{j\}}}{\eta_{\mathbf{v}}} < \infty,$$

then  $T_{\boldsymbol{\eta}}(\mathbf{g}, p)$  can be bounded independently of the dimension  $d$ .



*Proof.* Using the bound on  $T_\eta(\mathbf{g}, p)$  in Theorem 4 and inserting for  $\mathbf{g}$  the generating vector obtained from Algorithm 1, for which the bound on  $K_{b^m, d, \eta}(\mathbf{g})$  in Theorem 5 holds, yields

$$\begin{aligned} T_\eta(\mathbf{g}, p) &\leq 2 \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} ((b-1)m)^{|\mathbf{u}|} + \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \frac{\eta_{\mathbf{u}}}{b^m} (bm |\mathbf{u}|) \left( (b-1)m + \frac{b}{b-1} \right)^{|\mathbf{u}|} \\ &\leq \frac{1}{b^m} \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} \left( 2((b-1)m)^{|\mathbf{u}|} + bm \left( 2(b-1)m + \frac{2b}{b-1} \right)^{|\mathbf{u}|} \right), \end{aligned}$$

where we used that  $|\mathbf{u}| \leq 2^{|\mathbf{u}|}$ . This proves the first claim. Furthermore, we easily find that

$$\begin{aligned} b^m T_\eta(\mathbf{g}, p) &\leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} \left( 2((b-1)m)^{|\mathbf{u}|} + bm \left( 2(b-1)m + \frac{2b}{b-1} \right)^{|\mathbf{u}|} \right) \\ &\leq \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} (4bm)^{|\mathbf{u}|} + bm \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} (4bm)^{|\mathbf{u}|} \\ &= (1+bm) \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} (4bm)^{|\mathbf{u}|} \leq C(\delta/2) b^{m\delta/2} \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \eta_{\mathbf{u}} (4bm)^{|\mathbf{u}|} \end{aligned}$$

for arbitrary  $\delta > 0$ , where  $C(\delta/2)$  is a constant depending only on  $\delta$ . We can now directly use the result in [6, Lemma 4] with  $N = b^m$  and  $a = 4b/\log b$  to see that the sum in the last expression is of order  $\mathcal{O}(b^{m\delta/2})$ . This yields the claimed result.  $\square$

Theorem 6 immediately implies the following result with respect to the worst-case error.

**Corollary 1.** *Let  $b$  be prime, let  $m, d \in \mathbb{N}$  with  $m \geq 4$  and  $N = b^m$ , let  $p \in \mathbb{F}_b[x]$  be irreducible with  $\deg(p) = m$ , and let  $\gamma = (\gamma_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$  be positive weights, satisfying*

$$\sum_{j \geq 1} \max_{\mathbf{v} \subseteq \{1:j-1\}} \frac{\gamma_{\mathbf{v} \cup \{j\}}}{\gamma_{\mathbf{v}}} < \infty.$$

*Then, for any  $\delta > 0$  and each  $\alpha > 1$ , Algorithm 1, run for the weights  $\boldsymbol{\eta} = \boldsymbol{\gamma} = (\gamma_{\mathbf{u}})_{\mathbf{u} \subseteq \{1:d\}}$ , constructs a generating vector  $\mathbf{g} \in (G_{b,m}^*)^d$  such that the worst-case error  $e_{b^m, d, \alpha, \gamma^\alpha}(\mathbf{g})$  satisfies*

$$e_{b^m, d, \alpha, \gamma^\alpha}(\mathbf{g}) \leq \frac{1}{N^\alpha} \left( \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \gamma_{\mathbf{u}}^\alpha (2\mu_b(\alpha))^{|\mathbf{u}|} + (C_1(\gamma, \delta))^\alpha N^{\alpha\delta} \right),$$

*with a positive constant  $C_1(\gamma, \delta)$  independent of  $d$  and  $N$ .*

*Additionally, if*

$$\sum_{j \geq 1} \max_{\mathbf{v} \subseteq \{1:j-1\}} \left( \frac{\gamma_{\mathbf{v} \cup \{j\}}}{\gamma_{\mathbf{v}}} \right)^{1/\alpha} < \infty,$$

*then Algorithm 1, run for the weights  $\boldsymbol{\eta} = \boldsymbol{\gamma}^{1/\alpha} = (\gamma_{\mathbf{u}}^{1/\alpha})_{\mathbf{u} \subseteq \{1:d\}}$ , constructs a generating vector  $\tilde{\mathbf{g}} \in (G_{b,m}^*)^d$  such that the worst-case error  $e_{b^m, d, \alpha, \gamma}(\tilde{\mathbf{g}})$  satisfies*

$$e_{b^m, d, \alpha, \gamma}(\tilde{\mathbf{g}}) \leq \frac{1}{N^\alpha} \left( \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \gamma_{\mathbf{u}} (2\mu_b(\alpha))^{|\mathbf{u}|} + (C_2(\boldsymbol{\gamma}^{1/\alpha}, \delta))^\alpha N^{\alpha\delta} \right),$$

*with a positive constant  $C_2(\boldsymbol{\gamma}^{1/\alpha}, \delta)$  independent of  $d$  and  $N$ .*

*Proof.* We know by Proposition 1 that the worst-case error satisfies

$$e_{b^m, d, \alpha, \eta^\alpha}(\mathbf{g}) \leq \frac{1}{N^\alpha} \sum_{\emptyset \neq u \subseteq \{1:d\}} \eta_u^\alpha (2\mu_b(\alpha))^{|u|} + T_{\alpha, \eta^\alpha}(\mathbf{g}, p). \quad (10)$$

Then, using the fact that for  $\alpha \geq 1$  we have  $\sum_i x_i^\alpha \leq (\sum_i x_i)^\alpha$  for  $x_i \geq 0$  (which is sometimes referred to as Jensen's inequality) yields the estimate

$$\begin{aligned} T_{\alpha, \eta^\alpha}(\mathbf{g}, p) &= \sum_{\mathbf{0} \neq \mathbf{k} \in A_p(\mathbf{g})} (r_{\alpha, \eta^\alpha}(\mathbf{k}))^{-1} = \sum_{\mathbf{0} \neq \mathbf{k} \in A_p(\mathbf{g})} (r_{1, \eta}(\mathbf{k}))^{-\alpha} \\ &\leq \left( \sum_{\mathbf{0} \neq \mathbf{k} \in A_p(\mathbf{g})} (r_{1, \eta}(\mathbf{k}))^{-1} \right)^\alpha = (T_\eta(\mathbf{g}, p))^\alpha. \end{aligned} \quad (11)$$

By using the estimate obtained in Theorem 6, we can deduce from this that, for arbitrary  $\delta > 0$ ,

$$b^m T_\eta(\mathbf{g}, p) \leq C_1(\eta, \delta) N^\delta, \quad (12)$$

where  $C_1(\eta, \delta)$  is a constant depending only on  $\delta$  and  $\eta$ . Combining the obtained estimates in (11) and (12) with (10), yields the claimed inequality for the choice  $\eta = \gamma$ . The result for the choice  $\eta = \gamma^{1/\alpha}$  follows analogously.  $\square$

The result in Corollary 1 consists of two statements regarding the worst-case error behavior of generating vectors constructed by Algorithm 1. On the one hand, when run with weights  $\gamma^{1/\alpha}$ , and hence depending on the parameter  $\alpha$ , the algorithm yields typical error bounds for the worst-case error in the space  $W_{d, \gamma}^\alpha$ . We emphasize that this type of result could also be obtained by formulating and using the CBC algorithm common in the literature (see below), which is instead directly based on the search criterion  $e_{b^m, d, \alpha, \gamma}$ . On the other hand, when run with weights  $\gamma$ , and thus independently of  $\alpha$ , the algorithm produces generating vectors for which bounds on the worst-case errors in the spaces  $W_{d, \gamma}^\alpha$  hold simultaneously for all  $\alpha > 1$ .

*Remark 1.* In the recent article [1], it is shown that polynomial lattice rules which were constructed for the weighted Walsh space  $W_{d, \gamma}^\alpha$  can also achieve the almost optimal convergence rate for the space  $W_{d, \gamma'}^{\alpha'}$  with different smoothness parameter  $\alpha' > 1$  and weight sequence  $\gamma'$ , provided that certain conditions on both weight sequences are satisfied. While in [1] the relation between the different weight sequences and smoothness parameters may allow to transfer certain results, our algorithm (when run with weights  $\gamma$ ) is independent of  $\alpha$  and delivers QMC rules for which error bounds hold simultaneously for all  $\alpha > 1$ . Nevertheless, it would be interesting to investigate how the condition on the weight sequence  $\gamma$  obtained here (see Corollary 1) compares to the conditions in [1]. We leave this question open for future research.

## 4 Fast implementation of the CBC construction

The fast component-by-component construction was first introduced in [15] for the case of (ordinary) lattice rules with a prime number of points and with the corresponding worst-case error as the quality criterion. A corresponding fast implementation for polynomial lattice rules was first analyzed in [16]. In this section, we discuss the efficient implementation of Algorithm 1 and analyze its complexity. Throughout the next two sections, we will consider the implementation of the CBC construction for the special case of product weights  $\gamma_u = \prod_{j \in u} \gamma_j$  for a sequence of positive reals  $(\gamma_j)_{j \geq 1}$ .

## 4.1 Implementation and cost analysis of the CBC algorithm

By using the same technique as in [6], it is possible to rewrite the quality function  $K_{b^m,d,\gamma}$  in Definition 3 for product weights  $\gamma_{\mathbf{u}} = \prod_{j \in \mathbf{u}} \gamma_j$  with  $(\gamma_j)_{j \geq 1} \subset \mathbb{R}_+$ . For this purpose, let  $N = b^m$  and consider  $\mathbf{g} \in G_{b,m}^d$  with prime  $b$  and positive integer  $m$ . Then, we see that  $K_{N,d,\gamma}$  equals

$$\begin{aligned} K_{N,d,\gamma}(\mathbf{g}) &= \sum_{\emptyset \neq \mathbf{u} \subseteq \{1:d\}} \gamma_{\mathbf{u}} \sum_{n=1}^{N-1} L_{\mathbf{u}} \left( v_m \left( \frac{n(x) \mathbf{g}_{\mathbf{u}}(x)}{p(x)} \right) \right) \\ &= \sum_{n=1}^{N-1} \left[ -1 + \prod_{j=1}^d \left( 1 + \gamma_j L_{\{j\}} \left( v_m \left( \frac{n(x) g_j(x)}{p(x)} \right) \right) \right) \right] \\ &= -(N-1) + \sum_{n=1}^{N-1} \prod_{j=1}^d \left( 1 + \gamma_j \left( (1-b) \left\lfloor \log_b \left( v_m \left( \frac{n(x) g_j(x)}{p(x)} \right) \right) \right\rfloor - b \right) \right). \end{aligned}$$

For short, we define the quantity

$$\bar{K}_{N,d,\gamma}(\mathbf{g}) := \sum_{n=1}^{N-1} \prod_{j=1}^d \left( 1 + \gamma_j \left( (1-b) \left\lfloor \log_b \left( v_m \left( \frac{n(x) g_j(x)}{p(x)} \right) \right) \right\rfloor - b \right) \right).$$

We observe that as the term  $N-1$  is constant, we can equivalently minimize the function  $\bar{K}_{N,s,\gamma}(\mathbf{g})$  instead of  $K_{N,s,\gamma}(\mathbf{g})$  in each step of Algorithm 1. Noting that this function has the same structure as the worst-case error expression which is minimized in the standard CBC algorithm, see, e.g., [15], we can employ the same machinery to obtain a fast implementation of Algorithm 1. We summarize the computational cost of this fast implementation in the following proposition.

**Proposition 2.** *Let  $m, d \in \mathbb{N}$  and set  $N = b^m$  with prime  $b$ . For a given sequence of positive weights  $\gamma = (\gamma_j)_{j=1}^d$ , a generating vector  $\mathbf{g} = (g_1, \dots, g_d)$  can be computed via Algorithm 1 using  $\mathcal{O}(dN \log N)$  operations.*

Since the fast implementation of Algorithm 1 can be done entirely analogously as for the standard CBC construction, we omit further implementation details and refer the reader to [3,16].

## 5 Numerical results

In this section, we present the results of some numerical experiments. Firstly, we compare the worst-case errors of generating vectors constructed by the CBC algorithm with smoothness-independent quality function  $K_{b^m,d,\gamma}$ , i.e., Algorithm 1, and the standard CBC algorithm for polynomial lattice rules (with the worst-case error as the quality function, as given in, e.g., [14,16]) for several choices of positive weight sequences. Finally, we compare the computational costs of implementations of Algorithm 1 and the standard CBC algorithm for polynomial lattice rules.

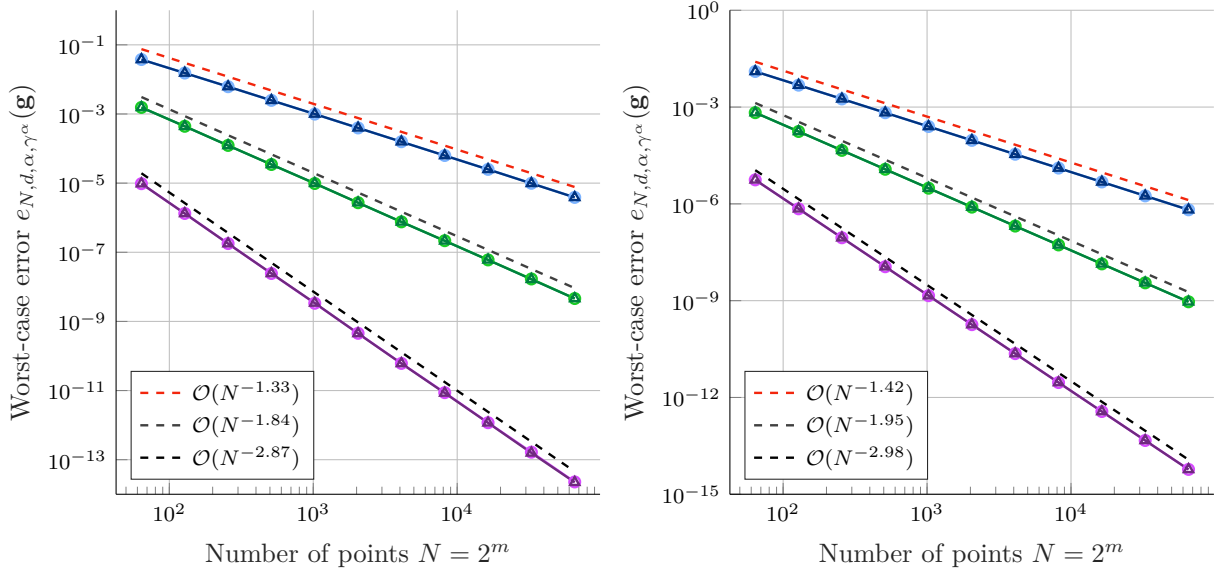
The algorithms considered were all implemented using Python 3.6.3. The implementations are available in double-precision as well as in arbitrary-precision floating-point arithmetic with the latter provided by the multiprecision Python library mpmath.

### 5.1 Error convergence behavior

We considered the convergence rate of the worst-case error corresponding to a generating vector  $\mathbf{g}$  constructed by Algorithm 1 and compared it to the error rates for polynomial lattice rules constructed by the standard CBC algorithm for different positive weight sequences  $\gamma = (\gamma_j)_{j \geq 1}$ . In particular, we computed the error  $e_{b^m,d,\alpha,\gamma^\alpha}(\mathbf{g})$  for dimension  $d = 100$  for different values of

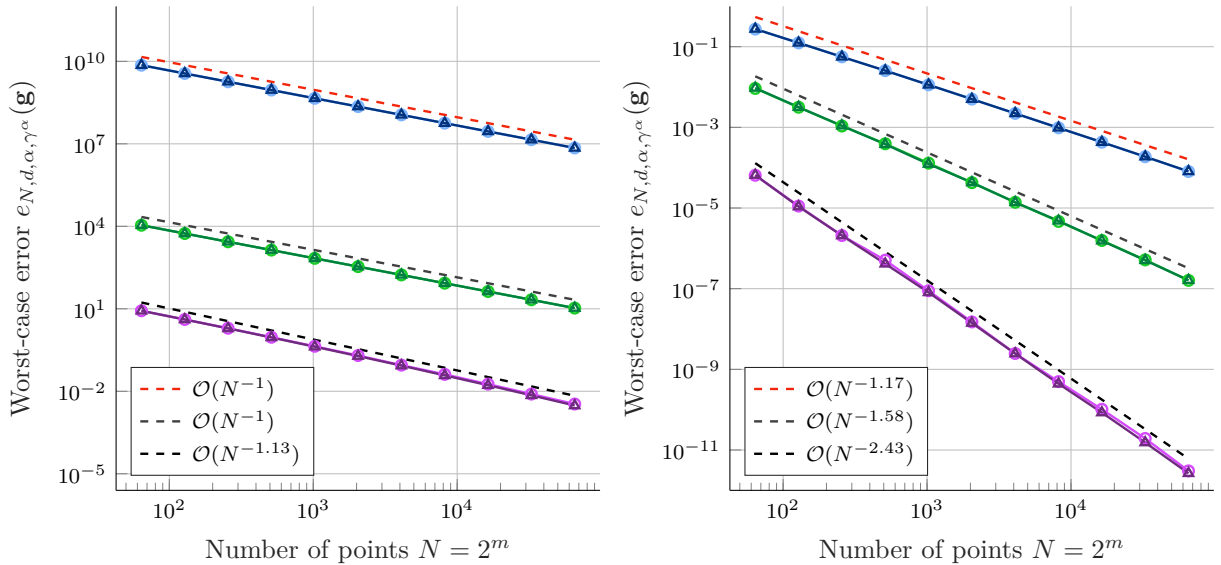
$m$  and different values of the smoothness parameter  $\alpha$ . We would like to illustrate the almost optimal error rates of  $\mathcal{O}(N^{-\alpha+\delta})$  which can be achieved according to Corollary 1, but may not be visible for the considered weights and the range of  $m$  in our numerical experiments. Therefore, the presented graphs are to be understood as an illustration of the pre-asymptotic error behavior.

**Error convergence in the space  $W_{d,\gamma}^\alpha$  with  $d = 100, \alpha = 1.5, 2, 3$ .**



(a) Weight sequence  $\gamma = (\gamma_j)_{j=1}^d$  with  $\gamma_j = 1/j^2$ .

(b) Weight sequence  $\gamma = (\gamma_j)_{j=1}^d$  with  $\gamma_j = 1/j^3$ .



(c) Weight sequence  $\gamma = (\gamma_j)_{j=1}^d$  with  $\gamma_j = (0.95)^j$ .

(d) Weight sequence  $\gamma = (\gamma_j)_{j=1}^d$  with  $\gamma_j = (0.7)^j$ .

—○— Algorithm 1    —▲— standard CBC    ■  $\alpha = 1.5$     ■  $\alpha = 2$     ■  $\alpha = 3$

Figure 1: Convergence results for the worst-case error  $e_{b^m,d,\alpha,\gamma^\alpha}(\mathbf{g})$  in the weighted space  $W_{d,\gamma}^\alpha$  for smoothness parameters  $\alpha = 1.5, 2, 3$  with dimension  $d = 100$ . The generating vectors  $\mathbf{g}$  were constructed via the component-by-component algorithms for polynomial lattice rules with  $N = 2^m$  points using  $K_{N,d,\gamma}$  and  $e_{N,d,\alpha,\gamma^\alpha}$ , respectively, as quality functions.

The graphs in Figure 1 show that the CBC algorithm with smoothness-independent quality function  $K_{b^m,d,\gamma}$  (Algorithm 1) constructs good generators of polynomial lattice rules with

worst-case errors that are almost identical to those of polynomial lattice rules constructed by the standard CBC algorithm with the worst-case error  $e_{b^m, d, \alpha, \gamma^\alpha}$  as the quality measure. We stress that while Algorithm 1 needs to be run only once (as the quality function is independent of the particular  $\alpha$ ), the standard CBC construction returns one generating vector for each choice of  $\alpha$  individually and is therefore run three times, once for each weight sequence considered.

## 5.2 Computational complexity

We illustrate the computational complexity of the component-by-component construction in Algorithm 1 which was stated in Proposition 2. To this end, we measure and compare the computation times of implementations of the smoothness-independent fast CBC construction (Algorithm 1) and the standard fast CBC algorithm for polynomial lattice rules with irreducible modulus  $p \in \mathbb{F}_2[x]$ . For  $m, d \in \mathbb{N}$  we use the weight sequence  $\gamma = (\gamma_j)_{j=1}^d$  with  $\gamma_j = j^{-2}$  and note that the chosen weights do not influence the computation times.

Table 1: Computation times (in seconds) for constructing the generating vector  $\mathbf{g}$  of a polynomial lattice rule with  $2^m$  points in  $d$  dimensions using the fast implementation of Algorithm 1 (**bold font**) and the standard fast CBC construction (normal font). For the standard fast CBC algorithm we constructed the polynomial lattice rules with smoothness parameter  $\alpha = 2$ .

|          | $d = 50$                | $d = 200$               | $d = 500$                 | $d = 1000$                | $d = 2000$                |
|----------|-------------------------|-------------------------|---------------------------|---------------------------|---------------------------|
| $m = 10$ | 0.007<br><b>0.007</b>   | 0.026<br><b>0.025</b>   | 0.051<br><b>0.056</b>     | 0.1<br><b>0.106</b>       | 0.2<br><b>0.207</b>       |
| $m = 12$ | 0.024<br><b>0.025</b>   | 0.088<br><b>0.09</b>    | 0.195<br><b>0.198</b>     | 0.384<br><b>0.389</b>     | 0.75<br><b>0.755</b>      |
| $m = 14$ | 0.112<br><b>0.115</b>   | 0.422<br><b>0.413</b>   | 0.939<br><b>0.928</b>     | 1.834<br><b>1.834</b>     | 3.938<br><b>3.751</b>     |
| $m = 16$ | 0.589<br><b>0.589</b>   | 1.99<br><b>2.023</b>    | 4.672<br><b>4.789</b>     | 9.544<br><b>9.787</b>     | 19.126<br><b>19.385</b>   |
| $m = 18$ | 2.716<br><b>2.751</b>   | 9.202<br><b>9.152</b>   | 22.046<br><b>21.809</b>   | 43.436<br><b>43.001</b>   | 86.343<br><b>85.279</b>   |
| $m = 20$ | 13.512<br><b>13.794</b> | 44.219<br><b>43.782</b> | 103.711<br><b>103.704</b> | 204.551<br><b>204.414</b> | 404.873<br><b>406.196</b> |

In Figure 2 below we display the measured computation times for the smoothness-independent CBC construction (Algorithm 1) graphically. Since the timings for both considered algorithms are almost identical, we only display the results for Algorithm 1 here.

The timings displayed in Table 1 and Figure 1 confirm that the computational complexity of both algorithms depends on  $m$  and  $d$  in a similar way and the measured times are in accordance with Proposition 2. Additionally, the linear dependence of the construction cost on the dimension  $d$  is well observable. The measured construction time for the fast implementation of Algorithm 1 is slightly higher than for the standard fast CBC algorithm but in general both algorithms can be executed in comparable time.

## Computation times for the fast implementation of Algorithm 1.

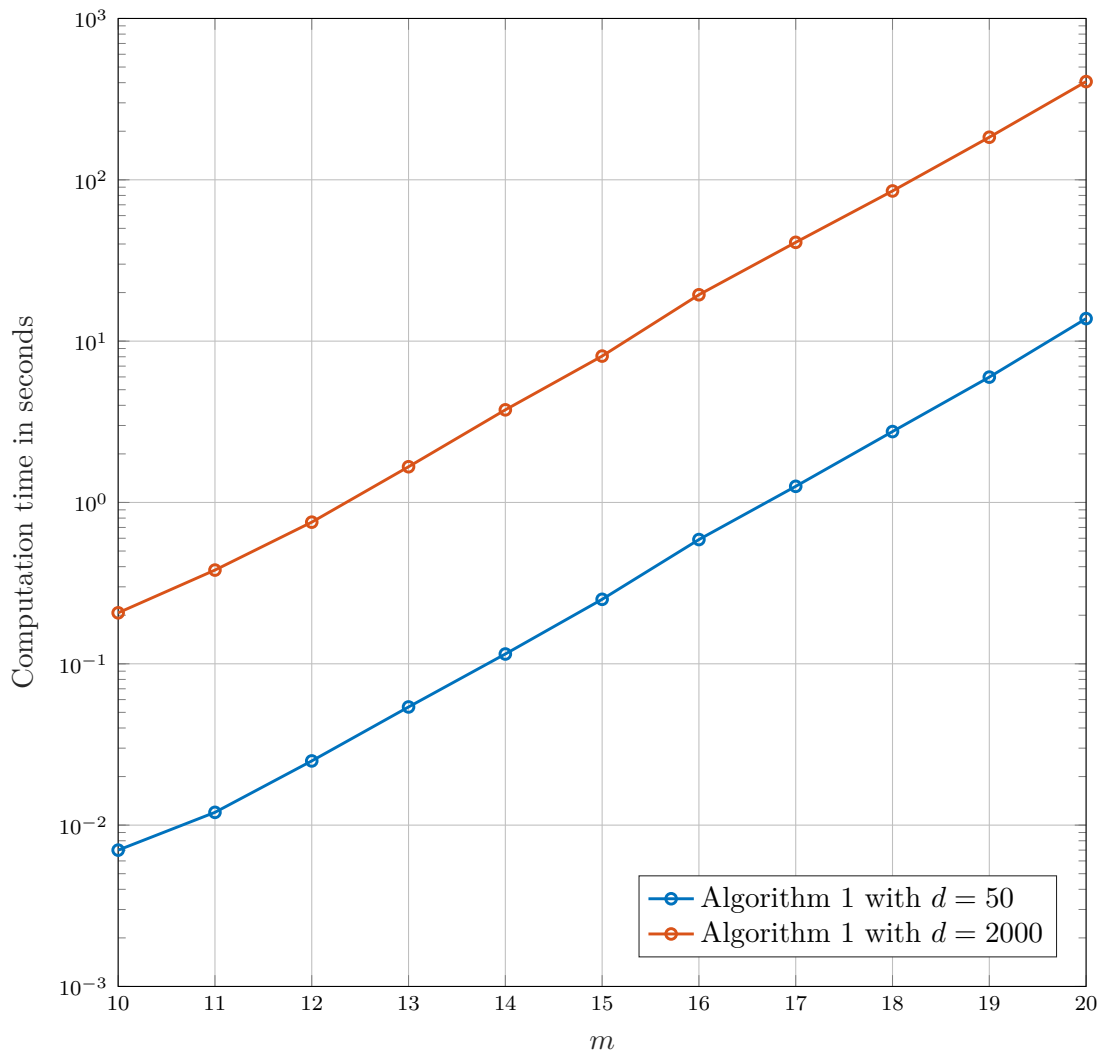


Figure 2: Computation times (in seconds) for constructing the generating vector  $\mathbf{g}$  of a polynomial lattice rule with  $2^m$  points in  $d \in \{50, 2000\}$  dimensions using the fast implementation of Algorithm 1.

## 6 Conclusion

In this paper we studied the construction of good polynomial lattice rules based on a quality measure which is independent of the smoothness of the underlying function space and can be used as a search criterion for polynomial lattice rules in weighted Walsh spaces. Based on these findings, we presented a component-by-component algorithm for constructing good polynomial lattice rules that is independent of the value of the smoothness parameter  $\alpha$  and yields an error convergence rate that is arbitrarily close to the optimal convergence rate in the studied function space. Under suitable summability conditions on the weight sequences, these error bounds were shown to be independent of the dimension. Furthermore, we studied a fast implementation of the algorithm which has the same computational complexity as the state-of-the-art standard fast CBC algorithm. Numerical experiments with respect to the error behavior and the computational complexity of the algorithm confirmed our theoretical findings.

## References

- [1] J. Dick, T. Goda. Stability of lattice rules and polynomial lattice rules constructed by the component-by-component algorithm. *J. Comput. Appl. Math.* 382, 113062, 2021.
- [2] J. Dick, F.Y. Kuo, F. Pillichshammer, I.H. Sloan. Construction algorithms for polynomial lattice rules for multivariate integration. *Math. Comp.* 74, 1895–1921, 2005.
- [3] J. Dick, F.Y. Kuo, I.H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numer.* 22, 133–288, 2013.
- [4] J. Dick, F. Pillichshammer. Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces. *J. Complexity* 21, 149–195, 2005.
- [5] J. Dick, F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, Cambridge 2010.
- [6] A. Ebert, P. Kritzer, D. Nuyens, O. Osisiogu. Digit-by-digit and component-by-component constructions of lattice rules for periodic functions with unknown smoothness. Submitted, 2020.
- [7] A. Ebert, P. Kritzer, O. Osisiogu, T. Stepaniuk. Component-by-component digit-by-digit construction of good polynomial lattice rules in weighted Walsh spaces. Submitted, 2020.
- [8] E. Hlawka. Zur angenäherten Berechnung mehrfacher Integrale. *Monatshefte für Mathematik* 66, 140–151, 1962.
- [9] N.M. Korobov. Approximate evaluation of repeated integrals. *Dokl. Akad. Nauk SSSR*, 124, 1207–1210, 1959. In Russian.
- [10] N.M. Korobov. Number-theoretic methods in approximate analysis. *Goz. Izdat. Fiz.-Math.*, 1963. In Russian.
- [11] H. Niederreiter. Point sets and sequences with small discrepancy. *Monatsh. Math.* 104, 273–337, 1987.
- [12] H. Niederreiter. Low-discrepancy point sets obtained by digital constructions over finite fields. *Czechoslovak Math. J.* 42, 143–166, 1992.
- [13] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.
- [14] D. Nuyens. The construction of good lattice rules and polynomial lattice rules. In: P. Kritzer, H. Niederreiter, F. Pillichshammer, A. Winterhof (eds.). *Uniform Distribution and Quasi-Monte Carlo Methods: Discrepancy, Integration and Applications*, 223–255, De Gruyter, Berlin, 2014.
- [15] D. Nuyens, R. Cools. Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. *Math. Comp.* 75, 903–920, 2006.
- [16] D. Nuyens, R. Cools. Fast component-by-component construction, a reprise for different kernels. In: H. Niederreiter, D. Talay (eds.). *Monte Carlo and Quasi-Monte Carlo Methods 2004*, 373–387, Springer, Berlin, 2006.
- [17] I.H. Sloan, S. Joe. *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford, 1994.

- [18] I.H. Sloan, V.A. Reztsov. Component-by-component construction of good lattice rules. *Math. Comp.* 71, 263–273, 2002.
- [19] I.H. Sloan, H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high-dimensional problems? *J. Complexity* 14, 1–33, 1998.

**Authors' addresses:**

Adrian Ebert

Johann Radon Institute for Computational and Applied Mathematics (RICAM)  
Austrian Academy of Sciences  
Altenbergerstr. 69, 4040 Linz, Austria.  
`adrian.ebert@oeaw.ac.at`

Peter Kritzer

Johann Radon Institute for Computational and Applied Mathematics (RICAM)  
Austrian Academy of Sciences  
Altenbergerstr. 69, 4040 Linz, Austria.  
`peter.kritzer@oeaw.ac.at`

Onyekachi Osisioogu

Johann Radon Institute for Computational and Applied Mathematics (RICAM)  
Austrian Academy of Sciences  
Altenbergerstr. 69, 4040 Linz, Austria.  
`onyekachi.osisioogu@oeaw.ac.at`

Tetiana Stepaniuk

Institute of Mathematics  
University of Lübeck  
Ratzeburger Allee 160, 23562 Lübeck, Germany,  
`stepaniuk@math.uni-luebeck.de`