www.oeaw.ac.at

Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

Constructing lattice points for numerical integration by a reduced fast successive coordinate search algorithm

A. Ebert, P. Kritzer

RICAM-Report 2018-09

Constructing lattice points for numerical integration by a reduced fast successive coordinate search algorithm

Adrian Ebert, Peter Kritzer

April 4, 2018

Abstract

In this paper, we study an efficient algorithm for constructing node sets of high-quality quasi-Monte Carlo integration rules for weighted Korobov, Walsh, and Sobolev spaces. The algorithm presented is a reduced fast successive coordinate search (SCS) algorithm, which is adapted to situations where the weights in the function space show a sufficiently fast decay. The new SCS algorithm is designed to work for the construction of lattice points, and, in a modified version, for polynomial lattice points, and the corresponding integration rules can be used to treat functions in different kinds of function spaces. We show that the integration rules constructed by our algorithms satisfy error bounds of optimal convergence order. Furthermore, we give details on efficient implementation such that we obtain a considerable speed-up of previously known SCS algorithms. This improvement is illustrated by numerical results. Furthermore, our main theorems yield previously unknown generalizations of earlier results.

Keywords: Numerical integration; lattice points; polynomial lattice points; quasi-Monte Carlo methods; weighted function spaces; component-by-component construction; successive coordinate search algorithm; fast implementations.
2010 MSC: 65D30, 65D32, 41A55, 41A63.

1 Introduction

Quasi-Monte Carlo (QMC) rules are equal-weight integration rules that are used for approximating integrals of functions over $[0,1]^s$,

$$\frac{1}{N}\sum_{n=0}^{N-1} f(\boldsymbol{x}_n) \approx \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

As opposed to Monte Carlo rules, where the integration nodes $\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{N-1}$ are selected at random, QMC integration is based on the idea of deterministically choosing the integration node set $\mathcal{P}=\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{N-1}\}$; here, the set $\mathcal{P}$ is interpreted as a multi-set, i.e., points are considered taking their multiplicity into account. For introductions to QMC methods and their applications we refer to [6, 9, 18, 19, 21].

Modern approaches to efficient QMC methods usually consider numerical integration for elements of Banach spaces, or, using a narrower setting, as in the present paper, for elements of certain reproducing kernel Hilbert spaces $(\mathcal{H},\|\cdot\|_{\mathcal{H}})$. For further information on reproducing kernel Hilbert spaces, see [1], and for details on the relation between such spaces and QMC theory, we refer to [28, 29]. In this context, the criterion considered for assessing the quality of a QMC integration rule based on a node set $\mathcal{P}=\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{N-1}\}$ for integration in a space

$(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ is the worst-case error,

$$e_{N,s}(\mathcal{H},\mathcal{P}) := \sup_{\substack{f\in\mathcal{H}\\ \|f\|_{\mathcal{H}}\le 1}} \left| \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{n=0}^{N-1} f(\boldsymbol{x}_n) \right|.$$

In this paper, we consider special types of QMC rules, namely lattice rules and polynomial lattice rules. Lattice rules (see, e.g., [21, 27] for introductions) are based on the choice of a positive integer N and a so-called generating vector $\boldsymbol{z} \in \{0,1,\ldots,N-1\}^s$. Using these parameters, an N-element lattice point set is given by the points

$$\boldsymbol{x}_n := \left\{ \frac{n\boldsymbol{z}}{N} \right\}, \quad 0 \le n \le N-1.$$

Here, we write $\{x\} = x - \lfloor x \rfloor$ for real numbers x, and apply $\{\cdot\}$ componentwise for vectors. Further details on these point sets and the function spaces whose elements can be integrated numerically using lattice rules will be given below in Section 2.

Polynomial lattice rules, see, e.g., [9, 21] are of a similar structure as lattice rules, but arithmetic over the reals is replaced by arithmetic of polynomials over finite fields. We will give further details on polynomial lattice rules in Section 6.

Returning to lattice rules, the crucial question regarding these integration rules is how to find a generating vector $\boldsymbol{z}$ that guarantees a low worst-case error of integration in a given function space. In general, there are no explicit constructions of good generating vectors for dimensions $s \ge 3$. One way to find good generating vectors is the component-by-component (CBC) construction, which is based on greedy algorithms choosing one component of the generating vector at a time. It was shown in [17] for prime N and in [3] for non-prime N that it is possible to find generating vectors yielding essentially optimal results for certain spaces of s-variate functions by the CBC construction. Furthermore, it was shown in [25, 26] that the computational cost of these algorithms is of order $\mathcal{O}(sN\log N)$. While the technique outlined in [25, 26] is very sophisticated, and the computational cost of order $\mathcal{O}(sN\log N)$ is excellent in comparison to previously known results, there is one drawback that remains. For s and N that are simultaneously large this cost may be still too high to construct $\boldsymbol{z}$. In the paper [4] it was therefore shown that this order of magnitude can be reduced further under suitable circumstances. The idea underlying the main result in [4] is to use the concept of weighted function spaces in the CBC construction. We will now shortly comment on weighted spaces and tractability, in order to describe the general idea of the paper [4] and also of the present paper.

The idea to use weighted function spaces in the context of quasi-Monte Carlo methods was introduced in the seminal paper [28]. Motivated by applications from financial mathematics, where different variables may have very different influence on a computational problem, Sloan and Woźniakowski introduced additional parameters in the definition of the function spaces under consideration, namely weights. These are given by a set of nonnegative real numbers $(\gamma_{\mathfrak{u}})_{\mathfrak{u}\subseteq[s]}$. Here and in the following, we write $[s]$ to denote the index set $\{1,\ldots,s\}$. The weight $\gamma_{\mathfrak{u}}$ models the importance of the projection of a given integrand f in the function space onto the variables x_j with $j \in \mathfrak{u}$. A small value of $\gamma_{\mathfrak{u}}$ means that the corresponding group of variables has only little influence on the problem, whereas a large value of $\gamma_{\mathfrak{u}}$ means the opposite. A special but important subcase is the case of product weights, where $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ for a (usually non-increasing) sequence $(\gamma_j)_{j\ge 1}$ of positive integers. In this case, γ_j can be thought of as modeling the influence of the variable γ_j.

The effect of studying weighted spaces in integration problems is that, if the influence of the variables (or, in other words, the weights) in the problem decay sufficiently fast for coordinates with high indices, one can vanquish the curse of dimensionality that is inherent to many high-dimensional problems. Indeed, under certain summability conditions on the weights, it is even possible to obtain bounds on the integration error that do not depend on the dimension of the problem at all. This is a property known as tractability, and we refer to the trilogy of Novak and Woźniakowski [22]–[24] for extensive information on this subject.

The paper [4] incorporated the weights of a given function space in the CBC construction of lattice rules that yield a low integration error for the same function space. Indeed, depending on the weights, the size of the search space for each component of the generating vector $\boldsymbol{z}$ was adjusted to the corresponding coordinate weight. This reduction is the motivation for calling the modified CBC algorithm from [4] a "reduced" CBC construction. It was also shown in [4] that the reduced CBC construction can be adapted to the existing fast CBC construction of Nuyens and Cools. Furthermore, it was shown that in the case of sufficiently fast decaying product weights the computational cost of the resulting reduced fast CBC construction can be independent of the dimension. These results also hold analogously for the case of polynomial lattice rules.

A different modification of the fast CBC construction was presented in the recent paper [11], where a so-called successive coordinate search (SCS) algorithm was presented. In this approach, one starts with a given generating vector $\boldsymbol{z}^0$ of a lattice rule. Then, the single components of this starting vector are improved on a step-by-step basis. The difference in the SCS approach, as opposed to the CBC approach, is that the algorithm has the starting vector as an input and the generating vector is not constructed from scratch. In particular, one could use the output of the fast CBC algorithm (or alternatively, of a previous instance of the SCS algorithm) as the input for the SCS algorithm and thereby further improve on the quality of the corresponding lattice rule. It is also possible to have a fast implementation of the SCS algorithm which has a computational cost of $\mathcal{O}(sN\log N)$, which is the same as that of the fast CBC construction. The paper [11] contains, apart from a theoretical analysis of the algorithm, also numerical results on the performance of the SCS algorithms. The numerical results show that the SCS algorithm can yield a significant improvement of the CBC algorithm for particular parameter settings (in particular, the performance is influenced by the choice of weights $\gamma_{\mathfrak{u}}$ in the problem).

In the present paper, we would like to combine the approaches in [4] and [11], and present a reduced fast SCS algorithm. This algorithm should be particularly well suited for situations in which one requires the construction of a large number of lattice points in high dimensions, with sufficiently fast decaying weights. The reduced fast SCS algorithm will again work by improving on a given starting vector $\boldsymbol{z}^0$, on a step-by-step basis (one component after the other). In comparison to the usual SCS algorithm presented in [11], however, the search spaces for the single components of the output vector will be reduced according to the coordinate weights, thus speeding up the construction method. We are going to show that for suitable choices of weights the construction cost of the reduced fast SCS algorithm can be made independent of the dimension, and that its result can be at least as good as that of the reduced fast CBC construction presented in [4]. Our results will be shown for integration algorithms for functions in weighted Korobov spaces, but, as we shall see below, they also can be transferred to hold for certain Sobolev spaces of functions. Apart from introducing and analyzing the reduced SCS algorithm for the construction of lattice points, our results imply a generalization of the results that have been presented in the paper [11], in the sense that the SCS algorithm now works for N being a prime power, and for general coordinate weights (as opposed to prime N and product weights in [11]).

We will also show that the SCS algorithm, as well as the reduced (fast) SCS algorithm can be adapted for constructing polynomial lattice rules which can be used for integrating functions in Walsh spaces and again certain Sobolev spaces. We stress that the present paper is the first paper where SCS algorithms for the polynomial lattice rule case are analyzed.

Moreover, we will present numerical results demonstrating that the reduced SCS algorithm constructs lattice rules which exhibit the same error convergence rate as the (reduced) CBC construction provided the weights decay sufficiently fast. Additionally, we will demonstrate the speed of the reduced SCS algorithm via timings.

The rest of the paper is structured as follows. In Section 2, we introduce Korobov spaces and point out how results for these are related to results for Sobolev spaces. Section 3 contains our main results regarding the reduced SCS construction for lattice rules. This is followed by remarks on how to obtain a fast implementation of the reduced SCS construction in Section 4 and numerical results for lattice rules in Section 5. We conclude the paper with a section on corresponding results for polynomial lattice rules.

2 Korobov spaces and related Sobolev spaces

We consider a weighted Korobov space with general weights as studied in [10, 23]. Let us first introduce some notation. We denote by $\mathbb{Z}$ the set of integers, by $\mathbb{Z}_*$ the set of integers excluding 0, and by $\mathbb{N}$ the set of positive integers. As above, for $s \in \mathbb{N}$ we write $[s] = \{1, 2, \ldots, s\}$. For a vector $\boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1]^s$ and for $\mathfrak{u} \subseteq [s]$, we write $\boldsymbol{x}_{\mathfrak{u}} = (x_j)_{j \in \mathfrak{u}} \in [0,1]^{|\mathfrak{u}|}$ and $(\boldsymbol{x}_{\mathfrak{u}}, \boldsymbol{0}) \in [0,1]^s$ for the vector $(y_1, \ldots, y_s)$ with $y_j = x_j$ if $j \in \mathfrak{u}$ and $y_j = 0$ if $j \notin \mathfrak{u}$. For integer vectors $\boldsymbol{h} \in \mathbb{Z}^s$, and $\mathfrak{u} \subseteq [s]$, we analogously write $\boldsymbol{h}_{\mathfrak{u}}$ to denote the projection of $\boldsymbol{h}$ onto those components with indices in $\mathfrak{u}$.

As outlined in the introduction, the importance of the different components or groups of components of the functions from the Korobov space to be defined is specified by a sequence of positive weights $\boldsymbol{\gamma} = (\gamma_{\mathfrak{u}})_{\mathfrak{u} \subseteq [s]}$, where we may assume that $\gamma_{\emptyset} = 1$. The smoothness of the functions in the space is described with a parameter $\alpha > 1$.

The weighted Korobov space, denoted by $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$, is a reproducing kernel Hilbert space with kernel function

$$K_{s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y}) = 1 + \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \prod_{j \in \mathfrak{u}} \left(\sum_{h \in \mathbb{Z}_*} \frac{\exp(2\pi \mathtt{i} h (x_j - y_j))}{|h|^{\alpha}} \right)$$
$$= 1 + \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathbb{Z}_*^{|\mathfrak{u}|}} \frac{\exp(2\pi \mathtt{i} \boldsymbol{h}_{\mathfrak{u}} \cdot (\boldsymbol{x}_{\mathfrak{u}} - \boldsymbol{y}_{\mathfrak{u}}))}{\prod_{j \in \mathfrak{u}} |h_j|^{\alpha}}.$$

The corresponding inner product is

$$\langle f, g \rangle_{K_{s,\alpha,\boldsymbol{\gamma}}} = \sum_{\mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{-1} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathbb{Z}_*^{|\mathfrak{u}|}} \left(\prod_{j \in \mathfrak{u}} |h_j|^{\alpha} \right) \widehat{f}((\boldsymbol{h}_{\mathfrak{u}}, \boldsymbol{0})) \overline{\widehat{g}((\boldsymbol{h}_{\mathfrak{u}}, \boldsymbol{0}))},$$

where $\widehat{f}(\boldsymbol{h}) = \int_{[0,1]^s} f(\boldsymbol{t}) \exp(-2\pi \mathtt{i} \boldsymbol{h} \cdot \boldsymbol{t}) \, \mathrm{d}\boldsymbol{t}$ is the $\boldsymbol{h}$-th Fourier coefficient of f.

For $h \in \mathbb{Z}_*$, we define $\rho_\alpha(h) = |h|^{-\alpha}$, and for $\boldsymbol{h} = (h_1, \ldots, h_s) \in \mathbb{Z}_*^s$ let $\rho_\alpha(\boldsymbol{h}) = \prod_{j=1}^s \rho_\alpha(h_j)$.

It is known (see, e.g., [10]) that the squared worst-case error of a lattice rule generated by a vector $\boldsymbol{z} \in \mathbb{Z}^s$ in the weighted Korobov space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ is given by

$$e_{N,s}^2(\boldsymbol{z}) = \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathcal{D}_{\mathfrak{u}}} \rho_\alpha(\boldsymbol{h}_{\mathfrak{u}}), \tag{1}$$

where

$$\mathcal{D}_{\mathfrak{u}} := \left\{ \boldsymbol{h}_{\mathfrak{u}} \in \mathbb{Z}_*^{|\mathfrak{u}|} \ : \ \boldsymbol{h}_{\mathfrak{u}} \cdot \boldsymbol{z}_{\mathfrak{u}} \equiv 0 \pmod{N} \right\}$$

is called the dual lattice of the lattice generated by $\boldsymbol{z}$. In order to avoid too many parameters in the notation, we do not include the weights $\boldsymbol{\gamma}$ when referring to the worst-case error $e_{N,s}$, unless this is essential for the context.

The worst-case error of lattice rules in a Korobov space can be related to the worst-case error in certain Sobolev spaces. Indeed, consider a tensor product Sobolev space $\mathcal{H}^{\text{sob}}_{s,\boldsymbol{\gamma}}$ of absolutely continuous functions whose mixed partial derivatives of order 1 in each variable are square integrable, with norm (see [13])

$$\|f\|_{\mathcal{H}^{\text{sob}}_{s,\boldsymbol{\gamma}}} = \left(\sum_{\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{-1} \int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{x}_{\mathfrak{u}}} f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}_{[s]\setminus\mathfrak{u}} \right)^2 \mathrm{d}\boldsymbol{x}_{\mathfrak{u}} \right)^{1/2},$$

where $\partial^{|\mathfrak{u}|} f/\partial \boldsymbol{x}_{\mathfrak{u}}$ denotes the mixed partial derivative with respect to all variables $j \in \mathfrak{u}$. As pointed out in [6, Section 5], the root mean square worst-case error $\widehat{e}_{N,s,\boldsymbol{\gamma}}$ for QMC integration in $\mathcal{H}^{\text{sob}}_{s,\boldsymbol{\gamma}}$ using randomly shifted lattice rules $(1/N)\sum_{k=0}^{N-1} f\left(\left\{\frac{k}{N}\boldsymbol{z} + \boldsymbol{\Delta}\right\}\right)$, i.e.,

$$\widehat{e}_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z}) = \left(\int_{[0,1)^s} e^2_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z},\boldsymbol{\Delta}) \,\mathrm{d}\boldsymbol{\Delta} \right)^{1/2},$$

where $e_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z},\boldsymbol{\Delta})$ is the worst-case error for QMC integration in $\mathcal{H}^{\text{sob}}_{s,\boldsymbol{\gamma}}$ using a randomly shifted integration lattice, is essentially the same as the worst-case error $e^{(2)}_{N,s,\boldsymbol{\gamma}}$ in the weighted Korobov space $\mathcal{H}(K_{s,2,\boldsymbol{\gamma}})$ using the unshifted version of the lattice rules. In fact, we have

$$\widehat{e}_{N,s,2\pi^2\boldsymbol{\gamma}}(\boldsymbol{z}) = e^{(2)}_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z}), \tag{2}$$

where $2\pi^2\boldsymbol{\gamma}$ denotes the weights $((2\pi^2)^{|\mathfrak{u}|}\gamma_{\mathfrak{u}})_{\emptyset\neq\mathfrak{u}\subseteq[s]}$. For a connection to the so-called anchored Sobolev space see, e.g., [14, Section 4].

In a slightly different setting, the random shift can be replaced by the tent transform $\phi(x) = 1 - |1 - 2x|$ in each variable. For a vector $\boldsymbol{x} \in [0,1]^s$ let $\phi(\boldsymbol{x})$ be defined component-wise. Let $\widetilde{e}_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z})$ be the worst-case error in the unanchored weighted Sobolev space $\mathcal{H}^{\text{sob}}_{s,\boldsymbol{\gamma}}$ using the QMC rule $(1/N)\sum_{k=0}^{N-1} f\left(\phi\left(\left\{\frac{k}{N}\boldsymbol{z}\right\}\right)\right)$. Then it is known due to [7] that

$$\widetilde{e}_{N,s,\pi^2\boldsymbol{\gamma}}(\boldsymbol{z}) = e^{(2)}_{N,s,\boldsymbol{\gamma}}(\boldsymbol{z}), \tag{3}$$

where $\pi^2\boldsymbol{\gamma} = (\pi^{2|\mathfrak{u}|}\gamma_{\mathfrak{u}})_{\emptyset\neq\mathfrak{u}\subseteq[s]}$. Hence we also have a direct connection between integration in the Korobov space using lattice rules and integration in the unanchored Sobolev space using tent-transformed lattice rules.

Thus, the results that will be shown in the following are valid for the root mean square worst-case error and the worst-case error using tent-transformed lattice rules in the unanchored Sobolev space as well as for the worst-case error in the Korobov space. Hence it suffices to state them only for $e_{N,s}$. Equation (2) can be used to obtain results also for $\widehat{e}_{N,s,\boldsymbol{\gamma}}$ and Equation (3) can be used to obtain results for $\widetilde{e}_{N,s,\boldsymbol{\gamma}}$.

What is more, there is also a connection between the worst-case errors for numerical integration using polynomial lattice rules in the Walsh space that we will introduce in Section 6 and the anchored [5, Section 5] and unanchored [8, Section 6] Sobolev space.

3 The reduced successive coordinate search algorithm

Let the number of quadrature points $N = b^m$ be a power of a prime number b, and $m \in \mathbb{N}$. Furthermore, we assume general weights $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$.

We further assume we are given integers w_j ordered in a non-decreasing fashion, i.e., $w_1 \leq w_2 \leq w_3 \leq \cdots$. Additionally, we set s^* as the largest j such that $w_j < m$. Next we define the reduced search space $\mathcal{Z}_{N,w_j}$ for the j-th component of the generating vector as

$$\mathcal{Z}_{N,w_j} = \begin{cases} \{z \in \{1,2,\dots,b^{m-w_j}-1\} : \gcd(z,N)=1\} & \text{if } w_j < m, \\ \{1\} & \text{if } w_j \geq m, \end{cases}$$

and $Y_j = b^{w_j}$ for $j \in \{1,\dots,s\}$. Then we consider the following algorithm for the construction of the generating vector $\boldsymbol{z}$ based on some initial vector $\boldsymbol{z}^0$.

Algorithm 1. *Let $N = b^m$ be a prime power, let $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$, be general weights, and let the worst-case error $e_{N,s}$ in the weighted Korobov space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ be defined as in Section 2. Let $w_1 \leq w_2 \leq \cdots \leq w_s$ and $Y_j = b^{w_j}$ for $j \in \{1,\dots,s\}$. Then we construct the generating vector $\boldsymbol{z} = (Y_1 z_1, \dots, Y_s z_s)$ as follows.*

- ***Input:*** *Starting vector $\boldsymbol{z}^0 = (z_1^0,\dots,z_s^0) \in \{0,1,\dots,N-1\}^s$.*
- *For $d \in [s]$ assume $z_1,\dots,z_{d-1}$ have already been selected. Then choose $z_d \in \mathcal{Z}_{N,w_d}$ such that $e_{N,s}^2((Y_1 z_1,\dots,Y_{d-1}z_{d-1},Y_d z_d, z_{d+1}^0,\dots,z_s^0))$ is minimized as a function of z_d.*
- *Increase d until $z_1,\dots,z_s$ are found.*

Theorem 1. *Let the assumptions in Algorithm 1 hold. Let $\boldsymbol{z} = (Y_1 z_1,\dots,Y_s z_s)$ be constructed by Algorithm 1. Then, for $\lambda \in (\frac{1}{\alpha}, 1]$, the squared worst-case error $e_{N,s}^2(\boldsymbol{z})$ satisfies*

$$e_{N,s}^2((Y_1 z_1,\dots,Y_s z_s)) \leq \left(\sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}} \right)^{\frac{1}{\lambda}}.$$

Proof. By (1), we have for $\boldsymbol{\xi} = (\xi_1,\dots,\xi_s) \in \{1,\dots,N-1\}^s$,

$$e_{N,s}^2(\boldsymbol{\xi}) = \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathcal{D}_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}})} \rho_\alpha(\boldsymbol{h}_{\mathfrak{u}})$$

with $\mathcal{D}_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}}) = \{\boldsymbol{h}_{\mathfrak{u}} \in \mathbb{Z}_*^{|\mathfrak{u}|} : \boldsymbol{h}_{\mathfrak{u}} \cdot \boldsymbol{\xi}_{\mathfrak{u}} \equiv 0 \pmod N\}$. We introduce the following notation:

$$g_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}}) := \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathcal{D}_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}})} \rho_\alpha(\boldsymbol{h}_{\mathfrak{u}}),$$

$$R_d(\boldsymbol{\xi}) := \sum_{d \in \mathfrak{u} \subseteq [d]} g_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}}),$$

and hence we obtain

$$e_{N,s}^2(\boldsymbol{\xi}) = \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} g_{\mathfrak{u}}(\boldsymbol{\xi}_{\mathfrak{u}}) = \sum_{d=1}^{s} R_d(\boldsymbol{\xi}).$$

In the following we write, for $d \in \{1,\dots,s\}$, $\boldsymbol{z}^{(d)} := (Y_1 z_1,\dots,Y_{d-1}z_{d-1},Y_d z_d, z_{d+1}^0,\dots,z_s^0)$. As minimizing $e_{N,s}^2(\boldsymbol{z}^{(d)})$ as a function of z_d is equivalent to minimizing only those parts that depend on z_d, namely

$$\theta_d(\boldsymbol{z}^{(d)}) := \sum_{d \in \mathfrak{u} \subseteq [s]} g_{\mathfrak{u}}(\boldsymbol{z}_{\mathfrak{u}}^{(d)}),$$

we consider this quantity for all d and note that

$$e_{N,s}^2(\boldsymbol{z}) = \sum_{d=1}^{s} R_d(Y_1 z_1, \ldots, Y_s z_s) = \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [d]} g_{\mathfrak{u}}(\boldsymbol{z}_{\mathfrak{u}}) \le \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} g_{\mathfrak{u}}(\boldsymbol{z}_{\mathfrak{u}}^{(d)}) = \sum_{d=1}^{s} \theta_d(\boldsymbol{z}^{(d)}).$$

We shall now make use of an inequality which is sometimes referred to as Jensen's inequality (see [12, 15]):

$$\sum_{i=1}^{M} a_i \le \left(\sum_{i=1}^{M} a_i^p\right)^{1/p} \qquad \text{for } 0 \le p \le 1 \text{ and } a_1, \ldots, a_M \ge 0.$$

Using Jensen's inequality we obtain, for $\lambda \in (\frac{1}{\alpha}, 1]$,

$$\left(e_{N,s}^2(\boldsymbol{z})\right)^{\lambda} \le \left(\sum_{d=1}^{s} \theta_d(\boldsymbol{z}^{(d)})\right)^{\lambda} \le \sum_{d=1}^{s} \theta_d^{\lambda}(\boldsymbol{z}^{(d)}).$$

By the standard averaging argument we obtain that, since the best choice for z_d is at least as good as the average,

$$\begin{aligned} \theta_d^{\lambda}(\boldsymbol{z}^{(d)}) &= \theta_d^{\lambda}(Y_1 z_1, \ldots, Y_d z_d, z_{d+1}^0, \ldots, z_s^0) \\ &\le \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \theta_d^{\lambda}(Y_1 z_1, \ldots, Y_{d-1} z_{d-1}, Y_d z, z_{d+1}^0, \ldots, z_s^0). \end{aligned}$$

We now use the notation $\hat{\boldsymbol{z}}^{(d)} = \hat{\boldsymbol{z}}^{(d)}(z) := (Y_1 z_1, \ldots, Y_{d-1} z_{d-1}, Y_d z, z_{d+1}^0, \ldots, z_s^0)$. For the sake of readability, we will sometimes write $\hat{\boldsymbol{z}}^{(d)} = (\hat{z}_1, \ldots, \hat{z}_s)$ for short. Next, we establish an upper estimate for the quantity $\theta_d^{\lambda}(\hat{\boldsymbol{z}}^{(d)})$ for each $d \in \{1, \ldots, s\}$,

$$\begin{aligned} \theta_d^{\lambda}(\hat{\boldsymbol{z}}^{(d)}) = \left(\sum_{d \in \mathfrak{u} \subseteq [s]} g_{\mathfrak{u}}(\hat{\boldsymbol{z}}_{\mathfrak{u}}^{(d)})\right)^{\lambda} &\le \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\lambda} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathcal{D}_{\mathfrak{u}}(\hat{\boldsymbol{z}}_{\mathfrak{u}}^{(d)})} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{u}}) \\ &= \gamma_{\{d\}}^{\lambda} \sum_{h_d \in \mathcal{D}_{\{d\}}(Y_d z)} \rho_{\alpha\lambda}(h_d) + \sum_{\substack{\emptyset \ne \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(h_d) \sum_{\substack{\boldsymbol{h}_{\mathfrak{v}} \in \mathbb{Z}_*^{|\mathfrak{v}|} \\ \sum_{j \in \mathfrak{v}} h_j \hat{z}_j \equiv -h_d Y_d z \ (N)}} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}), \end{aligned}$$

where we used Jensen's inequality twice to obtain the first estimate, and where we write (N) to denote mod N for short. This implies in turn that

$$\theta_d^{\lambda}(\boldsymbol{z}^{(d)}) \le \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \theta_d^{\lambda}(\hat{\boldsymbol{z}}^{(d)}) \le T_1 + T_2,$$

where

$$T_1 = \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \gamma_{\{d\}}^{\lambda} \sum_{h_d \in \mathcal{D}_{\{d\}}(Y_d z)} \rho_{\alpha\lambda}(h_d)$$

and

$$T_2 = \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \sum_{\substack{\emptyset \ne \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(h_d) \sum_{\substack{\boldsymbol{h}_{\mathfrak{v}} \in \mathbb{Z}_*^{|\mathfrak{v}|} \\ \sum_{j \in \mathfrak{v}} h_j \hat{z}_j \equiv -h_d Y_d z \ (N)}} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}).$$

For T_1 we consider the two possible cases $w_d \ge m$ and $w_d < m$. Then we obtain:

- If $w_d \geq m$, then $z \in \mathcal{Z}_{N,w_d} = \{1\}$ and $b^m = N$ is a divisor of b^{w_d} so that

$$T_1 = \gamma_{\{d\}}^{\lambda} \sum_{\substack{h_d \in \mathbb{Z}_* \\ b^{w_d} h_d \equiv 0 \ (N)}} \rho_{\alpha\lambda}(h_d) = \gamma_{\{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(h_d) = \gamma_{\{d\}}^{\lambda} 2\zeta(\alpha\lambda) = \gamma_{\{d\}}^{\lambda} \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}}.$$

- If $w_d < m$, then $h_d b^{w_d} z \equiv 0 \ (N)$, i.e., $b^{w_d} h_d z = k b^m$ for some $k \in \mathbb{Z}$, is equivalent to $h_d z = k b^{m-w_d}$ for some $k \in \mathbb{Z}$. Now if $z | k\, b^{m-w_d}$ then, since $z \in \mathcal{Z}_{N,w_d}$, we have $b^\ell \not| z$ for all $\ell = 1, \ldots, m - w_d$ which implies that $z|k$, i.e., $k = k'z$ for some $k' \in \mathbb{Z}$. Hence

$$\begin{aligned} h_d b^{w_d} z \equiv 0 \ (N) &\Leftrightarrow b^{w_d} h_d z = k b^m \Leftrightarrow h_d z = k b^{m-w_d} \Leftrightarrow h_d z = k' z b^{m-w_d} \\ &\Leftrightarrow h_d = k' b^{m-w_d} \Leftrightarrow b^{m-w_d} | h_d, \end{aligned}$$

and we obtain

$$\begin{aligned} T_1 &= \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \gamma_{\{d\}}^{\lambda} \sum_{\substack{h_d \in \mathbb{Z}_* \\ Y_d h_d z \equiv 0 \ (N)}} \rho_{\alpha\lambda}(h_d) = \frac{1}{|\mathcal{Z}_{N,w_d}|} \sum_{z \in \mathcal{Z}_{N,w_d}} \gamma_{\{d\}}^{\lambda} \sum_{\substack{h_d \in \mathbb{Z}_* \\ b^{m-w_d} | h_d}} \rho_{\alpha\lambda}(h_d) \\ &= \gamma_{\{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(b^{m-w_d} h_d) = \gamma_{\{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} b^{-\alpha\lambda(m-w_d)} \rho_{\alpha\lambda}(h_d) \\ &= \gamma_{\{d\}}^{\lambda} b^{-\alpha\lambda(m-w_d)} 2\zeta(\alpha\lambda) \leq \gamma_{\{d\}}^{\lambda} \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}}. \end{aligned}$$

Therefore, in both possible cases, it holds that

$$T_1 \leq \gamma_{\{d\}}^{\lambda} \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}}.$$

Similarly, we investigate the term T_2 for both cases.

- If $w_d \geq m$, then $z \in \mathcal{Z}_{N,w_d} = \{1\}$ and $b^{\max(0,m-w_d)} = b^0 = 1$, and so

$$\begin{aligned} T_2 &= \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(h_d) \sum_{\substack{\boldsymbol{h}_\mathfrak{v} \in \mathbb{Z}_*^{|\mathfrak{v}|} \\ \sum_{j \in \mathfrak{v}} h_j \hat{z}_j \equiv -h_d Y_d \ (N)}} \rho_{\alpha\lambda}(\boldsymbol{h}_\mathfrak{v}) \\ &\leq \sum_{h_d \in \mathbb{Z}_*} \rho_{\alpha\lambda}(h_d) \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{\boldsymbol{h}_\mathfrak{v} \in \mathbb{Z}_*^{|\mathfrak{v}|}} \rho_{\alpha\lambda}(\boldsymbol{h}_\mathfrak{v}) \\ &= \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}} \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{\boldsymbol{h}_\mathfrak{v} \in \mathbb{Z}_*^{|\mathfrak{v}|}} \rho_{\alpha\lambda}(\boldsymbol{h}_\mathfrak{v}) \\ &= \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}} \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} (2\zeta(\alpha\lambda))^{|\mathfrak{v}|} = \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \frac{(2\zeta(\alpha\lambda))^{|\mathfrak{v}|+1}}{b^{\max(0,m-w_d)}} \\ &= \sum_{\substack{\{d\} \neq \mathfrak{u} \subseteq [s] \\ d \in \mathfrak{u}}} \gamma_{\mathfrak{u}}^{\lambda} \frac{(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}}. \end{aligned}$$

- If $w_d < m$, then, with $|\mathcal{Z}_{N,w_d}| = b^{m-w_d-1}(b-1)$, we get

$$T_2 = \frac{1}{b^{m-w_d-1}(b-1)} \left[\sum_{z \in \mathcal{Z}_{N,w_d}} \sum_{\substack{\emptyset \neq \mathfrak{v} \subseteq [s] \\ d \notin \mathfrak{v}}} \gamma_{\mathfrak{v} \cup \{d\}}^{\lambda} \sum_{\substack{h_d \in \mathbb{Z}_* \\ h_d \equiv 0 \ (b^{m-w_d})}} \rho_{\alpha\lambda}(h_d) \sum_{\substack{\boldsymbol{h}_\mathfrak{v} \in \mathbb{Z}_*^{|\mathfrak{v}|} \\ \sum_{j \in \mathfrak{v}} h_j \hat{z}_j \equiv -h_d Y_d z \ (N)}} \rho_{\alpha\lambda}(\boldsymbol{h}_\mathfrak{v}) \right.$$

$$
+\sum_{z\in\mathcal{Z}_{N,w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\not\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv -h_dY_dz\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\Bigg]
$$

$$
=T_{2,1}+T_{2,2},
$$

where

$$
T_{2,1}=\frac{1}{b^{m-w_d-1}(b-1)}\sum_{z\in\mathcal{Z}_{N,w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv -h_dY_dz\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})
$$

and

$$
T_{2,2}=\frac{1}{b^{m-w_d-1}(b-1)}\sum_{z\in\mathcal{Z}_{N,w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\not\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv -h_dY_dz\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}).
$$

For $T_{2,1}$ we see that if $h_d\equiv 0\ (b^{m-w_d})$ then $h_dY_dz\equiv 0\ (N)$. Thus

$$
\begin{aligned}
T_{2,1}&=\frac{1}{b^{m-w_d-1}(b-1)}\sum_{z\in\mathcal{Z}_{N,w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\\
&=\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{h_d\in\mathbb{Z}_*}\rho_{\alpha\lambda}(b^{m-w_d}h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\\
&=\frac{2\zeta(\alpha\lambda)}{(b^{m-w_d})^{\alpha\lambda}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\\
&\leq\frac{4\zeta(\alpha\lambda)}{b^{m-w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}).
\end{aligned}
$$

For $T_{2,2}$ we see that since $h_d\not\equiv 0\ (b^{m-w_d})$, which is equivalent to $h_d\neq kb^{m-w_d}$, for any $k\in\mathbb{Z}$, and since $z\in\mathcal{Z}_{N,w_d}$, it holds that $h_dY_dz\not\equiv 0\ (N)$. Also if $z_1\neq z_2$ and $z_1,z_2\in\mathcal{Z}_{N,w_d}$ then $h_dY_dz_1\not\equiv h_dY_dz_2\ (N)$.

Hence we obtain

$$
\begin{aligned}
T_{2,2}&=\frac{1}{b^{m-w_d-1}(b-1)}\sum_{z\in\mathcal{Z}_{N,w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\not\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\equiv -h_dY_dz\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\\
&\leq\frac{1}{b^{m-w_d-1}(b-1)}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{\substack{h_d\in\mathbb{Z}_*\\ h_d\not\equiv 0\ (b^{m-w_d})}}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\not\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})\\
&\leq\frac{2}{b^{m-w_d}}\sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}}\gamma^{\lambda}_{\mathfrak{v}\cup\{d\}}\sum_{h_d\in\mathbb{Z}_*}\rho_{\alpha\lambda}(h_d)\sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}}h_j\hat{z}_j\not\equiv 0\ (N)}}\rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})
\end{aligned}
$$

$$= \frac{4\zeta(\alpha\lambda)}{b^{m-w_d}} \sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}} \gamma_{\mathfrak{v}\cup\{d\}}^{\lambda} \left(\sum_{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}) - \sum_{\substack{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}\\ \sum_{j\in\mathfrak{v}} h_j\hat{z}_j\equiv 0\ (N)}} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}}) \right).$$

Thus, we obtain for T_2,

$$T_2 = T_{2,1} + T_{2,2} \leq \frac{4\zeta(\alpha\lambda)}{b^{m-w_d}} \sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}} \gamma_{\mathfrak{v}\cup\{d\}}^{\lambda} \sum_{\boldsymbol{h}_{\mathfrak{v}}\in\mathbb{Z}_*^{|\mathfrak{v}|}} \rho_{\alpha\lambda}(\boldsymbol{h}_{\mathfrak{v}})$$

$$= \sum_{\substack{\emptyset\neq\mathfrak{v}\subseteq[s]\\ d\notin\mathfrak{v}}} \gamma_{\mathfrak{v}\cup\{d\}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{v}|+1}}{b^{\max(0,m-w_d)}} = \sum_{\substack{\{d\}\neq\mathfrak{u}\subseteq[s]\\ d\in\mathfrak{u}}} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}}.$$

Combining both cases, T_2 is always bounded by

$$T_2 \leq \sum_{\substack{\{d\}\neq\mathfrak{u}\subseteq[s]\\ d\in\mathfrak{u}}} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}}.$$

Hence, for the quantity $\theta_d^{\lambda}(\boldsymbol{z}^{(d)})$ we see that

$$\theta_d^{\lambda}(\boldsymbol{z}^{(d)}) \leq T_1 + T_2 \leq \gamma_{\{d\}}^{\lambda} \frac{2\zeta(\alpha\lambda)}{b^{\max(0,m-w_d)}} + \sum_{\substack{\{d\}\neq\mathfrak{u}\subseteq[s]\\ d\in\mathfrak{u}}} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}} \leq \sum_{d\in\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}},$$

and so the squared worst-case error is bounded by

$$\left(e_{N,s}^2(\boldsymbol{z})\right)^{\lambda} \leq \sum_{d=1}^{s} \theta_d^{\lambda}(\boldsymbol{z}^{(d)}) \leq \sum_{d=1}^{s} \sum_{d\in\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}}$$

which proves the claim. □

Corollary 2. *Let the assumptions in Algorithm 1 hold. Let* $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_s z_s)$ *be constructed by Algorithm 1. Then we have for all* $\delta \in (0, \frac{\alpha-1}{2}]$ *that*

$$e_{N,s}(\boldsymbol{z}) \leq C_{s,\alpha,\boldsymbol{\gamma},\delta}\, N^{-\alpha/2+\delta},$$

where

$$C_{s,\alpha,\boldsymbol{\gamma},\delta} := \left(2 \sum_{d=1}^{s} \sum_{d\in\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha-2\delta}} \left(2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)^{|\mathfrak{u}|} b^{w_d} \right)^{\alpha/2-\delta}.$$

For $\delta \in (0, \frac{\alpha-1}{2}]$ *and* $q \geq 0$*, define*

$$C_{\delta,q} := \sup_{s\in\mathbb{N}} \left[\frac{2}{s^q} \sum_{d=1}^{s} \sum_{d\in\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha-2\delta}} \left(2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)^{|\mathfrak{u}|} b^{w_d} \right].$$

If $C_{\delta,q} < \infty$ *for some* $\delta \in (0, \frac{\alpha-1}{2}]$ *and* $q \geq 0$ *then*

$$e_{N,s}(\boldsymbol{z}) \leq (s^q C_{\delta,q})^{\alpha/2-\delta} N^{-\alpha/2+\delta}.$$

If $C_{\delta,0} < \infty$ *for some* $\delta \in (0, \frac{\alpha-1}{2}]$ *then*

$$e_{N,s}(\boldsymbol{z}) \leq (C_{\delta,0})^{\alpha/2-\delta} N^{-\alpha/2+\delta}.$$

Proof. By Theorem 1 we have, for $\lambda \in (\frac{1}{\alpha}, 1]$, that

$$\left(e^2_{N,s}(\boldsymbol{z})\right)^\lambda \le \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^\lambda \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}}$$

and thus, since $b^{-\max(0,m-w_d)} = b^{\min(0,w_d-m)} = b^{-m}\, b^{\min(m,w_d)} \le b^{-m}\, b^{w_d} = \frac{1}{N}\, b^{w_d}$,

$$\left(e^2_{N,s}(\boldsymbol{z})\right)^\lambda \le \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^\lambda \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0,m-w_d)}} \le \frac{2}{N} \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^\lambda (2\zeta(\alpha\lambda))^{|\mathfrak{u}|} b^{w_d}.$$

Setting $\frac{1}{\lambda} = \alpha - 2\delta$, this shows the first assertion in the corollary. The proof of the further assertions is straightforward. □

Let us, for the next corollary, assume that we have product weights, i.e., $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for $\mathfrak{u} \subseteq [s]$, where the γ_j are elements of an infinite, non-increasing sequence of positive reals, $(\gamma_j)_{j \ge 1}$.

Corollary 3. *Let the assumptions in Algorithm 1 hold. Let $\boldsymbol{z} = (Y_1 z_1, \ldots, Y_s z_s)$ be constructed by Algorithm 1. Then we have for all $\delta \in (0, \frac{\alpha-1}{2}]$ that*

$$e_{N,s}(\boldsymbol{z}) \le C_{s,\alpha,\boldsymbol{\gamma},\delta}\, N^{-\alpha/2+\delta},$$

where

$$C_{s,\alpha,\boldsymbol{\gamma},\delta} := \left(\left(\sum_{d=1}^{s} \gamma_d^{\frac{1}{\alpha-2\delta}} b^{w_d} \right) \left(4\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right) \prod_{d=1}^{s-1} \left(1 + \gamma_d^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)\right)^{\alpha/2-\delta}.$$

Furthermore, the constant $C_{s,\alpha,\boldsymbol{\gamma},\delta}$ is bounded independently of the dimension s if

$$\sum_{d=1}^{\infty} \gamma_d^{\frac{1}{\alpha-2\delta}}\, b^{w_d} < \infty.$$

Proof. Similar to the proof of Corollary 2, we see that

$$\left(e^2_{N,s}(\boldsymbol{z})\right)^\lambda \le \frac{2}{N} \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^\lambda (2\zeta(\alpha\lambda))^{|\mathfrak{u}|} b^{w_d}.$$

Thus,

$$\begin{aligned}
\left(e^2_{N,s}(\boldsymbol{z})\right)^\lambda &\le \frac{2}{N} \sum_{d=1}^{s} \left(\sum_{\mathfrak{u} \subseteq [s]\setminus\{d\}} \gamma_{\mathfrak{u}}^\lambda (2\zeta(\alpha\lambda))^{|\mathfrak{u}|} \right) \left(\gamma_d^\lambda 2\,\zeta(\alpha\lambda)\, b^{w_d}\right) \\
&\le \frac{2}{N} \sum_{d=1}^{s} \left(\gamma_d^\lambda\, b^{w_d}\right) (2\,\zeta(\alpha\lambda)) \max_{d=1,\ldots,s} \left(\sum_{\mathfrak{u} \subseteq [s]\setminus\{d\}} \gamma_{\mathfrak{u}}^\lambda (2\zeta(\alpha\lambda))^{|\mathfrak{u}|} \right) \\
&= \frac{2}{N} \sum_{d=1}^{s} \left(\gamma_d^\lambda\, b^{w_d}\right) (2\,\zeta(\alpha\lambda)) \max_{d=1,\ldots,s} \left(\prod_{\substack{j=1\\ j\ne d}}^{s} \left(1 + \gamma_j^\lambda 2\zeta(\alpha\lambda)\right) \right) \\
&= \frac{2}{N} \sum_{d=1}^{s} \left(\gamma_d^\lambda\, b^{w_d}\right) (2\,\zeta(\alpha\lambda)) \prod_{j=1}^{s-1} \left(1 + \gamma_j^\lambda 2\zeta(\alpha\lambda)\right).
\end{aligned}$$

Hence we have that

$$e_{N,s}(\boldsymbol{z}) \le \left(\frac{1}{N}\right)^{\frac{1}{2\lambda}} \left(\sum_{d=1}^{s} \left(\gamma_d^{\lambda}\, b^{w_d}\right) (4\,\zeta(\alpha\lambda)) \prod_{j=1}^{s-1} \left(1+\gamma_j^{\lambda} 2\zeta(\alpha\lambda)\right)\right)^{\frac{1}{2\lambda}},$$

and setting $\frac{1}{\lambda} = \alpha - 2\delta$ this gives

$$\begin{aligned} e_{N,s}(\boldsymbol{z}) &\le N^{-\frac{\alpha}{2}+\delta} \left(\sum_{d=1}^{s} \left(\gamma_d^{\frac{1}{\alpha-2\delta}}\, b^{w_d}\right) \left(4\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right) \prod_{j=1}^{s-1} \left(1+\gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)\right)^{\alpha/2-\delta} \\ &= C_{s,\alpha,\boldsymbol{\gamma},\delta}\, N^{-\frac{\alpha}{2}+\delta}. \end{aligned}$$

Furthermore, note that since

$$\prod_{j=1}^{s-1} \left(1+\gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right) = \exp\left(\log\left(\prod_{j=1}^{s-1} \left(1+\gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)\right)\right),$$

and (as $\log(1+x) \le x$)

$$\begin{aligned} \log\left(\prod_{j=1}^{s-1} \left(1+\gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)\right) &= \sum_{d=1}^{s-1} \log\left(1+\gamma_j^{\frac{1}{\alpha-2\delta}} 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right) \\ &\le 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right) \sum_{d=1}^{s-1} \gamma_j^{\frac{1}{\alpha-2\delta}} \\ &\le 2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right) \sum_{d=1}^{\infty} \gamma_j^{\frac{1}{\alpha-2\delta}} b^{w_d}, \end{aligned}$$

the constant $C_{s,\alpha,\boldsymbol{\gamma},\delta}$ is finite, and therefore bounded independently of the dimension s, if

$$\sum_{d=1}^{\infty} \gamma_j^{\frac{1}{\alpha-2\delta}} b^{w_d} < \infty.$$

□

A straightforward but important consequence of Algorithm 1 and Theorem 1 is that we obtain a generalization of one of the main results in [11]. In that paper, the (unreduced) SCS algorithm was considered for prime N and for product weights. The following theorem generalizes this result to prime powers N and to arbitrary weights.

Theorem 4. *Let $N = b^m$ be a prime power, let $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$, be general weights, and let the worst-case error $e_{N,s}$ in the weighted Korobov space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ be defined as in Section 2. Let $\boldsymbol{z}^0 \in \{0, 1, \ldots, N-1\}^s$ be an arbitrary initial vector. Then Algorithm 1 applied with $w_1 = \cdots = w_s = 0$ constructs $\boldsymbol{z} = (z_1, \ldots, z_s)$ such that, for $\lambda \in (\frac{1}{\alpha}, 1]$, the squared worst-case error $e^2_{N,s}(\boldsymbol{z})$ satisfies*

$$e^2_{N,s}((z_1, \ldots, z_s)) \le \left(\sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(2\zeta(\alpha\lambda))^{|\mathfrak{u}|}}{b^m}\right)^{\frac{1}{\lambda}}.$$

In particular, $e_{N,s}(\boldsymbol{z}) \in \mathcal{O}(N^{-\alpha/2+\delta})$ for δ arbitrarily close to zero, where the implied constant is independent of s if

$$C_\delta := \sup_{s \in \mathbb{N}} \left[2 \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha-2\delta}} \left(2\zeta\left(\frac{\alpha}{\alpha-2\delta}\right)\right)^{|\mathfrak{u}|}\right] < \infty.$$

Proof. The result follows immediately by considering Algorithm 1, Theorem 1, and Corollary 2 for the special case $w_1 = w_2 = \cdots = w_d = 0$. □

4 Fast SCS construction for product weights

For product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_s) \in \{0, 1, \ldots, N-1\}^s$, the squared worst-case error can be written as

$$e^2_{N,s}(\boldsymbol{\xi}) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left(1 + \gamma_j\, \omega\left(\left\{\frac{k\xi_j}{N}\right\}\right)\right),$$

where ω is a real-valued function satisfying $\omega(x) = \omega(1-x)$ for $x \in [0,1]$, cf., e.g., [16] and [17]. For our purposes, we assume that the function ω can be evaluated in N distinct arguments at a cost of at most $\mathcal{O}(N \log N)$; this assumption is justified for the setting studied in this paper, see, e.g., [19]. Now, for one step of the reduced SCS algorithm with $w_d < m$, we need to find a component $z_d \in \mathcal{Z}_{N,w_d}$ such that $e^2_{N,s}((Y_1 z_1, \ldots, Y_{d-1} z_{d-1}, Y_d z_d, z^0_{d+1}, \ldots, z^0_s))$ is minimized as a function of z_d. This is obviously equivalent to minimizing

$$\sum_{k=0}^{N-1} \left(1 + \gamma_d\, \omega\left(\left\{\frac{kY_d z_d}{N}\right\}\right)\right) q_d(k) = \sum_{k=0}^{N-1} q_d(k) + \gamma_d \sum_{k=0}^{N-1} \omega\left(\left\{\frac{kY_d z_d}{N}\right\}\right) q_d(k)$$

as a function of z_d, where $Y_d = b^{w_d}$ and

$$q_d(k) := \left[\prod_{j=1}^{d-1} \left(1 + \gamma_j\, \omega\left(\left\{\frac{kY_j z_j}{N}\right\}\right)\right)\right] \left[\prod_{j=d+1}^{s} \left(1 + \gamma_j\, \omega\left(\left\{\frac{kz^0_j}{N}\right\}\right)\right)\right].$$

Thus, the component z_d is given by the $z \in \mathcal{Z}_{N,w_d}$ which minimizes

$$T_d(z) = \sum_{k=0}^{N-1} \omega\left(\frac{kb^{w_d} z \bmod N}{N}\right) q_d(k).$$

In the following we write $\mathbb{Z}_N$ to denote the set of integers $\{0, 1, \ldots, N-1\}$. We note that $T_d(z)$ can be calculated simultaneously for all $z \in \mathcal{Z}_{N,w_d}$ as the matrix-vector product of the reduced matrix

$$\Omega_{b^m,w} := \left[\omega\left(\frac{kb^w z \bmod N}{N}\right)\right]_{\substack{z\in\mathcal{Z}_{N,w}\\ k\in\mathbb{Z}_N}} = \left[\omega\left(\frac{kb^w z \bmod b^m}{b^m}\right)\right]_{\substack{z\in\mathcal{Z}_{N,w}\\ k\in\mathbb{Z}_{b^m}}}$$

with $w = w_d$, and the vector $\boldsymbol{q}_d = (q_d(0), q_d(1), \ldots, q_d(N-1)) \in \mathbb{R}^N$.

4.1 The block-circulant structure of $\Omega_{b^m,w}$

Due to the reduction of the search space from $\mathbb{U}_{b^m} = \{z \in \{1, 2, \ldots, b^m - 1\} : \gcd(z, b) = 1\}$ to

$$\mathcal{Z}_{N,w} = \{z \in \{1, 2, \ldots, b^{m-w} - 1\} : \gcd(z, b) = 1\} = \mathbb{U}_{b^{m-w}},$$

with $w < m$, the matrix $\Omega_{b^m,w}$ is of special block-circulant structure which allows a fast computation of the above matrix-vector product. The following two theorems, which will be shown in a combined proof, illustrate this structure for the cases $b \neq 2$ and $b = 2$, respectively.

In the following, for $t, r \geq 1$, we denote by $\langle\langle g \rangle\rangle_{b^r}$ the set $\{g^i \bmod b^r \mid 0 \leq i \leq \frac{\varphi(b^r)}{2} - 1\}$, and furthermore set $\mathbf{1}_t \otimes A$ and $A \otimes \mathbf{1}_t$ as the vertical and horizontal stacking of t instances of the matrix A, respectively.

Theorem 5. *For $b \neq 2$, $w < m$, and $\omega : [0,1] \to \mathbb{R}$ such that $\omega(x) = \omega(1-x)$, the reduced matrix*

$$\Omega_{b^m,w} := \left[\omega\left(\frac{kb^w z \bmod b^m}{b^m}\right)\right]_{\substack{z\in\mathcal{Z}_{N,w}\\ k\in\mathbb{Z}_{b^m}}}$$

can, with respect to a generator g of $\mathbb{U}_{b^m}$, be reordered to

$$\Omega^{\langle g\rangle}_{b^m,w} := \left[\mathbf{1}_{b^0} \otimes B^{\langle g\rangle}_{b^{m-w}} \,\middle|\, \mathbf{1}_{b^1} \otimes B^{\langle g\rangle}_{b^{m-w-1}} \,\middle|\, \ldots \,\middle|\, \mathbf{1}_{b^{m-w-1}} \otimes B^{\langle g\rangle}_{b^1} \,\middle|\, \mathbf{1}_{\varphi(b^{m-w})} \otimes (\omega(0) \otimes \mathbf{1}_{b^w})\right],$$

where for $\ell \in \{w+1, w+2, \ldots, m\}$ and $r \in \{1, \ldots, m\}$ we define

$$B^{\langle g\rangle}_{b^{\ell-w}} := \begin{bmatrix} M^{\langle g\rangle}_{b^{\ell-w}} \otimes \mathbf{1}_{2b^w} \\ M^{\langle g\rangle}_{b^{\ell-w}} \otimes \mathbf{1}_{2b^w} \end{bmatrix} \quad \text{and} \quad M^{\langle g\rangle}_{b^r} := \left[\omega\left(\frac{kz \bmod b^r}{b^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}}.$$

Thus, $B^{\langle g\rangle}_{b^{\ell-w}}$, and therefore also $\Omega^{\langle g\rangle}_{b^m,w}$, consists of circulant blocks $M^{\langle g\rangle}_{b^{\ell-w}}$.

Theorem 6. *For $b = 2$, $w < m$, and $\omega : [0,1] \to \mathbb{R}$ such that $\omega(x) = \omega(1-x)$, the reduced matrix*

$$\Omega_{2^m,w} := \left[\omega\left(\frac{k2^w z \bmod 2^m}{2^m}\right)\right]_{\substack{z\in\mathcal{Z}_{N,w}\\ k\in\mathbb{Z}_{2^m}}}$$

can be reordered with respect to the divisors of 2^m and $g = 5$ as

$$\Omega^{\langle g\rangle}_{2^m,w} := \left[\mathbf{1}_{2^0} \otimes B^{\langle g\rangle}_{2^{m-w}} \,\middle|\, \mathbf{1}_{2^1} \otimes B^{\langle g\rangle}_{2^{m-w-1}} \,\middle|\, \ldots \,\middle|\, \mathbf{1}_{2^{m-w-2}} \otimes B^{\langle g\rangle}_{2^2} \,\middle|\, \mathbf{1}_{2^{m-w-1}} \otimes (\omega(1/2) \otimes \mathbf{1}_{2^w}) \,\middle|\, \mathbf{1}_{2^{m-w-1}} \otimes (\omega(0) \otimes \mathbf{1}_{2^w})\right],$$

where for $\ell \in \{w+2, w+3, \ldots, m\}$ and $r \in \{1, \ldots, m\}$ we define

$$B^{\langle g\rangle}_{2^{\ell-w}} := \begin{bmatrix} M^{\langle g\rangle}_{2^{\ell-w}} \otimes \mathbf{1}_{2^{w+1}} \\ M^{\langle g\rangle}_{2^{\ell-w}} \otimes \mathbf{1}_{2^{w+1}} \end{bmatrix} \quad \text{and} \quad M^{\langle g\rangle}_{2^r} := \left[\omega\left(\frac{kz \bmod 2^r}{2^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{2^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{2^r}}}.$$

Thus, $B^{\langle g\rangle}_{2^{\ell-w}}$, and therefore also $\Omega^{\langle g\rangle}_{2^m,w}$, consists of circulant blocks $M^{\langle g\rangle}_{2^{\ell-w}}$.

Proof. To prove Theorems 5 and 6 consider Theorems 4.2 and 4.3 in [2] which show how the unreduced matrix $\Omega_{b^m} = \Omega_{b^m,0}$ can be reordered with respect to the divisors of b^m based on the circulant matrices $M^{\langle g\rangle}_{b^r}$. Since the matrix $\Omega_{b^m,w}$ can be obtained from Ω_{b^m} by replacing k by kb^w and only using the rows for which $z \in \mathbb{U}_{b^{m-w}}$, the matrix $\Omega_{b^m,w}$ inherits the structure of Ω_{b^m}. This becomes evident by considering the above substitution for the circulant matrices $M^{\langle g\rangle}_{b^r}$. For $0 \le w < r$ we obtain that

$$\left[\omega\left(\frac{kzb^w \bmod b^r}{b^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}} = \left[\omega\left(\frac{kz \bmod b^{r-w}}{b^{r-w}}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}}. \tag{4}$$

Next, note that for $b = 2$ and $b \neq 2$ the set $\mathbb{U}_{b^r}$ can be written as

$$\mathbb{U}_{b^r} = \langle\langle g\rangle\rangle_{b^r} \cup (-1)\langle\langle g\rangle\rangle_{b^r},$$

where $(-1)\langle\langle g\rangle\rangle_{b^r} = \{-g^i \bmod b^r \mid 0 \le i \le \frac{\varphi(b^r)}{2} - 1\}$. For $b \ne 2$ this follows from the fact that for the cyclic group $\mathbb{U}_{b^r}$ with generator g we always have that $-1 \equiv g^{\varphi(b^r)/2}$. Hence, coming back to Equation (4), we see that the two variables z and k iterate through the sets

$$\underbrace{\langle\langle a\rangle\rangle_{b^{r-w}} \cup (-1)\langle\langle a\rangle\rangle_{b^{r-w}} \cup \langle\langle a\rangle\rangle_{b^{r-w}} \cup \ldots \cup \langle\langle a\rangle\rangle_{b^{r-w}} \cup (-1)\langle\langle a\rangle\rangle_{b^{r-w}}}_{b^w \text{ times}}$$

for $a = g$ and $a = g^{-1}$, respectively. Thus, for $0 \le w \le r-2$, the matrices $M_{b^r}^{\langle g\rangle}$ with respect to the substitution $\tilde{k} = kb^w$ are given by

$$\begin{aligned}
M_{b^r}^{\langle g\rangle} &= \left[\omega\left(\frac{kzb^w \bmod b^r}{b^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}} = \left[\omega\left(\frac{kz \bmod b^{r-w}}{b^{r-w}}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}} \\
&= \underbrace{\left.\begin{bmatrix} M_{b^{r-w}}^{\langle g\rangle} & M_{b^{r-w}}^{\langle g\rangle} & \cdots & M_{b^{r-w}}^{\langle g\rangle} \\ M_{b^{r-w}}^{\langle g\rangle} & M_{b^{r-w}}^{\langle g\rangle} & \cdots & M_{b^{r-w}}^{\langle g\rangle} \\ \vdots & \vdots & \ddots & \vdots \\ M_{b^{r-w}}^{\langle g\rangle} & M_{b^{r-w}}^{\langle g\rangle} & \cdots & M_{b^{r-w}}^{\langle g\rangle} \end{bmatrix}\right\}}_{b^w \text{ times}} b^w \text{ times} = \left.\begin{bmatrix} M_{b^{r-w}}^{\langle g\rangle} \otimes \mathbf{1}_{b^w} \\ M_{b^{r-w}}^{\langle g\rangle} \otimes \mathbf{1}_{b^w} \\ \vdots \\ M_{b^{r-w}}^{\langle g\rangle} \otimes \mathbf{1}_{b^w} \end{bmatrix}\right\} b^w \text{ times},
\end{aligned}$$

where the penultimate equality follows through the reasoning above and since $\omega(x) = \omega(1-x)$. The same statement holds true for $w = r-1$ and $b \ne 2$. For the case $b = 2$ and $w = r-1$ we obtain a special case since then the above substitution yields

$$\begin{aligned}
M_{2^r}^{\langle g\rangle} &= \left[\omega\left(\frac{kz2^{r-1} \bmod 2^r}{2^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{2^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{2^r}}} = \left[\omega\left(\frac{kz \bmod 2}{2}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{2^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{2^r}}} \\
&= \omega(1/2)\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{2^{r-2}\times 2^{r-2}}.
\end{aligned}$$

For $w \ge r$ the matrix $M_{b^r}^{\langle g\rangle}$ reduces to

$$M_{b^r}^{\langle g\rangle} = \left[\omega\left(\frac{kzb^w \bmod b^r}{b^r}\right)\right]_{\substack{z\in\langle\langle g\rangle\rangle_{b^r}\\ k\in\langle\langle g^{-1}\rangle\rangle_{b^r}}} = \omega(0)\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{\frac{\varphi(b^r)}{2}\times\frac{\varphi(b^r)}{2}}.$$

For the special case $b = 2$ there occurs the additional term $B_{2^1}^{\langle g\rangle} = [\omega(1/2)]$ in Thm. 4.3 of [2], however, for $w \ge 1$ and the substitution $\tilde{k} = kb^w$ this results in $\omega(0)$. Now, using the theorems in [2] and putting all derived cases together we obtain the structure of $\Omega_{b^m,w}^{\langle g\rangle}$ as given in Theorems 5 and 6. □

4.2 Computational complexity of the reduced SCS construction

Firstly, denote by s^* the largest integer such that $w_{s^*} < m$. In order to achieve a low computational complexity, we consider initial vectors $\boldsymbol{z}^0$ of the form

$$\boldsymbol{z}^0 = (z_1^0, \ldots, z_s^0) = (Y_1\bar{z}_1, \ldots, Y_s\bar{z}_s) \equiv (Y_1\bar{z}_1, \ldots, Y_{s^*}\bar{z}_{s^*}, 0, \ldots, 0) \bmod N \tag{5}$$

with $\bar{z}_j \in \mathcal{Z}_{N,w_j}$ for all $j \in \{1, \ldots, s\}$. The fast implementation of the reduced successive coordinate search algorithm can then be formulated as follows.

Algorithm 2 (Reduced fast SCS algorithm)**.**

1. *Precomputation:*

   (a) *Compute* $\omega\left(\frac{k}{b^m}\right)$ *for* $k = 0, 1, \ldots, b^m - 1$ *and store the results.*

   (b) *For* $\boldsymbol{z}^0$ *as in (5) and* $k = 0, 1, \ldots, b^m - 1$ *initialize* $\boldsymbol{q} = (q(0), \ldots, q(b^m - 1))$ *as*

$$q(k) := \prod_{j=1}^{s} \left(1 + \gamma_j\, \omega\left(\frac{k z_j^0 \bmod b^m}{b^m}\right)\right).$$

   (c) *Set* $d = 1$ *and* s^* *to be the largest integer such that* $w_{s^*} < m$.

While $d \leq \min\{s, s^*\}$*:*

2. *Set* $\boldsymbol{q}_d$ *via* $\boldsymbol{q}$ *by dividing out the initial choice* z_d^0 *(for* $k = 0, 1, \ldots, b^m - 1$*) such that*

$$q_d(k) = \left[\prod_{j=1}^{d-1} \left(1 + \gamma_j\, \omega\left(\left\{\frac{kY_j z_j}{N}\right\}\right)\right)\right] \left[\prod_{j=d+1}^{s} \left(1 + \gamma_j\, \omega\left(\left\{\frac{k z_j^0}{N}\right\}\right)\right)\right].$$

3. *Partition the vector* $\boldsymbol{q}_d$ *into* b^{w_d} *vectors* $\boldsymbol{q}_d^{(1)}, \ldots, \boldsymbol{q}_d^{(b^{w_d})}$ *of length* b^{m-w_d}*, where*

$$\boldsymbol{q}_d^{(\ell)} = (q_d(1 + (\ell - 1)b^{m-w_d}), \ldots, q_d(\ell\, b^{m-w_d})) \quad \text{for} \quad \ell = 1, \ldots, b^{w_d}$$

   and set $\boldsymbol{q}_d' = \boldsymbol{q}_d^{(1)} + \cdots + \boldsymbol{q}_d^{(b^{w_d})}$.

4. *Calculate* $T_d(z) = \Omega_{b^m, w}\, \boldsymbol{q}_d'$ *for all* $z \in \mathcal{Z}_{N, w_d}$ *using FFT.*

5. *Set* $z_d = \arg\min_{z \in \mathcal{Z}_{N,w_d}} T_d(z)$.

6. *Update* $\boldsymbol{q}$ *via* $\boldsymbol{q}_d$ *by multiplying with the chosen* z_d *(for* $k = 0, 1, \ldots, b^m - 1$*) such that*

$$q(k) = \left[\prod_{j=1}^{d} \left(1 + \gamma_j\, \omega\left(\left\{\frac{kY_j z_j}{N}\right\}\right)\right)\right] \left[\prod_{j=d+1}^{s} \left(1 + \gamma_j\, \omega\left(\left\{\frac{k z_j^0}{N}\right\}\right)\right)\right].$$

7. *Increase* d *by* 1.

If $s > s^*$*, then set* $z_{s^*+1} = \cdots = z_s = 1$*. The squared worst-case error is then given as*

$$e_{N,s}^2(Y_1 z_1, \ldots, Y_s z_s) = -1 + \frac{1}{b^m} \sum_{k=0}^{b^m - 1} q(k).$$

Theorem 7. *The computational complexity of Algorithm 2 is*

$$\mathcal{O}\left(m b^m + \min\{s, s^*\}\, b^m + \sum_{d=1}^{\min\{s,s^*\}} (m - w_d) b^{m-w_d}\right).$$

Proof. The first term originates from the precalculation in (a) which requires $\mathcal{O}(mb^m)$ operations. Due to the chosen form of initial vectors as in (5), the initialization of $\boldsymbol{q}$ in (b) only requires $\mathcal{O}(\min\{s, s^*\} b^m)$ operations since for $k = 0, 1, \ldots, b^m - 1$

$$q(k) = \prod_{j=1}^{s^*} \left(1 + \gamma_j\, \omega\left(\frac{k z_j^0 \bmod b^m}{b^m}\right)\right) \prod_{j=s^*+1}^{s} (1 + \gamma_j\, \omega(0)).$$

Furthermore, the updates for $\boldsymbol{q}_d$ and $\boldsymbol{q}$ in the Steps 2 and 6, respectively, can likewise be done in $\mathcal{O}(\min\{s, s^*\}b^m)$ operations. The additions in Step 3 similarly require $\mathcal{O}(\min\{s, s^*\}b^m)$ calculations. Lastly, the matrix-vector product in Step 4 can be computed in only $\mathcal{O}((m - w_d)b^{m-w_d})$ operations using FFT (see, e.g., [25, 26]) and the results of Theorems 5 and 6. This then gives the last term and proves the theorem. □

Remark 8. *Note that in the implementation of Algorithm 2, the vector* $\boldsymbol{q}$ *also has to be ordered with respect to a generator* g *as in Theorems 5 and 6 in order to exploit the special structure of the matrix* $\Omega_{b^m,w}$.

Furthermore, for initial vectors $\boldsymbol{z}^0$ as in (5), we obtain the following useful result.

Theorem 9. *Let the initial vector* $\boldsymbol{z}^0 \in \{0, 1, \ldots, N-1\}^s$ *be of the form (5) and denote by* $\boldsymbol{z}$ *the result of Algorithm 1 seeded with* $\boldsymbol{z}^0$. *Then the generating vector* $\boldsymbol{z}$ *satisfies*

$$e_{N,s}(\boldsymbol{z}) \le e_{N,s}(\boldsymbol{z}^0),$$

i.e., the constructed vector $\boldsymbol{z}$ *is always at least as good as the initial vector* $\boldsymbol{z}^0$ *with respect to the associated worst-case error.*

Proof. For this special choice of initial vectors, the statement follows directly from the formulation of Algorithm 1. Since the value of z_d^0 is in each minimization step $d \in [s]$ amongst the candidates for z_d the worst-case error $e_{N,s}$ never grows. □

5 Numerical results

In this section, the results from Sections 3 and 4 which led to the reduced fast SCS construction, stated in Algorithm 2, will be illustrated via numerical experiments. Here we consider the construction of rank-1 lattices in weighted Korobov spaces $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ of smoothness $\alpha > 1$, and, as in Section 4, we assume product weights $\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j$. For $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_s) \in \{0, 1, \ldots, N-1\}^s$ the worst-case error is then given by

$$e_{N,s}^2(\boldsymbol{\xi}) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left(1 + \gamma_j \sum_{h\in\mathbb{Z}_*} \frac{\exp(2\pi \mathtt{i} hk\xi_j/N)}{|h|^\alpha}\right), \tag{6}$$

and it is easy to check that the symmetry assumption which was previously imposed on ω is satisfied. For an even smoothness parameter α, the sum of exponentials in (6) simplifies to the Bernoulli polynomial $B_\alpha(\{k\xi_j/N\})$ modulo a constant, see, e.g., [6]. For ease of implementation, we will therefore restrict our experiments to the case $\alpha = 2$ so that the worst-case error reads

$$e_{N,s}^2(\boldsymbol{\xi}) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^{s} \left(1 + 2\pi^2 \gamma_j B_2\left(\left\{\frac{k\xi_j}{N}\right\}\right)\right).$$

Due to the connection between Korobov and (unanchored) Sobolev spaces pointed out in Section 2, the presented results remain also valid for integration in weighted Sobolev spaces using randomly shifted or tent-transformed lattice rules.

The subsequent sections are devoted to illustrating the key features of the reduced fast SCS algorithm, i.e., the error convergence rate of the constructed lattices, the computational complexity of the algorithm, and the precise worst-case errors. In order to carry out a rigorous analysis, we will always compare the obtained results with those of the reduced and unreduced CBC construction and the unreduced SCS construction as in [11]. The different algorithms have all been implemented using Matlab R2016b.

5.1 Error convergence behavior

We consider the convergence rate of the worst-case error $e_{N,s}$ for different weight sequences $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ and reduction indices w_j of the form $w_j = \lfloor c \log_b j \rfloor$ with $c > 0$. According to Corollary 3, the almost optimal error convergence rate of $\mathcal{O}(N^{-1+\delta})$ for the reduced CBC and SCS algorithm will always be achieved for $N \to \infty$. Additionally, Corollary 3 implies that a constant independent of s can be achieved provided that the chosen weights γ_j satisfy

$$\sum_{j=1}^{\infty} \gamma_j^{\frac{1}{2(1-\delta)}} \, b^{w_j} \leq \sum_{j=1}^{\infty} \gamma_j^{\frac{1}{2}} \, b^{w_j} < \infty. \tag{7}$$

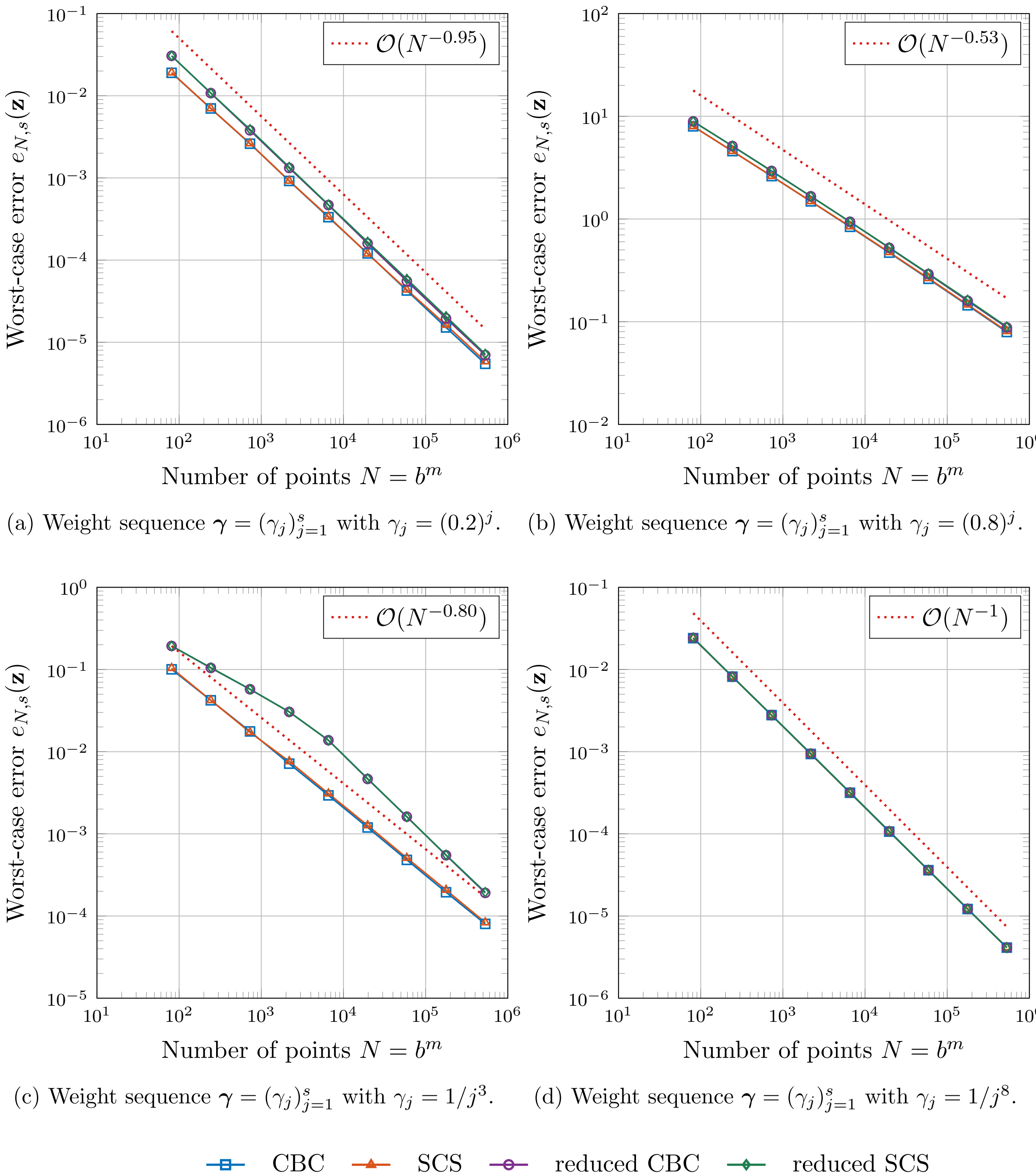
It is to be expected that parameter choices which satisfy the condition in (7) will yield a nicer error behavior also in numerical experiments, since the negative influence of high s is not present anymore. In particular, if (7) is satisfied, there should not be much difference in the error behavior of the vectors obtained by reduced and unreduced algorithms, respectively, since the negative influence of the w_j washes away. Nevertheless, there are situations where the almost optimal convergence order $\mathcal{O}(N^{-1+\delta})$ is only visible for larger values of N than those considered in our numerical experiments. In that sense, our numerical results are to be understood as illustrating a kind of "pre-asymptotic" error behavior.

Here, we consider two common types of weight sequences with the general form $\gamma_j = q^j$ with $0 < q < 1$ or $\gamma_j = 1/j^a$ with $a > 1$. For the former type of weights, Corollary 3 assures the optimal error convergence rate, with constant independent of s, for any q. For the latter type, we see that since

$$\sum_{j=1}^{\infty} \gamma_j^{\frac{1}{2}} \, b^{w_j} = \sum_{j=1}^{\infty} j^{-\frac{a}{2}} \, b^{\lfloor c \log_b j \rfloor} \asymp \sum_{j=1}^{\infty} j^{-\frac{a}{2}} \, b^{c \log_b j} = \sum_{j=1}^{\infty} j^{c-\frac{a}{2}},$$

the convergence of the series on the right-hand side of (7) is only guaranteed for small δ if $a > 2(1+c)$. In Figures 1 and 2 we display the results of numerical experiments using different weights γ_j for a moderate and rapid reduction of $w_j = \lfloor 2 \log_b j \rfloor$ and $\lfloor \frac{7}{2} \log_b j \rfloor$, respectively. The generating vectors $\boldsymbol{z}$ are constructed by the reduced and unreduced versions of both the CBC and the SCS construction, where the initial vector for the reduced and unreduced SCS algorithm is fixed to $\boldsymbol{z}^0 = (Y_1, \ldots, Y_s)$ and $\boldsymbol{z}^0 = (1, \ldots, 1)$, respectively.

Error convergence in the Korobov space with $s = 100, \alpha = 2, b = 3, w_j = \lfloor 2 \log_b j \rfloor$.



(a) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = (0.2)^j$.

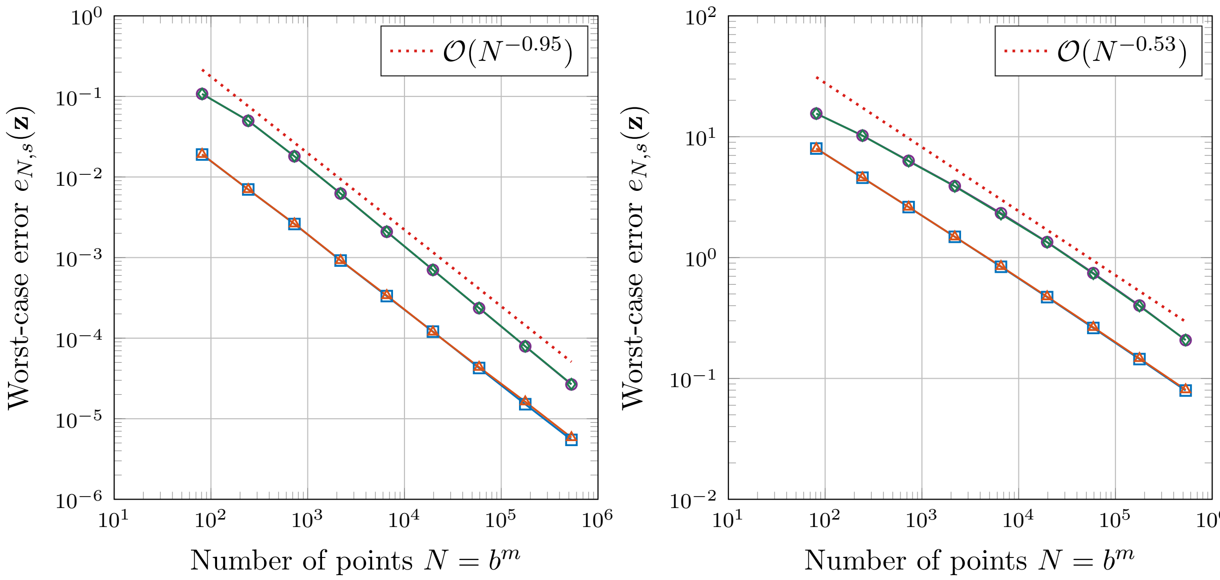
(b) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = (0.8)^j$.

(c) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = 1/j^3$.

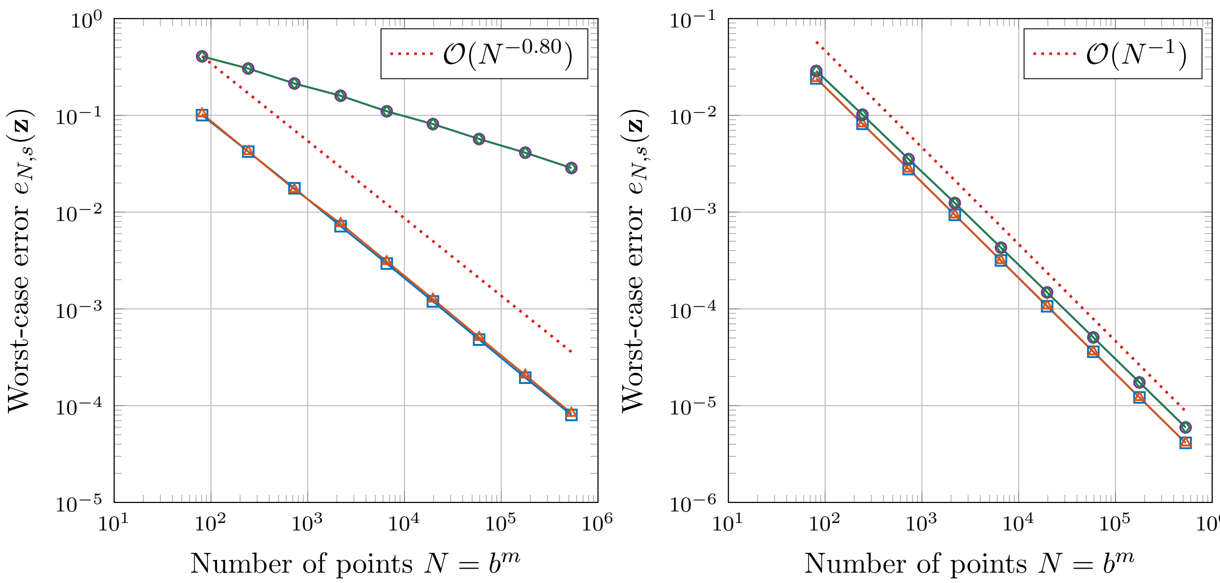
(d) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = 1/j^8$.

Figure 1: Convergence of the worst-case error $e_{N,s}(\boldsymbol{z})$ in the weighted Korobov space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ of smoothness $\alpha = 2$ with $s = 100, b = 3$ and integer sequence $w_j = \lfloor 2 \log_b j \rfloor$. The generating vector $\boldsymbol{z}$ is constructed via the reduced and unreduced CBC construction and the reduced and unreduced SCS algorithm, respectively.

Error convergence in the Korobov space with $s = 100, \alpha = 2, b = 3, w_j = \lfloor \frac{7}{2} \log_b j \rfloor$.



(a) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = (0.2)^j$. (b) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = (0.8)^j$.



(c) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = 1/j^3$. (d) Weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^s$ with $\gamma_j = 1/j^8$.

Figure 2: Convergence of the worst-case error $e_{N,s}(\boldsymbol{z})$ in the weighted Korobov space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ of smoothness $\alpha = 2$ with $s = 100, b = 3$ and integer sequence $w_j = \lfloor \frac{7}{2} \log_b j \rfloor$. The generating vector $\boldsymbol{z}$ is constructed via the reduced and unreduced CBC construction and the reduced and unreduced SCS algorithm, respectively.

As expected, Figures 1 and 2 illustrate that for weight sequences of geometric decay the convergence order is the same for the reduced and unreduced algorithms (see Cases (a) and (b) in both figures). Furthermore, note that the weights $\gamma_j = 1/j^3$ do not satisfy condition (7) for any of the chosen w_j such that, as we expected, the pre-asymptotic convergence order displayed by the unreduced CBC and SCS constructions is better than that of the reduced CBC and SCS constructions. This becomes evident by considering Case (c) in Figures 1 and 2. Note that the choice of $\gamma_j = 1/j^8$ and $w_j = \lfloor \frac{7}{2} \log_b j \rfloor$ also does not satisfy $a > 2(1+c)$, however, we still observe almost no difference between the reduced and the unreduced algorithms (see Case (d) in Figure 2). As can be seen from Figures 1 and 2, the error rates of the reduced and unreduced version of the algorithms are the same if the condition from Corollary 3 holds, but they differ by a multiplicative constant. The larger the w_j, the larger this multiplicative constant is (compare between Figures 1 and 2). An explanation for this observation is given by identifying the observed constant with the constant $C_{s,\alpha,\gamma,\delta}$ of Corollary 3.

5.2 Timings for the reduced fast SCS algorithm

Here, we illustrate the computational complexity of the reduced fast SCS construction in Algorithm 2 which was stated in Theorem 7. For that purpose, let $b = 2$ and $N = b^m$ and let the weight sequence $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$ be given by $\gamma_j = (0.7)^j$. Note that the choice of the weights γ_j does not influence the construction cost of the considered algorithms. In Tables 1, 2 and Tables 3, 4 below, we report on the computation times for the construction of the generating vector $\boldsymbol{z}$ via the four considered algorithms for the two reductions given by $w_j = \lfloor \frac{3}{2} \log_b j \rfloor$ and $w_j = \lfloor 3 \log_b j \rfloor$, respectively. Again, the two SCS algorithms (cf. Tables 2 and 4) are seeded with initial vectors $\boldsymbol{z}^0 = (Y_1, \ldots, Y_s)$ and $\boldsymbol{z}^0 = (1, \ldots, 1)$, respectively. We emphasize that the used algorithms solely construct the generating vector $\boldsymbol{z}$ but do not calculate the worst-case error $e_{N,s}(\boldsymbol{z})$, which allows for an unbiased comparison between the considered algorithms. The computations and timings were performed on an Intel Core i5-2400S CPU with 2.5GHz using Matlab.

Table 1: Computation times (in seconds) for constructing the generating vector $\boldsymbol{z}$ using the unreduced (normal font) and reduced CBC (**bold font**) construction. The associated lattice can be used for integration in the Korobov space with $\alpha = 2, b = 2, \gamma_j = (0.7)^j$ and $w_j = \lfloor \frac{3}{2} \log_b j \rfloor$.

	$s = 50$	$s = 100$	$s = 500$	$s = 1000$	$s = 2000$
$m = 10$	0.0183 **0.00963**	0.0329 **0.00999**	0.163 **0.0106**	0.32 **0.00994**	0.64 **0.0102**
$m = 12$	0.0319 **0.0139**	0.0485 **0.0178**	0.239 **0.0295**	0.475 **0.0279**	0.944 **0.0273**
$m = 14$	0.0476 **0.0216**	0.0899 **0.0308**	0.425 **0.0806**	0.861 **0.0944**	1.74 **0.0915**
$m = 16$	0.129 **0.0428**	0.24 **0.0744**	1.21 **0.264**	2.46 **0.448**	4.72 **0.619**
$m = 18$	0.424 **0.108**	0.829 **0.178**	4.14 **0.696**	8.45 **1.33**	16.8 **2.65**
$m = 20$	2.23 **0.484**	4.22 **0.839**	21.5 **3.91**	43.2 **7.11**	87.2 **14.2**

Table 2: Computation times (in seconds) for constructing the generating vector $\boldsymbol{z}$ using the unreduced (normal font) and reduced SCS (**bold font**) construction. The associated lattice can be used for integration in the Korobov space with $\alpha = 2, b = 2, \gamma_j = (0.7)^j$ and $w_j = \lfloor \frac{3}{2} \log_b j \rfloor$.

	$s = 50$	$s = 100$	$s = 500$	$s = 1000$	$s = 2000$
$m = 10$	0.0311 **0.0186**	0.0524 **0.0155**	0.258 **0.0159**	0.509 **0.016**	1.02 **0.0155**
$m = 12$	0.0458 **0.0249**	0.0763 **0.0316**	0.381 **0.0469**	0.744 **0.0468**	1.5 **0.0458**
$m = 14$	0.088 **0.0427**	0.14 **0.0668**	0.687 **0.175**	1.37 **0.205**	2.75 **0.196**
$m = 16$	0.202 **0.0838**	0.397 **0.148**	1.93 **0.536**	3.88 **0.89**	7.73 **1.18**
$m = 18$	0.685 **0.217**	1.33 **0.376**	6.58 **1.54**	13.2 **2.91**	27.1 **5.69**
$m = 20$	3.33 **1.06**	6.62 **1.9**	33.5 **8.7**	65.9 **16.7**	135 **32.5**

Table 3: Computation times (in seconds) for constructing the generating vector $\boldsymbol{z}$ using the unreduced (normal font) and reduced CBC (**bold font**) construction. The associated lattice can be used for integration in the Korobov space with $\alpha = 2, b = 2, \gamma_j = (0.7)^j$ and $w_j = \lfloor 3 \log_b j \rfloor$.

	$s = 50$	$s = 100$	$s = 500$	$s = 1000$	$s = 2000$
$m = 10$	0.0173 **0.00298**	0.0329 **0.00206**	0.16 **0.00218**	0.323 **0.00222**	0.636 **0.00241**
$m = 12$	0.0256 **0.00358**	0.0481 **0.00365**	0.241 **0.0037**	0.48 **0.00354**	0.953 **0.00439**
$m = 14$	0.0469 **0.00803**	0.0851 **0.00761**	0.438 **0.0105**	0.856 **0.00712**	1.88 **0.00747**
$m = 16$	0.14 **0.0237**	0.239 **0.0233**	1.33 **0.0233**	2.49 **0.0227**	5.05 **0.0251**
$m = 18$	0.443 **0.0798**	0.832 **0.0897**	4.44 **0.0915**	8.54 **0.091**	17.1 **0.09**
$m = 20$	2.17 **0.38**	4.17 **0.623**	21.5 **0.643**	42.4 **0.636**	84.3 **0.628**

Table 4: Computation times (in seconds) for constructing the generating vector $\boldsymbol{z}$ using the unreduced (normal font) and reduced SCS (**bold font**) construction. The associated lattice can be used for integration in the Korobov space with $\alpha = 2, b = 2, \gamma_j = (0.7)^j$ and $w_j = \lfloor 3 \log_b j \rfloor$.

	$s = 50$	$s = 100$	$s = 500$	$s = 1000$	$s = 2000$
$m = 10$	0.0275 **0.00408**	0.0516 **0.00327**	0.256 **0.00354**	0.516 **0.00347**	1.03 **0.00329**
$m = 12$	0.0418 **0.00592**	0.0751 **0.00504**	0.383 **0.00612**	0.756 **0.00516**	1.56 **0.00794**
$m = 14$	0.0792 **0.014**	0.14 **0.0136**	0.767 **0.0163**	1.39 **0.0138**	2.82 **0.0138**
$m = 16$	0.204 **0.0441**	0.388 **0.0434**	2.09 **0.0434**	4.05 **0.0423**	8.04 **0.0462**
$m = 18$	0.686 **0.16**	1.35 **0.177**	6.89 **0.182**	13.7 **0.183**	26.8 **0.187**
$m = 20$	3.28 **0.843**	6.71 **1.4**	34.4 **1.51**	67.4 **1.37**	132 **1.36**

According to Theorem 7 and [4], the reduced fast CBC and SCS algorithm both construct a generating vector in $\mathcal{O}(mb^m + \min\{s, s^*\}\, b^m)$ operations while the unreduced constructions require $\mathcal{O}(smb^m)$ operations. Tables 1 to 4 illustrate a drastic reduction of the construction cost between the classic (fast) unreduced CBC and SCS constructions and their reduced counterparts. The magnitude of this speed-up depends on the chosen reduction indices w_j. For values of $N = 2^{18}$ and $N = 2^{20}$ and dimensions $s = 1000$ and $s = 2000$, the reduction factor ranges from 4 to 6.5 and 49 to 190 for reductions of $w_j = \left\lfloor \frac{3}{2} \log_b j \right\rfloor$ and $w_j = \lfloor 3 \log_b j \rfloor$, respectively. The higher the dimension s is, the larger the reduction of the computational cost becomes. We note that the speed-up for the reduced fast CBC construction is higher than for the reduced fast SCS algorithm, however, both lie in similar ranges. Our results show that the reduced constructions yield a considerable reduction of the computational cost while the deterioration of the associated error values is only marginal (see Subsections 5.1 and 5.3). Similar results have been observed in [4] for the reduced fast CBC construction. We would like to stress that, as we expected, our implementations of the different CBC constructions in Matlab appear to be much faster than the CBC algorithms used in [4] (cf. Table 1) which were implemented in Mathematica.

5.3 Analysis of the worst-case errors

We investigate the precise values of the worst-case errors $e_{N,s}(\boldsymbol{z})$ for generating vectors $\boldsymbol{z}$ constructed by the reduced and unreduced component-by-component constructions and the reduced SCS construction. Based on the results in Subsection 5.1, we expect the error values of the two reduced algorithms to be very similar. Let $b = 3$ and $N = b^m$ and consider a dimension of $s = 100$. In Tables 5 and 6 we display the results of numerical tests for different weight sequences $\boldsymbol{\gamma} = (\gamma_j)_{j \in \mathbb{N}}$ and reduction indices $w_j = \left\lfloor \frac{3}{2} \log_b j \right\rfloor$ and $w_j = \left\lfloor \frac{5}{2} \log_b j \right\rfloor$. For the construction of $\boldsymbol{z}$ via the reduced SCS algorithm we have to choose suitable initial vectors $\boldsymbol{z}^0$ of the form (5). In our experiments we thus consider q different seed vectors with

$$\boldsymbol{z}^0 = (Y_1 \bar{z}_1, \ldots, Y_s \bar{z}_s),$$

where the $\bar{z}_j$ are uniform random draws from the set $\mathcal{Z}_{N,w_j}$ for all $j \in \{1, \ldots, s\}$. The reduced fast SCS algorithm is then applied to all of these q seed vectors $\boldsymbol{z}_1^0, \ldots, \boldsymbol{z}_q^0$ yielding generating vectors

$z_1, \dots, z_q$. The smallest associated worst-case error of these q vectors is then displayed in the tables below. Note that the construction cost for this procedure is $\mathcal{O}(qmb^m + q \min\{s, s^*\} b^m)$, which is feasible for small q.

Table 5: $\log_{10}$-worst-case errors $\log_{10} e_{N,s}(\boldsymbol{z})$ with generating vector $\boldsymbol{z}$ being either constructed via the unreduced (normal font) and reduced (underlined) CBC construction or being the best vector constructed by the reduced SCS (**bold font**) algorithm. The errors are computed for the Korobov space with $\alpha = 2, s = 100, b = 3, q = 100$ and $w_j = \lfloor \frac{3}{2} \log_b j \rfloor$.

	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	$m=11$
	-0.4281	-0.7065	-0.9928	-1.283	-1.58	-1.881
$\gamma_j = (0.7)^j$	-0.4033	-0.685	-0.9783	-1.265	-1.564	-1.869
	-0.418	**-0.6934**	**-0.9783**	**-1.266**	**-1.559**	**-1.865**
	-1.442	-1.804	-2.162	-2.521	-2.889	-3.271
$\gamma_j = (0.5)^j$	-1.404	-1.771	-2.145	-2.502	-2.879	-3.254
	-1.422	**-1.783**	**-2.138**	**-2.497**	**-2.863**	**-3.236**
	-1.754	-2.146	-2.532	-2.923	-3.317	-3.711
$\gamma_j = 1/j^3$	-1.602	-2.008	-2.452	-2.817	-3.258	-3.66
	-1.618	**-2.037**	**-2.441**	**-2.851**	**-3.245**	**-3.635**
	-2.44	-2.904	-3.364	-3.83	-4.286	-4.75
$\gamma_j = 1/j^6$	-2.439	-2.904	-3.364	-3.828	-4.288	-4.749
	-2.44	**-2.904**	**-3.365**	**-3.831**	**-4.288**	**-4.749**

Table 6: $\log_{10}$-worst-case errors $\log_{10} e_{N,s}(\boldsymbol{z})$ with generating vector $\boldsymbol{z}$ being either constructed via the unreduced (normal font) and reduced (underlined) CBC construction or being the best vector constructed by the reduced SCS (**bold font**) algorithm. The errors are computed for the Korobov space with $\alpha = 2, s = 100, b = 3, q = 100$ and $w_j = \lfloor \frac{5}{2} \log_b j \rfloor$.

	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$	$m=11$
	-0.4281	-0.7065	-0.9928	-1.283	-1.58	-1.881
$\gamma_j = (0.7)^j$	-0.1983	-0.5021	-0.807	-1.122	-1.426	-1.747
	-0.2023	**-0.5136**	**-0.8233**	**-1.129**	**-1.437**	**-1.747**
	-1.442	-1.804	-2.162	-2.521	-2.889	-3.271
$\gamma_j = (0.5)^j$	-1.113	-1.515	-1.901	-2.33	-2.703	-3.11
	-1.116	**-1.524**	**-1.931**	**-2.317**	**-2.709**	**-3.094**
	-1.754	-2.146	-2.532	-2.923	-3.317	-3.711
$\gamma_j = 1/j^3$	-0.9724	-1.181	-1.391	-1.622	-1.919	-2.396
	-0.973	**-1.182**	**-1.392**	**-1.622**	**-1.92**	**-2.396**
	-2.44	-2.904	-3.364	-3.83	-4.286	-4.75
$\gamma_j = 1/j^6$	-2.361	-2.81	-3.268	-3.728	-4.191	-4.657
	-2.362	**-2.811**	**-3.269**	**-3.728**	**-4.191**	**-4.657**

The results in Tables 5 and 6 show that the reduced fast SCS algorithm generates lattice rules with similar errors as the reduced CBC algorithm as was to be expected from the results in Section 5.1. Furthermore, we note that it is possible to obtain better error values than with the reduced CBC construction, at the price of increased computational costs. The loss of accuracy as compared to the classic CBC construction is for both reduced algorithms only marginal. The only exception to this is the case $\gamma_j = 1/j^3$ in Table 6. As discussed in Subsection 5.1, this is most likely due to the fact that the weights γ_j do not decay fast enough (cf. Case (c) in Figure 2).

6 Walsh spaces and polynomial lattice point sets

6.1 Walsh spaces

Similar results to those for lattice point sets from the previous sections can be shown for polynomial lattice point sets over finite fields $\mathbb{F}_b$ of prime order b with modulus x^m. Here we only sketch these results and the necessary notation, as they are in analogy to those for Korobov spaces and lattice point sets.

As a quality criterion we use the worst-case error of QMC rules in a weighted Walsh space, as introduced in [8] for the case of product weights, with general weights. For a prime number b and $h \in \mathbb{N}$ define $\psi_b(h) := \lfloor \log_b(h) \rfloor$. We furthermore write

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

for $h \in \mathbb{N}_0$ and set

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

For the multivariate case with dimension $s \in \mathbb{N}$ and $\boldsymbol{h} = (h_1, \ldots, h_s)$ we set $r_\alpha(\boldsymbol{h}) = \prod_{j=1}^s r_\alpha(h_j)$. Moreover, for a nonnegative integer h, we define the h-th Walsh function ${}_b\mathrm{wal}_h : [0,1) \to \mathbb{C}$ by

$${}_b\mathrm{wal}_h(x) := \mathrm{e}^{2\pi \mathtt{i}(x_1 h_0 + \cdots + x_{a+1} h_a)/b}$$

with $x \in [0,1)$, and base b representations $h = h_0 + h_1 b + \cdots + h_a b^a$, with $h_i \in \{0, 1, \ldots, b-1\}$, and $x = \frac{x_1}{b} + \frac{x_2}{b^2} + \cdots$ (unique in the sense that infinitely many of the x_i must be different from $b-1$).

For dimension $s \geq 2$ and vectors $\boldsymbol{h} = (h_1, \ldots, h_s) \in \mathbb{N}_0^s$, and $\boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1)^s$ we define ${}_b\mathrm{wal}_{\boldsymbol{h}} : [0,1)^s \to \mathbb{C}$ by

$${}_b\mathrm{wal}_{\boldsymbol{h}}(\boldsymbol{x}) := \prod_{j=1}^{s} {}_b\mathrm{wal}_{h_j}(x_j).$$

In the following, we will consider the prime number b as fixed, and then simply write wal_h or $\mathrm{wal}_{\boldsymbol{h}}$ instead of ${}_b\mathrm{wal}_h$ or ${}_b\mathrm{wal}_{\boldsymbol{h}}$, respectively.

The weighted Walsh space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}})$ is a reproducing kernel Hilbert space with kernel function of the form

$$K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}}(\boldsymbol{x}, \boldsymbol{y}) = 1 + \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|}} r_\alpha(\boldsymbol{h}_{\mathfrak{u}}) \mathrm{wal}_{\boldsymbol{h}_{\mathfrak{u}}}(\boldsymbol{x}_{\mathfrak{u}}) \overline{\mathrm{wal}_{\boldsymbol{h}_{\mathfrak{u}}}(\boldsymbol{y}_{\mathfrak{u}})},$$

and inner product

$$\langle f, g \rangle_{K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}}} = \sum_{\mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{-1} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|}} (r_\alpha(\boldsymbol{h}_{\mathfrak{u}}))^{-1} \widetilde{f}((\boldsymbol{h}_{\mathfrak{u}}, \boldsymbol{0})) \overline{\widetilde{g}((\boldsymbol{h}_{\mathfrak{u}}, \boldsymbol{0}))},$$

where $\widetilde{f}(\boldsymbol{h}) = \int_{[0,1]^s} f(\boldsymbol{t})\overline{\mathrm{wal}_{\boldsymbol{h}}(\boldsymbol{t})}\,\mathrm{d}\boldsymbol{t}$ is the $\boldsymbol{h}$-th Walsh coefficient of f and $(\boldsymbol{h}_{\mathfrak{u}}, \boldsymbol{0}) \in \mathbb{N}^s$ denotes the vector whose j-th component is equal to the corresponding one of $\boldsymbol{h}_{\mathfrak{u}}$ if $j \in \mathfrak{u}$ and zero if $j \notin \mathfrak{u}$.

For integration in $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}})$ we use a special instance of polynomial lattice point sets over the finite field $\mathbb{F}_b$ with prime b. Polynomial lattice point sets are special examples of (t, m, s)-nets in base b, which were proposed by Niederreiter in [20] (see also [21, Ch. 4.4]). Let $\mathbb{F}_b((x^{-1}))$ be the field of formal Laurent series over $\mathbb{F}_b$ with elements of the form

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

where w is an arbitrary integer and all $t_\ell \in \mathbb{F}_b$. Note that the field of rational functions is a subfield of $\mathbb{F}_b((x^{-1}))$. We further denote by $\mathbb{F}_b[x]$ the set of all polynomials over $\mathbb{F}_b$ and define the map $\nu : \mathbb{F}_b((x^{-1})) \to [0, 1)$ by

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

There is a close connection between the base b expansions of natural numbers and the polynomial ring $\mathbb{F}_b[x]$. For $n \in \mathbb{N}_0$ with b-adic expansion $n = n_0 + n_1 b + \cdots + n_a b^a$, we associate n with the polynomial

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

Now, for given integers $m \geq 1$ and $s \geq 2$, choose $f \in \mathbb{F}_b[x]$ with $\deg(f) = m$, and let $g_1, \ldots, g_s \in \mathbb{F}_b[x]$. Then the point set $\mathcal{P}(\boldsymbol{g}, f)$ is defined as the collection of the b^m points

$$\boldsymbol{x}_n := \left(\nu\left(\frac{n\ g_1}{f}\right), \ldots, \nu\left(\frac{n\ g_s}{f}\right)\right) \quad \text{for } n \in \mathbb{F}_b[x] \text{ with } \deg(n) < m.$$

Note that one can restrict the choice of g_j for $j = 1, \ldots, s$ to the set

$$\{g \in \mathbb{F}_b[x] : \deg(g) < m\}.$$

Due to the construction principle, $\mathcal{P}(\boldsymbol{g}, f)$ is often called a polynomial lattice and a QMC rule using the point set $\mathcal{P}(\boldsymbol{g}, f)$ is referred to as a polynomial lattice rule (modulo f). The vector $\boldsymbol{g} = (g_1, \ldots, g_s)$ is called the generating vector.

For our purposes, we only consider a special case of lattice rules, namely the special choice of $f(x) = x^m$ as the modulus. With a slight misuse of notation, we shall often write x^m instead of f. Let now $\mathcal{P}(\boldsymbol{g}, x^m)$, where $\boldsymbol{g} = (g_1, \ldots, g_s) \in (\mathbb{F}_b[x])^s$, be the b^m-element polynomial lattice consisting of

$$\boldsymbol{x}_n := \left(\nu\left(\frac{n\ g_1}{x^m}\right), \ldots, \nu\left(\frac{n\ g_s}{x^m}\right)\right) \quad \text{for } n \in \mathbb{F}_b[x] \text{ with } \deg(n) < m,$$

where for $v \in \mathbb{F}_b[x]$, $v(x) = a_0 + a_1 x + \cdots + a_r x^r$, with $\deg(v) = r$, the map ν is in this particular case given by

$$\nu\left(\frac{v}{x^m}\right) := \frac{a_{\min(r,m-1)}}{b^{m-\min(r,m-1)}} + \cdots + \frac{a_1}{b^{m-1}} + \frac{a_0}{b^m} \in [0, 1).$$

Note that $\nu(v/x^m) = \nu((v \,(\mathrm{mod}\ x^m))/x^m)$. We refer to [9, Chapter 10] for more information on polynomial lattice point sets.

In the following we write, for a nonnegative integer h with base b representation $\sum_{i=0}^{a} h_i b^i$,

$$\mathrm{tr}_m(h) = \mathrm{tr}_m(h)(x) := h_0 + h_1 x + \cdots + h_{m-1} x^{m-1},$$

where the h_i with $i > a$ are set equal to zero. For vectors of nonnegative integers $\boldsymbol{h} \in \mathbb{N}^s$, $\mathrm{tr}_m(\boldsymbol{h}) \in (\mathbb{F}_b[x])^s$ is defined component-wise.

The worst-case error of a polynomial lattice rule based on $\mathcal{P}(\boldsymbol{g}, x^m)$ with $\boldsymbol{g} \in (\mathbb{F}_b[x])^s$ in the weighted Walsh space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}})$ is given by (see [5])

$$e_{N,s}^2(\boldsymbol{g}) = \sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}} \sum_{\boldsymbol{h}_{\mathfrak{u}} \in \mathcal{D}_{\mathfrak{u}}} \prod_{j \in \mathfrak{u}} b^{-\alpha \psi_b(h_j)},$$

where

$$\mathcal{D}_{\mathfrak{u}}(\boldsymbol{g}_{\mathfrak{u}}) = \mathcal{D}_{\mathfrak{u}} := \left\{ \boldsymbol{h}_{\mathfrak{u}} \in \mathbb{N}^{|\mathfrak{u}|} : \ \mathrm{tr}_m(\boldsymbol{h}_{\mathfrak{u}}) \cdot \boldsymbol{g}_{\mathfrak{u}} \equiv 0 \, (x^m) \right\},$$

and for $\boldsymbol{v} = (v_1, \ldots, v_s)$ and $\boldsymbol{u} = (u_1, \ldots, u_s)$ in $(\mathbb{F}_b[x])^s$ we define the vector product by $\boldsymbol{v} \cdot \boldsymbol{u} := \sum_{i=1}^{s} v_i u_i$.

6.2 The reduced SCS algorithm for polynomial lattice rules

Let us now assume again that $f(x) = x^m$ for some integer m and that we are given weights $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$, and a non-decreasing sequence of integers w_j with $w_1 \le w_2 \le w_3 \le \cdots$.

Then we define the restricted search set for the j-th component of the generating vector $\boldsymbol{g}$ as

$$\mathcal{G}_{N,w_j} = \begin{cases} \{g \in \mathbb{F}_b[x] : 0 \le \deg(g) < m - w_j \text{ and } \gcd(g,f) = 1\} & \text{if } w_j < m, \\ \{1 \in \mathbb{F}_b[x]\} & \text{if } w_j \ge m, \end{cases}$$

and note that these sets depend on the integers w_j. Additionally, we note that the cardinality of the search space is $|\mathcal{G}_{N,w_j}| = b^{m-w_j-1}$. Moreover, we put $Y_j(x) = x^{w_j}$, and again with a misuse of notation we sometimes write $Y_j = x^{w_j}$.

We then consider the following algorithm for the construction of the generating vector $\boldsymbol{g}$ based on some initial vector $\boldsymbol{g}^0 \in (\mathbb{F}_b[x])^s$.

Algorithm 3. *Let $f \in \mathbb{F}_b[x]$, $f(x) = x^m$ for a fixed $m \in \mathbb{N}$, let $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$ be general weights, and let the worst-case error $e_{N,s}$ in the weighted Walsh space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}}^{\mathrm{wal}})$ be defined as above. Furthermore, let $w_1 \le w_2 \le \cdots \le w_s$ and $Y_j(x) = x^{w_j}$ for $j \in \{1, \ldots, s\}$. Then we construct the generating vector $\boldsymbol{g} = (Y_1 g_1, \ldots, Y_s g_s)$ as follows.*

- ***Input:*** *Starting vector $\boldsymbol{g}^0 = (g_1^0, \ldots, g_s^0) \in \{g \in \mathbb{F}_b[x] : \deg(g) < m\}^s$.*
- *For $d \in [s]$ assume $g_1, \ldots, g_{d-1}$ have already been selected. Then choose $g_d \in \mathcal{G}_{N,w_d}$ such that $e_{N,s}^2((Y_1 g_1, \ldots, Y_{d-1} g_{d-1}, Y_d g_d, g_{d+1}^0, \ldots, g_s^0))$ is minimized as a function of g_d.*
- *Increase d until $g_1, \ldots, g_s$ are found.*

Theorem 10. *Let the assumptions in Algorithm 3 hold. Let $\boldsymbol{g} = (Y_1 g_1, \ldots, Y_s g_s)$ be constructed by Algorithm 3. Then, for $\lambda \in (\frac{1}{\alpha}, 1]$, the squared worst-case error $e_{N,s}^2(\boldsymbol{g})$ satisfies*

$$e_{N,s}^2((Y_1 g_1, \ldots, Y_s g_s)) \le \left(\sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(\mu_b(\alpha\lambda))^{|\mathfrak{u}|}}{b^{\max(0, m-w_d)}} \right)^{\frac{1}{\lambda}}.$$

Proof. The proof works analogously to the proof of Theorem 1. □

The following corollary is derived in a similar way from Theorem 10 as Corollary 2 is derived from Theorem 1.

Corollary 11. *Let the assumptions in Algorithm 3 hold. Let $\boldsymbol{g} = (Y_1 g_1, \ldots, Y_s g_s)$ be constructed by Algorithm 3. Then we have for all $\delta \in (0, \frac{\alpha-1}{2}]$ that*

$$e_{N,s}(\boldsymbol{g}) \leq C_{s,\alpha,\boldsymbol{\gamma},\delta}\, N^{-\alpha/2+\delta},$$

where

$$C_{s,\alpha,\boldsymbol{\gamma},\delta} := \left(2 \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha-2\delta}} \left(\mu_b\left(\frac{\alpha}{\alpha-2\delta}\right)\right)^{|\mathfrak{u}|} b^{w_d}\right)^{\alpha/2-\delta}.$$

For $\delta \in (0, \frac{\alpha-1}{2}]$ and $q \geq 0$, define

$$C_{\delta,q} := \sup_{s \in \mathbb{N}} \left[\frac{2}{s^q} \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha-2\delta}} \left(\mu_b\left(\frac{\alpha}{\alpha-2\delta}\right)\right)^{|\mathfrak{u}|} b^{w_d}\right].$$

If $C_{\delta,q} < \infty$ for some $\delta \in (0, \frac{\alpha-1}{2}]$ and $q \geq 0$ then

$$e_{N,s}(\boldsymbol{g}) \leq (s^q C_{\delta,q})^{\alpha/2-\delta} N^{-\alpha/2+\delta}.$$

If $C_{\delta,0} < \infty$ for some $\delta \in (0, \frac{\alpha-1}{2}]$ then

$$e_{N,s}(\boldsymbol{g}) \leq (C_{\delta,0})^{\alpha/2-\delta} N^{-\alpha/2+\delta}.$$

For the following corollary, which is shown analogously to Corollary 3, we again assume product weights, i.e., $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$ for $\mathfrak{u} \subseteq [s]$, where the γ_j are elements of an infinite, non-increasing sequence of positive reals, $(\gamma_j)_{j \geq 1}$.

Corollary 12. *Let the assumptions in Algorithm 3 hold. Let $\boldsymbol{g} = (Y_1 g_1, \ldots, Y_s g_s)$ be constructed by Algorithm 3. Then we have for all $\delta \in (0, \frac{\alpha-1}{2}]$ that*

$$e_{N,s}(\boldsymbol{g}) \leq C_{s,\alpha,\boldsymbol{\gamma},\delta}\, N^{-\alpha/2+\delta},$$

where

$$C_{s,\alpha,\boldsymbol{\gamma},\delta} := \left(\left(\sum_{d=1}^{s} \gamma_d^{\frac{1}{\alpha-2\delta}} b^{w_d}\right)\left(2\mu_b\left(\frac{\alpha}{\alpha-2\delta}\right)\right) \prod_{d=1}^{s-1} \left(1 + \gamma_d^{\frac{1}{\alpha-2\delta}} \mu_b\left(\frac{\alpha}{\alpha-2\delta}\right)\right)\right)^{\alpha/2-\delta}.$$

Furthermore, the constant $C_{s,\alpha,\boldsymbol{\gamma},\delta}$ is bounded independently of the dimension s if

$$\sum_{d=1}^{\infty} \gamma_d^{\frac{1}{\alpha-2\delta}} b^{w_d} < \infty.$$

By setting all w_j equal to zero in Algorithm 3, we obtain an (unreduced) SCS algorithm, and the corresponding analogous results in Theorem 10 and Corollaries 11 and 12. We would like to point out that an SCS algorithm for the polynomial lattice case has not existed previously.

Theorem 13. *Let $f \in \mathbb{F}_b[x]$, $f(x) = x^m$ for a fixed $m \in \mathbb{N}$, let $\gamma_{\mathfrak{u}}$, $\mathfrak{u} \subseteq [s]$, be general weights, and let the worst-case error $e_{N,s}$ in the weighted Walsh space $\mathcal{H}(K_{s,\alpha,\boldsymbol{\gamma}})$ be defined as above. Let $\boldsymbol{g}^0 \in (\mathbb{F}_b[x])^s$ be an arbitrary initial vector. Then Algorithm 3 applied with $w_1 = \cdots = w_s = 0$ constructs $\boldsymbol{g} = (g_1, \ldots, g_s)$ such that, for $\lambda \in (\frac{1}{\alpha}, 1]$, the squared worst-case error $e^2_{N,s}(\boldsymbol{g})$ satisfies*

$$e^2_{N,s}((g_1, \ldots, g_s)) \leq \left(\sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\lambda} \frac{2(\mu_b(\alpha\lambda))^{|\mathfrak{u}|}}{b^m} \right)^{\frac{1}{\lambda}}.$$

In particular, $e_{N,s}(\boldsymbol{g}) \in \mathcal{O}(N^{-\alpha/2+\delta})$ for δ arbitrarily close to zero, where the implied constant is independent of s if

$$C_\delta := \sup_{s \in \mathbb{N}} \left[2 \sum_{d=1}^{s} \sum_{d \in \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{\frac{1}{\alpha - 2\delta}} \left(\mu_b \left(\frac{\alpha}{\alpha - 2\delta} \right) \right)^{|\mathfrak{u}|} \right] < \infty.$$

6.3 Fast implementation of the reduced SCS algorithm for polynomial lattice points

By using the same theory that was used in [4, Section 5], it is possible to obtain a fast implementation of the SCS algorithm also for the polynomial lattice rule case. Indeed, the precomputation outlined in Algorithm 2 can be done similarly for polynomial lattice points by using an analogous error expression that was shown in [5]. Furthermore, as outlined for the reduced CBC construction of polynomial lattice rules in [4], the matrix-vector multiplication can be implemented such that it uses a number of operations that exceeds the order of magnitude in the lattice case only by one logarithmic factor. These observations lead to the following theorem.

Theorem 14. *Algorithm 3 can be implemented such that its computational cost is of order*

$$\mathcal{O} \left(mb^m + \min\{s, s^*\}\, b^m + \sum_{j=1}^{\min\{s,s^*\}} (m - w_d)^2 b^{m-w_d} \right).$$

7 Conclusion

In this paper, we studied a combination of the SCS algorithm introduced in [11], and the reduced construction approach introduced in [4], with the goal of pooling the advantages of these two methods: by the reduced construction method, we can drastically reduce the computational cost compared to the unreduced algorithm, and by an SCS construction we may obtain better numerical error values for the corresponding integration rules. We showed that our new algorithm yields generating vectors of lattice rules achieving an almost optimal convergence rate, where the weights in the function space can help in overcoming the curse of dimensionality. By our new results, we extended previous results to arbitrary weights and non-prime numbers of points. Furthermore, the considered algorithms were implemented in an efficient way using a modern programming language; numerical tests confirm our main results. Similar observations hold for the case of polynomial lattice rules. It would be interesting to study further improvements on CBC or SCS algorithms, for example the choice of good initial vectors $\boldsymbol{z}^0$. These problems are left open for future research.

Acknowledgements

P. Kritzer is supported by the Austrian Science Fund (FWF): Project F5506-N26, which is part of the Special Research Program "Quasi-Monte Carlo Methods: Theory and Applications". P.

Kritzer furthermore gratefully acknowledges the partial support of the Erwin Schrödinger International Institute for Mathematics and Physics (ESI) in Vienna under the thematic programme "Tractability of High Dimensional Problems and Discrepancy".

The authors acknowledge the support of the National Science Foundation (NSF) under Grant DMS-1638521 to the Statistical and Applied Mathematical Sciences Institute.

References

[1] N. Aronszajn. *Theory of reproducing kernels.* Trans. Amer. Math. Soc. 68 (1950) 337–404.

[2] R. Cools, F.Y. Kuo, D. Nuyens. *Constructing embedded lattice rules for multivariate integration.* SIAM Journal on Scientific Computing, 28 (16), 2162–2188, 2006.

[3] J. Dick. *On the convergence rate of the component-by-component construction of good lattice rules.* J. Complexity 20, 493–522, 2004.

[4] J. Dick, P. Kritzer, G. Leobacher, F. Pillichshammer. *A reduced fast component-by-component construction of lattice points for integration in weighted spaces with fast decreasing weights.* J. Comput. Appl. Math. 276, 1–15, 2015.

[5] J. Dick, F.Y. Kuo, F. Pillichshammer, and I.H. Sloan. *Construction algorithms for polynomial lattice rules for multivariate integration.* Math. Comp. 74, 1895–1921, 2005.

[6] J. Dick, F.Y. Kuo, I.H. Sloan. *High-dimensional integration—the quasi-Monte Carlo way.* Acta Numer. 22, 133–288, 2013.

[7] J. Dick, D. Nuyens, and F. Pillichshammer. *Lattice rules for nonperiodic smooth integrands.* Numer. Math. 126, 259–291, 2014.

[8] J. Dick and F. Pillichshammer. *Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces.* J. Complexity 21, 149–195, 2005.

[9] J. Dick and F. Pillichshammer. *Digital Nets and Sequences. Discrepancy Theory and Quasi-Monte Carlo Integration.* Cambridge University Press, Cambridge, 2010.

[10] J. Dick, I.H. Sloan, X. Wang, and H. Woźniakowski. *Good lattice rules in weighted Korobov spaces with general weights.* Numer. Math. 103, 63–97, 2006.

[11] A. Ebert, H. Leövey, D. Nuyens. *Successive Coordinate Search and Component-by-Component Construction of Rank-1 Lattice Rules.* To appear in: P. Glynn, A. Owen (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2016.* Springer, 2018.

[12] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities.* Cambridge University Press, Cambridge, 1934.

[13] F.J. Hickernell. *A generalized discrepancy and quadrature error bound.* Math. Comp. 67, 299–322, 1998.

[14] F.J. Hickernell and H. Woźniakowski. *Integration and approximation in arbitrary dimensions. High dimensional integration.* Adv. Comput. Math. 12, 25–58, 2000.

[15] J. L. W. V. Jensen. *Sur les fonctions convexes et les inégalités entre les valeurs moyennes.* Acta Math. 30, 175–193, 1906.

[16] F.Y. Kuo and S. Joe. *Component-by-component construction of good QMC rules with a composite number of quadrature points.* J. Complexity 18, 943–976, 2002.

[17] F.Y. Kuo. *Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces.* J. Complexity 19, 301–320, 2003.

[18] C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling.* Springer Series in Statistics, Springer, New York, 2009.

[19] G. Leobacher, F. Pillichshammer. *Introduction to Quasi-Monte Carlo Integration and Applications.* Compact Textbooks in Mathematics, Birkhäuser, Cham, 2014.

[20] H. Niederreiter. *Low discrepancy point sets obtained by digital constructions over finite fields.* Czechoslovak Math. J. 42, 143–166, 1992.

[21] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM, Philadelphia, 1992.

[22] E. Novak, H. Woźniakowski. *Tractability of Multivariate Problems, Volume I: Linear Information.* EMS, Zurich, 2008.

[23] E. Novak, H. Woźniakowski. *Tractability of Multivariate Problems, Volume II: Standard Information for Functionals.* EMS, Zurich, 2010.

[24] E. Novak, H. Woźniakowski. *Tractability of Multivariate Problems, Volume III: Standard Information for Operators.* EMS, Zurich, 2012.

[25] 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.

[26] D. Nuyens, R. Cools. *Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points.* J. Complexity 22, 4–28, 2006.

[27] I.H. Sloan, S. Joe. *Lattice Methods for Multiple Integration.* Oxford University Press, New York and Oxford, 1994.

[28] I.H. Sloan, H. Woźniakowski. *When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals?* J. Complexity 14, 1–33, 1998.

[29] I.H. Sloan, H. Woźniakowski. *Tractability of multivariate integration for weighted Korobov classes.* J. Complexity 17, 697–721, 2001.

Authors' addresses:

Adrian Ebert
Department of Computer Science
KU Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium.
`adrian.ebert@cs.kuleuven.be`

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`