Function integration, reconstruction and approximation using rank-1 lattices

Frances y Kuo, Giovanni Migliorati, Fabio Nobile, Dirk Nuyens

Function integration, reconstruction and approximation using rank-1 lattices

Frances Y. Kuo Giovanni Migliorati Fabio Nobile Dirk Nuyens*

Abstract

We consider rank-1 lattices for integration and reconstruction of functions with series expansion supported on a finite index set. We explore the connection between the periodic Fourier space and the non-periodic cosine space and Chebyshev space, via tent transform and then cosine transform, to transfer known results from the periodic setting into new insights for the non-periodic settings. Fast discrete cosine transform can be applied for the reconstruction phase. To reduce the size of the auxiliary index set in the associated component-by-component (CBC) construction for the lattice generating vectors, we work with a bi-orthonormal set of basis functions, leading to three methods for function reconstruction in the non-periodic settings. We provide new theory and efficient algorithmic strategies for the CBC construction. We also interpret our results in the context of general function approximation and discrete least-squares approximation.

Keywords: Exact integration and approximation on finite index sets, Quasi-Monte Carlo methods, Rank-1 lattice points, Fourier space, Cosine space, Chebyshev space, Component-by-component construction.

AMS Subject classifications: 41A10 (Approximation by polynomials), 42A10 (Trigonometric approximation), 41A63 (Multidimensional problems), 42B05 (Fourier series and coefficients), 65D30 (Numerical integration), 65D32 (Quadrature and cubature formulas), 65D15 (Algorithms for functional approximation).

1 Introduction

In this paper we consider function integration, reconstruction and approximation in the periodic and non-periodic settings using rank-1 lattices. We explore the connection between three function space settings to transfer known results on rank-1 lattices from the periodic setting to the non-periodic settings. We obtain necessary and sufficient conditions on rank-1 lattices to achieve the exactness properties we require in each setting, and we develop efficient algorithms to construct the generating vectors for rank-1 lattices that satisfy these conditions.

More precisely, we consider functions with absolutely convergent series expansions with respect to an orthonormal basis, written in the generic form

$$f = \sum_{\boldsymbol{k}} \widehat{f}_{\boldsymbol{k}}\, \alpha_{\boldsymbol{k}}. \tag{1}$$

A large part of this paper is devoted to functions which are fully supported on a finite index set Λ, i.e.,

$$f = \sum_{\boldsymbol{k}\in\Lambda} \widehat{f}_{\boldsymbol{k}}\, \alpha_{\boldsymbol{k}}. \tag{2}$$

*Addresses: Frances Y. Kuo (f.kuo@unsw.edu.au), UNSW Sydney, Australia; Giovanni Migliorati (migliorati@ljll.math.upmc.fr), Sorbonne University, France; Fabio Nobile (fabio.nobile@epfl.ch), EPFL, Switzerland; Dirk Nuyens (dirk.nuyens@cs.kuleuven.be, corresponding author), KU Leuven, Belgium.

We develop methods based on rank-1 lattices to exactly integrate such functions (2), and to exactly reconstruct all series coefficients $\widehat{f}_{\boldsymbol{k}}$ in (2). We also consider the approximation problem for functions (1) which are not finitely supported on Λ.

The three function space settings we consider are as follows:

- The **Fourier space** contains all absolutely convergent Fourier series in the unit cube $[0,1]^d$, with exponential basis functions $e_{\boldsymbol{h}}(\boldsymbol{x}) = e^{2\pi \mathrm{i}\, \boldsymbol{h}\cdot\boldsymbol{x}}$ and indices $\boldsymbol{h} \in \mathbb{Z}^d$.
- The **cosine space** contains all absolutely convergent cosine series in $[0,1]^d$, with half-period cosine basis functions $\phi_{\boldsymbol{k}}$ (see (10) below) and nonnegative indices $\boldsymbol{k} \in \mathbb{N}_0^d$.
- The **Chebyshev space** consists of all absolutely convergent Chebyshev series in the larger domain $[-1,1]^d$, under the Chebyshev measure, with Chebyshev basis functions $\eta_{\boldsymbol{k}}$ (see (27) below) and also nonnegative indices $\boldsymbol{k} \in \mathbb{N}_0^d$.

To avoid excessive notation we keep to generic notation for the three spaces wherever possible, including the same 'hat' notation for series coefficients. However, to effectively describe the connection between spaces, we often distinguish the basis functions $e_{\boldsymbol{h}}$, $\phi_{\boldsymbol{k}}$, $\eta_{\boldsymbol{k}}$, and we often use $\boldsymbol{h}$ and $\boldsymbol{k}$ to contrast indices containing integers $\mathbb{Z}$ or only nonnegative integers $\mathbb{N}_0$.

The Fourier space contains periodic functions while the cosine and Chebyshev spaces contain nonperiodic functions. The Fourier space is often referred to as the Wiener algebra; it is the standard setting for analyzing periodic functions, see, e.g., [11, 12, 13, 27]. The cosine space is connected to the Fourier space by the **tent transform** which is defined by $\varphi_{\text{tent}}(x) := 1-|2x-1|$ for $x \in [0,1]$ and is applied componentwise in d dimensions, see, e.g., [5, 7, 29]. We show that the composition $\phi_{\boldsymbol{k}} \circ \varphi_{\text{tent}}$ is the average over all of those exponential basis functions $e_{\boldsymbol{h}}$ for which $(|h_1|, \ldots, |h_d|) = \boldsymbol{k}$ (see (14) below). Consequently, *the tent-transformed cosine space is a subspace of the Fourier space.* Thus we can apply results from the Fourier space to the cosine space via tent transform.

The Chebyshev space is related to the cosine space by the **cosine transform**, given by $\boldsymbol{x} = \cos(\pi \boldsymbol{x}') \in [-1,1]^d$ for $\boldsymbol{x}' \in [0,1]^d$, where the cosine function is applied componentwise, and we have $\eta_{\boldsymbol{k}}(\boldsymbol{x}) = \eta_{\boldsymbol{k}}(\cos(\pi \boldsymbol{x}')) = \phi_{\boldsymbol{k}}(\boldsymbol{x}')$. Thus *the cosine transform provides an isomorphism between the Chebyshev space and the cosine space.* Trivially all results from the cosine space can be carried over to the Chebyshev space.

Rank-1 lattices have been well studied for integration, reconstruction and approximation in the Fourier space; see, e.g., [4, 28] for integration, [9, 10] for reconstruction, and [1, 11, 12, 13, 14, 15, 18, 19, 35, 36] for approximation. Given the *generating vector* $\boldsymbol{z} \in \mathbb{Z}^d$, the n points of a rank-1 lattice are specified by

$$\boldsymbol{t}_i \;=\; \frac{i\boldsymbol{z} \bmod n}{n} \;\in\; [0,1]^d \qquad \text{for} \quad i = 0, \ldots, n-1.$$

For a Fourier space function f, the average of function values at the lattice points

$$Q_n(f) \;:=\; \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{t}_i)$$

is known as a rank-1 lattice rule which is an equal-weight cubature rule for approximating the integral

$$I(f) \;:=\; \int_{[0,1]^d} f(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.$$

Rank-1 lattices have an important property known as the "character property" (see (9) below) which states that the cubature sum of the exponential basis functions $Q_n(e_{\boldsymbol{h}})$ can only take the value of 1 or 0, depending on whether or not the dot product $\boldsymbol{h} \cdot \boldsymbol{z}$ is a multiple of n.

Since the integral of the basis function $I(e_{\boldsymbol{h}})$ is 1 if $\boldsymbol{h} = \boldsymbol{0}$ and is 0 otherwise, we easily deduce that a rank-1 lattice rule can exactly integrate a function $f = \sum_{\boldsymbol{h}\in\Lambda} \widehat{f}_{\boldsymbol{h}}\, e_{\boldsymbol{h}}$ whose Fourier series is supported on a finite set $\Lambda \subset \mathbb{Z}^d$ if and only if $\boldsymbol{h}\cdot\boldsymbol{z}$ is not a multiple of n for all nonzero vectors $\boldsymbol{h} \in \Lambda$. This condition in turn leads to an efficient algorithm to construct a generating vector $\boldsymbol{z}$ with the exactness property in a component-by-component fashion. This result is stated later in Lemma 4, see also [4], and it can be said to be the starting point of all results in this paper. Indeed, the result extends to function reconstruction on Λ where we evaluate all the Fourier coefficients $\widehat{f}_{\boldsymbol{h}}$ for $\boldsymbol{h} \in \Lambda$ by a rank-1 lattice rule, and the evaluations can be done using the fast Fourier transform. Using the character property one can deduce a necessary and sufficient condition when these Fourier coefficients can be recovered exactly, thus leading to a constructive algorithm to find suitable generating vectors by working with the "difference set" $\Lambda \ominus \Lambda$ which is obtained by forming all differences of indices in Λ. We state this result later in Lemma 5, which was first proved with varying generality in [9, 10, 27]. The idea has been further extended to the construction of "multiple rank-1 lattices" in [13], where the benefits of multiple reconstruction lattices are combined strategically to achieve the same goal with a reduced overall number of sampling nodes; we do not go down this path.

The connection between the Fourier space and the cosine space allows us to apply the theory of rank-1 lattices to the cosine space by tent transform. We can obtain necessary and sufficient conditions for *tent-transformed rank-1 lattices* to achieve the integral exactness and function reconstruction properties in the cosine space, see Lemmas 10 and 11 below (see also [29] for part of Lemma 11). In the case of function reconstruction on a finite index set $\Lambda \subset \mathbb{N}_0^d$, we end up having to work with quite a large auxiliary index set $\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)$ in our component-by-component construction of the lattice generating vector, where $\mathcal{M}(\Lambda)$ denotes the "mirrored set" obtained from Λ by including all sign changes of indices in Λ, while the $\oplus$ then means that we form the sum of all indices from $\mathcal{M}(\Lambda)$; we call this **plan A**. To improve on the computational efficiency of plan A, we show by working with a bi-orthonormal set of basis functions that we can achieve function reconstruction on Λ with a weaker condition which means working with a smaller auxiliary index set $\Lambda \oplus \mathcal{M}(\Lambda)$, see Lemma 15 below; we call this **plan B**. We also relax the algorithm to not necessarily recover the normalization of the basis functions to arrive at **plan C**, which achieves the same reconstruction property at a lower computational cost, see Lemma 16 below. All three plans for function reconstruction in the cosine space can be computed using the fast discrete cosine transform.

The isomorphism between the cosine space and the Chebyshev space allows us to take all results from the cosine space to the Chebyshev space, including plans A, B, C. We arrive at *tent-transformed and then cosine-transformed rank-1 lattices*, which in the case of n being even is also known as "Chebyshev lattices", see, e.g., [6, 26], although we do not adopt this terminology. Our plan C for the Chebyshev space with even n is essentially the approach in [26]; in this paper we do not require n to be even.

1.1 Layout of the paper and highlight of new results

In Section 2 we review results on rank-1 lattices for integration and function reconstruction on a finite index set in the Fourier space, referencing essential results from [4, 9, 10, 27].

In Section 3 we introduce the cosine space and consider integration and function reconstruction, with three plans for achieving exact function reconstruction using rank-1 lattices with varying costs. Except for the if-part of Lemma 11 and Lemma 12 which was proved in [29], all remaining results in this section are new, including plan B and plan C for function reconstruction and the applicability of fast discrete cosine transform.

In Section 4 we present the corresponding results for the Chebyshev space. Lemma 21 for plan C with even n turns out to be equivalent to the approach in [26]. However, the precise connection to the cosine space via cosine transform and in turn the precise connection to the

Fourier space via tent transform are both new interpretations here, and they lead to broader implications in the Chebyshev space. In particular, the multiplicity of the transformed points under these interpretations are known explicitly for n both even and odd, and fast discrete cosine transform can be applied for all n.

Section 5 is devoted to the theory and algorithmic aspect of the component-by-component (CBC) construction for lattice generating vectors achieving various conditions needed for the exactness properties. As the theoretical justification for the CBC construction, Theorem 23 generalises previous results proved in [4, 9, 10] and provides a cheaper variant of the algorithm when building up the index set, while Theorem 26 is new and specific to plan C. The systematic way to combine two different approaches (namely, the "brute force" approach and the "elimination" approach, to be explained in Section 5) in a mixed CBC construction is new. Strategies for storage and a "smart lookup" to efficiently search through difference and/or sum involving mirrored sets are also new.

Finally in Section 6 we interpret our results in the context of approximation of general functions that are not necessarily supported on a finite index set, and compare them with discrete least-squares approximation as analysed in [2, 3, 21, 22, 25]. We mention other known results in function approximation based on rank-1 lattices (see, e.g., [16, 17, 33, 34] for general results and [1, 5, 11, 13, 14, 15, 18, 19, 35, 36] for rank-1 lattices).

We end the introduction with setting the notation on multiindices and introducing some special index sets.

1.2 Notation on multiindices and special index sets

Throughout this paper we use # to denote the cardinality of a set. For any multiindex $\boldsymbol{k} \in \mathbb{Z}^d$, we write $|\boldsymbol{k}|_0 := \#\{1 \le j \le d : k_j \ne 0\}$ for the number of nonzero indices in $\boldsymbol{k}$. For $\boldsymbol{k}, \boldsymbol{k}' \in \mathbb{Z}^d$, $\boldsymbol{k}' \le \boldsymbol{k}$ is to be interpreted componentwise, i.e., $k'_j \le k_j$ for all j.

For $\boldsymbol{\sigma} \in \{\pm 1\}^d$ and $\boldsymbol{k} \in \mathbb{Z}^d$, we write $\boldsymbol{\sigma}(\boldsymbol{k}) := (\sigma_1 k_1, \dots, \sigma_d k_d)$ to mean that we apply the sign changes in $\boldsymbol{\sigma}$ componentwise to $\boldsymbol{k}$. For any $\boldsymbol{k} \in \mathbb{Z}^d$ we use

$$\mathcal{S}_{\boldsymbol{k}} := \Big\{\boldsymbol{\sigma} \in \{\pm 1\}^d : \sigma_j = +1 \text{ for each } j \text{ for which } k_j = 0\Big\}$$

to denote a set of *unique sign changes* for $\boldsymbol{k}$. Then clearly we have $\#\mathcal{S}_{\boldsymbol{k}} = 2^{|\boldsymbol{k}|_0}$.

We will consider index sets with some special properties:

- An index set $\Lambda \subset \mathbb{N}_0^d$ is *downward closed* if $\boldsymbol{k}' \in \Lambda$ whenever $\boldsymbol{k}' \le \boldsymbol{k}$ and $\boldsymbol{k} \in \Lambda$. This means that from every $\boldsymbol{k} \in \Lambda$ we can move towards $\boldsymbol{0}$ along the coordinate axes without finding a $\boldsymbol{k}' \notin \Lambda$. Analogous definition holds with $\mathbb{N}_0^d$ replaced by $\mathbb{Z}^d$.
- An index set $\Lambda \subset \mathbb{Z}^d$ is *centrally symmetric* if $-\boldsymbol{k} \in \Lambda$ whenever $\boldsymbol{k} \in \Lambda$.
- An index set $\Lambda \subset \mathbb{Z}^d$ is *fully sign symmetric* if $\boldsymbol{\sigma}(\boldsymbol{k}) \in \Lambda$ whenever $\boldsymbol{k} \in \Lambda$ and $\boldsymbol{\sigma} \in \{\pm 1\}^d$.
- An index set $\Lambda \subset \mathbb{N}_0^d$ is an (*anisotropic*) *tensor product* set if there exist $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{N}_0^d$ such that $\Lambda = \{\boldsymbol{k} \in \mathbb{N}_0^d : \boldsymbol{a} \le \boldsymbol{k} \le \boldsymbol{b}\}$. Analogous definition holds with $\mathbb{N}_0^d$ replaced by $\mathbb{Z}^d$.

For any index set $\Lambda \subset \mathbb{N}_0^d$ or $\Lambda \subset \mathbb{Z}^d$, we denote its largest component in magnitude by

$$\max(\Lambda) := \max_{\boldsymbol{k} \in \Lambda} \max_{1 \le j \le d} |k_j|,$$

and we define

$$\begin{aligned}
\Lambda \oplus \Lambda &:= \{\boldsymbol{k} + \boldsymbol{k}' : \boldsymbol{k}, \boldsymbol{k}' \in \Lambda\} && \text{("sum set")},\\
\Lambda \ominus \Lambda &:= \{\boldsymbol{k} - \boldsymbol{k}' : \boldsymbol{k}, \boldsymbol{k}' \in \Lambda\} && \text{("difference set")},\\
\mathcal{M}(\Lambda) &:= \Big\{\boldsymbol{\sigma}(\boldsymbol{k}) : \boldsymbol{k} \in \Lambda, \boldsymbol{\sigma} \in \{\pm 1\}^d\Big\} = \bigcup_{\boldsymbol{k} \in \Lambda} \{\boldsymbol{\sigma}(\boldsymbol{k}) : \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}\} && \text{("mirrored set")}.
\end{aligned}$$

If $\Lambda \subset \mathbb{Z}^d$ is centrally symmetric, then $\Lambda \oplus \Lambda = \Lambda \ominus \Lambda$. If $\Lambda \in \mathbb{Z}^d$ is fully sign symmetric, then $\mathcal{M}(\Lambda) = \Lambda$. For $\Lambda \subset \mathbb{Z}^d$, $\Lambda \ominus \Lambda$ is always centrally symmetric (since both $\boldsymbol{k} - \boldsymbol{k}'$ and $\boldsymbol{k}' - \boldsymbol{k}$ belong to $\Lambda \ominus \Lambda$ when $\boldsymbol{k}, \boldsymbol{k}' \in \Lambda$).

Trivially we have

$$\#(\Lambda \oplus \Lambda) \,\leq\, (\#\Lambda)^2, \qquad \#(\Lambda \ominus \Lambda) \,\leq\, (\#\Lambda)^2, \qquad \text{and} \qquad \#\mathcal{M}(\Lambda) \,\leq\, \sum_{\boldsymbol{k}\in\Lambda} 2^{|\boldsymbol{k}|_0} \,\leq\, 2^d\,\#\Lambda.$$

The squaring effect in the upper bounds for the sum/difference set cannot be avoided in general, since even if all multiindices in Λ fall on the first two axes (i.e., all components of $\boldsymbol{k}$ are zero except for one of k_1 and k_2), the sum/difference sets will contain a large rectangle (so there is a lower bound of the same order). On the other hand, the 2^d factor in the upper bound for $\mathcal{M}(\Lambda)$ can sometimes be improved, as shown in the forthcoming Lemma 2 and Example 3 below. We also need the next Lemma 1, whose proof uses induction arguments from [23, 2].

Lemma 1. *In any dimension d, given any $\Lambda \subset \mathbb{Z}^d$ downward closed and any polynomial $p(n) = \sum_{k=0}^{\eta} b_k n^k$ of degree $\eta \geq 0$ with nonnegative coefficients $b_0 \leq 1$ and all $b_k \leq \binom{\eta+1}{k}$, it holds*

$$\sum_{\boldsymbol{k}\in\Lambda} \prod_{j=1}^{d} p(|k_j|) \leq (\#\Lambda)^{\eta+1}. \tag{3}$$

Moreover, it holds

$$\sum_{\boldsymbol{k}\in\Lambda} 2^{|\boldsymbol{k}|_0} \,\leq\, (\#\Lambda)^{\ln 3/\ln 2}. \tag{4}$$

Proof. When $\eta = 0$ the result holds true. Consider then the case $\eta \geq 1$. Every downward closed set $\Lambda \subset \mathbb{Z}^d$ can be seen as a set $\widetilde{\Lambda} \subset \mathbb{N}_0^{2d}$ constructed in the following way. Start with $\widetilde{\Lambda} = \emptyset$. For any $\boldsymbol{k} \in \Lambda$, define the sets $C_{\boldsymbol{k}} := \{1 \leq j \leq d : k_j < 0\} \subseteq \{1, \ldots, d\}$ and $U_{\boldsymbol{k}} := \{1, \ldots, d\} \setminus C_{\boldsymbol{k}}$. Then define $\widetilde{\boldsymbol{k}} \in \mathbb{N}_0^{2d}$ by setting $\widetilde{k}_j = k_j$ and $\widetilde{k}_{j+d} = 0$ for all $j \in U_{\boldsymbol{k}}$, and $\widetilde{k}_j = 0$ and $\widetilde{k}_{j+d} = -k_j$ for all $j \in C_{\boldsymbol{k}}$. Finally add $\widetilde{\boldsymbol{k}}$ to $\widetilde{\Lambda}$. Notice that this algorithm establishes a one-to-one correspondence between the elements of Λ and $\widetilde{\Lambda}$, and therefore $\#\Lambda = \#\widetilde{\Lambda}$. By construction $\widetilde{\Lambda}$ is also downward closed in $\mathbb{N}_0^{2d}$. Applying Theorem 1 from [23] to the set $\widetilde{\Lambda} \in \mathbb{N}_0^{2d}$ we obtain (3).

For the proof of (4), as above, starting from Λ we construct the downward closed set $\widetilde{\Lambda} \subset \mathbb{N}_0^{2d}$ such that $\#\widetilde{\Lambda} = \#\Lambda$, and then apply Lemma 3.3 from [2] to the set $\widetilde{\Lambda}$. □

Lemma 2. *If $\Lambda \subset \mathbb{Z}^d$ is downward closed then*

$$\max_{\boldsymbol{k}\in\Lambda} 2^{|\boldsymbol{k}|_0} \,\leq\, \#\Lambda, \quad \sum_{\boldsymbol{k}\in\Lambda} 2^{|\boldsymbol{k}|_0} \,\leq\, (\#\Lambda)^{\ln 3/\ln 2}, \quad \textit{and} \quad \#\mathcal{M}(\Lambda) \,\leq\, \min\left(2^d\,\#\Lambda, (\#\Lambda)^{\ln 3/\ln 2}\right).$$

Proof. For the first bound, since Λ is downward closed, for any $\boldsymbol{k} \in \Lambda$, the set Λ will include the hyper-rectangle with $\boldsymbol{k}$ and the origin as corners. Thus $\#\Lambda \geq \prod_{1\leq j\leq d,\, k_j\neq 0}(1 + |k_j|) \geq 2^{|\boldsymbol{k}|_0}$. The second bound is proved in Lemma 1. The third bound is an immediate consequence of the second bound. □

Since $\#\Lambda$ most likely grows with d, in general it is not obvious which of $2^d\,\#\Lambda$ or $(\#\Lambda)^{\ln 3/\ln 2}$ is a better bound for $\#\mathcal{M}(\Lambda)$. If $\#\Lambda$ can be bounded independently of d, then most likely so can $\#\mathcal{M}(\Lambda)$.

Example 3. *Consider a "weighted" index set of "degree" $m \in \mathbb{N}$ defined by (see, e.g., [4]) $\Lambda = \{\boldsymbol{k} \in \mathbb{N}_0^d : r(\boldsymbol{k}) \leq m\}$, where $r(\boldsymbol{k})$ is given by*

$$\max_{1\leq j\leq d} \frac{k_j}{\beta_j}, \quad \sum_{1\leq j\leq d} \frac{k_j}{\beta_j}, \quad \textit{or} \quad \prod_{j=1}^{d} \max\left(1, \frac{k_j}{\beta_j}\right), \tag{5}$$

with $1 = \beta_1 \geq \beta_2 \geq \cdots > 0$ *and* $\sum_{j=1}^\infty \beta_j < \infty$. *The first example is an anisotropic tensor product set and is the largest of the three examples. We have* $\#\Lambda = \prod_{j=1}^d (1 + \lfloor \beta_j m \rfloor) \leq \exp(m \sum_{j=1}^\infty \beta_j)$ *and* $\#\mathcal{M}(\Lambda) = \prod_{j=1}^d (1 + 2\lfloor \beta_j m \rfloor) \leq \exp(2m \sum_{j=1}^\infty \beta_j)$, *so both are bounded independently of* d. *Their ratio satisfies*

$$\frac{\#\mathcal{M}(\Lambda)}{\#\Lambda} \;=\; \prod_{j=1}^d \left(1 + \frac{\lfloor \beta_j m \rfloor}{1 + \lfloor \beta_j m \rfloor}\right) \;\leq\; \min\left(2^J, \exp\left(m \sum_{j=1}^\infty \beta_j\right)\right),$$

where J *is the "truncation dimension" such that* $\beta_j m < 1$ *for all* $j \geq J$. *If we have* $\lambda \in (0, 1]$ *such that* $\sum_{j=1}^\infty \beta_j^\lambda < \infty$, *then* $\beta_J < J^{-1/\lambda} (\sum_{j=1}^\infty \beta_j^\lambda)^{1/\lambda}$ *and it suffices to take* $J = m^\lambda (\sum_{j=1}^\infty \beta_j^\lambda)$. *Both upper bounds on the ratio grow exponentially with* m.

The third example in (5) *is the smallest of the three. Its mirror set is commonly referred to as the "Zaremba cross" or "hyperbolic cross", see, e.g., [4, 9]. For all* $\tau > 1$ *we have*

$$m + 1 \;\leq\; \#\Lambda \;\leq\; m^\tau \prod_{j=1}^d \left(1 + \zeta(\tau) \beta_j^\tau\right) \;\leq\; m^\tau \exp\left(\zeta(\tau) \sum_{j=1}^\infty \beta_j^\tau\right),$$

$$2m + 1 \leq \#\mathcal{M}(\Lambda) \;\leq\; m^\tau \prod_{j=1}^d \left(1 + 2\zeta(\tau) \beta_j^\tau\right) \;\leq\; (\#\Lambda)^\tau \exp\left(2\zeta(\tau) \sum_{j=1}^\infty \beta_j^\tau\right),$$

where $\zeta(\tau) := \sum_{k=1}^\infty k^{-\tau}$ *is the Riemann zeta function. Since* τ *can be arbitrarily close to* 1, $\#\mathcal{M}(\Lambda)$ *is essentially of the same order as* $\#\Lambda$, *both are bounded independently of* d. *The upper bound on* $\#\mathcal{M}(\Lambda)$ *is proved in [14].*

2 Periodic setting based on trigonometric polynomials

2.1 Fourier series

We start by considering periodic functions on $[0,1]^d$. Let $\mathcal{F}^{\text{Four}}$ denote the space of complex-valued functions defined on $[0,1]^d$ with absolutely converging Fourier series:

$$\mathcal{F}^{\text{Four}} := \left\{ f \in L^2 \;\middle|\; f : [0,1]^d \to \mathbb{C},\; f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \widehat{f}_{\boldsymbol{h}}\, e^{2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}} \text{ and } \sum_{\boldsymbol{h} \in \mathbb{Z}^d} |\widehat{f}_{\boldsymbol{h}}| < \infty \right\},$$

where $\boldsymbol{h} \cdot \boldsymbol{x} := h_1 x_1 + \cdots + h_d x_d$ is the usual dot product and $\widehat{f}_{\boldsymbol{h}}$ are the Fourier coefficients. We equip $\mathcal{F}^{\text{Four}}$ with the usual L^2 inner product

$$\langle f_1, f_2 \rangle \;:=\; \int_{[0,1]^d} f_1(\boldsymbol{x}) \overline{f_2(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x}. \tag{6}$$

The exponential functions form an orthonormal basis

$$e_{\boldsymbol{h}}(\boldsymbol{x}) \;:=\; e^{2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}}$$

satisfying $\langle e_{\boldsymbol{h}}, e_{\boldsymbol{h}'} \rangle = \delta_{\boldsymbol{h}, \boldsymbol{h}'}$, where the Kronecker delta function yields 1 if $\boldsymbol{h} = \boldsymbol{h}'$ and 0 if $\boldsymbol{h} \neq \boldsymbol{h}'$. The Fourier coefficients are given by

$$\widehat{f}_{\boldsymbol{h}} \;:=\; \langle f, e_{\boldsymbol{h}} \rangle \;=\; \int_{[0,1]^d} f(\boldsymbol{x})\, e^{-2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}}\, \mathrm{d}\boldsymbol{x}, \qquad \boldsymbol{h} \in \mathbb{Z}^d.$$

The norm of f satisfies $\|f\|^2 \;=\; \int_{[0,1]^d} |f(\boldsymbol{x})|^2\, \mathrm{d}\boldsymbol{x} \;=\; \sum_{\boldsymbol{h} \in \mathbb{Z}^d} |\widehat{f}_{\boldsymbol{h}}|^2$.

2.2 Fourier coefficients by cubature

For $f \in \mathcal{F}^{\text{Four}}$ we define the integral operator

$$I(f) := \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

Later we will seek a cubature formula $Q_n(f)$ which uses linear combinations of n evaluations of f to approximate $I(f)$. We define a discrete inner product

$$\langle f_1, f_2 \rangle_n := Q_n(f_1 \, \overline{f_2})$$

as an approximation to (6).

Given an arbitrary finite index set $\Lambda \subset \mathbb{Z}^d$, we consider the subspace $\mathcal{F}_\Lambda^{\text{Four}}$ of all functions whose Fourier series is supported solely on Λ, i.e.,

$$\text{for } f \in \mathcal{F}_\Lambda^{\text{Four}}: \qquad f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \Lambda} \widehat{f}_{\boldsymbol{h}} \, e^{2\pi \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{x}}. \tag{7}$$

Implicitly, this means that all other Fourier coefficients of f are zero, i.e., $\widehat{f}_{\boldsymbol{h}} = 0$ for $\boldsymbol{h} \notin \Lambda$.

In this paper we will demand one or both of the following related properties on the cubature formula:

- **Integral exactness.** We want our cubature formula to be exact for all functions which are supported solely on Λ, i.e., we want $Q_n(f) = I(f)$ for all $f \in \mathcal{F}_\Lambda^{\text{Four}}$. This holds if and only if

  $$Q_n(e_{\boldsymbol{h}}) = I(e_{\boldsymbol{h}}) = \delta_{\boldsymbol{h},\boldsymbol{0}} \qquad \text{for all } \boldsymbol{h} \in \Lambda,$$

  i.e., our cubature formula integrates exactly all basis functions $e^{2\pi \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{x}}$ with $\boldsymbol{h} \in \Lambda$.

- **Function reconstruction.** Instead of (7) we consider

  $$f^a(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \Lambda} \widehat{f}^a_{\boldsymbol{h}} \, e^{2\pi \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{x}},$$

  where each Fourier coefficient $\widehat{f}_{\boldsymbol{h}} = \langle f, e_{\boldsymbol{h}} \rangle = I(f \, e_{-\boldsymbol{h}})$ in (7) is replaced by the cubature formula $\widehat{f}^a_{\boldsymbol{h}} := \langle f, e_{\boldsymbol{h}} \rangle_n = Q_n(f \, e_{-\boldsymbol{h}})$. We demand the "non-aliasing" condition that

  $$\widehat{f}^a_{\boldsymbol{h}} = \widehat{f}_{\boldsymbol{h}} \qquad \text{for all } \boldsymbol{h} \in \Lambda \text{ and } f \in \mathcal{F}_\Lambda^{\text{Four}},$$

  so that f^a is a reconstruction of f. (If other coefficients $\widehat{f}_{\boldsymbol{h}'}$ with $\boldsymbol{h}' \neq \boldsymbol{h}$ contribute to $\widehat{f}^a_{\boldsymbol{h}}$ then this is called "aliasing".) Using the linearity of Q_n, we then have

  $$\begin{aligned} \widehat{f}^a_{\boldsymbol{h}} &= Q_n(f \, e_{-\boldsymbol{h}}) \\ &= Q_n\Bigg(\Bigg(\sum_{\boldsymbol{h}' \in \Lambda} \widehat{f}_{\boldsymbol{h}'} \, e_{\boldsymbol{h}'}\Bigg) e_{-\boldsymbol{h}}\Bigg) = \sum_{\boldsymbol{h}' \in \Lambda} \widehat{f}_{\boldsymbol{h}'} \, Q_n(e_{\boldsymbol{h}'-\boldsymbol{h}}) = \widehat{f}_{\boldsymbol{h}} \quad \text{for all } \boldsymbol{h} \in \Lambda \text{ and } f \in \mathcal{F}_\Lambda^{\text{Four}}. \end{aligned}$$

  This holds if and only if

  $$Q_n(e_{\boldsymbol{h}'-\boldsymbol{h}}) = \langle e_{\boldsymbol{h}'}, e_{\boldsymbol{h}} \rangle = \delta_{\boldsymbol{h},\boldsymbol{h}'} \qquad \text{for all } \boldsymbol{h}, \boldsymbol{h}' \in \Lambda,$$

  which is equivalent to

  $$Q_n(e_{\boldsymbol{h}}) = I(e_{\boldsymbol{h}}) = \delta_{\boldsymbol{h},\boldsymbol{0}} \qquad \text{for all } \boldsymbol{h} \in \Lambda \ominus \Lambda,$$

  i.e., our cubature formula integrates exactly all basis functions $e^{2\pi \mathrm{i} \, \boldsymbol{h} \cdot \boldsymbol{x}}$ with $\boldsymbol{h} \in \Lambda \ominus \Lambda$.

2.3 Rank-1 lattice rules

Consider now the cubature formula given by *rank*-1 *lattices*

$$Q_n(f) := \frac{1}{n}\sum_{i=0}^{n-1} f\left(\frac{i\boldsymbol{z} \bmod n}{n}\right), \tag{8}$$

where $\boldsymbol{z} \in \mathbb{Z}^d$ is an integer vector known as the *generating vector*. It is easy to verify the "character property" that for any $\boldsymbol{h} \in \mathbb{Z}^d$,

$$Q_n(e_{\boldsymbol{h}}) = \frac{1}{n}\sum_{i=0}^{n-1} e^{2\pi \mathrm{i}\, i\boldsymbol{h}\cdot\boldsymbol{z}/n} = \begin{cases} 1 & \text{if } \boldsymbol{h}\cdot\boldsymbol{z} \equiv_n 0, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where the notation $a \equiv_n b$ means that $(a \bmod n) = (b \bmod n)$. This leads to the well-known lattice cubature error formula for $f \in \mathcal{F}^{\mathrm{Four}}$

$$Q_n(f) - I(f) = \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d\setminus\{\boldsymbol{0}\} \\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv_n 0}} \widehat{f}_{\boldsymbol{h}}.$$

The set of integer vectors $\{\boldsymbol{h} \in \mathbb{Z}^d : \boldsymbol{h}\cdot\boldsymbol{z} \equiv_n 0\}$ is known as the *dual lattice*. Clearly the cubature rule is exact for a function f solely supported on Λ if and only if the dual lattice does not contain any index from $\Lambda \setminus \{\boldsymbol{0}\}$. We know how to obtain such a lattice rule generating vector using a component-by-component construction.

Lemma 4 (Integral exactness)**.** *Let $\Lambda \subset \mathbb{Z}^d$ be an arbitrary index set. A lattice rule with n points and generating vector $\boldsymbol{z}$ integrates exactly all functions $f \in \mathcal{F}_\Lambda^{\mathrm{Four}}$ solely supported on Λ if and only if*

$$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in \Lambda\setminus\{\boldsymbol{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\Lambda\setminus\{\boldsymbol{0}\})}{\kappa} + 1,\, \max(\Lambda)\right\},$$

with $\kappa = 2$ if Λ is centrally symmetric and $\kappa = 1$ otherwise.

Proof. The result for some standard anisotropic, downward closed and centrally symmetric sets Λ can be found in Cools, Kuo & Nuyens [4]. A proof for general index sets is provided later in Section 5, see Theorem 23 and Remark 24. A similar proof can be found in Kämmerer [10]. □

Lemma 5 (Function reconstruction)**.** *Let $\Lambda \subset \mathbb{Z}^d$ be an arbitrary index set. A lattice rule Q_n with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the Fourier coefficients of all functions $f \in \mathcal{F}_\Lambda^{\mathrm{Four}}$ solely supported on Λ, by*

$$\widehat{f}_{\boldsymbol{h}} = \widehat{f}^{\,a}_{\boldsymbol{h}} := Q_n(f\, e_{-\boldsymbol{h}}) \qquad \text{for all } \boldsymbol{h} \in \Lambda,$$

if and only if

$$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in (\Lambda \ominus \Lambda)\setminus\{\boldsymbol{0}\},$$

which is equivalent to

$$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n \boldsymbol{h}'\cdot\boldsymbol{z} \qquad \text{for all } \boldsymbol{h},\boldsymbol{h}' \in \Lambda \text{ with } \boldsymbol{h} \neq \boldsymbol{h}'.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\Lambda \ominus \Lambda) + 1}{2},\, 2\max(\Lambda)\right\}.$$

Proof. The result follows directly from Theorem 23, noting that the difference set $\Lambda \ominus \Lambda$ is centrally symmetric and contains $\mathbf{0}$, and therefore $\frac{1}{2}\#((\Lambda \ominus \Lambda) \setminus \{\mathbf{0}\}) + 1 = \frac{1}{2}(\#(\Lambda \ominus \Lambda) + 1)$. Alternatively, the result for Λ a hyperbolic cross index set can be found in Kämmerer [9], while the result for any arbitrary index set Λ can be found in Kämmerer [10] and Potts & Volkmer [27, Theorem 2.1]. □

We end this section by the very interesting property that mapping from function values to Fourier coefficients and the other way around can be done using a one-dimensional fast Fourier transform.

Lemma 6. *Let $\boldsymbol{z}$ be a generating vector for an n-point rank-1 lattice satisfying the reconstruction property on an arbitrary index set $\Lambda \subset \mathbb{Z}^d$ according to Lemma 5. For a function $f \in \mathcal{F}_\Lambda^{\mathrm{Four}}$ solely supported on Λ we can compute*

coefficients from function values:	*function values from coefficients:*
// prepare function value vector $\boldsymbol{f} \in \mathbb{C}^n$	*// prepare coefficient vector* $\boldsymbol{F} \in \mathbb{C}^n$
`for` $i \in \{0, \dots, n-1\}$`:`	$\boldsymbol{F} = \mathbf{0} \in \mathbb{C}^n$
$\quad f_i = f((i\boldsymbol{z} \bmod n)/n)$	`for` $\boldsymbol{h} \in \Lambda$`:`
	$\quad F_{(\boldsymbol{h}\cdot\boldsymbol{z} \bmod n)} = \widehat{f}_{\boldsymbol{h}}$
// compute coefficient vector $\boldsymbol{F} \in \mathbb{C}^n$	*// compute function value vector* $\boldsymbol{f} \in \mathbb{C}^n$
$\boldsymbol{F} = \mathrm{FFT}(\boldsymbol{f})$	$\boldsymbol{f} = \mathrm{IFFT}(\boldsymbol{F})$
// $\widehat{f}_{\boldsymbol{h}}$ *is given by* $F_{(\boldsymbol{h}\cdot\boldsymbol{z} \bmod n)}$	*//* f_i *gives the value of* $f((i\boldsymbol{z} \bmod n)/n)$

where $\boldsymbol{f} \in \mathbb{C}^n$ is a vector containing function values and $\boldsymbol{F} \in \mathbb{C}^n$ is a vector containing Fourier coefficients. Here FFT *and* IFFT *are the one-dimensional fast Fourier transform and its inverse, respectively, with a normalization $1/n$ for* FFT *and* 1 *for* IFFT*; both mappings have cost $\mathcal{O}(n \log(n))$.*

Proof. This follows from expanding the formula for $\widehat{f}^a_{\boldsymbol{h}}$ in Lemma 5. Each $\boldsymbol{h} \in \Lambda$ will correspond to a unique value of $\boldsymbol{h} \cdot \boldsymbol{z} \bmod n$ by the non-aliasing condition in Lemma 5. The other Fourier coefficients are zero by the assumption that $f \in \mathcal{F}_\Lambda^{\mathrm{Four}}$ is solely supported on Λ. □

Remark 7. *If the function f has wider support in the Fourier space than just Λ, then the vector $\boldsymbol{F}$ resulting from the evaluation $\boldsymbol{F} = \mathrm{FFT}(\boldsymbol{f})$ will not necessarily be zero at positions F_κ when κ does not correspond to a value of $\boldsymbol{h} \cdot \boldsymbol{z} \bmod n$ for some $\boldsymbol{h} \in \Lambda$. This is due to the aliasing effect from $\boldsymbol{h}$ outside of Λ and this will also contaminate all other components of $\boldsymbol{F}$. It is possible to extend the index set to full size n while still keeping the reconstruction property on the extended index set such that all values in $\boldsymbol{F}$ can be interpreted as Fourier coefficients. This technique has been used, e.g., in [19, 24, 30, 31].*

3 Nonperiodic setting based on half-period cosines

3.1 Cosine series

The cosine basis functions are a complete and orthonormal basis for $L^2([0,1]^d)$:

$$\phi_{\boldsymbol{k}}(\boldsymbol{x}) := \sqrt{2}^{|\boldsymbol{k}|_0} \prod_{j=1}^{d} \cos(\pi k_j x_j), \qquad \boldsymbol{k} \in \mathbb{N}_0^d, \tag{10}$$

where $|\boldsymbol{k}|_0$ denotes the count of the nonzero entries in the vector $\boldsymbol{k}$, and we have $\langle \phi_{\boldsymbol{k}}, \phi_{\boldsymbol{k}'} \rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$. The "cosine space" $\mathcal{F}^{\cos}$ consists of nonperiodic real-valued functions on $[0,1]^d$ with absolutely

converging cosine series:

$$\mathcal{F}^{\cos} := \left\{ f \in L^2 \;\middle|\; f : [0,1]^d \to \mathbb{R},\ f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \widehat{f}_{\boldsymbol{k}}\, \phi_{\boldsymbol{k}}(\boldsymbol{x}) \text{ and } \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} |\widehat{f}_{\boldsymbol{k}}| < \infty \right\}$$

where the cosine coefficients are

$$\widehat{f}_{\boldsymbol{k}} := \int_{[0,1]^d} f(\boldsymbol{x})\, \phi_{\boldsymbol{k}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.$$

This space was studied for integration and approximation in [7, 29, 5]. Even though the cosine basis is a complete orthonormal system for $L^2([0,1]^d)$, it does not allow the representation of arbitrary polynomials.

3.2 Cosine coefficients by cubature

As in Section 2, for a given finite index set $\Lambda \subset \mathbb{N}_0^d$ we consider the subspace $\mathcal{F}_\Lambda^{\cos}$ of all functions whose cosine series is supported solely on Λ, i.e.,

$$\text{for } f \in \mathcal{F}_\Lambda^{\cos}: \qquad f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \Lambda} \widehat{f}_{\boldsymbol{k}}\, \phi_{\boldsymbol{k}}(\boldsymbol{x}), \tag{11}$$

and we are interested in two related properties on the cubature formula:

- **Integral exactness.** We want $Q_n(f) = I(f)$ for all $f \in \mathcal{F}_\Lambda^{\cos}$, which holds if and only if

$$Q_n(\phi_{\boldsymbol{k}}) = I(\phi_{\boldsymbol{k}}) = \delta_{\boldsymbol{k},\boldsymbol{0}} \qquad \text{for all } \boldsymbol{k} \in \Lambda. \tag{12}$$

- **Function reconstruction.** We replace each cosine coefficient $\widehat{f}_{\boldsymbol{k}} = \langle f, \phi_{\boldsymbol{k}} \rangle = I(f\,\phi_{\boldsymbol{k}})$ in (11) by the cubature formula $\widehat{f}^a_{\boldsymbol{k}} := \langle f, \phi_{\boldsymbol{k}} \rangle_n = Q_n(f\,\phi_{\boldsymbol{k}})$, and demand the non-aliasing condition

$$\widehat{f}^a_{\boldsymbol{k}} = Q_n\left(\left(\sum_{\boldsymbol{k}' \in \Lambda} \widehat{f}_{\boldsymbol{k}'}\, \phi_{\boldsymbol{k}'} \right) \phi_{\boldsymbol{k}} \right) = \sum_{\boldsymbol{k}' \in \Lambda} \widehat{f}_{\boldsymbol{k}'}\, Q_n(\phi_{\boldsymbol{k}'}\, \phi_{\boldsymbol{k}}) = \widehat{f}_{\boldsymbol{k}} \quad \text{for all } \boldsymbol{k} \in \Lambda \text{ and } f \in \mathcal{F}_\Lambda^{\cos},$$

  which holds if and only if

$$Q_n(\phi_{\boldsymbol{k}}\, \phi_{\boldsymbol{k}'}) = \langle \phi_{\boldsymbol{k}}, \phi_{\boldsymbol{k}'} \rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'} \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda. \tag{13}$$

Unlike the Fourier case where a product of two basis functions is another basis function, here the condition (13) is not straightforward to simplify, except when the index set Λ is downward closed. In the next section we will obtain necessary and sufficient conditions for function reconstruction by connecting with the Fourier space, without the restriction to downward closed index sets.

3.3 Connection with the Fourier case via tent transform

Below we will obtain sufficient conditions to achieve (12) and (13) in the cosine space by utilizing a known connection with the Fourier case via the so-called "tent transform"(see, e.g., [8])

$$\varphi_{\text{tent}} : [0,1] \to [0,1], \qquad \varphi_{\text{tent}}(x) := 1 - |2x - 1|.$$

The tent transform is a Lebesgue preserving transformation and therefore a componentwise mapping of

$$\boldsymbol{x}' = \varphi_{\text{tent}}(\boldsymbol{x}) := (\varphi_{\text{tent}}(x_1), \ldots, \varphi_{\text{tent}}(x_d))$$

yields

$$I(f \circ \varphi_{\text{tent}}) \;=\; \int_{[0,1]^d} f(\varphi_{\text{tent}}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \;=\; \int_{[0,1]^d} f(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \;=\; I(f).$$

To get a sense of how this transformation works, it is informative to consider the univariate case:

$$\begin{aligned}\int_0^1 f(\varphi_{\text{tent}}(x))\,\mathrm{d}x \;&=\; \int_0^{1/2} f(2x)\,\mathrm{d}x + \int_{1/2}^1 f(2-2x)\,\mathrm{d}x \\ &=\; \int_0^1 f(x')\,(\tfrac{1}{2}\,\mathrm{d}x') + \int_1^0 f(x')\,(-\tfrac{1}{2}\,\mathrm{d}x') \;=\; \int_0^1 f(x')\,\mathrm{d}x'.\end{aligned}$$

In the following, we recall the definition of the "mirrored" index set associated with the index set Λ, $\mathcal{M}(\Lambda) := \{\boldsymbol{\sigma}(\boldsymbol{k}) : \boldsymbol{k} \in \Lambda, \boldsymbol{\sigma} \in \{\pm 1\}^d\} = \bigcup_{\boldsymbol{k}\in\Lambda}\{\boldsymbol{\sigma}(\boldsymbol{k}) : \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}\}$, where $\mathcal{S}_{\boldsymbol{k}}$ is the set of all unique sign changes of $\boldsymbol{k}$.

Lemma 8 (Integral exactness – sufficiency)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. If a cubature rule $Q_n^*(f) = \sum_{i=0}^{n-1} w_i^*\, f(\boldsymbol{t}_i^*)$ integrates exactly all Fourier basis functions $e_{\boldsymbol{h}}$ with $\boldsymbol{h} \in \mathcal{M}(\Lambda)$, then the cubature rule $Q_n(f) = \sum_{i=0}^{n-1} w_i\, f(\boldsymbol{t}_i)$ with $w_i = w_i^*$ and $\boldsymbol{t}_i = \varphi_{\text{tent}}(\boldsymbol{t}_i^*)$ integrates exactly all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ.*

Proof. For any $\boldsymbol{k} \in \mathbb{N}_0^d$ we can write

$$\phi_{\boldsymbol{k}}(\boldsymbol{x}) \;=\; \sqrt{2}^{|\boldsymbol{k}|_0} \prod_{j=1}^d \cos(\pi k_j x_j) \;=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \exp(\pi\,\mathrm{i}\,\boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{x}),$$

which follows from expanding the product of $\cos(\theta_j) = (e^{\mathrm{i}\theta_j} + e^{-\mathrm{i}\theta_j})/2$ for those $\theta_j \neq 0$. Furthermore, since $\cos(\pi k\,\varphi_{\text{tent}}(x)) = \cos(2\pi k x)$ for all $k \in \mathbb{N}_0$, we also have

$$\phi_{\boldsymbol{k}}(\varphi_{\text{tent}}(\boldsymbol{x})) \;=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} e_{\boldsymbol{\sigma}(\boldsymbol{k})}(\boldsymbol{x}). \tag{14}$$

Thus if we have a cubature rule Q_n^* which integrates exactly all Fourier basis functions $e_{\boldsymbol{\sigma}(\boldsymbol{k})}$ for all sign changes of $\boldsymbol{k} \in \Lambda$, then

$$\begin{aligned}Q_n(\phi_{\boldsymbol{k}}) \;:=\; Q_n^*(\phi_{\boldsymbol{k}} \circ \varphi_{\text{tent}}) \;&=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})}) \\ &=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \delta_{\boldsymbol{\sigma}(\boldsymbol{k}),\boldsymbol{0}} \;=\; \delta_{\boldsymbol{k},\boldsymbol{0}},\end{aligned} \tag{15}$$

as required for integral exactness in (12). The cubature rule Q_n is obtained from Q_n^* by applying the tent-transform to the points. □

Lemma 9 (Function reconstruction – sufficiency)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. If a cubature rule $Q_n^*(f) = \sum_{i=0}^{n-1} w_i^*\, f(\boldsymbol{t}_i^*)$ integrates exactly all Fourier basis functions $e_{\boldsymbol{h}}$ with $\boldsymbol{h} \in \mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)$, then the cubature rule $Q_n(f) = \sum_{i=0}^{n-1} w_i\, f(\boldsymbol{t}_i)$ with $w_i = w_i^*$ and $\boldsymbol{t}_i = \varphi_{\text{tent}}(\boldsymbol{t}_i^*)$ reconstructs exactly the cosine coefficients of all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ.*

Proof. For any $\boldsymbol{k}, \boldsymbol{k}' \in \mathbb{N}_0^d$ we have from (14) that

$$\phi_{\boldsymbol{k}}(\varphi_{\text{tent}}(\boldsymbol{x}))\,\phi_{\boldsymbol{k}'}(\varphi_{\text{tent}}(\boldsymbol{x})) \;=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0+|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \sum_{\boldsymbol{\sigma}'\in\mathcal{S}_{\boldsymbol{k}'}} e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k}')}(\boldsymbol{x}).$$

Thus if we have a cubature rule Q_n^* which integrates exactly all Fourier basis functions $e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k}')}$ for all sign changes of $\boldsymbol{k}, \boldsymbol{k}' \in \Lambda$, then

$$\begin{aligned} Q_n(\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}'}) \;:=\; Q_n^*((\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}'})\circ\varphi_{\text{tent}}) \;&=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0+|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \sum_{\boldsymbol{\sigma}'\in\mathcal{S}_{\boldsymbol{k}'}} Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k}')}) \\ &=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0+|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \sum_{\boldsymbol{\sigma}'\in\mathcal{S}_{\boldsymbol{k}'}} \delta_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k}'),\mathbf{0}} \\ &=\; \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0+|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}}} \sum_{\substack{\boldsymbol{\sigma}'\in\mathcal{S}_{\boldsymbol{k}'} \\ \sigma'_j=-\sigma_j \text{ when } k_j\neq 0}} \delta_{\boldsymbol{k},\boldsymbol{k}'} \qquad = \delta_{\boldsymbol{k},\boldsymbol{k}'}, \end{aligned} \tag{16}$$

which is the reconstruction property (13). In the penultimate step we used the property that $\boldsymbol{\sigma}(\boldsymbol{k}) + \boldsymbol{\sigma}'(\boldsymbol{k}') = \mathbf{0}$ if and only if $\boldsymbol{k} = \boldsymbol{k}'$ and $\sigma'_j = -\sigma_j$ whenever $k_j \neq 0$. □

Now we consider the situation where the cubature rule Q_n^* in Lemma 8 and Lemma 9 is a rank-1 lattice rule (8). In this case, the corresponding cubature rule $Q_n(f) = Q_n^*(f \circ \varphi_{\text{tent}})$ is often called a *tent-transformed lattice rule*, given explicitly by

$$Q_n(f) \;:=\; \frac{1}{n}\sum_{i=0}^{n-1} f\left(\varphi_{\text{tent}}\left(\frac{i\boldsymbol{z} \bmod n}{n}\right)\right).$$

The character property (9) of lattice rules enables us to conclude that the implications in Lemma 8 and Lemma 9 also hold in the opposite direction, and we obtain necessary and sufficient conditions for tent-transformed lattice rules to achieve our desired properties. Lemma 10 and the "only if" part of Lemma 11 have not been explicitly stated in the literature.

Lemma 10 (Integral exactness)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A tent-transformed lattice rule Q_n of a lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ integrates exactly all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ if and only if*

$$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in \mathcal{M}(\Lambda)\backslash\{\mathbf{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\mathcal{M}(\Lambda)\backslash\{\mathbf{0}\})}{2}+1,\, \max(\Lambda)\right\}.$$

Proof. The "if" direction follows by combining Lemma 4 with Lemma 8. To prove the "only if" direction, we observe from the character property (9) that the terms $Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})})$ on the right-hand side of (15) can only take the values of 1 or 0 so there can be no cancelation. In particular, when $\boldsymbol{k} \neq \mathbf{0}$, if $Q_n(\phi_{\boldsymbol{k}})$ is 0 on the left-hand side of (15) then necessarily all terms $Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})})$ are 0 on the right-hand side of (15), which implies $\boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z} \not\equiv_n 0$. When $\boldsymbol{k} = \mathbf{0}$ both sides of (15) are equal to 1, and trivially $Q_n(\phi_{\mathbf{0}}) = 1$ implies $Q_n^*(e_{\mathbf{0}}) = 1$. The CBC result follows from Theorem 23, noting that $\mathcal{M}(\Lambda)$ is centrally symmetric. □

Lemma 11 (Function reconstruction – plan A)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A tent-transformed lattice rule Q_n of a lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the cosine coefficients of all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ, by*

$$\widehat{f}_{\boldsymbol{k}} \;=\; \widehat{f}^{\,a}_{\boldsymbol{k}} \;:=\; Q_n(f\,\phi_{\boldsymbol{k}}) \;=\; Q_n^*((f\,\phi_{\boldsymbol{k}})\circ\varphi_{\text{tent}}) \qquad \text{for all } \boldsymbol{k}\in\Lambda,$$

if and only if

$$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in \mathcal{M}(\Lambda)\oplus\mathcal{M}(\Lambda)\backslash\{\mathbf{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)) + 1}{2},\, 2\max(\Lambda)\right\}.$$

Proof. As in the previous proof, the "if" direction follows by combining Lemma 5 with Lemma 9. When $\boldsymbol{k} \neq \boldsymbol{k}'$, if $Q_n(\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}'}) = 0$ on the left-hand side of (16), then necessarily all terms $Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k}')})$ are 0 on the right-hand side of (16), since the only permissible values are 0 or 1 due to the character property (9). When $\boldsymbol{k} = \boldsymbol{k}'$, if $Q_n(\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}}) = 1$ on the left-hand side of (16), then we have

$$\begin{aligned} 1 \;=\; Q_n(\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}}) \;&=\; \frac{1}{2^{|\boldsymbol{k}|_0}} \sum_{\substack{\boldsymbol{\sigma},\boldsymbol{\sigma}' \in \mathcal{S}_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})=\mathbf{0}}} Q_n^*(1) + \frac{1}{2^{|\boldsymbol{k}|_0}} \sum_{\substack{\boldsymbol{\sigma},\boldsymbol{\sigma}' \in \mathcal{S}_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})\neq\mathbf{0}}} Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})}) \\ &=\; 1 + \frac{1}{2^{|\boldsymbol{k}|_0}} \sum_{\substack{\boldsymbol{\sigma},\boldsymbol{\sigma}' \in \mathcal{S}_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})\neq\mathbf{0}}} Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})}). \end{aligned}$$

Necessarily, all terms $Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{\sigma}'(\boldsymbol{k})})$ must be zero for $\boldsymbol{\sigma}(\boldsymbol{k}) + \boldsymbol{\sigma}'(\boldsymbol{k}) \neq \mathbf{0}$. Hence we conclude that $(\boldsymbol{\sigma}(\boldsymbol{k}) + \boldsymbol{\sigma}'(\boldsymbol{k})) \cdot \boldsymbol{z} \not\equiv_n 0$ for all $\boldsymbol{k}, \boldsymbol{k}', \boldsymbol{\sigma}, \boldsymbol{\sigma}'$ satisfying $\boldsymbol{\sigma}(\boldsymbol{k}) + \boldsymbol{\sigma}'(\boldsymbol{k}') \neq \mathbf{0}$. The CBC result follows from Theorem 23. □

The tent transformation "stretches and folds the domain" so that essentially one half of the lattice points will land on top of the other half. This is given precisely by the property that

$$\varphi_{\text{tent}}(\boldsymbol{t}_i) \;=\; \varphi_{\text{tent}}(\boldsymbol{t}_{n-i}) \qquad \text{for } 1 \leq i < n/2.$$

There is one point at the origin $\boldsymbol{t}_0$ which will not be duplicated. When n is even, there is one other point $\varphi_{\text{tent}}(\boldsymbol{t}_{n/2})$ which will not be duplicated. All other points have a multiplicity of two provided that the generating vector includes at least one component z_j such that $\gcd(n, z_j) = 1$. Typically in a CBC construction we set $z_1 = 1$. This is sufficient to ensure uniqueness.

Lemma 12. *A rank-1 lattice rule with n points and generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$, where $\gcd(n, z_j) = 1$ for some j, has $\lfloor n/2 + 1 \rfloor$ unique points after tent transform.*

Proof. See Suryanarayana, Nuyens & Cools [29]. □

3.4 Alternative approach for function reconstruction

In this subsection we use an alternative approach for function reconstruction. There are two essential ingredients, which we will separate into **plan B** and **plan C** below. Firstly, we make use of the tent transform and bi-orthonormality to switch to a simpler set of functions. Secondly, we allow "self-aliasing" of the cubature rule to relax bi-normality and correct for this normalization afterward. We shall see in the next section that this alternative approach has a connection with the method in Potts & Volkmer [26].

Remark 13 (Orthonormal and bi-orthonormal families)**.** *For function reconstruction we demand that the inner product of all basis functions in our support set are exactly represented by replacing the integral by a cubature rule. An alternative is to be able to exactly represent the inner product of all basis functions with another set of orthogonal functions which have the bi-orthonormal property. In general, if $\{\mathrm{u}_{\boldsymbol{k}}\}$ is an orthonormal basis with $\langle \mathrm{u}_{\boldsymbol{k}}, \mathrm{u}_{\boldsymbol{k}'}\rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$, and $\{\mathrm{v}_{\boldsymbol{k}}\}$ is an orthogonal set with the bi-orthonormal property $\langle \mathrm{u}_{\boldsymbol{k}}, \mathrm{v}_{\boldsymbol{k}'}\rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$, while $\langle \mathrm{v}_{\boldsymbol{k}}, \mathrm{v}_{\boldsymbol{k}'}\rangle = d_{\boldsymbol{k}}\,\delta_{\boldsymbol{k},\boldsymbol{k}'}$ with $d_{\boldsymbol{k}}$ not necessarily equal to 1, then the coefficients of a function $f \in \operatorname{span}\{\mathrm{u}_{\boldsymbol{k}}\}$ can be calculated by the inner product against either $\{\mathrm{u}_{\boldsymbol{k}}\}$ or $\{\mathrm{v}_{\boldsymbol{k}}\}$ since $\langle f, \mathrm{u}_{\boldsymbol{k}}\rangle = \langle f, \mathrm{v}_{\boldsymbol{k}}\rangle$. A cubature rule which can exactly calculate the inner products $\langle \mathrm{u}_{\boldsymbol{k}}, \mathrm{v}_{\boldsymbol{k}'}\rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$ for all $\boldsymbol{k}, \boldsymbol{k}'$ in our support set then also has the reconstruction property. We will make use of such a bi-orthonormal property below.*

Lemma 14. *For any $f \in \mathcal{F}^{\cos}$ and $\boldsymbol{k} \in \mathbb{N}_0^d$, we can write the cosine coefficients in multiple ways*

$$\begin{aligned}\widehat{f}_{\boldsymbol{k}} &= \langle f, \phi_{\boldsymbol{k}} \rangle \\ &= \langle f \circ \varphi_{\text{tent}}, \phi_{\boldsymbol{k}} \circ \varphi_{\text{tent}} \rangle = \langle f \circ \varphi_{\text{tent}}, \sqrt{2}^{|\boldsymbol{k}|_0} e_{\boldsymbol{k}} \rangle = \langle f \circ \varphi_{\text{tent}}, \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet) \rangle \\ &= \langle f \circ \varphi_{\text{tent}}, \sqrt{2}^{|\boldsymbol{k}|_0} e_{\boldsymbol{\sigma}(\boldsymbol{k})} \rangle = \langle f \circ \varphi_{\text{tent}}, \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{\sigma}(\boldsymbol{k}) \cdot \bullet) \rangle \quad \text{for all } \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}.\end{aligned}$$

Proof. Using the Lebesgue preserving property of the tent transform and (14), we can write

$$\widehat{f}_{\boldsymbol{k}} = \int_{[0,1]^d} f(\varphi_{\text{tent}}(\boldsymbol{x}))\, \phi_{\boldsymbol{k}}(\varphi_{\text{tent}}(\boldsymbol{x}))\, \mathrm{d}\boldsymbol{x} = \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}} \int_{[0,1]^d} f(\varphi_{\text{tent}}(\boldsymbol{x}))\, e_{\boldsymbol{\sigma}(\boldsymbol{k})}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.$$

For each integral, we apply the change of variables $x_j' = x_j$ if $\sigma_j = 1$ and $x_j' = 1 - x_j$ if $\sigma_j = -1$, and use the properties $\exp(2\pi \mathrm{i}\, \boldsymbol{\sigma}(\boldsymbol{k}) \cdot \boldsymbol{x}) = \exp(2\pi \mathrm{i}\, \boldsymbol{k} \cdot \boldsymbol{\sigma}(\boldsymbol{x})) = \exp(2\pi \mathrm{i}\, \boldsymbol{k} \cdot \boldsymbol{x}')$ and $\varphi_{\text{tent}}(\boldsymbol{x}) = \varphi_{\text{tent}}(\boldsymbol{x}')$ to deduce that

$$\int_{[0,1]^d} f(\varphi_{\text{tent}}(\boldsymbol{x}))\, e_{\boldsymbol{\sigma}(\boldsymbol{k})}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \int_{[0,1]^d} f(\varphi_{\text{tent}}(\boldsymbol{x}'))\, e_{\boldsymbol{k}}(\boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}' \qquad \text{for all } \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}.$$

Thus all integrals are equal regardless of the sign changes on $\boldsymbol{k}$. Furthermore, since f is a real-valued function, all its cosine coefficients will be real. Hence we may replace the exponential function $e_{\boldsymbol{\sigma}(\boldsymbol{k})}$ by its real part $\cos(2\pi \boldsymbol{\sigma}(\boldsymbol{k}) \cdot \bullet)$. □

By considering the special case of $f = \phi_{\boldsymbol{k}'}$ in Lemma 14 we obtain

$$\delta_{\boldsymbol{k},\boldsymbol{k}'} = \langle \phi_{\boldsymbol{k}'}, \phi_{\boldsymbol{k}} \rangle = \langle \phi_{\boldsymbol{k}'} \circ \varphi_{\text{tent}}, \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet) \rangle \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \mathbb{N}_0^d, \tag{17}$$

i.e., the functions $\{\mathrm{u}_{\boldsymbol{k}} = \phi_{\boldsymbol{k}} \circ \varphi_{\text{tent}}\}$ and $\{\mathrm{v}_{\boldsymbol{k}} = \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)\}$ are bi-orthonormal in L^2. Instead of demanding a cubature rule with exactness for the first inner product in (17) (as we did in (13)), below we seek a cubature rule with exactness for the second inner product in (17) (see (21) below), thus preserving bi-orthonormality.

Lemma 15 (Function reconstruction – plan B)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the cosine coefficients of all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ, by*

$$\widehat{f}_{\boldsymbol{k}} = \widehat{f}_{\boldsymbol{k}}^{\,b} := Q_n^*((f \circ \varphi_{\text{tent}})\, \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)) \qquad \textit{for all } \boldsymbol{k} \in \Lambda, \tag{18}$$

if and only if

$$\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \qquad \textit{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda,\ \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'},\ \boldsymbol{\sigma}(\boldsymbol{k}') \neq \boldsymbol{k}, \tag{19}$$

which is equivalent to

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \textit{for all } \boldsymbol{h} \in (\Lambda \oplus \mathcal{M}(\Lambda)) \backslash \{\boldsymbol{0}\}. \tag{20}$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\#(\Lambda \oplus \mathcal{M}(\Lambda)),\, 2\max(\Lambda)\right\}.$$

Proof. Substituting the cosine series of f into (18), it follows that we have exact reconstruction of the cosine coefficients, i.e.,

$$\widehat{f}_{\boldsymbol{k}}^{\,b} = \sum_{\boldsymbol{k}' \in \Lambda} \widehat{f}_{\boldsymbol{k}'}\, Q_n^*((\phi_{\boldsymbol{k}'} \circ \varphi_{\text{tent}})\, \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)) = \widehat{f}_{\boldsymbol{k}} \quad \text{for all } f \in \mathcal{F}_\Lambda^{\cos} \text{ and } \boldsymbol{k} \in \Lambda,$$

if and only if

$$Q_n^*((\phi_{\boldsymbol{k}'} \circ \varphi_{\text{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)) \;=\; \delta_{\boldsymbol{k}',\boldsymbol{k}} \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda. \tag{21}$$

It remains to prove that (21) holds if and only if (19) holds.

Using (14) and $\cos(2\pi \boldsymbol{k} \cdot \bullet) = (e_{\boldsymbol{k}} + e_{-\boldsymbol{k}})/2$, we can write

$$\begin{aligned} Q_n^*((\phi_{\boldsymbol{k}'} \circ \varphi_{\text{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)) \;&=\; \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{2 \cdot \sqrt{2}^{|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'}} \Big(Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k}')+\boldsymbol{k}}) + Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k}')-\boldsymbol{k}}) \Big) \\ &=\; \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{2 \cdot \sqrt{2}^{|\boldsymbol{k}'|_0}} \sum_{\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'}} \Big(Q_n^*(e_{-\boldsymbol{\sigma}(\boldsymbol{k}')+\boldsymbol{k}}) + Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k}')-\boldsymbol{k}}) \Big), \end{aligned} \tag{22}$$

where it is valid to replace one $\boldsymbol{\sigma}(\boldsymbol{k}')$ by $-\boldsymbol{\sigma}(\boldsymbol{k}')$ since we sum over all unique sign changes.

Consider first the case $\boldsymbol{k} \neq \boldsymbol{k}'$. Then $\boldsymbol{\sigma}(\boldsymbol{k}') \neq \boldsymbol{k}$. By the character property (9) we know that $Q_n^*(e_{\pm(\boldsymbol{\sigma}(\boldsymbol{k}')-\boldsymbol{k})})$ can only take the values of 1 or 0. Thus (22) is equal to 0 if and only if all terms $Q_n^*(e_{\pm(\boldsymbol{\sigma}(\boldsymbol{k}')-\boldsymbol{k})})$ are 0, which holds following the character property if and only if $\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$.

Consider now the case $\boldsymbol{k} = \boldsymbol{k}'$. Then we can rewrite (22) as

$$Q_n^*((\phi_{\boldsymbol{k}} \circ \varphi_{\text{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet)) \;=\; 1 + \frac{1}{2} \sum_{\substack{\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k}) \neq \boldsymbol{k}}} \Big(Q_n^*(e_{-\boldsymbol{\sigma}(\boldsymbol{k})+\boldsymbol{k}}) + Q_n^*(e_{\boldsymbol{\sigma}(\boldsymbol{k})-\boldsymbol{k}}) \Big). \tag{23}$$

Using the character property as before, we conclude that (23) is equal to 1 if and only if all terms $Q_n^*(e_{\pm(\boldsymbol{\sigma}(\boldsymbol{k})-\boldsymbol{k})})$ are 0 whenever $\boldsymbol{\sigma}(\boldsymbol{k}) \neq \boldsymbol{k}$, and in turn this means that $\boldsymbol{\sigma}(\boldsymbol{k}) \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$ except for when $\boldsymbol{\sigma}(\boldsymbol{k}) = \boldsymbol{k}$. Combining all conditions, we conclude that (21) holds if and only if (19) holds.

Finally (19) is clearly equivalent to (20). The condition on n then follows from Theorem 23, noting that the set $\Lambda \oplus \mathcal{M}(\Lambda)$ includes $\mathbf{0}$ but is not centrally symmetric. □

In the next lemma we propose another modification which allows "self-aliasing" in the lattice rule with respect to sign changes (see $\boldsymbol{\sigma}(\boldsymbol{k}) \cdot \boldsymbol{z} \equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$ in (24) below). Consequently, the right-hand side of (21) for the case $\boldsymbol{k} = \boldsymbol{k}'$ can be an integer $c_{\boldsymbol{k}}$, not necessarily 1 (see (26) below). In other words, the cubature rule no longer preserves bi-normality, with normalization to be corrected by this factor $c_{\boldsymbol{k}}$.

Lemma 16 (Function reconstruction – plan C)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the cosine coefficients of all cosine space functions $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ, by*

$$\begin{aligned} \widehat{f}_{\boldsymbol{k}} \;=\; \widehat{f}_{\boldsymbol{k}}^{\,c} \;&:=\; \frac{Q_n^*((f \circ \varphi_{\text{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi \boldsymbol{k} \cdot \bullet))}{c_{\boldsymbol{k}}}, \quad \textit{with} \\ c_{\boldsymbol{k}} \;&:=\; \#\big\{ \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}} \;:\; \boldsymbol{\sigma}(\boldsymbol{k}) \cdot \boldsymbol{z} \equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \big\} \quad \textit{for all } \boldsymbol{k} \in \Lambda, \end{aligned} \tag{24}$$

if and only if

$$\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \qquad \textit{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda,\, \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'},\, \boldsymbol{k} \neq \boldsymbol{k}'. \tag{25}$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{ \#\Lambda\, \#\mathcal{M}(\Lambda),\, 2\max(\Lambda) \right\}.$$

Proof. Following the argument in the proof of Lemma 15, we now have exact reconstruction of the cosine coefficients if and only if (instead of (21))

$$Q_n^*((\phi_{\boldsymbol{k}'} \circ \varphi_{\mathrm{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0}\cos(2\pi\boldsymbol{k}\cdot\bullet)) \;=\; c_{\boldsymbol{k}}\,\delta_{\boldsymbol{k}',\boldsymbol{k}} \qquad \text{for all } \boldsymbol{k},\boldsymbol{k}' \in \Lambda. \tag{26}$$

The case $\boldsymbol{k} \neq \boldsymbol{k}'$ is the same as in Lemma 15. It suffices to reconsider the case $\boldsymbol{k} = \boldsymbol{k}'$. Instead of separating out the term $\boldsymbol{\sigma}(\boldsymbol{k}) = \boldsymbol{k}$ as in (23), we apply the character property (9) for $Q_n^*(e_{\pm(\boldsymbol{\sigma}(\boldsymbol{k}')-\boldsymbol{k})})$ in (22) with $\boldsymbol{k} = \boldsymbol{k}'$ to arrive at

$$Q_n^*((\phi_{\boldsymbol{k}} \circ \varphi_{\mathrm{tent}})\,\sqrt{2}^{|\boldsymbol{k}|_0}\cos(2\pi\boldsymbol{k}\cdot\bullet)) \;=\; \frac{1}{2} \sum_{\substack{\boldsymbol{\sigma}\in\mathcal{S}_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z}\equiv_n \boldsymbol{k}\cdot\boldsymbol{z}}} 2,$$

which is equal to $c_{\boldsymbol{k}}$ as required. The CBC result is proved in Theorem 26 later. □

3.5 Fast calculation of cosine coefficients and function values

Here we can also make use of a one-dimensional fast Fourier transform to map cosine coefficients to function values on the tent-transformed lattice points and vice versa.

Lemma 17. *Let $\boldsymbol{z}$ be a generating vector for an n-point rank-1 lattice satisfying the reconstruction property on an arbitrary index set $\Lambda \subset \mathbb{N}_0^d$ according to Lemma 11 (plan A), Lemma 15 (plan B) or Lemma 16 (plan C). For a function $f \in \mathcal{F}_\Lambda^{\cos}$ solely supported on Λ we can compute*

coefficients from function values:	*function values from coefficients:*
// prepare function value vector $\boldsymbol{f} \in \mathbb{R}^n$	*// prepare coefficient vector* $\boldsymbol{F} \in \mathbb{R}^n$
$f_0 = f(\boldsymbol{0})$	$\boldsymbol{F} = \boldsymbol{0} \in \mathbb{R}^n$
`for` $i \in \{1,\dots,\lfloor n/2 \rfloor\}$`:`	`for` $\boldsymbol{k} \in \Lambda$`:`
$\quad f_i = f(\varphi_{\mathrm{tent}}((i\boldsymbol{z} \bmod n)/n))$	$\quad$ `for` $\boldsymbol{\sigma} \in S_{\boldsymbol{k}}$`:`
$\quad f_{n-i} = f_i$	$\quad\quad F_{(\boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z} \bmod n)} = F_{(\boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z} \bmod n)} + \widehat{f}_{\boldsymbol{k}}/\sqrt{2}^{|\boldsymbol{k}|_0}$
// compute coefficient vector $\boldsymbol{F} \in \mathbb{R}^n$	*// compute function value vector* $\boldsymbol{f} \in \mathbb{R}^n$
$\boldsymbol{F} = \mathrm{FFT}(\boldsymbol{f})$	$\boldsymbol{f} = \mathrm{IFFT}(\boldsymbol{F})$
// $\widehat{f}_{\boldsymbol{k}}$ *is given by* $\sqrt{2}^{\lvert\boldsymbol{k}\rvert_0}\, F_{(\boldsymbol{k}\cdot\boldsymbol{z} \bmod n)}/c_{\boldsymbol{k}}$	*//* f_i *gives the value of* $f(\varphi_{\mathrm{tent}}((i\boldsymbol{z} \bmod n)/n))$

where $\boldsymbol{f} \in \mathbb{R}^n$ is a vector containing function values with the symmetry $f_i = f_{n-i}$ and $\boldsymbol{F} \in \mathbb{R}^n$ is a vector containing cosine coefficients with the symmetry $F_\kappa = F_{n-\kappa}$. Here FFT *and* IFFT *are the one-dimensional fast Fourier transform and its inverse, respectively, with a normalization $1/n$ for* FFT *and 1 for* IFFT*; both mappings have cost $\mathcal{O}(n\log(n))$. For plan A and B we set $c_{\boldsymbol{k}} = 1$ for all $\boldsymbol{k}$.*

Alternatively, for even $n = 2m$ a length $m+1$ DCT-I can be used, while for odd $n = 2m-1$ a length m DCT-V can be used. In this case the memory requirement and computational effort is halved (w.r.t. a real-to-real FFT).

Proof. We show the result by using the calculation of Lemma 16 (plan C). Plan A and plan B are essentially the same with $c_{\boldsymbol{k}} = 1$. For all plans we have the option to use the inner product with respect to $\cos(2\pi\, i\,\boldsymbol{k}\cdot\boldsymbol{z}/n)$ as given in Lemma 14 since the reconstruction property is guaranteed by the conditions of plan C which is a subset of the conditions of plan B which in turn is a subset of the conditions of plan A.

Let $f_i := f(\varphi_{\text{tent}}((i\boldsymbol{z} \bmod n)/n))$ for $i = 0, \ldots, n-1$. In all three plans we have for $\boldsymbol{k} \in \Lambda$,

$$\widehat{f}_{\boldsymbol{k}} = \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}} \frac{1}{n} \sum_{i=0}^{n-1} f_i \cos\Big(2\pi \frac{i\,\boldsymbol{k}\cdot\boldsymbol{z}}{n}\Big) = \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}} \frac{1}{n} \sum_{i=0}^{n-1} f_i \left[\cos\Big(2\pi \frac{i\,\boldsymbol{k}\cdot\boldsymbol{z}}{n}\Big) + \mathrm{i}\,\sin\Big(2\pi \frac{i\,\boldsymbol{k}\cdot\boldsymbol{z}}{n}\Big)\right]$$

$$= \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}} \underbrace{\frac{1}{n} \sum_{i=0}^{n-1} f_i \exp\Big(-2\pi\mathrm{i}\,\frac{i\,\boldsymbol{k}\cdot\boldsymbol{z}}{n}\Big)}_{=: F_{(\boldsymbol{k}\cdot\boldsymbol{z} \bmod n)}},$$

which follows because of the symmetry of $f_i = f_{n-i}$ due to the tent transform and the odd and even properties of the sine and cosine functions respectively. The last expression is a scaled discrete Fourier transform of length n in terms of i and $\kappa = \boldsymbol{k}\cdot\boldsymbol{z} \bmod n$. This shows the calculation of coefficients from function values by one-dimensional FFT.

Next we consider the evaluation of function values from coefficients. For each $i = 0, \ldots, n-1$, we have from (14) that

$$f_i = \sum_{\boldsymbol{k}\in\Lambda} \widehat{f}_{\boldsymbol{k}} \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}} \sum_{\boldsymbol{\sigma}\in S_{\boldsymbol{k}}} \exp\Big(2\pi\mathrm{i}\,\frac{i\,\boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z}}{n}\Big) = \sum_{\kappa=0}^{n-1} \underbrace{\Bigg(\sum_{\boldsymbol{k}\in\Lambda} \sum_{\substack{\boldsymbol{\sigma}\in S_{\boldsymbol{k}} \\ \boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z} \equiv_n \kappa}} \widehat{f}_{\boldsymbol{k}} \frac{1}{\sqrt{2}^{|\boldsymbol{k}|_0}}\Bigg)}_{=: F_\kappa} \exp\Big(2\pi\mathrm{i}\,\frac{i\,\kappa}{n}\Big),$$

where it is unfortunately not possible to avoid considering all sign changes of $\boldsymbol{k}$. For plan A and plan B there is only one κ associated with each $\sigma(\boldsymbol{k})$. For plan C it might occur that different sign changes on the same $\boldsymbol{k}$ map to the same value of κ, hence the summation in the algorithm to prepare the coefficient vector $\boldsymbol{F}$.

Now we explain how to make use of DCT using symmetry. We have $f_i = f_{n-i}$ due to the symmetry of φ_{tent}. In the formula below, we will write $f_{n/2}$ which is to be interpreted in the way we just defined for even n, and to be considered equal to zero for odd n. Then, by making use of the symmetry, we have for general n (odd or even)

$$\widehat{f}_{\boldsymbol{k}} = \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}} \frac{1}{n}\Bigg(f_0 + 2\sum_{i=1}^{\lfloor (n-1)/2\rfloor} f_i \cos\Big(2\pi\,\frac{i\,\kappa}{n}\Big) + f_{n/2}\,\cos(\pi\,\kappa)\Bigg),$$

where $f_{n/2}\,\cos(\pi\,\kappa)$ is only present for even n. Now for $n = 2m$ we have $\lfloor (n-1)/2\rfloor = n/2 - 1 = m-1$ and we find

$$\widehat{f}_{\boldsymbol{k}} = \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}}\,F_\kappa, \qquad \text{with} \qquad F_\kappa := \frac{1}{m}\Bigg(\frac{1}{2}\,f_0 + \sum_{i=1}^{m-1} f_i\,\cos\Big(\pi\,\frac{i\,\kappa}{m}\Big) + \frac{1}{2}\,f_m\,\cos(\pi\,\kappa)\Bigg),$$

which is the formula for the one-dimensional DCT-I of length $m+1$ to turn the sequence $f_0, f_1, \ldots, f_m$ into the sequence $F_0, F_1, \ldots, F_m$. For odd $n = 2m-1$ we have $\lfloor (n-1)/2\rfloor = m-1$ and we find

$$\widehat{f}_{\boldsymbol{k}} = \frac{\sqrt{2}^{|\boldsymbol{k}|_0}}{c_{\boldsymbol{k}}}\,F_\kappa, \qquad \text{with} \qquad F_\kappa := \frac{1}{2m-1}\Bigg(f_0 + 2\sum_{i=1}^{m-1} f_i \cos\Big(2\pi\,\frac{i\,\kappa}{2m-1}\Big)\Bigg),$$

which is the formula for the one-dimensional DCT-V of length m to turn the sequence $f_0, f_1, \ldots, f_{m-1}$ into the sequence $F_0, F_1, \ldots, F_{m-1}$. For DCT-I and DCT-V see Martucci [20] (definitions (A.1) and (A.5) therein).

Similarly, we have $F_\kappa = F_{n-\kappa}$ so we can write

$$f_i = F_0 + 2\sum_{\kappa=1}^{\lfloor (n-1)/2\rfloor} F_\kappa\,\cos\Big(2\pi\frac{i\,\kappa}{n}\Big) + F_{n/2}\,\cos(\pi\,i),$$

where $F_{n/2} := 0$ if n is odd. Therefore DCT-I and DCT-V work in an analogous way. □

We note that all coefficients calculated by the FFT are real because of the symmetry in the input, but in a typical implementation one would take the real part in case of a complex FFT routine to remove possible numerical noise. Alternatively one can make use of a special real to real FFT implementation or use a specific implementation for the corresponding DCT.

Similarly to Remark 7, one can also extend the set Λ such that $\sigma(\boldsymbol{k}) \cdot \boldsymbol{z} \bmod n$ covers as many values as possible in $\mathbb{Z}_n$ for functions which are not solely supported on Λ.

4 Nonperiodic setting based on Chebyshev polynomials

4.1 Chebyshev series

In the univariate case, the Chebyshev polynomials of the first kind for $|x| \le 1$ can be written

$$T_k(x) \;=\; \cos(k \arccos(x)), \qquad k = 0, 1, 2, \ldots, \qquad \text{for } x \in [-1, 1].$$

We have orthogonality with respect to the Chebyshev weight $(\sqrt{1-x^2})^{-1}$:

$$\int_{-1}^{1} T_k(x)\, T_{k'}(x)\, \frac{\mathrm{d}x}{\sqrt{1-x^2}} = \begin{cases} 0, & \text{if } k \ne k', \\ \pi, & \text{if } k = k' = 0, \\ \pi/2, & \text{if } k = k' \ne 0. \end{cases}$$

To obtain an orthonormal basis on $[-1, 1]$, we first normalize the measure to 1 by adjusting the Chebyshev weight to $(\pi\sqrt{1-x^2})^{-1}$. Then we define

$$\eta_k(x) := \begin{cases} T_0(x) = 1, & \text{if } k = 0, \\ \sqrt{2}\, T_k(x), & \text{if } k = 1, 2, \ldots, \end{cases}$$

and for the multivariate case we define the tensor product basis functions

$$\eta_{\boldsymbol{k}}(\boldsymbol{x}) := \prod_{j=1}^{d} \eta_{k_j}(x_j) = \sqrt{2}^{|\boldsymbol{k}|_0} \prod_{j=1}^{d} T_{k_j}(x_j), \tag{27}$$

where, as for cosine basis, $|\boldsymbol{k}|_0$ denotes the count of the nonzero entries in the vector $\boldsymbol{k}$.

Let $\mathcal{F}^{\mathrm{Cheb}}$ denote the space of real valued functions defined on $[-1, 1]^d$ with absolutely converging Chebyshev series:

$$\mathcal{F}^{\mathrm{Cheb}} := \Big\{ f \in L^2 \;\Big|\; f : [-1, 1]^d \to \mathbb{R},\; f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \widehat{f}_{\boldsymbol{k}}\, \eta_{\boldsymbol{k}}(\boldsymbol{x}) \text{ and } \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} |\widehat{f}_{\boldsymbol{k}}| < \infty \Big\},$$

where $\widehat{f}_{\boldsymbol{k}}$ are the Chebyshev coefficients of f. We equip $\mathcal{F}^{\mathrm{Cheb}}$ with the weighted L^2 inner product

$$\langle f_1, f_2 \rangle_\mu \;:=\; \int_{[-1,1]^d} f_1(\boldsymbol{x})\, f_2(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x}), \qquad \mu(\mathrm{d}\boldsymbol{x}) \;:=\; \frac{\mathrm{d}\boldsymbol{x}}{\prod_{j=1}^{d} \left(\pi\sqrt{1-x_j^2}\right)}.$$

The orthonormal Chebyshev basis functions satisfy $\langle \eta_{\boldsymbol{k}}, \eta_{\boldsymbol{k}'} \rangle_\mu = \delta_{\boldsymbol{k},\boldsymbol{k}'}$. The Chebyshev coefficients are

$$\widehat{f}_{\boldsymbol{k}} := \langle f, \eta_{\boldsymbol{k}} \rangle_\mu = \int_{[-1,1]^d} f(\boldsymbol{x})\, \eta_{\boldsymbol{k}}(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x}).$$

Given an arbitrary finite index set $\Lambda \subset \mathbb{N}_0^d$, we consider the subspace $\mathcal{F}_\Lambda^{\mathrm{Cheb}}$ of all functions whose Chebyshev series is supported solely on Λ, i.e.,

$$\text{for } f \in \mathcal{F}_\Lambda^{\mathrm{Cheb}}: \qquad f(\boldsymbol{x}) \;=\; \sum_{\boldsymbol{k} \in \Lambda} \widehat{f}_{\boldsymbol{k}}\, \eta_{\boldsymbol{k}}(\boldsymbol{x}).$$

4.2 Isomorphism with the cosine space via cosine transform

The Chebyshev basis functions and the cosine basis functions are related by the mapping

$$\eta_{\boldsymbol{k}}(\boldsymbol{x}) = \phi_{\boldsymbol{k}}(\boldsymbol{x}') \quad \iff \quad \arccos(\boldsymbol{x}) = \pi\,\boldsymbol{x}' \quad \iff \quad \boldsymbol{x} = \cos(\pi\,\boldsymbol{x}'),$$

where the cosine function and its inverse are applied componentwise. This provides an isomorphism between the Chebyshev setting and the cosine space, with

$$f \in \mathcal{F}^{\mathrm{Cheb}} \quad \iff \quad f_{\cos} := f(\cos(\pi\,\bullet)) \in \mathcal{F}^{\cos}.$$

To get a sense of how this transformation works, it is informative to consider the univariate case:

$$\int_{-1}^{1} f(x)\,\frac{\mathrm{d}x}{\pi\,\sqrt{1-x^2}} = \int_{1}^{0} f(\cos(\pi x'))\,\frac{-\pi\,\sin(\pi x')\,\mathrm{d}x'}{\pi\,\sqrt{1-\cos^2(\pi x')}} = \int_{0}^{1} f(\cos(\pi x'))\,\mathrm{d}x'.$$

For the multivariate case we have the integral operator

$$I_\mu(f) := \int_{[-1,1]^d} f(\boldsymbol{x})\,\mu(\mathrm{d}\boldsymbol{x}) = \int_{[0,1]^d} f(\cos(\pi\,\boldsymbol{x}'))\,\mathrm{d}\boldsymbol{x}' = I(f(\cos(\pi\,\bullet))) = I(f_{\cos}),$$

so that $I_\mu(\eta_{\boldsymbol{k}}) = I(\phi_{\boldsymbol{k}})$, $I_\mu(\eta_{\boldsymbol{k}}\,\eta_{\boldsymbol{k}'}) = I(\phi_{\boldsymbol{k}}\,\phi_{\boldsymbol{k}'})$, and

$$\langle f, \eta_{\boldsymbol{k}}\rangle_\mu = \langle f(\cos(\pi\,\bullet)), \phi_{\boldsymbol{k}}\rangle = \langle f_{\cos}, \phi_{\boldsymbol{k}}\rangle,$$

i.e., the Chebyshev coefficients of f are precisely the cosine coefficients of $f_{\cos}$.

4.3 Chebyshev coefficients by transformed rank-1 lattice rules

The isomorphism between the spaces means that we can bring all results from the cosine space over to the Chebyshev space. Noting the useful property

$$\cos(\pi\,\varphi_{\mathrm{tent}}(\boldsymbol{t})) = \cos(2\pi\,\boldsymbol{t}),$$

we then arrive at the results for a *tent-transformed and then cosine-transformed lattice rule*, which is given explicitly by

$$Q_n(f) := \frac{1}{n}\sum_{i=0}^{n-1} f\left(\cos\left(2\pi\frac{i\boldsymbol{z} \bmod n}{n}\right)\right) = \frac{1}{n}\sum_{i=0}^{n-1} f\left(\cos\left(2\pi\frac{i\boldsymbol{z}}{n}\right)\right).$$

Since $\cos(2\pi\,i\boldsymbol{z}/n) = \cos(2\pi\,(n-i)\boldsymbol{z}/n)$, the cubature points double up and we can write

$$Q_n(f) = \begin{cases} \dfrac{f(\boldsymbol{1})}{n} + \dfrac{f(-\boldsymbol{1})}{n} + \dfrac{2}{n}\displaystyle\sum_{i=1}^{n/2-1} f\left(\cos\left(2\pi\frac{i\boldsymbol{z}}{n}\right)\right) & \text{if } n \text{ is even,} \\ \dfrac{f(\boldsymbol{1})}{n} + \dfrac{2}{n}\displaystyle\sum_{i=1}^{(n-1)/2} f\left(\cos\left(2\pi\frac{i\boldsymbol{z}}{n}\right)\right) & \text{if } n \text{ is odd.} \end{cases}$$

Thus the cubature rule can be computed with $\lfloor n/2+1\rfloor$ function evaluations, where the cubature weight for $i=0$ and $i=n/2$ (if n is even) are $1/n$ and the others are $2/n$. In general there can be further duplication of points. However, if $\gcd(z_j, n) = 1$ for at least one $j = 1,\dots,d$, then all the $\lfloor n/2+1\rfloor$ points are distinct, see Lemma 12.

For even $n = 2m$ this point set has previously been called a "Chebyshev lattice", see, e.g., [6, 26], defined by $\{\cos(\pi i\boldsymbol{z}/m) : i = 0,\dots,m\}$. However, we prefer the interpretation as a tent-transformed and then cosine-transformed lattice, since then we can also use odd n, and the cubature weights are automatically correct according to the multiplicity of the points.

We now state the analogous results to Lemmas 10, 11, 15, 16, 17 from the cosine space.

Lemma 18 (Integral exactness)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A tent-transformed and then cosine transformed lattice rule with n points and generating vector $\boldsymbol{z}$ integrates exactly (against the Chebyshev density) all Chebyshev space functions $f \in \mathcal{F}_\Lambda^{\mathrm{Cheb}}$ solely supported on Λ if and only if*

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in \mathcal{M}(\Lambda)\backslash\{\boldsymbol{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\mathcal{M}(\Lambda)\backslash\{\boldsymbol{0}\})}{2} + 1,\, \max(\Lambda)\right\}.$$

Lemma 19 (Function reconstruction – plan A)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the Chebyshev coefficients of all Chebyshev space functions $f \in \mathcal{F}_\Lambda^{\mathrm{Cheb}}$ solely supported on Λ, by*

$$\widehat{f}_{\boldsymbol{k}} = \widehat{f}_{\boldsymbol{k}}^{\,a} := Q_n^*\left(f(\cos(2\pi\,\bullet))\,(\phi_{\boldsymbol{k}} \circ \varphi_{\mathrm{tent}})\right) \qquad \text{for all } \boldsymbol{k} \in \Lambda,$$

if and only if

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in \mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)\backslash\{\boldsymbol{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\frac{\#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)) + 1}{2},\, 2\max(\Lambda)\right\}$$

Lemma 20 (Function reconstruction – plan B)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the Chebyshev coefficients of all Chebyshev space functions $f \in \mathcal{F}_\Lambda^{\mathrm{Cheb}}$ solely supported on Λ, by*

$$\widehat{f}_{\boldsymbol{k}} = \widehat{f}_{\boldsymbol{k}}^{\,b} := Q_n^*\left(f(\cos(2\pi\,\bullet))\,\sqrt{2}^{|\boldsymbol{k}|_0}\cos(2\pi\boldsymbol{k}\cdot\bullet)\right) \qquad \text{for all } \boldsymbol{k} \in \Lambda,$$

if and only if

$$\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda,\, \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'},\, \boldsymbol{\sigma}(\boldsymbol{k}') \neq \boldsymbol{k},$$

which is equivalent to

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all } \boldsymbol{h} \in (\Lambda \oplus \mathcal{M}(\Lambda))\backslash\{\boldsymbol{0}\}.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\#(\Lambda \oplus \mathcal{M}(\Lambda)),\, 2\max(\Lambda)\right\}.$$

Lemma 21 (Function reconstruction – plan C)**.** *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set. A lattice rule Q_n^* with n points and generating vector $\boldsymbol{z}$ reconstructs exactly the Chebyshev coefficients of all Chebyshev space functions $f \in \mathcal{F}_\Lambda^{\mathrm{Cheb}}$ solely supported on Λ, by*

$$\begin{aligned} \widehat{f}_{\boldsymbol{k}} = \widehat{f}_{\boldsymbol{k}}^{\,c} &:= \frac{Q_n^*\big(f(\cos(2\pi\,\bullet))\,\sqrt{2}^{|\boldsymbol{k}|_0}\cos(2\pi\boldsymbol{k}\cdot\bullet)\big)}{c_{\boldsymbol{k}}}, \quad \text{with} \\ c_{\boldsymbol{k}} &:= \#\big\{\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}} \,:\, \boldsymbol{\sigma}(\boldsymbol{k}) \cdot \boldsymbol{z} \equiv_n \boldsymbol{k} \cdot \boldsymbol{z}\big\} \quad \text{for all } \boldsymbol{k} \in \Lambda, \end{aligned} \tag{28}$$

if and only if

$$\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda,\, \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'},\, \boldsymbol{k} \neq \boldsymbol{k}'.$$

Such a generating vector $\boldsymbol{z}$ can be constructed component-by-component if n is a prime satisfying

$$n > \max\left\{\#\Lambda\,\#\mathcal{M}(\Lambda),\, 2\max(\Lambda)\right\}.$$

Lemma 22. *We can use FFTs or DCTs to map Chebyshev coefficients to function values on tent-transformed and then cosine-transformed lattice points, and the other way round, for an n-point rank-1 lattice rule with generating vector $\boldsymbol{z}$ satisfying the non-aliasing conditions of Lemma 19 (plan A), Lemma 20 (plan B) or Lemma 21 (plan C) on an arbitrary index set $\Lambda \subset \mathbb{N}_0^d$ by replacing $\varphi_{\rm tent}(\bullet)$ by $\cos(\pi\,\varphi_{\rm tent}(\bullet))$ in the statement of Lemma 17.*

Lemma 21 with even $n = 2m$ in combination with the DCT-I in Lemma 22 is essentially the approach in Potts & Volkmer [26].

5 The CBC construction

5.1 Induction proof for the component-by-component construction

Let $\mathbb{Z}_n^* := \{1, \ldots, n-1\}$ for n prime. The necessary and sufficient conditions on the lattice rule generating vector $\boldsymbol{z} \in (\mathbb{Z}_n^*)^d$ for integral exactness and function reconstruction in most cases boil down to the same generic form of verifying for a given index set $\mathcal{A} \subset \mathbb{Z}^d$ that

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all} \quad \boldsymbol{h} \in \mathcal{A} \backslash \{\boldsymbol{0}\}. \tag{29}$$

The following theorem justifies a generic component-by-component (CBC) algorithm to find a $\boldsymbol{z}$ satisfying this condition. The inductive argument needs to work with projections of the index set $\mathcal{A}$ down to the lower coordinates. We consider two definitions for the projections since each has its advantages:

$$\text{either} \qquad \mathcal{A}_s := \{\boldsymbol{h} \in \mathbb{Z}^s : (\boldsymbol{h}, \boldsymbol{0}) \in \mathcal{A}\} \qquad \text{for } s = 1, \ldots, d, \tag{30}$$

$$\text{or} \qquad \mathcal{A}_s := \{\boldsymbol{h} \in \mathbb{Z}^s : (\boldsymbol{h}, \boldsymbol{h}_\perp) \in \mathcal{A} \text{ for some } \boldsymbol{h}_\perp \in \mathbb{Z}^{d-s}\} \qquad \text{for } s = 1, \ldots, d. \tag{31}$$

Both definitions yield $\mathcal{A}_d = \mathcal{A}$. The definition (30) includes only the indices whose higher components are zero; we shall refer to this as the 'zero' projection. The definition (31) includes all indices obtained by truncating the original indices; we shall refer to this as the 'full' projection. If the index set $\mathcal{A}$ is downward closed then they are the same; otherwise (30) is a subset of (31). The full projection (31) was used in [4, 9, 10]; the zero projection (30) is new in this paper.

The condition $n > \max(\mathcal{A})$ in the theorem guarantees that the components of $\boldsymbol{h} \in \mathcal{A}$ all satisfy $|h_j| < n$, and thus $h_j \equiv_n 0$ if and only if $h_j = 0$. This condition can be replaced by the direct assumption that there is no $\boldsymbol{h} \in \mathcal{A}$ with a nonzero component h_j that is a multiple of n.

Theorem 23. *Let $\mathcal{A} \subset \mathbb{Z}^d$ be an arbitrary index set, and let n be a prime number satisfying*

$$n > \max\Big\{\frac{\#(\mathcal{A}\backslash\{\boldsymbol{0}\})}{\kappa} + 1\,,\ \max(\mathcal{A})\Big\}, \tag{32}$$

with $\kappa = 2$ if $\mathcal{A}$ is centrally symmetric and $\kappa = 1$ otherwise. Define the projections $\mathcal{A}_s$ by (30) or (31). Then a generating vector $\boldsymbol{z}^* = (z_1, \ldots, z_d) \in (\mathbb{Z}_n^*)^d$ can be constructed component-by-component such that for all $s = 1, \ldots, d$ and $\boldsymbol{z} = (z_1, \ldots, z_s)$ we have

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \textit{for all} \quad \boldsymbol{h} \in \mathcal{A}_s \backslash \{\boldsymbol{0}\}. \tag{33}$$

Proof. The result for centrally symmetric and downward closed index sets (e.g. hyperbolic cross) or more general index sets with the full projection (31) has been proved in [4, 9, 10]. So we focus on proving the general result with the zero projection (30).

The proof is by induction on s. We will attempt to derive the condition (32) rather than assuming it from the beginning.

For $s = 1$, the condition $h_1 z_1 \not\equiv_n 0$ holds for all $z_1 \in \mathbb{Z}_n^*$ if $h_1 \not\equiv_n 0$, and fails for all z_1 if $h_1 \equiv_n 0$. To avoid the latter scenario we assume that $n > |h_1|$ always holds.

Suppose we already obtained the generating vector $\boldsymbol{z} \in (\mathbb{Z}_n^*)^{s-1}$ satisfying (33) for $\mathcal{A}_{s-1}$. For each $(\boldsymbol{h}, h_s) \in \mathcal{A}_s \backslash \{\mathbf{0}\}$, we will eliminate any 'bad' $z_s \in \mathbb{Z}_n^*$ that satisfies

$$(\boldsymbol{h}, h_s) \cdot (\boldsymbol{z}, z_s) \equiv_n \boldsymbol{h} \cdot \boldsymbol{z} + h_s z_s \equiv_n 0 \quad \Leftrightarrow \quad h_s z_s \equiv_n -\boldsymbol{h} \cdot \boldsymbol{z}. \tag{34}$$

We stress that $(\boldsymbol{h}, h_s) \in \mathcal{A}_s$ does not imply $\boldsymbol{h} \in \mathcal{A}_{s-1}$ under the zero projection (30). Depending on the value of h_s we have the following scenarios:

1. If $h_s = 0$ then $\boldsymbol{h} \in \mathcal{A}_{s-1} \backslash \{\mathbf{0}\}$; in turn the induction hypotheses (33) for $\mathcal{A}_{s-1}$ guarantees that $\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0$ and so (34) has no solution for z_s. There are $\#(\mathcal{A}_{s-1} \backslash \{\mathbf{0}\})$ such cases.

2. If $h_s \neq 0$ but $h_s \equiv_n 0$, then (34) has no solution for z_s if $\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0$, or it holds for all z_s if $\boldsymbol{h} \cdot \boldsymbol{z} \equiv_n 0$. To avoid the latter scenario we assume that $n > |h_s|$ always holds.

3. If $h_s \not\equiv_n 0$ then, since n is prime, (34) has a unique solution for $z_s \in \mathbb{Z}_n^*$ if $\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0$, or has no solution for $z_s \in \mathbb{Z}_n^*$ if $\boldsymbol{h} \cdot \boldsymbol{z} \equiv_n 0$. The latter scenario includes $\boldsymbol{h} = \mathbf{0}$ so there are at least $\#\mathcal{A}_s^\dagger$ such cases, where $\mathcal{A}_s^\dagger := \{h_s \in \mathbb{Z} : h_s \not\equiv_n 0 \text{ and } (\mathbf{0}, h_s) \in \mathcal{A}_s\}$.

Thus, provided that $n > |h_s|$, there is at most one bad z_s to be eliminated for each $(\boldsymbol{h}, h_s) \in \mathcal{A}_s \backslash \{\mathbf{0}\}$, and the total number of bad z_s we eliminate is at most $\#(\mathcal{A}_s \backslash \{\mathbf{0}\}) - \#(\mathcal{A}_{s-1} \backslash \{\mathbf{0}\}) - \#\mathcal{A}_s^\dagger$.

Hence, provided additionally that $\#\mathbb{Z}_n^* = n - 1 > \#\mathcal{A}_s - \#\mathcal{A}_{s-1} - \#\mathcal{A}_s^\dagger$, there is always a 'good' z_s remaining such that $(\boldsymbol{z}, z_s)$ will satisfy (33) for $\mathcal{A}_s$. Moreover if $\mathcal{A}$ is centrally symmetric, then all $\mathcal{A}_s$ are centrally symmetric, and both $(\boldsymbol{h}', h_s)$ and $(-\boldsymbol{h}, -h_s)$ will eliminate the same z_s if a solution for (34) exists. It then suffices to demand that $n - 1 > (\#\mathcal{A}_s - \#\mathcal{A}_{s-1} - \#\mathcal{A}_s^\dagger)/2$. By induction, to ensure that a good $\boldsymbol{z}^* \in (\mathbb{Z}_n^*)^d$ exists, it suffices to assume that

$$n > \max\Big\{ \max_{s=2,\ldots,d} \frac{\#\mathcal{A}_s - \#\mathcal{A}_{s-1} - \#\mathcal{A}_s^\dagger}{\kappa} + 1 \,,\ \max_{s=1,\ldots,d} \max_{\boldsymbol{h} \in \mathcal{A}_s} |h_s| \Big\},$$

with $\kappa = 2$ if $\mathcal{A}$ is centrally symmetric and $\kappa = 1$ otherwise. This leads to the simplified condition on n in the theorem.

Now for completeness we discuss briefly the case for the full projection (31). The proof is almost identical to the case for the zero projection but is slightly simpler. The subtle difference is that for each $(\boldsymbol{h}, h_s) \in \mathcal{A}_s$ we now have $\boldsymbol{h} \in \mathcal{A}_{s-1}$ regardless of the value of h_s, and the induction hypothesis (33) for $\mathcal{A}_{s-1}$ guarantees that $\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0$ if $h_s \neq 0$, thus simplifying the second and third scenarios above. □

5.2 Algorithmic aspects of the CBC construction

Remark 24. *For the projections $\mathcal{A}_s$ defined by either* (30) *or* (31)*, Theorem 23 and its proof justify two different approaches to carry out the component-by-component construction:*

- *Brute force approach: At step s, we search through $z_s \in \mathbb{Z}_n^*$ until we find one that satisfies* (33) *for all $\boldsymbol{h} \in \mathcal{A}_s \backslash \{\mathbf{0}\}$. The cost is $\mathcal{O}(n_{\text{fail}} \#\mathcal{A}_s)$, where n_{fail} is the number of different z_s that was checked. So the cost is at worst $\mathcal{O}(n \#\mathcal{A}_s)$, leading to a total cost of $\mathcal{O}(d\, n \#\mathcal{A})$.*

- *Elimination approach: At step s, we loop through every $\boldsymbol{h} \in \mathcal{A}_s \backslash \{\mathbf{0}\}$ and eliminate the corresponding $z_s \in \mathbb{Z}_n^*$ that fails* (33)*, if any. Then we take any remaining z_s. The cost is only $\mathcal{O}(\#\mathcal{A}_s)$, leading to a total cost of $\mathcal{O}(d \#\mathcal{A})$.*

In both approaches we have left out the $\mathcal{O}(s)$ factor in step s that arises from the evaluation of dot products; this is valid if we store and update the values of $\boldsymbol{h} \cdot \boldsymbol{z}$ for all $\boldsymbol{h}$ in each step. In the elimination approach we mark the bad choices of z_s in an array of length $n-1$ with pointers linking the previous and next good choices of z_s, so that it is $\mathcal{O}(1)$ cost to obtain a good z_s at the end.

The two approaches may be used for different steps in the algorithm if it is advantageous to mix them. Both approaches are guaranteed to succeed provided n is sufficiently large, see (32)*. We can run the algorithm with smaller values of n (or even composite values of n in the brute force approach), and be prepared to increase n when the algorithm fails. Once a $\boldsymbol{z}$ is found, we can systematically test and reduce the value of n by verifying whether* (33) *still holds for $\mathcal{A}$.*

In general the zero projections (30) *are subsets of the full projections* (31)*, and consequently the condition* (33) *is weaker and faster to check for* (30) *than for* (31)*. There are also algorithmic advantages in the data structure for iterating the sets based on* (30)*, namely, that the indices can be ordered according to the number of zeros at the end.*

We now apply Theorem 23 to the situation where the input set $\mathcal{A}$ is a difference set, i.e., $\mathcal{A} = \Lambda \ominus \Lambda$. Then the condition (33) is now explicitly given by

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all} \quad \boldsymbol{h} \in (\Lambda \ominus \Lambda)_s \setminus \{\boldsymbol{0}\}, \tag{35}$$

where $(\Lambda \ominus \Lambda)_s$ denote the projection of the difference set $\Lambda \ominus \Lambda$ defined according to (30) or (31). Since $\#(\Lambda \ominus \Lambda) \le (\#\Lambda)^2$, Remark 24 indicates that the cost of CBC construction for $\mathcal{A} = \Lambda \ominus \Lambda$ is $\mathcal{O}(d\,n\,(\#\Lambda)^2)$ or $\mathcal{O}(d\,(\#\Lambda)^2)$ for the two approaches, respectively.

Similarly, for all sign changes on an index set Λ we have $\#\mathcal{M}(\Lambda) \le 2^d\,\#\Lambda$ and therefore the cost of CBC construction for $\mathcal{A} = \mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)$ is $\mathcal{O}(d\,n\,2^{2d}\,(\#\Lambda)^2)$ or $\mathcal{O}(d\,2^{2d}\,(\#\Lambda)^2)$; the cost for $\mathcal{A} = \Lambda \oplus \mathcal{M}(\Lambda)$ is $\mathcal{O}(d\,n\,2^d\,(\#\Lambda)^2)$ or $\mathcal{O}(d\,2^d\,(\#\Lambda)^2)$.

Hence, we can apply Theorem 23 and Remark 24 to the Fourier space in Lemmas 4 and 5, noting that the difference set $\Lambda \ominus \Lambda$ is always centrally symmetric and contains the zero vector. Analogously, we can apply Theorem 23 and Remark 24 to the cosine space in Lemmas 10 and 11, as well as Lemma 15 – plan B, and correspondingly, to the Chebyshev space in Lemmas 18, 19, and 20.

However, plan C for the cosine space and Chebyshev space, see Lemmas 16 and 21, respectively, cannot be formulated in the same generic form (29). So we will need to develop a separate justification for it. We will return to this later in Subsection 5.5.

5.3 Smart lookup for the brute force approach with full projection

When $\mathcal{A}_s$ are full projections (31), we have the important property that the projection of the difference set equals the difference set of the projections, i.e.,

$$(\Lambda \ominus \Lambda)_s = \Lambda_s \ominus \Lambda_s \qquad \text{for the full projection (31).} \tag{36}$$

Thus the condition (35) becomes

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n 0 \qquad \text{for all} \quad \boldsymbol{h} \in (\Lambda_s \ominus \Lambda_s) \setminus \{\boldsymbol{0}\},$$

which is equivalent to

$$\boldsymbol{h} \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{h}' \cdot \boldsymbol{z} \qquad \text{for all} \quad \boldsymbol{h}, \boldsymbol{h}' \in \Lambda_s \text{ with } \boldsymbol{h} \ne \boldsymbol{h}'. \tag{37}$$

In other words, every dot product needs to have a unique value.

The following code snippet shows that it is possible to verify condition (37) for a given $\boldsymbol{z}$ with cost $\mathcal{O}(\#\Lambda_s)$ rather than $\mathcal{O}((\#\Lambda_s)^2)$, by marking a bit string of length n for the values of dot product modulo n that have occurred.

```
// Fourier space
// INPUT: z, n and Λ_s
// VERIFY: h·z ≢_n h'·z for all h,h' ∈ Λ_s, h ≠ h'
// COST: O(#Λ_s)
```

```
S = 0
for h ∈ Λ_s:
    α = h · z mod n
    if S[α] = 1: return FALSE
    S[α] = 1
return TRUE
```

Consequently, we can reduce the cost of the brute force CBC construction for $\mathcal{A} = \Lambda \ominus \Lambda$ from $\mathcal{O}(d\,n\,(\#\Lambda)^2)$ to $\mathcal{O}(d\,n\,\#\Lambda)$. We shall refer to this as the "smart lookup" trick.

We stress once again that (36) only holds when we have the full projection (31). Under the zero projection (30) we would have in general $(\Lambda \ominus \Lambda)_s \supseteq \Lambda_s \ominus \Lambda_s$; in this case the alternative formulation (37) would miss out on some indices.

Similar reduction in cost can be achieved for $\mathcal{A} = \mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)$ as we show in the code snippet below.

```
// Cosine space and Chebyshev space -- plan A
// INPUT: z, n and Λ_s
// VERIFY: σ'(k') · z ≢_n σ(k) · z for all k, k' ∈ Λ_s, σ ∈ S_k, σ' ∈ S_k', σ'(k') ≠ σ(k)
// COST: O(#M(Λ_s))

S = 0
for k ∈ Λ_s:
    for h ∈ {σ(k) : σ ∈ S_k}:
        α = h · z mod n
        if S[α] = 1: return FALSE
        S[α] = 1
return TRUE
```

We can save on half of the calculations, since if $\boldsymbol{k} \neq \boldsymbol{0}$ we can fix one of the signs for a non-zero element of $\boldsymbol{k}$ to get half of the sign changes and multiply by -1 to get the other half as shown below.

```
// Cosine space and Chebyshev space -- plan A -- halved
// INPUT: z, n and Λ_s
// VERIFY: σ'(k') · z ≢_n σ(k) · z for all k, k' ∈ Λ_s, σ ∈ S_k, σ' ∈ S_k', σ'(k') ≠ σ(k)
// COST: O(#M(Λ_s)/2)

S = 0
if 0 ∈ Λ_s: S[0] = 1
for k ∈ Λ_s \ {0}:
    i = min{j : k_j ≠ 0}
    for h ∈ {σ(k) : σ ∈ S_k, σ_i = +1}:
        α = h · z mod n // note: n − α = −h · z mod n
        if S[α] = 1: return FALSE
        S[α] = 1
        if S[n − α] = 1: return FALSE // note: it could happen that n − α = α
        S[n − α] = 1
return TRUE
```

We can achieve a similar reduction in cost for $\mathcal{A} = \Lambda \oplus \mathcal{M}(\Lambda)$, but this is more complicated because we need to distinguish between the dot products coming from the original indices and the dot products coming from sign changes of the indices. We do this by keeping two bit strings of length n as shown in the code snippet below: S_1 marks the original dot products, while S_2 marks the dot products from all sign changes thus including S_1. (We can also half the cost as above but we do not include that here.)

```
// Cosine space and Chebyshev space -- plan B
// INPUT: z, n and Λ_s
```

```
// VERIFY: σ(k')·z ≢_n k·z for all k,k' ∈ Λ_s, σ ∈ S_{k'}, σ(k') ≠ k
// COST: O(#M(Λ_s))

S_1 = 0
S_2 = 0
for k ∈ Λ_s:
    α = k·z mod n
    if S_2[α] = 1: return FALSE // note: the value of S_1[α] is also checked since S_1 ⊆ S_2
    S_2[α] = 1
    S_1[α] = 1
    for h ∈ {σ(k) : σ ∈ S_k} with h ≠ k:
        α' = h·z mod n
        if S_1[α'] = 1: return FALSE
        S_2[α'] = 1 // note: it does not matter if S_2[α'] is already set
return TRUE
```

The previous algorithm can now be modified to allow self-aliasing and to keep track of the constant $c_{\boldsymbol{k}}$, see Lemma 16 for cosine space and Lemma 21 for Chebyshev space.

```
// Cosine space and Chebyshev space -- plan C
// INPUT: z, n and Λ_s
// VERIFY: σ(k')·z ≢_n k·z for all k,k' ∈ Λ_s, σ ∈ S_{k'}, k' ≠ k
// COST: O(#M(Λ_s))

S_1 = 0
S_2 = 0
for k ∈ Λ_s:
    α = k·z mod n
    if S_2[α] = 1: return FALSE
    S_2[α] = 1
    c_k = 1
    for h ∈ {σ(k) : σ ∈ S_k} with h ≠ k:
        α' = h·z mod n
        if α' = α: c_k += 1
        if S_1[α'] = 1: return FALSE
        S_2[α'] = 1
    S_1[α] = 1
return TRUE with {c_k : k ∈ Λ_s}
```

The crucial difference between the last two code snippets is that in plan B the bit $S_1[\alpha]$ is marked before the dot products α' from the sign changes are checked against S_1, thus not allowing $\alpha' = \alpha$ (no aliasing), while in plan C the bit $S_1[\alpha]$ is marked only after all α' have been checked against S_1, thus allowing $\alpha' = \alpha$ and indeed counts the number of times this occurs in $c_{\boldsymbol{k}}$ (self-aliasing).

We summarize this subsection in the following remark.

Remark 25. *The cost for brute force CBC in step s with full projection* (31) *and smart lookup is*

$$\mathcal{O}(n_{\text{fail}}\,\#\Lambda_s), \qquad \mathcal{O}(n_{\text{fail}}\,\#\mathcal{M}(\Lambda_s)), \qquad \mathcal{O}(n_{\text{fail}}\,\#\mathcal{M}(\Lambda_s)),$$

for the index sets $\mathcal{A} = \Lambda \ominus \Lambda$, $\mathcal{A} = \mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda)$ *and* $\mathcal{A} = \Lambda \oplus \mathcal{M}(\Lambda)$, *respectively.*

Similar strategies have been implemented in the sparseFFTr1l software library of Toni Volkmer [32].

5.4 Mixed CBC construction

Combining Remark 25 with Remark 24, we see that there is advantage in mixing the two different approaches. As long as n_{fail} remains small it is advantageous to follow the brute force approach

with full projection (31) and smart lookup. We anticipate this to be the case for the initial dimensions.

Starting from $z_1 = 1$, at step s we begin our brute force search with the value $z_s = z_{s-1} + 1$. If this z_s fails then we increment again by 1 and do this repeatedly (if $n-1$ is reached then we continue from 1) until a valid z_s is found, while keeping a count on n_{fail}. Then gradually as the dimension increases and as we run out of choices, we expect the value of n_{fail} to increase until at some point the balance tips over the other way and it becomes cheaper to follow the elimination approach. From then on we switch over to the elimination approach in the generic formulation (29) with the zero projection (30) so that the sets are smaller (except for the case of plan C which we discuss in the next subsection).

We summarize our results for the different spaces in Table 1.

Table 1: Summary of CBC algorithms for function reconstruction

	Fourier space	Cosine space and Chebyshev space		
		Plan A	Plan B	Plan C
(a)	$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0$ for all $\boldsymbol{h} \in (\Lambda \ominus \Lambda)_s \setminus \{\boldsymbol{0}\}$	$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0$ for all $\boldsymbol{h} \in (\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda))_s \setminus \{\boldsymbol{0}\}$	$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n 0$ for all $\boldsymbol{h} \in (\Lambda \oplus \mathcal{M}(\Lambda))_s \setminus \{\boldsymbol{0}\}$	NA
(b)	$\boldsymbol{h}\cdot\boldsymbol{z} \not\equiv_n \boldsymbol{h}'\cdot\boldsymbol{z}$ for all $\boldsymbol{h},\boldsymbol{h}' \in \Lambda_s$, $\boldsymbol{h} \neq \boldsymbol{h}'$	$\boldsymbol{\sigma}'(\boldsymbol{k}')\cdot\boldsymbol{z} \not\equiv_n \boldsymbol{\sigma}(\boldsymbol{k})\cdot\boldsymbol{z}$ for all $\boldsymbol{k},\boldsymbol{k}' \in \Lambda_s$, $\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}}$, $\boldsymbol{\sigma}' \in \mathcal{S}_{\boldsymbol{k}'}$, $\boldsymbol{\sigma}'(\boldsymbol{k}') \neq \boldsymbol{\sigma}(\boldsymbol{k})$	$\boldsymbol{\sigma}(\boldsymbol{k}')\cdot\boldsymbol{z} \not\equiv_n \boldsymbol{k}\cdot\boldsymbol{z}$ for all $\boldsymbol{k},\boldsymbol{k}' \in \Lambda_s$, $\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'}$, $\boldsymbol{\sigma}(\boldsymbol{k}') \neq \boldsymbol{k}$	$\boldsymbol{\sigma}(\boldsymbol{k}')\cdot\boldsymbol{z} \not\equiv_n \boldsymbol{k}\cdot\boldsymbol{z}$ for all $\boldsymbol{k},\boldsymbol{k}' \in \Lambda_s$, $\boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'}$, $\boldsymbol{k} \neq \boldsymbol{k}'$
(c)	$n \sim \#(\Lambda \ominus \Lambda)_s$	$n \sim \#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda))_s$	$n \sim \#(\Lambda \oplus \mathcal{M}(\Lambda))_s$	$n \sim \#\Lambda_s \, \#\mathcal{M}(\Lambda_s)$
(d)	$\#(\Lambda \ominus \Lambda)_s$	$\#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda))_s$	$\#(\Lambda \oplus \mathcal{M}(\Lambda))_s$	NA
(e)	$n_{\text{fail}} \, \#(\Lambda \ominus \Lambda)_s$	$n_{\text{fail}} \, \#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda))_s$	$n_{\text{fail}} \, \#(\Lambda \oplus \mathcal{M}(\Lambda))_s$	NA
(f)	$(\#\Lambda_s)^2$	$(\#\mathcal{M}(\Lambda_s))^2$	$\#\Lambda_s \, \#\mathcal{M}(\Lambda_s)$	$\#\Lambda_s \, \#\mathcal{M}(\Lambda_s)$
(g)	$n_{\text{fail}} \, (\#\Lambda_s)^2$	$n_{\text{fail}} \, (\#\mathcal{M}(\Lambda_s))^2$	$n_{\text{fail}} \, \#\Lambda_s \, \#\mathcal{M}(\Lambda_s)$	
(h)	$n_{\text{fail}} \, \#\Lambda_s$	$n_{\text{fail}} \, \#\mathcal{M}(\Lambda_s)$	$n_{\text{fail}} \, \#\mathcal{M}(\Lambda_s)$	
(i)	$n_{\text{fail}} \sim \#\Lambda_s$	$n_{\text{fail}} \sim \#\mathcal{M}(\Lambda_s)$	$n_{\text{fail}} \sim \#\mathcal{M}(\Lambda_s)$	
(j)	$d \, (\#\Lambda)^2$	$d \, (\#\mathcal{M}(\Lambda))^2$	$d \, \#\Lambda \, \#\mathcal{M}(\Lambda)$	

(a) Standard formulation of the reconstruction condition at step s with full/zero projection
(b) Equivalent formulation of the reconstruction condition at step s with full projection
(c) Required size of n to guarantee success at step s (also need n to cover spread of index set)
(d) Cost of elimination approach at step s based on (a) with full/zero projection
(e) Cost of brute force approach at step s based on (a) with full/zero projection
(f) Cost of elimination approach at step s based on (b) with full projection
(g) Cost of brute force approach at step s based on (b) with full projection
(h) Cost of brute force approach at step s based on (b) with full projection and smart lookup
(i) Switching point on n_{fail} from brute force (h) to elimination (d)/(f)
(j) Total cost of mixed CBC: brute force until n_{fail} reaches switching point then elimination

5.5 A new CBC proof for plan C

Recall that the condition (25) is weaker than the condition (19), which is in turn equivalent to (20). Thus when we have the full projection (31), the condition on n in Theorem 23 guarantees the existence of z_s with the required property in step s. However, to prove that the CBC construction can find this vector, we need a new CBC proof.

Theorem 26. *Let $\Lambda \subset \mathbb{N}_0^d$ be an arbitrary index set, and let n be a prime number satisfying*

$$n > \max\Big\{\#\Lambda\,\#\mathcal{M}(\Lambda)\ ,\ 2\max(\Lambda)\Big\}. \tag{38}$$

Define Λ_s to be the full projection of Λ as in (31). Then a generating vector $\boldsymbol{z}^* = (z_1, \ldots, z_d) \in (\mathbb{Z}_n^*)^d$ can be constructed component-by-component such that for all $s = 1, \ldots, d$ and $\boldsymbol{z} = (z_1, \ldots, z_s)$ we have

$$\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z} \qquad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in \Lambda_s \text{ and } \boldsymbol{\sigma} \in \mathcal{S}_{\boldsymbol{k}'} \text{ with } \boldsymbol{k} \neq \boldsymbol{k}'. \tag{39}$$

Proof. The proof is by induction on s. For $s = 1$ and $k_1 \neq k_1'$, the condition $\sigma_1(k_1')z_1 \not\equiv_n k_1 z_1$ holds for all $z_1 \in \mathbb{Z}_n^*$ if $\sigma_1(k_1') - k_1 \not\equiv_n 0$, and fails for all z_1 if $\sigma_1(k_1') - k_1 \equiv_n 0$. To avoid the latter scenario we assume that $n > 2\max_{k_1 \in \Lambda_1} |k_1|$.

Suppose we already obtained the generating vector $\boldsymbol{z} \in (\mathbb{Z}_n^*)^{s-1}$ satisfying (39) for Λ_{s-1}. For each distinct pair $(\boldsymbol{k}, k_s), (\boldsymbol{k}', k_s') \in \Lambda_s$ and each $(\boldsymbol{\sigma}, \sigma_s) \in \mathcal{S}_{(\boldsymbol{k}', k_s')}$, we will eliminate any 'bad' $z_s \in \mathbb{Z}_n^*$ that satisfies

$$(\boldsymbol{\sigma}(\boldsymbol{k}'), \sigma_s(k_s')) \cdot (\boldsymbol{z}, z_s) \equiv_n (\boldsymbol{k}, k_s) \cdot (\boldsymbol{z}, z_s) \quad \Leftrightarrow \quad (\sigma_s(k_s') - k_s) z_s \equiv_n -(\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} - \boldsymbol{k} \cdot \boldsymbol{z}). \tag{40}$$

From the definition (31) we have $\boldsymbol{k}, \boldsymbol{k}' \in \Lambda_{s-1}$. We have the following scenarios:

1. If $\boldsymbol{k} \neq \boldsymbol{k}'$ then the induction hypotheses (39) for Λ_{s-1} guarantees that $\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$ for all $\boldsymbol{\sigma}$. Thus (40) has no solution for z_s if $\sigma_s(k_s') - k_s \equiv_n 0$, and (40) has a unique solution for z_s if $\sigma_s(k_s') - k_s \not\equiv_n 0$.

2. If $\boldsymbol{k} = \boldsymbol{k}'$ (thus $k_s \neq k_s'$) and $\boldsymbol{\sigma}$ satisfies $\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \not\equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$ then, as in the previous scenario, (40) has no solution for z_s if $\sigma_s(k_s') - k_s \equiv_n 0$, and (40) has a unique solution for z_s if $\sigma_s(k_s') - k_s \not\equiv_n 0$.

3. If $\boldsymbol{k} = \boldsymbol{k}'$ (thus $k_s \neq k_s'$) and $\boldsymbol{\sigma}$ satisfies $\boldsymbol{\sigma}(\boldsymbol{k}') \cdot \boldsymbol{z} \equiv_n \boldsymbol{k} \cdot \boldsymbol{z}$, then (40) has no solution for z_s if $\sigma_s(k_s') - k_s \not\equiv_n 0$, and (40) holds for all z_s if $\sigma_s(k_s') - k_s \equiv_n 0$. To avoid the latter scenario we assume that $n > |\sigma_s(k_s') - k_s|$. Note that it is not possible to have $\sigma_s(k_s') = k_s$ when $k_s \neq k_s'$ since both k_s and k_s' are nonnegative integers.

Thus, provided that $n > 2\max_{(\boldsymbol{k}, k_s) \in \Lambda_s} |k_s|$, there is at most one bad z_s to be eliminated for each distinct pair $(\boldsymbol{k}, k_s), (\boldsymbol{k}', k_s') \in \Lambda_s$ and each $(\boldsymbol{\sigma}, \sigma_s) \in \mathcal{S}_{(\boldsymbol{k}', k_s')}$, so the total number of bad z_s we eliminate is at most $\#\Lambda_s\,(\#\mathcal{M}(\Lambda_s) - 1)$.

Hence, provided additionally that $\#\mathbb{Z}_n^* = n - 1 > \#\Lambda_s\,(\#\mathcal{M}(\Lambda_s) - 1)$, there is always a 'good' z_s remaining such that $(\boldsymbol{z}, z_s)$ will satisfy (39) for Λ_s. By induction, to ensure that a good $\boldsymbol{z}^* \in (\mathbb{Z}_n^*)^d$ exists, it suffices to assume that n satisfies (38). This completes the proof. □

6 Approximation

We now discuss function approximation for all three settings under a unified framework. A major difference of this section compared to the previous sections is that the function f under consideration is no longer supported only on a finite index set. We cannot achieve exact function reconstruction and therefore an error analysis is needed.

6.1 Function approximation under a unified framework

We have an orthonormal basis $\{\alpha_{\boldsymbol{k}}\}$ for $L_\mu^2(\Omega)$, where $\mu(\Omega) = 1$, and consider functions with absolutely converging series expansions

$$f = \sum_{\boldsymbol{k}} \widehat{f}_{\boldsymbol{k}}\, \alpha_{\boldsymbol{k}},$$

where the sum is over $\mathbb{Z}^d$ (for Fourier space) or $\mathbb{N}_0^d$ (for cosine and Chebyshev spaces). Now consider a subset Λ of the indices and represent the exact L^2_μ projection of f, i.e., the best L^2_μ approximation on Λ, by

$$f_\Lambda \;=\; \sum_{\boldsymbol{k}\in\Lambda} \widehat{f}_{\boldsymbol{k}}\, \alpha_{\boldsymbol{k}}.$$

We cannot calculate these coefficients $\widehat{f}_{\boldsymbol{k}}$ exactly and will have to approximate them, leading to

$$f^{\square}_\Lambda \;=\; \sum_{\boldsymbol{k}\in\Lambda} \widehat{f}^{\square}_{\boldsymbol{k}}\, \alpha_{\boldsymbol{k}},$$

with $\square \in \{a, b, c\}$ denoting the approximation by plan A (including the Fourier case), plan B, or plan C.

We write

$$\varphi = \begin{cases} \mathrm{id} & \text{for Fourier space,} \\ \varphi_{\mathrm{tent}} & \text{for cosine space plan A, B, C,} \\ \cos(\pi\, \varphi_{\mathrm{tent}}(\bullet)) & \text{for Chebyshev space plan A, B, C,} \end{cases}$$

$$\alpha_{\boldsymbol{k}} \circ \varphi = \mathrm{u}_{\boldsymbol{k}} = \begin{cases} \exp(2\pi \mathrm{i}\, \boldsymbol{h}\cdot\boldsymbol{x}) & \text{for Fourier space,} \\ \sqrt{2}^{|\boldsymbol{k}|_0} \prod_{j=1}^d \cos(2\pi\, k_j x_j) & \text{for cosine/Chebyshev space plan A, B, C,} \end{cases}$$

$$\mathrm{v}_{\boldsymbol{k}} = \begin{cases} \mathrm{u}_{\boldsymbol{k}} & \text{for Fourier space, cosine/Chebyshev space plan A,} \\ \sqrt{2}^{|\boldsymbol{k}|_0} \cos(2\pi\, \boldsymbol{k}\cdot\boldsymbol{x}) & \text{for cosine/Chebyshev space plan B, C.} \end{cases}$$

Then we have $\langle \mathrm{u}_{\boldsymbol{k}}, \mathrm{u}_{\boldsymbol{k}'}\rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$, $\langle \mathrm{u}_{\boldsymbol{k}}, \mathrm{v}_{\boldsymbol{k}'}\rangle = \delta_{\boldsymbol{k},\boldsymbol{k}'}$, and $\langle \mathrm{v}_{\boldsymbol{k}}, \mathrm{v}_{\boldsymbol{k}'}\rangle = d_{\boldsymbol{k}}\, \delta_{\boldsymbol{k},\boldsymbol{k}'}$, with $d_{\boldsymbol{k}} = 2^{|\boldsymbol{k}|_0 - 1}$ for $\boldsymbol{k} \neq \boldsymbol{0}$ in the case of cosine or Chebyshev space plan B or C and $d_{\boldsymbol{k}} = 1$ otherwise.

We demand from our lattice rule that

$$Q_n(\mathrm{u}_{\boldsymbol{k}}\, \overline{\mathrm{v}_{\boldsymbol{k}'}}) = c_{\boldsymbol{k}}\, \delta_{\boldsymbol{k},\boldsymbol{k}'} \qquad \forall \boldsymbol{k}, \boldsymbol{k}' \in \Lambda, \tag{41}$$

$$Q_n(\mathrm{v}_{\boldsymbol{k}}\, \overline{\mathrm{v}_{\boldsymbol{k}'}}) = d_{\boldsymbol{k}}\, \delta_{\boldsymbol{k},\boldsymbol{k}'} \qquad \forall \boldsymbol{k}, \boldsymbol{k}' \in \Lambda, \tag{42}$$

where $c_{\boldsymbol{k}}$ in the case of cosine or Chebyshev space plan C (see (24) or (28)) can be a positive integer up to the number of unique sign changes of $\boldsymbol{k}$, i.e., $1 \le c_{\boldsymbol{k}} \le 2^{|\boldsymbol{k}|_0}$, and $c_{\boldsymbol{k}} = 1$ otherwise. Note that we do not necessarily have $Q_n(\mathrm{u}_{\boldsymbol{k}}\, \overline{\mathrm{u}_{\boldsymbol{k}'}}) = \delta_{\boldsymbol{k},\boldsymbol{k}'}$ (except for when $\mathrm{v}_{\boldsymbol{k}} = \mathrm{u}_{\boldsymbol{k}}$).

With the above unifying notation, and with $w_i = 1/n$ and $\boldsymbol{t}_i$ our lattice points, we can write our approximate coefficient as

$$\widehat{f}^{\square}_{\boldsymbol{k}} \;=\; Q_n(f\circ\varphi\, \overline{\mathrm{v}_{\boldsymbol{k}}})\, c_{\boldsymbol{k}}^{-1} \;=\; \sum_{i=0}^{n-1} w_i\, f(\varphi(\boldsymbol{t}_i))\, \overline{\mathrm{v}_{\boldsymbol{k}}(\boldsymbol{t}_i)}\, c_{\boldsymbol{k}}^{-1}, \qquad \boldsymbol{k}\in\Lambda\,. \tag{43}$$

In comparison, the exact coefficient is given by $\widehat{f}_{\boldsymbol{k}} = \langle f, \alpha_{\boldsymbol{k}}\rangle_\mu = \langle f\circ\varphi, \mathrm{u}_{\boldsymbol{k}}\rangle = \langle f\circ\varphi, \mathrm{v}_{\boldsymbol{k}}\rangle$.

6.2 Connection to discrete least squares

With a prescribed ordering of the elements in Λ, the approximate coefficients (43) for $\boldsymbol{k}\in\Lambda$ can be written in matrix-vector notation as

$$\widehat{\boldsymbol{f}}^{\square} \;=\; C^{-1}\, V^*\, W\, \boldsymbol{f}_\varphi,$$

with column vectors $\boldsymbol{f}_\varphi = [f(\varphi(\boldsymbol{t}_i))]_i$, $\widehat{\boldsymbol{f}}^{\square} = [\widehat{f}^{\square}_{\boldsymbol{k}}]_{\boldsymbol{k}}$, and matrices $V = [\mathrm{v}_{\boldsymbol{k}}(\boldsymbol{t}_i)]_{i,\boldsymbol{k}}$, $C = \mathrm{diag}(c_{\boldsymbol{k}})$, and $W = \mathrm{diag}(w_i)$. The conditions (41) and (42) can be expressed as

$$V^*\, W\, U = C \qquad \text{and} \qquad V^*\, W\, V = D,$$

with matrices $U = [\mathrm{u}_{\boldsymbol{k}}(\boldsymbol{t}_i)]_{i,\boldsymbol{k}}$ and $D = \mathrm{diag}(d_{\boldsymbol{k}})$.

For plan A (or for the Fourier case) we have $U = V$ and $C = D = I$, so

$$\widehat{\boldsymbol{f}}^a = U^* W \boldsymbol{f}_\varphi,$$

which is precisely the solution to the normal equations

$$(W^{1/2} U)^* W^{1/2} U \widehat{\boldsymbol{f}}^a = (W^{1/2} U)^* W^{1/2} \boldsymbol{f}_\varphi \Leftrightarrow (U^* W U) \widehat{\boldsymbol{f}}^a = U^* W \boldsymbol{f}_\varphi \Leftrightarrow \widehat{\boldsymbol{f}}^a = U^* W \boldsymbol{f}_\varphi,$$

which in turn solves the discrete least-squares problem

$$\min_{\widehat{\boldsymbol{f}}^a} \|W^{1/2} U \widehat{\boldsymbol{f}}^a - W^{1/2} \boldsymbol{f}_\varphi\|_2^2.$$

Plan B and plan C do *not* have the least-squares interpretation.

6.3 Stability to perturbation

Suppose that there is perturbation error in the function evaluations in (43) so that instead of $f(\varphi(\boldsymbol{t}_i))$ we have

$$f_{\mathrm{pert}}(\varphi(\boldsymbol{t}_i)) \;=\; f(\varphi(\boldsymbol{t}_i)) + \varepsilon_i, \qquad i = 0, \dots, n-1.$$

We denote the corresponding perturbed approximate coefficients by $\widehat{f}^{\square}_{\boldsymbol{k},\mathrm{pert}}$ and the corresponding approximate function over Λ by $f^{\square}_{\Lambda,\mathrm{pert}}$ for $\square \in \{a, b, c\}$. Using (43), we can write

$$\begin{aligned}
\|f^{\square}_{\Lambda,\mathrm{pert}} - f^{\square}_{\Lambda}\|^2_{L^2_\mu} \;&=\; \sum_{\boldsymbol{k}\in\Lambda} |\widehat{f}^{\square}_{\boldsymbol{k},\mathrm{pert}} - \widehat{f}^{\square}_{\boldsymbol{k}}|^2 \\
&= \sum_{\boldsymbol{k}\in\Lambda} \left| \sum_{i=0}^{n-1} w_i \, (f_{\mathrm{pert}} - f)(\varphi(\boldsymbol{t}_i)) \, \overline{\mathrm{v}_{\boldsymbol{k}}(\boldsymbol{t}_i)} \, c_{\boldsymbol{k}}^{-1} \right|^2 \;=\; \|C^{-1} V^* W \boldsymbol{\varepsilon}\|_2^2,
\end{aligned}$$

with column vector $\boldsymbol{\varepsilon} = [\varepsilon_i]_i$. We have

$$\|C^{-1} V^* W \boldsymbol{\varepsilon}\|_2^2 \;=\; \|C^{-1} \, D^{1/2} \, (D^{-1/2} \, V^* \, W^{1/2}) \, W^{1/2} \, \boldsymbol{\varepsilon}\|_2^2,$$

where

$$(D^{-1/2} \, V^* \, W^{1/2}) \, (W^{1/2} \, V \, D^{-1/2}) \;=\; D^{-1/2} \, V^* \, W \, V \, D^{-1/2} = I,$$

so that $\|D^{-1/2} \, V^* \, W^{1/2}\|_2 = 1$. Thus

$$\|f^{\square}_{\Lambda,\mathrm{pert}} - f^{\square}_{\Lambda}\|^2_{L^2_\mu} \;=\; \|C^{-1} V^* W \boldsymbol{\varepsilon}\|_2^2 \;\le\; \|C^{-1} \, D^{1/2}\|_2^2 \, \|W^{1/2} \, \boldsymbol{\varepsilon}\|_2^2 \;=\; \underbrace{\left(\max_{\boldsymbol{k}\in\Lambda} \frac{d_{\boldsymbol{k}}}{c_{\boldsymbol{k}}^2} \right)}_{=:\, \rho^{\square}_{\Lambda}} \left(\frac{1}{n} \sum_{i=0}^{n-1} |\varepsilon_i|^2 \right).$$

Here $\rho^{\square}_{\Lambda}$ is the *stability constant*, and we have

$$\rho^a_\Lambda \;=\; 1, \quad \rho^b_\Lambda \;=\; \max\left(1_{\mathbf{0}\in\Lambda}, \max_{\boldsymbol{k}\in\Lambda\setminus\{\mathbf{0}\}} 2^{|\boldsymbol{k}|_0 - 1} \right), \quad \rho^c_\Lambda \;=\; \max\left(1_{\mathbf{0}\in\Lambda}, \max_{\boldsymbol{k}\in\Lambda\setminus\{\mathbf{0}\}} \frac{2^{|\boldsymbol{k}|_0 - 1}}{c_{\boldsymbol{k}}^2} \right), \qquad (44)$$

where $1_{\mathbf{0}\in\Lambda}$ is 1 if $\mathbf{0} \in \Lambda$ and is 0 otherwise.

For plan A the approximation is perfectly stable.

For plan B and plan C we have the general upper bound $\rho^c_\Lambda \le \rho^b_\Lambda \le 2^{d-1}$ which might be too pessimistic. If we have a weighted index set with decaying weights (see Example 3) ρ^b_Λ could be much smaller. Alternatively, if Λ is downward closed then by Lemma 2 we have $\rho^c_\Lambda \le \rho^b_\Lambda \le \#\Lambda$. In any case, the stability constant for plan B is likely to be much bigger than 1 even if it is independent of d.

For plan C the values of $c_{\boldsymbol{k}}$ depend on the lattice rule and can potentially be as large as $2^{|\boldsymbol{k}|_0}$, giving hope that one may attempt to minimize the stability constant ρ^c_Λ as part of the CBC construction of the lattice generating vector. Unfortunately, numerical experiments show that not much improvement can be obtained because "self-aliasing" does not happen often enough.

6.4 Error analysis

We have for our three plans A (including the Fourier space), B, and C, annotated by $\square \in \{a, b, c\}$,

$$\|f - f_\Lambda^\square\|_{L^2_\mu}^2 = \underbrace{\|f - f_\Lambda\|_{L^2_\mu}^2}_{\text{truncation error}} + \underbrace{\|f_\Lambda - f_\Lambda^\square\|_{L^2_\mu}^2}_{\text{approximation error}} .$$

The first part is the truncation error for the finite index set Λ, and thus represents the best L^2_μ approximation error for the choice of Λ. In the *Information Based Complexity* (IBC) error analysis, this would be a complexity result using *arbitrary linear information*: if we know more about the smoothness class of our functions, then this bound is known in terms of a set Λ which is constructed according to the decay of the singular values of the approximation operator and this error is exactly the next singular value, see, e.g., [33]. The second part is how well we approximate this best possible approximation by our numerical algorithm which only uses function values; in IBC this is known as *standard information*, see, e.g., [34].

We proceed to analyze the second error $\|f_\Lambda - f_\Lambda^\square\|_{L^2_\mu}^2$. Since f_Λ is supported only on Λ, our reconstruction lattice can exactly compute its coefficients on Λ. Thus for $\boldsymbol{k} \in \Lambda$ we have

$$\widehat{f_{\boldsymbol{k}}} = \widehat{(f_\Lambda)}_{\boldsymbol{k}} = Q_n(f_\Lambda \circ \varphi\, \overline{\mathrm{v}_{\boldsymbol{k}}})\, c_{\boldsymbol{k}}^{-1} = \sum_{i=0}^{n-1} w_i\, f_\Lambda(\varphi(\boldsymbol{t}_i))\, \overline{\mathrm{v}_{\boldsymbol{k}}(\boldsymbol{t}_i)}\, c_{\boldsymbol{k}}^{-1}. \tag{45}$$

Using (45) and (43) and following the same argument as for the stability analysis, we obtain

$$\|f_\Lambda - f_\Lambda^\square\|_{L^2_\mu}^2 = \sum_{\boldsymbol{k}\in\Lambda} |\widehat{f_{\boldsymbol{k}}} - \widehat{f_{\boldsymbol{k}}^\square}|^2 = \sum_{\boldsymbol{k}\in\Lambda} \left| \sum_{i=0}^{n-1} w_i\, (f_\Lambda - f)(\varphi(\boldsymbol{t}_i))\, \overline{v_{\boldsymbol{k}}(\boldsymbol{t}_i)}\, c_{\boldsymbol{k}}^{-1} \right|^2 = \|C^{-1} V^* W \boldsymbol{g}\|_2^2,$$

with column vector $\boldsymbol{g} = [(f_\Lambda - f)(\varphi(\boldsymbol{t}_i))]_i$, and we arrive at

$$\|f_\Lambda - f_\Lambda^\square\|_{L^2_\mu}^2 \le \rho_\Lambda^\square\, \|f - f_\Lambda\|_n^2, \qquad \text{where} \quad \|h\|_n^2 := \frac{1}{n} \sum_{i=0}^{n-1} |h(\varphi(\boldsymbol{t}_i))|^2. \tag{46}$$

We summarize our combined result for function approximation in the following theorem.

Theorem 27. *Consider the problem of approximating a function $f \in \mathcal{F}$ from the Fourier, cosine, or Chebyshev space by $f_\Lambda^\square$, $\square \in \{a, b, c\}$, using a finite index set Λ and an n-point rank-1 lattice under plans A, B, or C as described in this paper. For sufficiently large n we have*

$$\|f - f_\Lambda^\square\|_{L^2_\mu}^2 \le \|f - f_\Lambda\|_{L^2_\mu}^2 + \rho_\Lambda^\square\, \|f - f_\Lambda\|_n^2, \tag{47}$$

with stability constant $\rho_\Lambda^\square$ given in (44) *and discrete seminorm $\|\cdot\|_n$ defined in* (46)*. We have $\rho_\Lambda^a = 1$, and if Λ is downward closed then $\rho_\Lambda^c \le \rho_\Lambda^b \le \min(2^{d-1}, \#\Lambda)$. A loose upper bound is*

$$\|f - f_\Lambda^\square\|_{L^2_\mu} \le \sqrt{1 + \rho_\Lambda^\square}\, \|f - f_\Lambda\|_{L^\infty}. \tag{48}$$

The requirement on n to achieve (47) *is proportional to $\#(\Lambda \ominus \Lambda)$ for the Fourier case, while for the cosine/Chebyshev case it is $\#(\mathcal{M}(\Lambda) \oplus \mathcal{M}(\Lambda))$ with plan A, $\#(\Lambda \oplus \mathcal{M}(\Lambda))$ with plan B, and $\#\Lambda\, \#\mathcal{M}(\Lambda)$ with plan C.*

6.5 Comparison with previous results from the literature and conclusions

Approximation by discrete least-squares has been analysed for different measures μ and bases $\{\alpha_{\boldsymbol{k}}\}$ in several works. We mention for instance [2, 3, 21] when using evaluations at random points, and in [22, 25] when random points are replaced by deterministic point sets. A common denominator in all the aforementioned analyses is the equivalence of the norm $\|\cdot\|_{L^2_\mu}$ and a suitably defined discrete seminorm $\|\cdot\|_n$, on the finite-dimensional space $\mathcal{F}_\Lambda$. More precisely, there exists $\delta \in [0,1)$ such that, under appropriate conditions on n, $\#\Lambda$ and δ, it holds that

$$(1-\delta)\,\|f\|^2_{L^2_\mu} \le \|f\|^2_n \le (1+\delta)\,\|f\|^2_{L^2_\mu}, \qquad \text{for all} \quad f \in \mathcal{F}_\Lambda. \tag{49}$$

Under the same conditions between n, $\#\Lambda$ and δ that ensure (49), the discrete least-squares approximation $\Pi_n f$ of any $f \in \mathcal{F}$ satisfies

$$\|f - \Pi_n f\|_{L^2_\mu} \;\le\; \sqrt{1+\frac{1}{1-\delta}}\,\inf_{v\in\mathcal{F}_\Lambda}\|f-v\|_{L^\infty}, \tag{50}$$

see [21, Proposition 1] for a proof.

Our results for the Fourier space and for plan A of the cosine and Chebyshev spaces achieve exactly $\delta = 0$; see also (48) with $\rho^a_\Lambda = 1$. For the Fourier case we obtain essentially a scaling of $n \ge (\#\Lambda)^2$. For the cosine and Chebyshev spaces we obtain essentially $n \ge 4^d\,(\#\Lambda)^2$ in general, and $n \ge \min(4^d\,(\#\Lambda)^2, (\#\Lambda)^{2\ln 3/\ln 2})$ for downward closed index sets. However, if the mirrored index set is a weighted hyperbolic cross with sufficiently fast decaying weights (see Example 3), then we obtain essentially $n \ge c_\tau\,(\#\Lambda)^{2\tau}$ for $\tau > 1$ arbitrarily close to 1.

With the Chebyshev space and for any downward closed set Λ, these results improve on [25] where it is proven that (49) holds true if $n \ge 2^{2d+1}d^2(\#\Lambda)^2$.

Moreover, we mention that in the case of uniform measure μ and expansion on the Legendre basis, the results in [22] show a scaling of $n/(\ln n)^d$ as $(\#\Lambda)^4$ for general downward closed sets, and a scaling of $n/(\ln n)^d$ as $(\#\Lambda)^2$ when $\mathcal{F}_\Lambda$ is an anisotropic tensor product space.

Our results for plans B and C in the cosine and Chebyshev spaces do not have the discrete least-squares interpretation. All three plans give exact function reconstruction in $\mathcal{F}_\Lambda$, but for a general $f \in \mathcal{F}$ not finitely supported on Λ, there is a trade-off between the approximation error and the requirement on n (e.g., plan A requires n to be larger but also has the smallest constant ρ^a_Λ). Therefore it is not easy to directly compare them without further analysis.

To proceed further from the very general result in Theorem 27, one would need to make further assumptions on, for example, the smoothness properties of the function space, and the knowledge of a corresponding index set that has been chosen to take advantage of such properties. Starting from the loose upper bound (48), if we know that the best L^∞ approximation error satisfies $\|f - f_\Lambda\|_{L^\infty} \le c_q\,(\#\Lambda)^{-q}$ for some $q > 0$, see, e.g., [16], then we arrive at $\|f - f^\square_\Lambda\|_{L^2_\mu} \le \sqrt{1+\rho^\square_\Lambda}\;c_q\,(\#\Lambda)^{-q}$. For the Fourier space we have $\rho^a_\Lambda = 1$ and n needs to be proportional to $\#(\Lambda\ominus\Lambda) \le (\#\Lambda)^2$, leading to $\|f - f^a_\Lambda\|_{L^2_\mu} = \mathcal{O}(n^{-q/2})$, where the implied constant is independent of d if c_q is independent of d. For the cosine or Chebyshev space, the result is more complicated because it depends on the size of the mirrored index set. If we have a weighted hyperbolic cross with sufficiently decaying weights (see Example 3) then the mirrored set itself is not of concern. However, for plan B or C we need to further take into account the value of ρ^b_Λ or ρ^c_Λ. In general ρ^b_Λ and ρ^c_Λ can be much worse than $\rho^a_\Lambda = 1$, but depending on the actual index set they might also be manageable.

Finally we stress that the L^2 approximation result based on *the estimate* (48) *is not sharp, and neither is* (50), because the best L^2 approximation error $\|f - f_\Lambda\|_{L^2_\mu}$, i.e., the first term on the right-hand side of (47), has been estimated by the best L^∞ approximation error $\|f - f_\Lambda\|_{L^\infty}$, which is generally half an order worse in the convergence rate (e.g., rate p for L^2 versus rate $p - 1/2 = q$ for L^∞), see, e.g., [17]. Moreover, a direct analysis on the discrete norm

$\|f - f_\Lambda\|_n$, i.e., the second term on the right-hand side of (47), based on properties of the lattice points, has a chance to improve upon the best L^∞ approximation error too. Indeed, function approximation based on rank-1 lattices has been analyzed in [5, 14, 15, 18] where the lattice generating vectors were constructed to minimize the approximation error directly, without the reconstruction property. It is known that if p is the rate of convergence for the best L^2 approximation error (rather than L^∞) then lattice generating vectors can be constructed to achieve $\|f - f_\Lambda^a\|_{L_\mu^2} = \mathcal{O}(n^{-p/2})$. There are also other approximation results using rank-1 or multiple rank-1 lattices, see, e.g., [1, 11, 12, 13, 19, 35, 36].

Rank-1 lattices are very attractive due to their simplicity and stability, and the availability of fast computation methods compared to other approximation algorithms.

References

[1] G. Byrenheid, L. Kämmerer, T. Ullrich, T. Volkmer, Tight error bounds for rank-1 lattice sampling in spaces of hybrid mixed smoothness, Numer. Math., 136:993–1034, 2017.

[2] A. Chkifa, A. Cohen, G. Migliorati, F. Nobile, R. Tempone, Discrete least-squares polynomial approximation with random evaluations – application to parametric and stochastic elliptic PDEs, ESAIM Math. Model. Numer. Anal., 49(3):815–837, 2015.

[3] A. Cohen, M. Davenport, D. Leviatan, On the stability and accuracy of least-squares approximations, Found. Comput. Math., 5:819–834, 2013.

[4] R. Cools, F.Y. Kuo, D. Nuyens, Constructing lattice rules based on weighted degree of exactness and worst case error, Computing, 87:63–89, 2010.

[5] R. Cools, F.Y. Kuo, D. Nuyens, G. Suryanarayana, Tent-transformed lattice rules for integration and approximation of multivariate non-periodic functions, J. Complexity, 36:166–181, 2016.

[6] R. Cools, K. Poppe, Chebyshev lattices, a unifying framework for cubature with Chebyshev weight function, BIT Numer. Math., 51:275–288, 2011.

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

[8] F. J. Hickernell, Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules, in K. T. Fang, F. J. Hickernell, and H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer, 274–289, 2002.

[9] L. Kämmerer, Reconstructing hyperbolic cross trigonometric polynomials from sampling along rank-1 lattices, SIAM J. Numer. Anal., 2773–2796, 2013.

[10] L. Kämmerer, Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices, in G.E. Fasshauer, L.L. Schumaker (Eds.), Approximation Theory XIV: San Antonio 2013, Springer, 255–271, 2014.

[11] L. Kämmerer, D. Potts, T. Volkmer, Approximation of multivariate periodic functions by trigonometric polynomials based on rank-1 lattice sampling, J. Complexity, 31:543–576, 2015.

[12] L. Kämmerer, S. Kunis, D. Potts, Interpolation lattices for hyperbolic cross trigonometric polynomials, J. Complexity, 28:76–92, 2012.

[13] L. Kämmerer, T. Volkmer, Approximation of multivariate periodic functions based on sampling along multiple rank-1 lattices, J. Approx. Theory, 246:1–27, 2019.

[14] F. Y. Kuo, I. H. Sloan, H. Woźniakowski, Lattice rules for multivariate approximation in the worst case setting, in H. Niederreiter and D. Talay (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2004, Springer, 289–330, 2006.

[15] F. Y. Kuo, I. H. Sloan, H. Woźniakowski, Lattice rule algorithms for multivariate approximation in the average case setting, J. Complexity, 24:283–323, 2008.

[16] F. Y. Kuo, G. W. Wasilkowski, H. Woźniakowski, Multivariate L_∞ approximation in the worst case setting over reproducing kernel Hilbert spaces, J. Approx. Theory, 152:135–160, 2008.

[17] F. Y. Kuo, G. W. Wasilkowski, H. Woźniakowski, On the power of standard information for multivariate approximation in the worst case setting, J. Approx. Theory, 158:97–125, 2009.

[18] F. Y. Kuo, G. W. Wasilkowski, H. Woźniakowski, Lattice algorithms for multivariate L_∞ approximation in the worst-case setting, Constr. Approx., 30:475–493, 2009.

[19] D. Li, F. J. Hickernell, Trigonometric spectral collocation methods on lattices, in S. Y. Cheng, C.-W. Shu, and T. Tang (Eds.), Recent Advances in Scientific Computing and Partial Differential Equations, AMS Series in Contemporary Mathematics, vol. 330, American Mathematical Society, Providence, Rhode Island, 121–132 (2003).

[20] S. A. Martucci, Symmetric convolution and the discrete sine and cosine transforms, IEEE Transactions on Signal Processing, 42:1038–1051, 1994.

[21] G. Migliorati, F. Nobile, E. von Schwerin, R.Tempone, Analysis of discrete L^2 projection on polynomial spaces with random evaluations, Found. Comput. Math. 14:419–456, 2014.

[22] G. Migliorati, F. Nobile, Analysis of discrete least squares on multivariate polynomial spaces with evaluations at low-discrepancy point sets, J. Complexity, 31(4):517–542, 2015.

[23] G. Migliorati, Multivariate Markov-type and Nikolskii-type inequalities for polynomials associated with downward closed multi-index sets, J. Approx. Theory, 189:137–159, 2015.

[24] H. Munthe-Kaas, T. Sørevik, Multidimensional pseudo-spectral methods on lattice grids, Appl. Numer. Math., 62:155–165, 2012.

[25] A. Narayan, Z. Xu, T. Zhou, Multivariate discrete least-squares approximations with a new type of collocation grid, SIAM J. Sci. Comput., 36(5):A2401–A2422, 2014.

[26] D. Potts, T. Volkmer, Fast and exact reconstruction of arbitratry multivariate algebraic polynomials in Chebyshev form, 2015 International Conference on Sampling Theory and Applications (SampTA), IEEE, 392–396, 2015.

[27] D. Potts, T. Volkmer, Sparse high-dimensional FFT based on rank-1 lattice sampling, Appl. Comput. Harmon. Anal. 41:713–748, 2016.

[28] I. H. Sloan, S. Joe, Lattice methods for multiple integration, Oxford University Press, Oxford, 1994.

[29] G. Suryanarayana, D. Nuyens, R. Cools, Reconstruction and collocation of a class of non-periodic functions by sampling along tent-transformed rank-1 lattices, Journal of Fourier Analysis and Applications, 22(1):187–214, 2016.

[30] Y. Suzuki, G. Suryanarayana, D. Nuyens, Strang splitting in combination with rank-1 and rank-r lattices for the time-dependent Schrödinger equation, SIAM J. Sci. Comput., 41(6):B1254–B1283, 2019.

[31] Y. Suzuki, D. Nuyens, Rank-1 lattices and higher-order exponential splitting for the time-dependent Schrödinger equation, in B. Tuffin, and P. L'Ecuyer (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2018, Springer, 485–502, 2020.

[32] T. Volkmer, sparseFFTr1l software library, https://www-user.tu-chemnitz.de/~tovo/software.php.en

[33] G. W. Wasilkowski, H. Woźniakowski, Weighted tensor product algorithms for linear multivariate problems, J. Complexity, 15:402–447, 1999.

[34] G. W. Wasilkowski, H. Woźniakowski, On the power of standard information for weighted approximation, Found. Comput. Math., 1:417–434, 2001.

[35] X. Y. Zeng, K. T. Leung, F. J. Hickernell, Error analysis of splines for periodic problems using lattice designs, in H. Niederreiter and D. Talay (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2004, Springer, 501–514, 2006.

[36] X. Y. Zeng, P. Kritzer, F. J. Hickernell, Spline methods using integration lattices and digital nets, Constr. Approx., 30: 529–555, 2009.