Hiroshima Math. J.
49 (2019), 139–159

Enumeration of the Chebyshev-Frolov lattice points in axis-parallel boxes

Kosuke Suzuki and Takehito Yoshiki

(Received March 20, 2018)
(Revised January 8, 2019)

Abstract. For a positive integer d, the d-dimensional Chebyshev-Frolov lattice is the $\mathbb{Z}$-lattice in $\mathbb{R}^d$ generated by the Vandermonde matrix associated to the roots of the d-dimensional Chebyshev polynomial. It is important to enumerate the points from the Chebyshev-Frolov lattices in axis-parallel boxes when $d = 2^n$ for a non-negative integer n, since the points are used as the nodes of Frolov's cubature formula, which achieves the optimal rate of convergence for many spaces of functions with bounded mixed derivatives and compact support. Kacwin, Oettershagen and Ullrich suggested an enumeration algorithm for such points and later Kacwin improved it, which are claimed to be efficient up to dimension $d = 16$. In this paper we suggest a new algorithm which enumerates such points in realistic time for $d = 2^n$, up to $d = 32$. Our algorithm is faster than theirs by a constant factor.

1. Introduction

Let d be a positive integer and $\mathbb{X} \subset \mathbb{R}^d$ be a d-dimensional lattice, i.e., there exists an invertible $d \times d$ matrix T over $\mathbb{R}$ such that

$$\mathbb{X} = T(\mathbb{Z}^d) = \{T\boldsymbol{k} \mid \boldsymbol{k} \in \mathbb{Z}^d\}.$$

The lattice $\mathbb{X}$ is said to be admissible if

$$\rho(\mathbb{X}) := \inf\left\{ \prod_{i=1}^{d} |x_i| \;\middle|\; (x_1, \dots, x_d)^\top \in \mathbb{X} \backslash \{\boldsymbol{0}\} \right\} > 0.$$

Thus, for an admissible lattice $\mathbb{X}$, the region $|x_1 \dots x_d| < \rho(\mathbb{X})$ contains no lattice points other than the origin. Using an admissible lattice $\mathbb{X} = T(\mathbb{Z}^d)$, Frolov's cubature formula approximates the integral

$$I(f) := \int_{[-1/2, 1/2]^d} f(\boldsymbol{x}) d\boldsymbol{x}$$

The research of the authors was supported under the Australian Research Councils Discovery Projects funding scheme (project number DP150101770).

2010 *Mathematics Subject Classification.* 11P21, 11Y16, 65D30, 65D32.

Key words and phrases. Lattice points, Chebyshev-Frolov lattice points, Frolov's cubature formula.

of a function $f : [-1/2, 1/2]^d \to \mathbb{R}$ by

$$Q_{a^{-1}T}(f) := |\det(a^{-1}T)| \sum_{x \in a^{-1}\mathbb{X} \cap [-1/2, 1/2]^d} f(x) \qquad \text{for } a \geq 1. \tag{1}$$

Thus the nodes are the shrunk lattice points $a^{-1}\mathbb{X}$ inside the box $[-1/2, 1/2]^d$. Frolov's cubature formula was first proposed by Frolov [4] and has been studied in many papers, see [1, 2, 3, 8, 10, 11, 12, 13, 14, 15]. One prominent feature of the formula is that it achieves the optimal rate of convergence for various spaces of functions with bounded mixed derivatives and compact support. This means that the approximation is automatically good, even without knowing specific information about the integrands. The constraint of compact supportness can be removed using some modification, see [9].

The implementation of Frolov's cubature formula requires one to enumerate the points in the set $a^{-1}\mathbb{X} \cap [-1/2, 1/2]^d$, or equivalently, the points in the set $\mathbb{X} \cap [-a/2, a/2]^d$. However, the enumeration is a difficult task even in moderate dimensions. Recently, an efficient enumeration algorithm for the so-called Chebyshev-Frolov lattices up to $d = 16$ was proposed by Kacwin, Oettershagen and Ullrich [7]. Since such lattices are admissible when $d = 2^n$, it is possible to implement Frolov's cubature formula for $d = 2^n$, up to $d = 16$. Based on the algorithm, numerical experiments to measure the performance of Frolov's cubature formula are given in [5] and the recent preprint [6]. Our contribution in this paper is to suggest a new efficient enumeration algorithm for the Chebyshev-Frolov lattices for $d = 2^n$. It is efficient up to $d = 32$.

The Chebyshev-Frolov lattices for $d = 2^n$ are examples of admissible lattices, suggested by Temlyakov [11, IV.4]. Let P_d be a rescaled d-dimensional Chebyshev polynomial defined by

$$P_d(x) = 2\cos(d \arccos(x/2)) \qquad \text{for } |x| < 2. \tag{2}$$

Its roots are given by

$$\zeta_{n,k} = 2\cos\left(\frac{\pi(2k-1)}{2d}\right), \qquad k = 1, \dots, 2^n. \tag{3}$$

With these roots, we define a Vandermonde matrix T by

$$T = (\zeta_{n,i}^{j-1})_{i,j=1}^{d}.$$

Now the d-dimensional Chebyshev-Frolov lattice is defined as the lattice $T(\mathbb{Z}^d)$. It is known that the lattice $T(\mathbb{Z}^d)$ is admissible if and only if $d = 2^n$. This is a special case of a general construction method for admissible lattices for any d elaborated in [11], see also Section 2. An advantage of the

Chebyshev-Frolov lattices is that the generating matrices are explicitly given. Using other kinds of Chebyshev polynomials, we can similarly construct admissible lattices for d with $d+1$ or $2d+1$ being prime. However, this is out of the scope of this paper.

We now briefly recall results in [5] and [7]. The paper [7] established an enumeration algorithm of the lattice points in $[-a/2, a/2]^d$, for any orthogonal lattices. The approach is as follows. The enumeration of the lattice points $T(\mathbb{Z}^d) \cap [-a/2, a/2]^d$ with a $d \times d$ matrix T is equivalent to the enumeration of the points $\mathbb{Z}^d \cap T^{-1}([-a/2, a/2]^d)$. They used a "bounding set" $B \supset T^{-1}([-a/2, a/2]^d)$ which allows for an easy enumeration. Since different T may give the same lattice points, we need to choose T carefully. The idea is that if T is orthogonal then we can take a comparably small bounding set; for the sphere S of radius $a\sqrt{d}/2$ with center at origin, which includes $[-a/2, a/2]^d$, we can take the ellipsoid $T^{-1}(S)$ as a bounding set. Since all the axes of the ellipsoid are aligned with the coordinate axes, it allows for an easy enumeration. They further discovered that Chebyshev-Frolov lattices are orthogonal, hence this approach is applicable to the desired enumeration. They claimed that it is efficient up to $d = 16$. It is improved in the master thesis of Kacwin [5, Algorithm 2] by taking a reduced bounding set.

Our algorithms are based on another property particular to the Chebyshev-Frolov lattices. Our key observation is that the 2^n-dimensional Chebyshev-Frolov lattice with a certain permutation of coordinates is generated by a matrix A_n which satisfies a recursive property as in formula (4) in Section 3. This property reduces the 2^n-dimensional enumeration to a number of 2^{n-1}-dimensional enumerations as in Lemma 2. This recursion implies Algorithm 1. By applying this repeatedly, eventually the enumeration is reduced to nested 1-dimensional enumerations, which can be implemented as 2^n-nested for-loops, see Theorem 3 and Algorithm 2. In other words, we do not need a bounding set: The set $A_n^{-1}([-a/2, a/2]^d)$ already allows for an easy enumeration. This strongly supports the fastness of our algorithm. We will describe our algorithms in Section 3.

Let us compare the pros and cons of the algorithms in [5, 7] and our Algorithm 2. Firstly, their algorithms are more widely applicable. They are applicable to any orthogonal lattices, and in particular to the construction of Frolov cubature rules not only for the dimension $d = 2^n$ but also for d with $2d+1$ or $d+1$ being prime, whereas our algorithm is only for the dimension $d = 2^n$. Secondly, our algorithm is faster than theirs. As far as we observed, the execution time of both algorithms linearly depends on the scaling parameter N and exponentially depends on the dimension d. We observed that our algorithm is faster by a constant factor for a given d, which is about $10, 6, 8, 10^3, 6 \times 10^5$ for $d = 2, 4, 8, 16, 32$, respectively. Hence our algorithm is much better when $d = 16, 32$. All results of our experiments are written in Section 5. Thirdly, another advantage of our algorithm is that it can enumerate the Chebyshev-Frolov lattice points in arbitrary axis-parallel boxes. This helps us to implement not only Frolov's cubature formula but also its randomization. Randomized Frolov's cubature formula was introduced by Krieg and Novak [8] and studied further by Ullrich [14]. It inherits the prominent convergence behavior of the deterministic version as well as it is unbiased. Further it also has the optimal order of convergence in the randomized sense for Sobolev spaces with isotropic and mixed smoothness. We will explain how to enumerate the integration nodes of the deterministic and randomized versions with our algorithm in Section 4.

Throughout this paper we use the following notation. The symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the set of non-negative integers, integers, rational numbers and real numbers, respectively. For $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$, $(\boldsymbol{x}_1; \boldsymbol{x}_2) \in \mathbb{R}^{2d}$ denotes the vector where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are vertically concatenated. We denote by $SL_d(\mathbb{Z})$ the special linear group of degree d over $\mathbb{Z}$, i.e., the set of matrices over $\mathbb{Z}$ whose determinant is 1. For $x_1, \dots, x_d \in \mathbb{R}$, $\mathrm{diag}(x_1, \dots, x_d)$ denotes the diagonal matrix with $(x_1, \dots, x_d)$ at the diagonal. For a vector $\boldsymbol{b} = (b_1, \dots, b_d)^\top \in \mathbb{R}^d$ and $\boldsymbol{c} = (c_1, \dots, c_d)^\top \in \mathbb{R}^d$, we define $[\boldsymbol{b}, \boldsymbol{c}] := \prod_{i=1}^d [b_i, c_i]$ and $\max(\boldsymbol{b}, \boldsymbol{c}) := (\max(b_i, c_i))_{i=1}^d \in \mathbb{R}^d$, and write $\boldsymbol{b} \le \boldsymbol{c}$ if $b_i \le c_i$ holds for all $1 \le i \le d$.

2. Construction method of admissible lattices

One general construction scheme for admissible lattices is the one studied in Temlyakov [11, IV.4]. Let $p_d(x) \in \mathbb{Z}[x]$ be a d-dimensional polynomial with integer coefficients satisfying the following three properties: (i) its leading coefficient is 1, (ii) it is irreducible over $\mathbb{Q}$, (iii) it has d distinct real roots, say $\zeta_1, \dots, \zeta_d \in \mathbb{R}$. With these roots, we define a Vandermonde matrix T by

$$T = (\zeta_i^{j-1})_{i,j=1}^d.$$

Then the lattice $T(\mathbb{Z}^d)$ generated by T is admissible. Frolov used $q_d(x) = -1 + \prod_{j=1}^d (x - 2j + 1)$ in his paper [4]. Note that he originally used the lattice made from $q_d(x)$ not for T in (1) but for its dual lattice. However, later it was shown that $T(\mathbb{Z}^d)$ itself is admissible if and only if its dual lattice is admissible, see [10, Lemma 3.1] and also [15, Lemma 2.1] for a Vandermonde matrix. One disadvantage of the choice of q_d is that its roots are not given explicitly.

In [11] Temlyakov proposed to use the rescaled Chebyshev polynomials P_d as in (2) when $d = 2^n$ for a non-negative integer n. It is shown that P_d satisfies the conditions (i) and (iii), and its roots are given as in (3). Further

P_d is irreducible if and only if $d = 2^n$. Thus the Chebyshev-Frolov lattice, i.e., the lattice constructed as above using $P_d(x)$, is admissible if and only if $d = 2^n$. It is also known that Chebyshev-Frolov lattices are orthogonal.

THEOREM 1 ([7, Theorem 1.1]). *For any positive integer d, the d-dimensional Chebyshev-Frolov lattice $T(\mathbb{Z}^d)$ is orthogonal. In particular, there exists a lattice representation $\tilde{T} = TS$ with some $S \in SL_d(\mathbb{Z})$ such that*

- *For each component $t_{i,j}$ of $\tilde{T}$, it holds that $|t_{i,j}| \le 2$,*
- *$\tilde{T}^\top \tilde{T} = \mathrm{diag}(d, 2d, \ldots, 2d)$.*

3. Enumeration of the Chebyshev-Frolov lattice points

3.1. Recursive property of generating matrices. We consider coordinate-permuted Chebyshev-Frolov lattices. We define $\sigma(k) \in \mathbb{Z}$ for $k \in \mathbb{N}$ recursively by $\sigma(1) = 1$ and

$$\sigma(k) = 2^{n+1} + 1 - \sigma(k - 2^n)$$

for k with $2^n + 1 \le k \le 2^{n+1}$, $n \in \mathbb{N}$. For all $n \in \mathbb{N}$, the map $\sigma(\cdot)$ is a permutation on $\{1, \ldots, 2^n\}$, which is shown by induction on n as follows. The case $n = 0$ is trivial. We assume this holds for n. By the definition of $\sigma(k)$ and induction assumption, $\sigma(\cdot)$ is a permutation on $\{1, \ldots, 2^n\}$ and also a permutation on $\{2^n + 1, \ldots, 2^{n+1}\}$. This proves the result for $n + 1$.

Let $n \in \mathbb{N}$ and put $d = 2^n$. We now define $\xi_{n,k} \in \mathbb{R}$ by

$$\xi_{n,k} = 2\cos\left(\frac{\pi(2\sigma(k) - 1)}{2d}\right) \qquad \text{for } k = 1, \ldots, d,$$

and consider a Vandermonde matrix $V_n \in \mathbb{R}^{d \times d}$ defined by

$$V_n := (\xi_{n,i}^{j-1})_{i,j=1}^{d} = \begin{pmatrix} 1 & \xi_{n,1} & \cdots & \xi_{n,1}^{d-1} \\ 1 & \xi_{n,2} & \cdots & \xi_{n,2}^{d-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{n,d} & \cdots & \xi_{n,d}^{d-1} \end{pmatrix}.$$

Comparing $\xi_{n,k}$'s and $\zeta_{n,k}$'s defined as in (3), we find that $\xi_{n,k}$'s are also the roots of $P_d(x)$ since $\sigma(\cdot)$ is a permutation on $\{1, \ldots, d\}$. Thus the lattice $V_n(\mathbb{Z}^d)$ is a coordinate permutation of the usual Chebyshev-Frolov lattice.

Further we define a diagonal matrix $D_n \in \mathbb{R}^{d \times d}$ by

$$D_n := \mathrm{diag}(\xi_{n+1,1}, \ldots, \xi_{n+1,d}).$$

We are now ready to define a matrix $A_n \in \mathbb{R}^{d\times d}$ recursively by $A_0 = 1$ and

$$A_{n+1} = \begin{pmatrix} A_n & D_n A_n \\ A_n & -D_n A_n \end{pmatrix}. \tag{4}$$

The following lemma shows that A_n can be used as a generating matrix of the Chebyshev-Frolov lattices, i.e., $V_n(\mathbb{Z}^d) = A_n(\mathbb{Z}^d)$.

LEMMA 1. *For all $n \in \mathbb{N}$, there exists $S_n \in \mathbb{Z}^{2^n\times 2^n}$ such that* $\det S_n = \pm 1$ *and $V_n S_n = A_n$.*

PROOF. We prove the lemma by induction on n. The case $n = 0$ is trivial since $V_0 = A_0 = 1$. Now we assume that the assertion holds for n and prove it for $n+1$. Put $d = 2^n$. Define a matrix $V'_{n+1} \in \mathbb{R}^{2d\times 2d}$ obtained by column swapping of V_{n+1} as

$$V'_{n+1} = \begin{pmatrix} 1 & \xi_{n+1,1}^2 & \cdots & \xi_{n+1,1}^{2(d-1)} & \xi_{n+1,1} & \xi_{n+1,1}^3 & \cdots & \xi_{n+1,1}^{2d-1} \\ 1 & \xi_{n+1,2}^2 & \cdots & \xi_{n+1,2}^{2(d-1)} & \xi_{n+1,2} & \xi_{n+1,2}^3 & \cdots & \xi_{n+1,2}^{2d-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_{n+1,2d}^2 & \cdots & \xi_{n+1,2d}^{2(d-1)} & \xi_{n+1,2d} & \xi_{n+1,2d}^3 & \cdots & \xi_{n+1,2d}^{2d-1} \end{pmatrix}.$$

Since V'_{n+1} is obtained by column swapping of V_{n+1}, there exists $W_{n+1} \in \mathbb{Z}^{2d\times 2d}$ such that $\det W_{n+1} = \pm 1$ and $V'_{n+1} = V_{n+1} W_{n+1}$.

Define $U_n = (u_{i,j})_{i,j=1}^d \in \mathbb{Z}^{d\times d}$ by

$$u_{i,j} = (-2)^{j-i}\binom{j-1}{i-1},$$

where $\binom{j}{i}$ is a binomial coefficient and is defined to be zero if $i > j$. Since U_n is upper-triangular and all the diagonal entries are 1, $U_n \in SL_d(\mathbb{Z})$ holds. We now compute $V'_{n+1}\begin{pmatrix} U_n & \mathbf{0} \\ \mathbf{0} & U_n \end{pmatrix}$, where $\mathbf{0} \in \mathbb{R}^{d\times d}$ is the zero matrix. First we note that we have $\xi_{n+1,i+d} = -\xi_{n+1,i}$ for $1 \le i \le d$ by using $\cos(\theta+\pi) = -\cos\theta$. Hence, denoting by $\tilde{V}_n$ the $d\times d$ upper-left submatrix of V'_{n+1}, we have

$$V'_{n+1} = \begin{pmatrix} \tilde{V}_n & D_n\tilde{V}_n \\ \tilde{V}_n & -D_n\tilde{V}_n \end{pmatrix}.$$

Further, using the formula $\cos 2\theta = 2\cos^2\theta - 1$, we have $\xi_{n,i} = \xi_{n+1,i}^2 - 2$ for $1 \le i \le d$ and thus $\xi_{n,i}^l = (\xi_{n+1,i}^2 - 2)^l$ for all $l \in \mathbb{N}$. Thus, using the binomial expansion of this equality, we have $\tilde{V}_n U_n = V_n$. Therefore we have

$$V'_{n+1}\begin{pmatrix} U_n & \mathbf{0} \\ \mathbf{0} & U_n \end{pmatrix} = \begin{pmatrix} \tilde{V}_n & D_n\tilde{V}_n \\ \tilde{V}_n & -D_n\tilde{V}_n \end{pmatrix}\begin{pmatrix} U_n & \mathbf{0} \\ \mathbf{0} & U_n \end{pmatrix} = \begin{pmatrix} V_n & D_n V_n \\ V_n & -D_n V_n \end{pmatrix}.$$

By induction assumption, there exists $S_n \in \mathbb{Z}^{d\times d}$ such that $\det S_n = \pm 1$ and $V_n S_n = A_n$. Hence

$$\begin{pmatrix} V_n & D_n V_n \\ V_n & -D_n V_n \end{pmatrix} \begin{pmatrix} S_n & \mathbf{0} \\ \mathbf{0} & S_n \end{pmatrix} = \begin{pmatrix} A_n & D_n A_n \\ A_n & -D_n A_n \end{pmatrix} = A_{n+1}.$$

Thus we have shown that $V_{n+1} S_{n+1} = A_{n+1}$ with

$$S_{n+1} = W_{n+1} \begin{pmatrix} U_n & \mathbf{0} \\ \mathbf{0} & U_n \end{pmatrix} \begin{pmatrix} S_n & \mathbf{0} \\ \mathbf{0} & S_n \end{pmatrix}.$$

This shows that the assertion holds for $n+1$. □

3.2. Recursive enumeration. In this subsection we give a recursive algorithm to obtain the Chebyshev-Frolov lattice points $A_n(\mathbb{Z}) \cap [\boldsymbol{b}, \boldsymbol{c}] = \{A_n \boldsymbol{k} \mid \boldsymbol{k} \in \mathbb{Z}^n, \boldsymbol{b} \le A_n \boldsymbol{k} \le \boldsymbol{c}\}$ for $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$. We start with the definition of functions which are used to state Lemma 2. Then we reduce a 2^{n+1}-dimensional enumeration to 2^n-dimensional enumerations.

Definition 1. Let $n \in \mathbb{N}$ and $d := 2^n$. Let $\boldsymbol{a}_1, \boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathbb{R}^d$ and $\boldsymbol{b} = (\boldsymbol{b}_1; \boldsymbol{b}_2)$, $\boldsymbol{c} := (\boldsymbol{c}_1; \boldsymbol{c}_2) \in \mathbb{R}^{2d}$. We define functions $\boldsymbol{\rho}_n(\boldsymbol{b})$, $\boldsymbol{\phi}_n(\boldsymbol{a}_1, \boldsymbol{b}, \boldsymbol{c})$ and $\boldsymbol{\psi}_n(\boldsymbol{a}_1, \boldsymbol{b}, \boldsymbol{c})$ by

$$\boldsymbol{\rho}_n(\boldsymbol{b}) = (\boldsymbol{b}_1 + \boldsymbol{b}_2)/2 \in \mathbb{R}^d,$$

$$\boldsymbol{\phi}_n(\boldsymbol{a}_1, \boldsymbol{b}, \boldsymbol{c}) = D_n^{-1} \max(\boldsymbol{b}_1 - \boldsymbol{a}_1, -\boldsymbol{c}_2 + \boldsymbol{a}_1) \in \mathbb{R}^d,$$

$$\boldsymbol{\psi}_n(\boldsymbol{a}_1, \boldsymbol{b}, \boldsymbol{c}) = D_n^{-1} \min(\boldsymbol{c}_1 - \boldsymbol{a}_1, -\boldsymbol{b}_2 + \boldsymbol{a}_1) \in \mathbb{R}^d.$$

Lemma 2. *Let $n \in \mathbb{N}$ and put $d = 2^n$. Let $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$ and define $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^{2d}$ by $\boldsymbol{b} = (\boldsymbol{b}_1; \boldsymbol{b}_2)$, $\boldsymbol{c} := (\boldsymbol{c}_1; \boldsymbol{c}_2)$ and $\boldsymbol{x} := (\boldsymbol{x}_1; \boldsymbol{x}_2)$. Then the inequality $\boldsymbol{b} \le A_{n+1}\boldsymbol{x} \le \boldsymbol{c}$ is equivalent to the simultaneous inequalities*

$$\begin{cases} \boldsymbol{\rho}_n(\boldsymbol{b}) \le A_n \boldsymbol{x}_1 \le \boldsymbol{\rho}_n(\boldsymbol{c}), & (5) \\ \boldsymbol{\phi}_n(A_n\boldsymbol{x}_1, \boldsymbol{b}, \boldsymbol{c}) \le A_n \boldsymbol{x}_2 \le \boldsymbol{\psi}_n(A_n\boldsymbol{x}_1, \boldsymbol{b}, \boldsymbol{c}). & (6) \end{cases}$$

Proof. From (4), $\boldsymbol{b} \le A_{n+1}\boldsymbol{x} \le \boldsymbol{c}$ is equivalent to

$$\begin{cases} \boldsymbol{b}_1 \le A_n \boldsymbol{x}_1 + D_n A_n \boldsymbol{x}_2 \le \boldsymbol{c}_1, \\ \boldsymbol{b}_2 \le A_n \boldsymbol{x}_1 - D_n A_n \boldsymbol{x}_2 \le \boldsymbol{c}_2. \end{cases} \tag{7}$$

By adding the inequalities in (7) we have

$$\boldsymbol{\rho}_n(\boldsymbol{b}) \le A_n \boldsymbol{x}_1 \le \boldsymbol{\rho}_n(\boldsymbol{c}). \tag{8}$$

On the other hand, (7) is equivalent to

$$\begin{cases} \boldsymbol{b}_1 - A_n\boldsymbol{x}_1 \le D_nA_n\boldsymbol{x}_2 \le \boldsymbol{c}_1 - A_n\boldsymbol{x}_1, \\ -\boldsymbol{c}_2 + A_n\boldsymbol{x}_1 \le D_nA_n\boldsymbol{x}_2 \le -\boldsymbol{b}_2 + A_n\boldsymbol{x}_1, \end{cases}$$

which is equivalent to

$$\max(\boldsymbol{b}_1 - A_n\boldsymbol{x}_1, -\boldsymbol{c}_2 + A_n\boldsymbol{x}_1) \le D_nA_n\boldsymbol{x}_2 \le \min(\boldsymbol{c}_1 - A_n\boldsymbol{x}_1, -\boldsymbol{b}_2 + A_n\boldsymbol{x}_1).$$

Since D_n is a diagonal matrix whose diagonal entries are positive, this inequality is equivalent to

$$\boldsymbol{\phi}_n(A_n\boldsymbol{x}_1, \boldsymbol{b}, \boldsymbol{c}) \le A_n\boldsymbol{x}_2 \le \boldsymbol{\psi}_n(A_n\boldsymbol{x}_1, \boldsymbol{b}, \boldsymbol{c}). \tag{9}$$

Thus we have

$$(7) \Leftrightarrow (7) \text{ and } (8) \Leftrightarrow (9) \text{ and } (8),$$

which is what we desired to prove. □

Let $n \in \mathbb{N}$, $d := 2^n$ and $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$. We define

$$\mathscr{P}_n(\boldsymbol{b}, \boldsymbol{c}) := \{\boldsymbol{k} \in \mathbb{Z}^d \mid \boldsymbol{b} \le A_n\boldsymbol{k} \le \boldsymbol{c}\}.$$

Lemma 2 implies the following theorem that utilizes the definition of $\mathscr{P}_n(\boldsymbol{b}, \boldsymbol{c})$.

Theorem 2. *Let $n \in \mathbb{N}$, $d := 2^n$ and $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^{2d}$. Then we have*

$$\mathscr{P}_{n+1}(\boldsymbol{b}, \boldsymbol{c}) = \left\{ \begin{pmatrix} \boldsymbol{k}_1 \\ \boldsymbol{k}_2 \end{pmatrix} \in \mathbb{Z}^{2d} \;\middle|\; \begin{array}{l} \boldsymbol{k}_1 \in \mathscr{P}_n(\boldsymbol{\rho}_n(\boldsymbol{b}), \boldsymbol{\rho}_n(\boldsymbol{c})), \\ \boldsymbol{k}_2 \in \mathscr{P}_n(\boldsymbol{\phi}_n(A_n\boldsymbol{k}_1, \boldsymbol{b}, \boldsymbol{c}), \boldsymbol{\psi}_n(A_n\boldsymbol{k}_1, \boldsymbol{b}, \boldsymbol{c})) \end{array} \right\}.$$

This theorem reduces an enumeration in dimension 2^{n+1} to enumerations in dimension 2^n. Further the case $n = 0$ is easy to solve, since $k \in \mathscr{P}_0(b, c)$ for $k \in \mathbb{Z}$ and $b, c \in \mathbb{R}$ is equivalent to $b \le k \le c$. This justifies Algorithm 1, which gives the set $\mathscr{P}_n(\boldsymbol{b}, \boldsymbol{c})$.

3.3. Sequential enumeration. One disadvantage of Algorithm 1 is that it requires a large amount of memory. That is, while expanding recursions in Algorithm 1, all of $\mathrm{Set}(n, \boldsymbol{b}, \boldsymbol{c})$ have to be memorized. In this subsection, to overcome this disadvantage we derive simultaneous inequalities equivalent to $\boldsymbol{b} \le A_n\boldsymbol{x} \le \boldsymbol{c}$ by applying Lemma 2 repeatedly and then we give a sequential enumeration algorithm.

We begin with an illustration for the case $n = 2$. Fix $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^4$ and let $\boldsymbol{x} = (x_1; x_2; x_3; x_4)$. Our aim is to obtain simultaneous inequalities which are equivalent to $\boldsymbol{b} \le A_2\boldsymbol{x} \le \boldsymbol{c}$. From Lemma 2, it is reduced to

$$\begin{cases} \boldsymbol{\beta}_{1,1} \le A_1(x_1; x_2) \le \boldsymbol{\gamma}_{1,1}, & (10) \\ \boldsymbol{\beta}_{1,2} \le A_1(x_3; x_4) \le \boldsymbol{\gamma}_{1,2}. & (11) \end{cases}$$

Algorithm 1 Recursive algorithm to obtain the set $\mathscr{P}_n(\boldsymbol{b}, \boldsymbol{c})$

```
procedure ALGORITHM1(n, b, c)                                    ▷ Output the set P_n(b, c)
    SET(n, b, c)
end procedure

function SET(n, b, c)                                            ▷ Output the set P_n(b, c)
    if n = 0 then
        return {k ∈ Z | ⌈b⌉ ≤ k ≤ ⌊c⌋}                         ▷ In this case b and c are scalar
    else
        P ← empty set                                            ▷ Initialize P as the empty set
        for all k_1 ∈ SET(n − 1, ρ_{n−1}(b), ρ_{n−1}(c)) do
            for all k_2 ∈ SET(n − 1, φ_{n−1}(A_{n−1}k_1, b, c), ψ_{n−1}(A_{n−1}k_1, b, c)) do
                append (k_1; k_2) to P                           ▷ Append a point to the set P
            end for
        end for
        return P
    end if
end function
```

where we put $\boldsymbol{\beta}_{1,1} := \boldsymbol{\rho}_1(\boldsymbol{b})$, $\boldsymbol{\gamma}_{1,1} := \boldsymbol{\rho}_1(\boldsymbol{c})$, $\boldsymbol{\beta}_{1,2} := \boldsymbol{\phi}_1(A_1(x_1; x_2), \boldsymbol{b}, \boldsymbol{c})$ and $\boldsymbol{\gamma}_{1,2} := \boldsymbol{\psi}_1(A_1(x_1; x_2), \boldsymbol{b}, \boldsymbol{c})$. Whereas $\boldsymbol{\beta}_{1,2}$ and $\boldsymbol{\gamma}_{1,2}$ are not determined until x_1 and x_2 are fixed, $\boldsymbol{\beta}_{1,1}$ and $\boldsymbol{\gamma}_{1,1}$ are determined using only $\boldsymbol{b}$ and $\boldsymbol{c}$. Hence we first consider (10). Again from Lemma 2, (10) is reduced to

$$\begin{cases} \beta_{0,1} \le A_0 x_1 \le \gamma_{0,1}, & (12) \\ \beta_{0,2} \le A_0 x_2 \le \gamma_{0,2}, & (13) \end{cases}$$

where we put $\beta_{0,1} := \boldsymbol{\rho}_0(\boldsymbol{\beta}_{1,1})$, $\gamma_{0,1} := \boldsymbol{\rho}_0(\boldsymbol{\gamma}_{1,1})$, $\beta_{0,2} := \boldsymbol{\phi}_0(A_0 x_1, \boldsymbol{\beta}_{1,1}, \boldsymbol{\gamma}_{1,1})$ and $\gamma_{0,2} := \boldsymbol{\psi}_0(A_0 x_1, \boldsymbol{\beta}_{1,1}, \boldsymbol{\gamma}_{1,1})$. Whereas $\beta_{0,2}$ and $\gamma_{0,2}$ are not determined until x_1 is fixed, $\beta_{0,1}$ and $\gamma_{0,1}$ are determined using only $\boldsymbol{b}$ and $\boldsymbol{c}$. Thus we can fix x_1 satisfying (12). Once x_1 is fixed, $\beta_{0,2}$ and $\gamma_{0,2}$ are determined and thus we can fix x_2 with (13). Once x_2 is fixed, then $\boldsymbol{\beta}_{1,2}$ and $\boldsymbol{\gamma}_{1,2}$ are determined, and again from Lemma 2, Inequality (11) is reduced to

$$\begin{cases} \beta_{0,3} \le A_0 x_3 \le \gamma_{0,3}, & (14) \\ \beta_{0,4} \le A_0 x_4 \le \gamma_{0,4}, & (15) \end{cases}$$

where we put $\beta_{0,3} := \boldsymbol{\rho}_0(\boldsymbol{\beta}_{1,2})$, $\gamma_{0,3} := \boldsymbol{\rho}_0(\boldsymbol{\gamma}_{1,2})$, $\beta_{0,4} := \boldsymbol{\phi}_0(A_0 x_3, \boldsymbol{\beta}_{1,2}, \boldsymbol{\gamma}_{1,2})$ and $\gamma_{0,4} := \boldsymbol{\psi}_0(A_0 x_3, \boldsymbol{\beta}_{1,2}, \boldsymbol{\gamma}_{1,2})$. Now $\beta_{0,3}$ and $\gamma_{0,3}$ are determined and we can fix x_3 with (14). Once x_3 is fixed, $\beta_{0,4}$ and $\gamma_{0,4}$ are determined and thus we can fix x_4 with (15). In this way, we have shown that $\boldsymbol{b} \le A_2 \boldsymbol{x} \le \boldsymbol{c}$ is equivalent to the simultaneous inequalities (12)–(15), where $\beta_{0,1}$ and $\gamma_{0,1}$ are already determined and $\beta_{0,i}$ and $\gamma_{0,i}$ are determined when $x_1, \ldots, x_{i-1}$ are fixed ($i = 2, 3, 4$). This equivalence allows us to implement the enumeration of the vectors $\boldsymbol{k} \in \mathbb{Z}^4$ with $\boldsymbol{b} \le A_2\boldsymbol{k} \le \boldsymbol{c}$ by 4-nested for-loops or an equivalent tail recursion.

We now generalize the procedure to any $n \in \mathbb{N}$. Hereafter, to clarify which coordinates we consider, we use the following notation.

DEFINITION 2. Let $n, L, a \in \mathbb{N}$ with $0 \le L \le n$, $1 \le a \le 2^{n-L}$ and $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$. We define

$$\boldsymbol{x}_{L,a} := (x_{(a-1)2^L+1}, \dots, x_{a2^L})^\top \in \mathbb{Z}^{2^L},$$

$$\boldsymbol{\alpha}_{L,a} := A_L \boldsymbol{x}_{L,a} \in \mathbb{R}^{2^L}.$$

Put $d := 2^n$ and fix $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^d$. Our aim is to reduce $\boldsymbol{b} \le A_n \boldsymbol{x}_{n,1} \le \boldsymbol{c}$ to simultaneous 1-dimensional inequalities. Put $\boldsymbol{\beta}_{n,1} := \boldsymbol{b}$ and $\boldsymbol{\gamma}_{n,1} := \boldsymbol{c}$. From Lemma 2, for all $0 \le L \le n$ and $1 \le a \le 2^{n-L}$, an inequality $\boldsymbol{\beta}_{L,a} \le A_L \boldsymbol{x}_{L,a} \le \boldsymbol{\gamma}_{L,a}$ is reduced to

$$\begin{cases} \boldsymbol{\beta}_{L-1,2a-1} \le A_{L-1}\boldsymbol{x}_{L-1,2a-1} \le \boldsymbol{\gamma}_{L-1,2a-1}, \\ \boldsymbol{\beta}_{L-1,2a} \le A_{L-1}\boldsymbol{x}_{L-1,2a} \le \boldsymbol{\gamma}_{L-1,2a}, \end{cases}$$

where $\boldsymbol{\beta}_{L,a}, \boldsymbol{\gamma}_{L,a} \in \mathbb{R}^{2^L}$ are defined by

$$\boldsymbol{\beta}_{L-1,2a-1} = \boldsymbol{\rho}_{L-1}(\boldsymbol{\beta}_{L,a}), \tag{16}$$

$$\boldsymbol{\gamma}_{L-1,2a-1} = \boldsymbol{\rho}_{L-1}(\boldsymbol{\gamma}_{L,a}), \tag{17}$$

$$\boldsymbol{\beta}_{L-1,2a} = \boldsymbol{\phi}_{L-1}(\boldsymbol{\alpha}_{L-1,2a-1}, \boldsymbol{\beta}_{L,a}, \boldsymbol{\gamma}_{L,a}), \tag{18}$$

$$\boldsymbol{\gamma}_{L-1,2a} = \boldsymbol{\psi}_{L-1}(\boldsymbol{\alpha}_{L-1,2a-1}, \boldsymbol{\beta}_{L,a}, \boldsymbol{\gamma}_{L,a}). \tag{19}$$

We have seen that $\boldsymbol{\alpha}_{L,a}$'s, $\boldsymbol{\beta}_{L,a}$'s and $\boldsymbol{\gamma}_{L,a}$'s depend on each other and some of them are not determined until some of x_i's are fixed. The dependency between $\boldsymbol{\alpha}_{L,a}$'s is given as follows. For $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^{2^L}$, define

$$\boldsymbol{\tau}_{L+1}(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) := (\boldsymbol{\alpha}_1 + D_L\boldsymbol{\alpha}_2; \boldsymbol{\alpha}_1 - D_L\boldsymbol{\alpha}_2) \in \mathbb{R}^{2^{L+1}}.$$

Then for $1 \le L \le n$ and $1 \le a \le 2^{n-L}$ it follows from (4) that

$$\boldsymbol{\alpha}_{L,a} = \boldsymbol{\tau}_L(\boldsymbol{\alpha}_{L-1,2a-1}, \boldsymbol{\alpha}_{L-1,2a}). \tag{20}$$

We now study how those values are determined. We define the sets of indices $\mathscr{A}_i$ and $\mathscr{B}_i$ for $i \in \mathbb{N}$, $0 \le i \le 2^n$ by

$$\mathscr{A}_i = \{(L, a) \in \mathbb{N} \times (\mathbb{N}\backslash\{0\}) \mid 2^L a \le i\},$$

$$\mathscr{B}_i = \{(L, a) \in \mathbb{N} \times (\mathbb{N}\backslash\{0\}) \mid 2^L(a-1) \le i\}.$$

Hence we have

$$\mathscr{A}_0 := \varnothing, \qquad \mathscr{B}_0 := \{(j,1) \mid j \in \mathbb{N},\ 0 \le j \le n\},$$

and, for $i = 2^r p$ where $r \in \mathbb{N}$ and p is an odd integer,

$$\begin{aligned} \mathscr{A}_i \backslash \mathscr{A}_{i-1} &= \{(L,a) \in \mathbb{N} \times (\mathbb{N}\backslash\{0\}) \mid 2^L a = i\} \\ &= \{(j, 2^{r-j}p) \mid j \in \mathbb{N},\ 0 \le j \le r\}, \\ \mathscr{B}_i \backslash \mathscr{B}_{i-1} &= \{(L,a) \in \mathbb{N} \times (\mathbb{N}\backslash\{0\}) \mid 2^L (a-1) = i\} \\ &= \{(j, 2^{r-j}p+1) \mid j \in \mathbb{N},\ 0 \le j \le r\}. \end{aligned}$$

The following lemmas show that these sets control the determination of the values and that we can compute the determined values efficiently.

Lemma 3. *Let $i \in \mathbb{N}$, $0 \le i \le 2^n$. Let $x_1, \dots, x_i$ be fixed. If $(L,a) \in \mathscr{A}_i$ holds, then $\boldsymbol{a}_{L,a}$ is determined.*

Proof. We prove the lemma by induction on i. If $i = 0$, we have nothing to prove. Now let $i = 2^r p > 0$ where $r \in \mathbb{N}$ and p is an odd integer and assume that the result holds for $i-1$. Let $x_1, \dots, x_i$ be fixed. By induction assumption, for all $(L,a) \in \mathscr{A}_{i-1}$ the value $\boldsymbol{a}_{L,a}$ is determined. Thus it remains to show the assertion for $(L,a) \in \mathscr{A}_i \backslash \mathscr{A}_{i-1}$. Since x_i is fixed, $\boldsymbol{a}_{0,2^r p} = x_i$ is determined. Further, by induction assumption, for all $0 \le j < r$ we have $(j, 2^{r-j}p - 1) \in \mathscr{A}_{2^r p - 2^j} \subset \mathscr{A}_{i-1}$ and thus $\boldsymbol{a}_{j,2^{r-j}p-1}$ is determined. By using these results and applying (20) with $(L,a) = (j, 2^{r-j}p)$ for $j = 1, \dots, r$, $\boldsymbol{a}_{j,2^{r-j}p}$ is sequentially determined for all $0 \le j \le r$. This proves the result for i. □

We remark that the lemma is directly shown as follows: The condition that $x_1, \dots, x_i$ are fixed implies that $\boldsymbol{x}_{L,a}$ is fixed for all $(L,a) \in \mathscr{A}_i$ and thus $\boldsymbol{a}_{L,a} = A_L \boldsymbol{x}_{L,a}$ is determined. The procedure shown in the proof, however, can save the cost to compute the values in the similar way that the fast Fourier transform does.

Lemma 4. *Let $i \in \mathbb{N}$, $0 \le i < 2^n$. Let $x_1, \dots, x_i$ be fixed. If $(L,a) \in \mathscr{B}$ holds, then $\boldsymbol{\beta}_{L,a}$ and $\boldsymbol{\gamma}_{L,a}$ are determined.*

Proof. We prove the lemma by induction on i. First assume $i = 0$, i.e., none of x_j are fixed for $1 \le j \le 2^n$. Even then, $\boldsymbol{\beta}_{n,1}$ and $\boldsymbol{\gamma}_{n,1}$ are determined as $\boldsymbol{\beta}_{n,1} = \boldsymbol{b}$ and $\boldsymbol{\gamma}_{n,1} = \boldsymbol{c}$. Hence, using (16) and (17) repeatedly, $\boldsymbol{\beta}_{j,1}$ and $\boldsymbol{\gamma}_{j,1}$ are determined for all $0 \le j \le n$. This proves the result for $i = 0$.

Now we assume that the lemma holds for $i-1$. Let $x_1, \ldots, x_i$ be fixed. By induction assumption, $\boldsymbol{\beta}_{L,a}$ and $\boldsymbol{\gamma}_{L,a}$ are determined for all $(L,a) \in \mathscr{B}_{i-1}$. Thus it remains to show the assertion for $(L,a) \in \mathscr{B} \backslash \mathscr{B}_{i-1}$. We decompose i as $i = 2^r p$ where $r \in \mathbb{N}$ and p is an odd integer. Lemma 3 implies that $\boldsymbol{\alpha}_{r,p}$ is determined. Further, by induction assumption we have $(r+1, (p+1)/2) \in \mathscr{B}_{2^r(p-1)} \subset \mathscr{B}_{i-1}$ and thus $\boldsymbol{\beta}_{r+1,(p+1)/2}$ and $\boldsymbol{\gamma}_{r+1,(p+1)/2}$ are determined. Then $\boldsymbol{\beta}_{r,p+1}$ and $\boldsymbol{\gamma}_{r,p+1}$ are determined from these results, (18) and (19). Thus, by using (16) and (17) with $(L,a) = (r-j, 2^j p + 1)$ for $j = 0, \ldots, r-1$, $\boldsymbol{\beta}_{r-j,2^j p+1}$ and $\boldsymbol{\gamma}_{r-j,2^j p+1}$ are sequentially determined for all $1 \le j \le r$. This proves the result for i. □

Since $(0, i+1) \in \mathscr{B}_i$, Lemma 4 implies that $\boldsymbol{\beta}_{0,i+1}$ and $\boldsymbol{\gamma}_{0,i+1}$ are determined when $x_1, \ldots, x_i$ are fixed. Thus we have shown the following equivalence in summary.

THEOREM 3. *The inequality $\boldsymbol{b} \le A_n \boldsymbol{x} \le \boldsymbol{c}$ is equivalent to 2^n simultaneous inequalities*

$$\boldsymbol{\beta}_{0,i} \le x_i \le \boldsymbol{\gamma}_{0,i} \qquad \text{for } 1 \le i \le 2^n,$$

where $\boldsymbol{\beta}_{0,1}$ and $\boldsymbol{\gamma}_{0,1}$ are already determined and $\boldsymbol{\beta}_{0,i}$ and $\boldsymbol{\gamma}_{0,i}$ are determined when $x_1, \ldots, x_{i-1}$ are fixed, as in Lemmas 3 and 4.

Lemmas 3–4 and Theorem 3 justify Algorithm 2, a tail recursive enumeration of all the Chebyshev-Frolov lattice points $A_n \boldsymbol{k}$ with $\boldsymbol{k} \in \mathbb{Z}^{2^n}$ in the box $[\boldsymbol{b}, \boldsymbol{c}]$. Algorithm 2 is equivalent to 2^n-nested for-loops. We remark that this theorem implies that the set $A_n^{-1}([\boldsymbol{b}, \boldsymbol{c}])$ already allows for an easy enumeration.

REMARK 1. *If your task is only to approximate the integration value, replace Line 21 in Algorithm 2 by the evaluation of the integrand. You do not need to store any of the Chebyshev-Frolov lattice points.*

4. Frolov's cubature formula and its randomization

In this section we revisit Frolov's cubature formula and its randomization, and in particular we show how to enumerate the integration nodes using Algorithm 2.

Let $\boldsymbol{v} \in \mathbb{R}^d$ and take a matrix $T \in \mathbb{R}^{d \times d}$ which generates an admissible lattice $T(\mathbb{Z}^d)$. We define the set

$$X(T, \boldsymbol{v}) := \{T(\boldsymbol{k} + \boldsymbol{v}) \mid \boldsymbol{k} \in \mathbb{Z}^d\} \cap [-1/2, 1/2]^d$$

Algorithm 2 Enumerate the lattice points in the box $[\boldsymbol{b}, \boldsymbol{c}]$

procedure ALGORITHM2($n, \boldsymbol{b}, \boldsymbol{c}$) ▷ Give the lattice points in the box
  for $i = 1$ to 2^n **do** ▷ Preparation for updating
    store $r(i), p(i) \in \mathbb{N}$ as $i = 2^{r(i)} p(i)$
  end for ▷ Finish preparation
  $\boldsymbol{\beta}_{n,1} \leftarrow \boldsymbol{b}$ ▷ Update $\boldsymbol{\beta}_{L,a}$ and $\boldsymbol{\gamma}_{L,a}$ with $\mathscr{B}_0$
  $\boldsymbol{\gamma}_{n,1} \leftarrow \boldsymbol{c}$
  for $j = n-1$ to 0 **do**
    $\boldsymbol{\beta}_{j,1} \leftarrow \boldsymbol{\rho}_j(\boldsymbol{\beta}_{j+1,1})$
    $\boldsymbol{\gamma}_{j,1} \leftarrow \boldsymbol{\rho}_j(\boldsymbol{\gamma}_{j+1,1})$
  end for ▷ Finish updating
  ENUM(1)
end procedure

function ENUM(i) ▷ Enumerate the i-th coordinate k_i
  for $k_i = \lceil \boldsymbol{\beta}_{0,i} \rceil$ to $\lfloor \boldsymbol{\gamma}_{0,i} \rfloor$ **do**
    if $i \neq 2^n$ **then**
      UPDATEALPHA(i)
      UPDATEBETAGAMMA(i)
      ENUM($i+1$)
    else ▷ That is, if $i = 2^n$
      UPDATEALPHA(2^n)
      Output $\boldsymbol{\alpha}_{n,1}$ ▷ $\boldsymbol{\alpha}_{n,1} = A_n \boldsymbol{k}$ is a lattice point
    end if
  end for
end function

function UPDATEALPHA(i) ▷ Update $\boldsymbol{\alpha}_{L,a}$ with $\mathscr{A}$
  $\boldsymbol{\alpha}_{0,i} \leftarrow k_i$
  for $j = 1$ to $r(i)$ **do**
    $\boldsymbol{\alpha}_{j,2^{r(i)-j}p(i)} \leftarrow \boldsymbol{\tau}_j(\boldsymbol{\alpha}_{j-1,2^{r(i)-j+1}p(i)-1}, \boldsymbol{\alpha}_{j-1,2^{r(i)-j+1}p(i)})$
  end for
end function

function UPDATEBETAGAMMA(i) ▷ Update $\boldsymbol{\beta}_{L,a}$ and $\boldsymbol{\gamma}_{L,a}$ with $\mathscr{B}$
  $\boldsymbol{\beta}_{r(i),p(i)+1} \leftarrow \boldsymbol{\phi}_{r(i)}(\boldsymbol{\alpha}_{r(i),p(i)}, \boldsymbol{\beta}_{r(i)+1,(p(i)+1)/2}, \boldsymbol{\gamma}_{r(i)+1,(p(i)+1)/2})$
  $\boldsymbol{\gamma}_{r(i),p(i)+1} \leftarrow \boldsymbol{\psi}_{r(i)}(\boldsymbol{\alpha}_{r(i),p(i)}, \boldsymbol{\beta}_{r(i)+1,(p(i)+1)/2}, \boldsymbol{\gamma}_{r(i)+1,(p(i)+1)/2})$
  for $j = r(i)-1$ to 0 **do**
    $\boldsymbol{\beta}_{j,2^{r(i)-j}p(i)+1} \leftarrow \boldsymbol{\rho}_j(\boldsymbol{\beta}_{j+1,2^{r(i)-j-1}p(i)+1})$
    $\boldsymbol{\gamma}_{j,2^{r(i)-j}p(i)+1} \leftarrow \boldsymbol{\rho}_j(\boldsymbol{\gamma}_{j+1,2^{r(i)-j-1}p(i)+1})$
  end for
end function

and the cubature rule for a function $f(\boldsymbol{x})$ on $[-1/2,1/2]^d$ as

$$Q_{T,\boldsymbol{v}}(f) = |\det T| \sum_{\boldsymbol{x} \in X(T,\boldsymbol{v})} f(\boldsymbol{x}).$$

As mentioned in the introduction, Frolov's cubature formula is of the form $Q_{a^{-1}T,\boldsymbol{0}}(f)$ for $a > 1$. For the number of integration nodes, it is known from [10] that

$$\lim_{a\to\infty} \det(a^{-1}T)|X(a^{-1}T,\boldsymbol{0})| = 1. \tag{21}$$

We roughly explain the error analysis of Frolov's cubature formula $Q_{a^{-1}T,\boldsymbol{0}}(f)$. Let H^s_{mix} be the Sobolev space of mixed smoothness on $[0,1]^d$ equipped with the norm

$$\|f\|_{s,\mathrm{mix}} := \sum_{\substack{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_d)\in\mathbb{N}^d \\ \sum_i \alpha_i \le s}} \|D^{\boldsymbol{\alpha}} f\|^2_{L_2}$$

where $D^{\boldsymbol{\alpha}}$ stands for the usual partial derivative operator. Let $f \in H^s_{\mathrm{mix}}$. We denote by $\hat{f}$ the Fourier transform of f (here f is extended by zero to $\mathbb{R}^d$). Then it follows from Poisson summation formula that

$$Q_{a^{-1}T,\boldsymbol{0}}(f) = \sum_{\boldsymbol{x}\in aT^{-\top}(\mathbb{Z}^d)} \hat{f}(\boldsymbol{x}), \tag{22}$$

where $T^{-\top}$ is the inverse of the transpose of T. We note that $aT^{-\top}(\mathbb{Z}^d)$ is the dual lattice of $a^{-1}T(\mathbb{Z}^d)$ and that having T admissible implies that $T^{-\top}$ is also admissible. From (22) the integration error is bounded as

$$|I(f) - Q_{a^{-1}T,\boldsymbol{0}}(f)| \le \sum_{\boldsymbol{x}\in aT^{-\top}(\mathbb{Z}^d)\setminus\{\boldsymbol{0}\}} |\hat{f}(\boldsymbol{x})|. \tag{23}$$

An important fact is, roughly speaking, that $|\hat{f}(\boldsymbol{x})|$ is small if $\prod_{i=1}^d |x_i|$ is large. Recalling that an admissible lattice have no lattice points other than the origin with small $\prod_{i=1}^d |x_i|$, we can show that the right hand side of (23) is small. More precisely we have

$$\sum_{\boldsymbol{x}\in aT^{-\top}(\mathbb{Z}^d)\setminus\{\boldsymbol{0}\}} |\hat{f}(\boldsymbol{x})| \le C_{s,d} a^{-sd} (\log a)^{(s-1)/2} \|f\|_{s,\mathrm{mix}}$$

for large enough a, where $C_{s,d}$ is a constant depending only on s and d. This means that the convergence rate of the integration error with respect to the number of the nodes is $O(n^{-s}(\log n)^{(s-1)/2})$, which is shown to be optimal. It is shown that Frolov's cubature formula $Q_{a^{-1}T,\mathbf{0}}(f)$ also achieves the optimal rate of convergence in Besov-Triebel-Lizorkin spaces.

Following [7], we use scaled (and coordinate-permuted) Chebyshev-Frolov lattices as admissible lattices for Frolov's cubature formula. Let $n \in \mathbb{N}$ and let A_n be defined as in (4). For a scaling parameter $N \in \mathbb{R}$ with $N > 0$, we define the value $s(N) := (|\det(A_n)|N)^{-1/d}$ and the matrix

$$A_{n,N} := s(N)A_n,$$

which satisfies $|\det(A_{n,N})| = 1/N$. From (21), N is an approximation for the number of the nodes. From Theorem 1, we have $|\det(A_n)| = (2d)^{d/2}/\sqrt{2}$. We consider Frolov's cubature formula $Q_{a^{-1}T,\mathbf{0}}(f)$ for $N \in \mathbb{N}$. To find the integration nodes, we can use Algorithm 2 and the bijection

$$\{A_n\boldsymbol{k} \mid \boldsymbol{k} \in \mathbb{Z}^d\} \cap [\boldsymbol{b}, \boldsymbol{c}] \to X(A_{n,N}, \mathbf{0}), \qquad \boldsymbol{x} \mapsto s(N)\boldsymbol{x},$$

where $\boldsymbol{b} := -s(N)^{-1}(1/2, \dots, 1/2)^\top$ and $\boldsymbol{c} := -\boldsymbol{b} = s(N)^{-1}(1/2, \dots, 1/2)^\top$.

Randomized Frolov's cubature formula was introduced by Krieg and Novak [8], and studied further by Ullrich [14]. Our algorithm introduced below follows the exposition in [14], but note that $A_{n,N}$ in this paper corresponds to $B_N^{-\top}$ in [14]. Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two independent random vectors that are uniformly distributed in $[1/2, 3/2]^d$ and $[0,1]^d$, respectively. Let $U := \mathrm{diag}(\boldsymbol{u})$. We define randomized Frolov's cubature formula M_N using $A_{n,N}$ by

$$M_N(f) := Q_{U^{-1}A_{n,N},\boldsymbol{v}}(f).$$

How can we enumerate the nodes of the formula $M_N(f)$? We have

$$\begin{aligned} \boldsymbol{x} \in X(U^{-1}A_{n,N}, \boldsymbol{v}) &\Leftrightarrow \boldsymbol{x} = U^{-1}A_{n,N}(\boldsymbol{k} + \boldsymbol{v}) \in [-1/2, 1/2]^d \\ &\Leftrightarrow A_n\boldsymbol{k} \in s(N)^{-1}U[-1/2, 1/2]^d - A_n\boldsymbol{v}. \end{aligned}$$

Hence, defining $\boldsymbol{h} := (1/2, \dots, 1/2)^\top \in \mathbb{R}^d$, $\boldsymbol{b} = -s(N)^{-1}U\boldsymbol{h} - A_n\boldsymbol{v}$ and $\boldsymbol{c} = s(N)^{-1}U\boldsymbol{h} - A_n\boldsymbol{v}$, we have the following bijective map

$$\{A_n\boldsymbol{k} \mid \boldsymbol{k} \in \mathbb{Z}^d\} \cap [\boldsymbol{b}, \boldsymbol{c}] \to X(U^{-1}A_{n,N}, \boldsymbol{v}), \qquad \boldsymbol{x} \mapsto s(N)U^{-1}(\boldsymbol{x} + A_n\boldsymbol{v}).$$

Thus we can use Algorithm 2 to enumerate the nodes of randomized Frolov's cubature formula. We remark that the vector $A_n\boldsymbol{v}$ can be quickly computed, as with the computation of $\boldsymbol{a}_{n,1}$.

5. Numerical efficiency of the algorithm

In this section we numerically show the efficiency of our Algorithm 2. We counted the number of the nodes of Frolov's cubature formula using the Chebyshev-Frolov lattices, for dimensions $d = 2, 4, 8, 16, 32$ and for the scaling parameter $N = 2^m$ with $m = 1, \dots, 30$, based on our Algorithm 2 and Kacwin's algorithm [5, Algorithm 2]. More precisely, for our algorithm we replaced Line 21 in Algorithm 2 by incrementing a counter for the number of the nodes[1]. For Kacwin's algorithm, he kindly shared his codes with us and we used it with a slight modification. We conducted the experiments on an HPC cloud environment[2] provided by Information Media Center, Hiroshima University. We used an Intel Xeon E5-2697 v3 2.6GHz 8 cores CPU. Our codes are implemented in C and Kacwin's ones are in C++. They are compiled by GCC 4.4.7 with −O3 optimization flag. We used the function clock_gettime in C standard library for obtaining the execution time.

The result is summarized in Tables 1 and 2. Table 1 shows the number of the nodes. Table 2 shows the execution time of both algorithms. In Table 2, "Error" means that an error of type 'class std::bad_alloc' occurred, which would be due to our modification, and the blanks mean that we did not conduct the computation due to time constraint. We can see that, for a fixed dimension d, the execution time of both algorithms increases linearly with respect to the scaling parameter N. Our algorithm is faster by a constant factor than Kacwin's as far as we observed. For $d = 2, 4, 8, 16, 32$, the constant factor is about $10, 6, 8, 10^3, 6 \times 10^5$, respectively. Hence our algorithm is much faster when $d \geq 16$. For a fixed N, the execution time increases rapidly with respect to d. We can also see that the scaling parameter N does not well approximate the number of nodes when $d = 32$, for $N \leq 2^{30}$.

We remark on the accuracy of Algorithm 2. It requires many floating-point arithmetic operations, so it might have some errors. The following observations and experiments, however, support that our algorithm is sufficiently accurate in practical use. Firstly, We confirm that the number of the enumerated points given in Table 1 coincide with the result in [7, Appendix], which gives those for $d = 2, 4, 8, 16$ and $N = 4^m$ with $3 \leq m \leq 10$. Secondly, Kacwin's algorithm also enumerates the same number of points as far as we observed as in Table 2. Thirdly, we also conducted our experiment with quadruple-precision arithmetic. We confirmed that for $d = 32$ and $N \leq 2^m$ with $1 \leq m \leq 23$, we obtained the same number of points as those given in Table 1. Thus we can conclude that our algorithm is sufficiently accurate in practical use.

[1] The code we used can be found at https://github.com/tttyoyoyttt/the_Chebyshev_Frolov_lattice_points.

[2] https://www.media.hiroshima-u.ac.jp/services/hpc/hpcc

Table 1. The number of the nodes of Chebyshev-Frolov's cubature formula for $N = 2^m$ with $m = 1, \dots, 30$ and $d = 2, 4, 8, 16, 32$ is given.

m	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$
1	3	5	19	77	3377
2	5	5	19	127	4105
3	7	11	23	151	5041
4	15	15	27	223	6371
5	31	31	45	295	8915
6	65	71	79	423	11867
7	131	123	167	539	15291
8	257	261	271	967	20651
9	513	513	529	1377	29215
10	1027	1025	1067	2043	42323
11	2049	2049	2107	3503	61997
12	4095	4099	4113	5835	88645
13	8191	8201	8283	10451	128269
14	16383	16385	16413	18901	186749
15	32767	32775	32823	36085	278961
16	65539	65533	65645	69353	430037
17	131075	131095	131183	136839	679287
18	262145	262143	262263	267257	1102547
19	524289	524281	524341	530333	1799443
20	1048579	1048609	1048779	1054837	2990409
21	2097153	2097143	2097107	2106165	5079585
22	4194307	4194355	4194399	4207997	8757305
23	8388611	8388589	8388843	8402385	15442557
24	16777215	16777221	16777535	16797845	27637841
25	33554429	33554439	33554807	33577467	50306689
26	67108861	67108867	67108777	67135425	92921093
27	134217727	134217723	134217783	134246629	173897749
28	268435457	268435461	268435889	268458047	328647641
29	536870913	536870913	536871467	536891351	627372745
30	1073741827	1073741807	1073742019	1073829043	1208920345

Table 2. The execution time of the proposed algorithm (Algorithm 2) and Kacwin's one for $N = 2^m$ with $m = 1, \ldots, 30$ and $d = 2, 4, 8, 16, 32$ is given in seconds. "Error" means that an error of type 'class std::bad_alloc' occurred. The blanks mean that we did not conduct the computation due to time constraint.

	$d = 2$		$d = 4$		$d = 8$		$d = 16$		$d = 32$	
m	Proposed	Kacwin	Proposed	Kacwin	Proposed	Kacwin	Proposed	Kacwin	Proposed	Kacwin
1	0.000053	0.000075	0.000050	0.000084	0.000054	0.000159	0.000156	0.034146	0.024060	4565.822161
2	0.000045	0.000080	0.000049	0.000068	0.000063	0.000206	0.000208	0.060339	0.032049	7033.384453
3	0.000041	0.000073	0.000044	0.000082	0.000066	0.000299	0.000239	0.077497	0.043997	10979.072244
4	0.000041	0.000078	0.000049	0.000080	0.000064	0.000404	0.000327	0.174110	0.063395	23146.858947
5	0.000043	0.000073	0.000039	0.000078	0.000093	0.000692	0.000454	0.260097	0.084701	39957.242851
6	0.000059	0.000078	0.000054	0.000106	0.000074	0.000943	0.000618	0.579767	0.134083	89084.765037
7	0.000053	0.000085	0.000073	0.000121	0.000108	0.001497	0.000990	0.956735	0.191159	133461.487911
8	0.000055	0.000116	0.000064	0.000155	0.000132	0.002222	0.001595	1.461850	0.288220	
9	0.000059	0.000142	0.000077	0.000217	0.000218	0.003517	0.002324	2.748688	0.495829	
10	0.000053	0.000191	0.000095	0.000328	0.000335	0.005606	0.003639	4.570040	0.754119	
11	0.000062	0.000311	0.000144	0.000588	0.000476	0.009317	0.006007	7.091587	1.126724	
12	0.000082	0.000518	0.000184	0.000856	0.000948	0.014830	0.010352	12.574445	1.781347	
13	0.000136	0.000835	0.000299	0.001517	0.001394	0.024815	0.016905	22.246927	2.818438	
14	0.000186	0.001639	0.000508	0.002709	0.002441	0.040768	0.028669	38.803520	4.245921	
15	0.000332	0.003057	0.001056	0.005096	0.004302	0.068813	0.050016	73.292351	6.768067	

16	0.000595	0.006064	0.001616	0.009477	0.007706	0.116134	0.080850	126.019780	10.933249	
17	0.001149	0.011979	0.003049	0.017258	0.013942	0.175368	0.142785	211.570833	17.869626	
18	0.002229	0.022873	0.005740	0.032724	0.025371	0.292535	0.246138	387.480818	29.147994	
19	0.004380	0.044854	0.010877	0.061618	0.045617	0.509542	0.459063	689.401104	47.707398	
20	0.008753	0.088895	0.020685	0.118855	0.078319	0.853430	0.750615	1178.161018	79.972279	
21	0.017232	0.176927	0.040112	0.223835	0.147387	1.470535	1.341655	2087.259125	130.570463	
22	0.034288	0.359689	0.077001	0.437086	0.256223	2.624878	2.383631	3645.979307	221.130876	
23	0.068257	0.710619	0.147774	0.839913	0.476147	4.565089	4.469382	6698.425143	371.520567	
24	0.135184	1.352417	0.278379	1.602821	0.886616	7.987438	7.924161	11582.177697	647.578808	
25	0.261187	2.493735	0.514475	3.221815	1.663163	14.392347	14.375711	20721.980316	1117.627054	
26	0.498398	5.025748	1.004894	6.343853	3.173570	25.573729	25.611510	Error	1911.914483	
27	0.969649	9.959484	2.032257	12.416560	6.104644	Error	46.829467	Error	3287.846831	
28	1.906489	20.126803	3.936894	Error	12.456399	Error	85.948624	Error	5803.747645	
29	3.795645	Error	7.806856	Error	21.965344	Error	158.118302	Error	10227.261284	
30	8.118048	Error	15.749159	Error	42.132804	Error	297.322952	Error	17969.952262	

Acknowledgement

The authors are grateful to Christopher Kacwin, Mario Ullrich and Tino Ullrich for their valuable comments about the theory and experiments for Frolov's cubature formula, and to Christopher Kacwin for sharing his code. The authors would like to thank Hiroki Kajiura for helping them with the time comparison experiment.

References

[1] V. V. Dubinin, Optimal quadrature formulas for classes of functions with a bounded mixed difference, Mat. Zametki **49**(1) (1991), 149–151.

[2] V. V. Dubinin, Cubature formulas for Besov classes, Izv. Ross. Akad. Nauk Ser. Mat. **61**(2) (1997), 27–52.

[3] D. Dũng, V. N. Temlyakov and T. Ullrich, Hyperbolic cross approximation, ArXiv preprint arXiv:1601.03978 [math.NA].

[4] K. K. Frolov, Upper bounds for the errors of quadrature formulae on classes of functions, Dokl. Akad. Nauk SSSR **231**(4) (1976), 818–821.

[5] C. Kacwin, Realization of the Frolov cubature formula via orthogonal Chebyshev-Frolov lattices, Master's thesis, Mathematisch-Naturwissenschaftliche Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn (2016), http://wissrech.ins.uni-bonn.de/teaching/master/masterthesis_kacwin.pdf

[6] C. Kacwin, J. Oettershagen, M. Ullrich and T. Ullrich, Numerical performance of optimized Frolov lattices in tensor product reproducing kernel Sobolev spaces, ArXiv preprint arXiv:1601.03978 [math.NA].

[7] C. Kacwin, J. Oettershagen and T. Ullrich, On the orthogonality of the Chebyshev-Frolov lattice and applications, Monatsh. Math. **184**(3) (2017), 425–441.

[8] D. Krieg and E. Novak, A Universal Algorithm for Multivariate Integration, Found. Comput. Math. **17**(4) (2017), 895–916.

[9] V. K. Nguyen, M. Ullrich and T. Ullrich, Change of variable in spaces of mixed smoothness and numerical integration of multivariate functions on the unit cube, Constr. Approx. **46**(1) (2017), 69–108.

[10] M. M. Skriganov, Constructions of uniform distributions in terms of geometry of numbers, Algebra i Analiz **6**(3) (1994), 200–230.

[11] V. N. Temlyakov, Approximation of periodic functions, Computational Mathematics and Analysis Series. Nova Science Publishers, Inc., Commack, NY (1993).

[12] V. N. Temlyakov, Cubature formulas, discrepancy, and nonlinear approximation, J. Complexity **19**(3) (2003), 352–391.

[13] M. Ullrich, On "Upper error bounds for quadrature formulas on function classes" by K. K. Frolov, In: R. Cools, D. Nuyens (eds.) Monte Carlo and quasi-Monte Carlo methods 2014, Springer Proc. Math. Stat., vol. 163, pp. 571–582. Springer, Heidelberg (2016).

[14] M. Ullrich, A Monte Carlo Method for Integration of Multivariate Smooth Functions, SIAM J. Numer. Anal. **55**(3) (2017), 1188–1200.

[15] M. Ullrich and T. Ullrich, The role of Frolov's cubature formula for functions with bounded mixed derivative, SIAM J. Numer. Anal. **54**(2) (2016), 969–993.

Kosuke Suzuki
Graduate School of Science
Hiroshima University
1-3-1 *Kagamiyama, Higashihiroshima* 739-8526*, Japan*
E-mail: kosuke-suzuki@hiroshima-u.ac.jp

Takehito Yoshiki
Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University
36-1 *Yoshida-Honmachi, Sakyo-ku, Kyoto* 606-8501*, Japan*
E-mail: yoshiki.takehito.47x@st.kyoto-u.ac.jp