Sparse high-dimensional FFT based on rank-1 lattice sampling

Daniel Potts* Toni Volkmer†

In this paper, we suggest approximate algorithms for the reconstruction of sparse high-dimensional trigonometric polynomials, where the support in frequency domain is unknown. Based on ideas of constructing rank-1 lattices component-by-component, we adaptively construct the index set of frequencies belonging to the non-zero Fourier coefficients in a dimension incremental way. When we restrict the search space in frequency domain to a full grid $[-N, N]^d \cap \mathbb{Z}^d$ of refinement $N \in \mathbb{N}$ and assume that the cardinality of the support of the trigonometric polynomial in frequency domain is bounded by the sparsity $s \in \mathbb{N}$, our method requires $\mathcal{O}(d\,s^2 N)$ samples and $\mathcal{O}(d\,s^3 + d\,s^2 N \log(s\,N))$ arithmetic operations in the case $\sqrt{N} \lesssim s \lesssim N^d$. Moreover, we discuss possibilities to reduce the number of samples and arithmetic operations by applying methods from compressed sensing and a version of Prony's method. For the latter, the number of samples is reduced to $\mathcal{O}(d\,s + d\,N)$ and the number of arithmetic operations is $\mathcal{O}(d\,s^3)$. Various numerical examples demonstrate the efficiency of the suggested method.

Keywords and phrases : trigonometric polynomials, lattice rule, rank-1 lattice, sparse fast Fourier transform, approximation of multivariate functions, FFT.

2000 AMS Mathematics Subject Classification : 65T, 65T40, 42A10.

*Technische Universität Chemnitz, Faculty of Mathematics, 09107 Chemnitz, Germany
potts@mathematik.tu-chemnitz.de, Phone:+49-371-531-32150, Fax:+49-371-531-832150

†Technische Universität Chemnitz, Faculty of Mathematics, 09107 Chemnitz, Germany
toni.volkmer@mathematik.tu-chemnitz.de, Phone:+49-371-531-39999, Fax:+49-371-531-839999

1 Introduction

We consider the approximation of high-dimensional multivariate periodic functions $f \in L^1(\mathbb{T}^d)$ by trigonometric polynomials $p \in \Pi_I := \operatorname{span}\{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\circ} \colon \boldsymbol{k} \in I\}$ with frequencies supported on an unknown index set $I \subset \mathbb{Z}^d$ of finite cardinality,

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in I} \hat{p}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}, \quad \hat{p}_{\boldsymbol{k}} \in \mathbb{C}, \tag{1.1}$$

where $\mathbb{T}^d \simeq [0,1)^d$ is the d-dimensional torus. If such a function f fulfills certain smoothness conditions, characterized by the decay of its Fourier coefficients $\hat{f}_{\boldsymbol{k}} := \int_{\mathbb{T}^d} f(\boldsymbol{x})\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}\mathrm{d}\boldsymbol{x}$, $\boldsymbol{k} \in \mathbb{Z}^d$, one theoretical possibility for a good approximation by a trigonometric polynomial $p \in \Pi_I$ is using the Fourier partial sum $S_I f := \sum_{\boldsymbol{k}\in I} \hat{f}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\circ} \in \Pi_I$ of f. In this case, error estimates are well-known.

A more practical approach than using the Fourier partial sum $S_I f$ of the exact Fourier coefficients $\hat{f}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of a function f is to construct a trigonometric polynomial $p \in \Pi_I$ by approximately computing the Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ of f from sampling values using a suitable sampling scheme. In higher dimensions (e.g. $d > 4$), (generalized) sparse grids [46, 31, 12, 14, 13] are often used, where the corresponding frequency index sets I are (generalized) hyperbolic crosses. However, the discrete/fast Fourier transform [1, 15, 13] for the approximate computation of the Fourier coefficients $\hat{f}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of f from sparse grid samples may be numerically unstable, cf. [26]. Recently in [29, 28], another sampling scheme was considered, so-called reconstructing rank-1 lattices, which allow for the fast, constructive and perfectly stable approximate reconstruction of the Fourier coefficients $\hat{f}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of f, where $I \subset \mathbb{Z}^d$, $|I| < \infty$, is an arbitrary frequency index set. Lattice rules have extensively been investigated for the integration of functions of many variables for a long time, cf. e.g., [44, 6, 7] and the extensive reference list therein. Especially, rank-1 lattice rules have also been studied for the approximation of multivariate functions of suitable smoothness and similar error estimates are obtained like when sampling at sparse grid nodes, cf. [45, 35, 33, 36]. Using the ideas of a component-by-component construction of lattice rules for the numerical integration of trigonometric polynomials from [6], a generalized component-by-component construction method for reconstructing rank-1 lattices was presented in [22, 25].

For both sampling schemes, the sparse grids and the rank-1 lattices, a suitable frequency index set I has to be given in order to obtain a good approximation of the function f or a reasonable function class for f has to be known. In that case, very precise error estimates can be shown, see [13, 29]. In this paper, we assume that we do not have (exact) knowledge of a suitable frequency index set I and that we only know a relatively large superset $\Gamma \supset I$. Possible applications for our approach may be pseudo-spectral methods for solving partial differential equations as it was considered in the case of sparse grids discretization in [10, 11, 43].

Even if we do not consider a general function f but a trigonometric polynomial $p \in \Pi_I$, reconstructing the non-zero Fourier coefficients $\mathbf{0} \neq \hat{p}_{\boldsymbol{k}} \in \mathbb{C}$ of p from sampling values becomes very hard in higher dimensions d if the frequency index set I is very large. In the following, we consider the recovery of all the frequencies $\boldsymbol{k} \in I$ belonging to non-zero Fourier coefficients $\mathbf{0} \neq \hat{p}_{\boldsymbol{k}} \in \mathbb{C}$ as well as the Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in \mathbb{C}$ themselves from sampling values of a high-dimensional trigonometric polynomial p in (1.1). We assume that the (unknown) support $\operatorname{supp}\hat{p} := \{\boldsymbol{k} \in I \colon \hat{p}_{\boldsymbol{k}} \neq 0\} \subset \mathbb{Z}^d$ of p (in frequency domain) lies within a superset $I = \Gamma$ of finite cardinality, $\operatorname{supp}\hat{p} \subseteq \Gamma$, and is "sparse" in some sense. In this setting, determining

the unknown support $\operatorname{supp}\hat{p}$ and the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp}\hat{p}$, of the trigonometric polynomial p is equivalent to solving the sparse recovery problem

$$\|\tilde{\hat{\boldsymbol{p}}}\|_0 = |\{\boldsymbol{k} \in \Gamma : \tilde{\hat{p}}_{\boldsymbol{k}} \neq 0\}| \xrightarrow{\tilde{\hat{\boldsymbol{p}}}} \min \quad \text{subject to} \quad \boldsymbol{A}\tilde{\hat{\boldsymbol{p}}} = \boldsymbol{p}, \tag{1.2}$$

for the under-determined case $|\Gamma| > |\mathcal{X}|$, where $\mathcal{X} := \{\boldsymbol{x}_0, \dots, \boldsymbol{x}_{L-1}\}$ is the set of sampling nodes, $|\mathcal{X}| = L$, $\boldsymbol{p} := (p(\boldsymbol{x}_\ell))_{\boldsymbol{x}_\ell \in \mathcal{X}}$ is the vector of sampling values, $\boldsymbol{A} := \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}_\ell}\right)_{\boldsymbol{x}_\ell \in \mathcal{X},\, \boldsymbol{k}\in\Gamma}$ is the Fourier matrix and $\tilde{\hat{\boldsymbol{p}}} := (\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k}\in\Gamma}$ is the vector of (computed) Fourier coefficients.

In the case where the *search space* Γ (in frequency domain) is the full grid $\hat{G}_N^d := \{\boldsymbol{k} \in \mathbb{Z}^d \colon \|\boldsymbol{k}\|_\infty \leq N\}$, the straighforward approach would be using a d-dimensional discrete Fourier transform (DFT) of length $(2N+1, \dots, 2N+1)^\top \in \mathbb{N}^d$ to obtain Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \hat{G}_N^d$, which can be computed efficiently by the fast Fourier transform (FFT) in $\mathcal{O}(N^d \log N)$ arithmetic operations. The frequency index set $I = \operatorname{supp}\hat{p}$ is then obtained by $I := \{\boldsymbol{k} \in \hat{G}_N^d \colon \tilde{\hat{p}}_{\boldsymbol{k}} \neq 0\}$ and the (non-zero) Fourier coefficients of p are given by $\hat{p}_{\boldsymbol{k}} := \tilde{\hat{p}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$. Clearly, this approach suffers severely from the curse of dimensions since $|\hat{G}_N^d| = (2N+1)^d$ many sampling values are used and $|\hat{G}_N^d| = (2N+1)^d$ many Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \hat{G}_N^d$, are computed as an intermediate result.

One alternate approach to determine the unknown support $\operatorname{supp}\hat{p}$ and the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp}\hat{p}$, from a smaller amount of samples is applying random sampling in compressed sensing [8, 2, 9]. Provided a so-called restricted isometry condition is fulfilled, the sparse recovery problem can be solved efficiently using ℓ_1 minimization, cf. [3, 40, 41, 42, 37, 32]. The restricted isometry condition is fulfilled with probability at least $1-\eta$ if the number of samples $L \geq C\,|\operatorname{supp}\hat{p}|\,\log^4(|\Gamma|)\,\log(1/\eta)$, where C is an absolute constant independent of the dimension d. The arithmetic complexity is then $\mathcal{O}(L\,|\Gamma|)$, e.g., see [27, Sec. 3.4] and the references therein, and hence impractical for large search spaces Γ, e.g., $\Gamma = \hat{G}_N^d$.

Another possibility is using the so-called sparse fast Fourier transform, cf. [17, 16, 19, 18]. In [16], an algorithm is presented for the one-dimensional case and $\Gamma = \hat{G}_N^1$, which allows to determine the (unknown) support $\operatorname{supp}\hat{p}$ and the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ from $\mathcal{O}(|\operatorname{supp}\hat{p}|\log N)$ samples with an arithmetic complexity of $\mathcal{O}(|\operatorname{supp}\hat{p}|\log N)$, as well as a second algorithm, which allows the s-sparse ℓ_2 best approximation of the Fourier coefficients of p from $\mathcal{O}(s\,\log(N)\,\log(N/s))$ samples with an arithmetic complexity of $\mathcal{O}(s\,\log(N)\,\log(N/s))$. In [19], another variant was discussed, where the number of samples is $\mathcal{O}(s\,\log N)\,(\log\log N)^{\mathcal{O}(1)}$ and the arithmetic complexity is $\mathcal{O}(s\,\log^2 N)\,(\log\log N)^{\mathcal{O}(1)}$. Recently in [18], a result was presented for the multivariate case with $\Gamma = \hat{G}_N^d$, where the number of required samples is $\mathcal{O}(s\,\log N)$ for constant d and the arithmetic complexity is $\mathcal{O}(N^d \log^{\mathcal{O}(1)} N)$. In general the exact constants, especially the dependence on d, are unknown due to missing implementations. For instance the sample complexity $\mathcal{O}(s\,\log N)$ of the last mentioned algorithm contains a factor of $d^{\mathcal{O}(d)}$, see [18, Sec. 4].

Moreover, a deterministic sparse Fourier transform algorithm, using the Chinese Remainder Theorem, was presented in [20] for the univariate case and in [21] for the multivariate case, which takes $\mathcal{O}(d^4 s^2 \log^4(dN))$ samples and arithmetic operations. This means there is neither a exponential/super-exponential dependency on the dimension $d \in \mathbb{N}$ nor a dependency on a failure probability in the asymptotics of the number of samples and arithmetic operations for this method. Besides this deterministic algorithm, there also exists a randomized version which only requires $\mathcal{O}(d^4 s \log^4(dN))$ samples and arithmetic operations.

Recently, another one-dimensional sparse Fourier transform algorithm, which is based on a multiscale approach, was presented in [5] as an extension of the method [34]. Their algo-

rithm is able to handle (additive) noise and requires $\mathcal{O}(|\operatorname{supp}\hat{p}|\log|\operatorname{supp}\hat{p}|\log(N/|\operatorname{supp}\hat{p}|))$ on average.

For a given trigonometric polynomial p, the main idea of this paper is a dimension incremental construction of the frequency index set $\operatorname{supp}\hat{p}$ of the non-zero Fourier coefficients $\mathbf{0} \neq \hat{p}_{\boldsymbol{k}}$ of p. This idea is motivated by the component-by-component construction of reconstructing rank-1 lattices. We stress the fact that our method reconstructs first the (projected) Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ and selects then the index set I, whereas all the methods mentioned above determine first the index set I and then the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$. Since we use reconstructing rank-1 lattices for the sampling and one-dimensional FFTs for the computation of the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, the numerical computations are stable. Assuming $\Gamma \subseteq \hat{G}_N^d$, we require $\mathcal{O}(d\,s^2N)$ many samples and $\mathcal{O}(d\,s^3 + d\,s^2N\log(s\,N))$ arithmetic operations in the case $\sqrt{N} \lesssim s \lesssim N^d$ as well as $\mathcal{O}(d\,N^2)$ many samples and $\mathcal{O}(d\,N^2\log N)$ arithmetic operations in the case $s \lesssim \sqrt{N}$, where the asymptotics have no additional dependence on the dimension d. Furthermore, we apply ℓ_1 minimization with sub-sampling on rank-1 lattices and sampling on generated sets [23] using the SPGL1 algorithm [48, 47], which results in a reduction of the number of samples to only $\mathcal{O}(d\,s\log^4(s\,N) + d\,N)$ for $s \geq |\operatorname{supp}\hat{p}|$. When using the ℓ_1 minimization with sampling on generated sets, we obtain a method with an overall arithmetic complexity of $\mathcal{O}\left(d\,R\,s\,N\log(s\,N) + d\,R\,s\log^5(s\,N)\right)$, where $R \in \mathbb{N}$ is the number of iterations for SPGL1. Additionally, we use a version of Prony's method [39] with sub-sampling on rank-1 lattices, and we obtain an algorithm which only requires $\mathcal{O}(d\,s + d\,N)$ many samples for $s \geq \operatorname{supp}\hat{p}$ as well as $\mathcal{O}(d\,s^3)$ and $\mathcal{O}(d\,s\,N + d\,N\log N)$ arithmetic operations in the case $\sqrt{N} \lesssim s \lesssim N^d$ and $s \lesssim \sqrt{N}$, respectively. In numerical examples, we verify the approach for the reconstruction of sparse high-dimensional trigonometric polynomials with frequencies supported within a subset of the full grid $\Gamma = \hat{G}_N^d$ from samples and also use it for the s-sparse ℓ_2 approximation of the Fourier coefficients of a high-dimensional 1-periodic function $f\colon \mathbb{T}^d \to \mathbb{C}$.

The remaining sections of this paper are structured as follows. In Section 2, we discuss the reconstruction of high-dimensional trigonometric polynomials with frequencies supported within a subset of the search space Γ from samples. For this, we briefly explain the fast, exact and perfectly stable reconstruction of trigonometric polynomials with frequencies supported on a known index I with $I \subset \Gamma$ using sampling values at rank-1 lattice nodes in Section 2.1. Using these results, we introduce a method for detecting the frequencies of a trigonometric polynomial belonging to non-zero Fourier coefficients by a dimension incremental method in Section 2.2.1. In Section 2.2.2, we discuss conditions when the frequency detection succeeds and scenarios where it may fail. The number of samples and arithmetic complexity of our approach is given in Section 2.2.3 for the case where the search space Γ is a full grid $\hat{G}_N^d$. As a possibility to reduce the number of samples and the arithmetic complexity, we briefly discuss using ℓ_1 minimization with sub-sampling on rank-1 lattices and generated sets for the dimension incremental reconstruction of trigonometric polynomials from samples in Section 2.3. Moreover, we apply Prony's method with sub-sampling on rank-1 lattices in Section 2.4 in order to reduce the number of samples. We verify the presented methods using numerical tests in Section 3 and particularly in Section 3.3, we approximately reconstruct the largest Fourier coefficients of a 10-dimensional periodic function of dominating mixed smoothness, which has infinitely many non-zero Fourier coefficients. Moreover, we test the robustness to noise in Section 3.4, where we apply the method from Section 2.2.1 to samples of sparse trigonometric polynomials perturbed by white Gaussian noise. Finally, we summarize the results of this paper in Section 4.

2 Reconstruction of trigonometric polynomials

2.1 Reconstructing rank-1 lattices for known frequency index sets I

As discussed in [25], for a given frequency index set $I \subset \mathbb{Z}^d$ of finite cardinality, we are able to exactly reconstruct the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of an arbitrarily chosen trigonometric polynomial $p(\boldsymbol{x}) := \sum_{\boldsymbol{k}\in I} \hat{p}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi \mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$ with frequencies supported on I from sampling values $p(\boldsymbol{x}_j)$. As sampling nodes $\boldsymbol{x}_j$, $j = 0, \ldots, M-1$, we use the nodes of a rank-1 lattice $\Lambda(\boldsymbol{z}, M) := \{\frac{j}{M}\boldsymbol{z} \bmod \boldsymbol{1} \colon j = 0, \ldots, M-1\}$ with generating vector $\boldsymbol{z} \in \mathbb{Z}^d$ of size $M \in \mathbb{N}$, i.e., we set the sampling nodes $\boldsymbol{x}_j := \frac{j}{M}\boldsymbol{z} \bmod \boldsymbol{1}$, $j = 0, \ldots, M-1$. Formally, the Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in \mathbb{C}$ of the trigonometric polynomial p are given by the Fourier transform of p,

$$\hat{p}_{\boldsymbol{k}} := \int_{\mathbb{T}^d} p(\boldsymbol{x})\, \mathrm{e}^{-2\pi \mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{x}, \quad \boldsymbol{k} \in I,$$

and we approximate these integrals by the (rank-1) lattice rule

$$\frac{1}{M}\sum_{j=0}^{M-1} p(\boldsymbol{x}_j) \mathrm{e}^{-2\pi \mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}_j} = \frac{1}{M}\sum_{j=0}^{M-1} p\left(\frac{j}{M}\boldsymbol{z}\right) \mathrm{e}^{-2\pi \mathrm{i} j\boldsymbol{k}\cdot\boldsymbol{z}/M} =: \tilde{\hat{p}}_{\boldsymbol{k}}.$$

Now, we ask for the exactness of this cubature formula, i.e., when is $\hat{p}_{\boldsymbol{k}} = \tilde{\hat{p}}_{\boldsymbol{k}}$ $\forall \boldsymbol{k} \in I$. Since we have

$$\tilde{\hat{p}}_{\boldsymbol{k}} = \frac{1}{M}\sum_{j=0}^{M-1}\sum_{\boldsymbol{k}'\in I} \hat{p}_{\boldsymbol{k}'}\, \mathrm{e}^{2\pi \mathrm{i} j\boldsymbol{k}'\cdot\boldsymbol{z}/M} \mathrm{e}^{-2\pi \mathrm{i} j\boldsymbol{k}\cdot\boldsymbol{z}/M} = \sum_{\boldsymbol{k}'\in I} \hat{p}_{\boldsymbol{k}'} \frac{1}{M}\sum_{j=0}^{M-1} \mathrm{e}^{2\pi \mathrm{i} j(\boldsymbol{k}'-\boldsymbol{k})\cdot\boldsymbol{z}/M},$$

we need the condition

$$\frac{1}{M}\sum_{j=0}^{M-1} \mathrm{e}^{2\pi \mathrm{i} j(\boldsymbol{k}'-\boldsymbol{k})\cdot\boldsymbol{z}/M} = \begin{cases} 1 & \text{for } \boldsymbol{k} = \boldsymbol{k}' \\ 0 & \text{for } \boldsymbol{k} \neq \boldsymbol{k}', \boldsymbol{k}, \boldsymbol{k}' \in I, \end{cases}$$

to be fulfilled. This is the case if and only if $\boldsymbol{k}\cdot\boldsymbol{z} \not\equiv \boldsymbol{k}'\cdot\boldsymbol{z} \pmod{M}$ $\forall \boldsymbol{k}, \boldsymbol{k}' \in I, \boldsymbol{k} \neq \boldsymbol{k}'$, see [25, Section 2]. Introducing the difference set $\mathcal{D}(I)$ for the index set I, $\mathcal{D}(I) := \{\boldsymbol{k} - \boldsymbol{k}' \colon \boldsymbol{k}, \boldsymbol{k}' \in I\}$, we can rewrite the above conditions as

$$\boldsymbol{m}\cdot\boldsymbol{z} \not\equiv 0 \pmod{M} \; \forall \boldsymbol{m} \in \mathcal{D}(I) \setminus \{\boldsymbol{0}\}. \tag{2.1}$$

A rank-1 lattice $\Lambda(\boldsymbol{z}, M)$ which fulfills the *reconstruction property* (2.1) for a given frequency index set I will be called *reconstructing rank-1 lattice* $\Lambda(\boldsymbol{z}, M, I)$ *for* I.

We remark that an arbitrarily chosen trigonometric polynomial p with frequencies supported on the index set I can be quickly evaluated at all nodes of an arbitrary rank-1 lattice $\Lambda(\boldsymbol{z}, M)$ in $\mathcal{O}(M \log M + d|I|)$ arithmetic operations using a single one-dimensional fast Fourier transform, cf. [35]. Moreover, the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of p can be exactly reconstructed from sampling values of p at the nodes $\boldsymbol{x}_j := \frac{j}{M}\boldsymbol{z} \bmod \boldsymbol{1}$, $j = 0, \ldots, M-1$, of a reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M, I)$ for I in $\mathcal{O}(M \log M + d|I|)$ arithmetic operations as discussed in [25]. For this, we compute

$$\hat{p}_\ell := \frac{1}{M}\sum_{j=0}^{M-1} p\left(\frac{j}{M}\boldsymbol{z} \bmod \boldsymbol{1}\right) \mathrm{e}^{-2\pi \mathrm{i} j\ell/M}, \quad \ell = 0, \ldots, M-1,$$

using a single one-dimensional inverse fast Fourier transform of length M and we set $\hat{p}_{\boldsymbol{k}} := \hat{p}_{\boldsymbol{k}\cdot\boldsymbol{z} \bmod M}$ for $\boldsymbol{k} \in I$, i.e., we additionally compute the scalar products $\boldsymbol{k}\cdot\boldsymbol{z}$ for $\boldsymbol{k} \in I$. These two computation steps will be called *inverse rank-1 lattice FFT* in the following.

Theorem 2.1. *For a given frequency index set $I \subset \mathbb{Z}^d$, $1 \leq |I| < \infty$, and any prime rank-1 lattice size*

$$M \geq \max\left\{\frac{|\mathcal{D}(I)|+3}{2}, \max\{2\|\boldsymbol{k}\|_\infty + 1 \colon \boldsymbol{k} \in I\}\right\}, \tag{2.2}$$

there always exists a generating vector $\boldsymbol{z} \in \mathbb{Z}^d$ such that $\Lambda(\boldsymbol{z}, M) = \Lambda(\boldsymbol{z}, M, I)$ is a reconstructing rank-1 lattice for I. Moreover, there always exist a prime rank-1 lattice size M,

$$\begin{aligned} |I| \leq M &\leq \max\left\{\frac{2}{3}(|\mathcal{D}(I)|+7), \max\{3\|\boldsymbol{k}\|_\infty \colon \boldsymbol{k} \in I\}\right\} \\ &\leq \max\left\{\frac{2}{3}(|I|^2 - |I| + 8), \max\{3\|\boldsymbol{k}\|_\infty \colon \boldsymbol{k} \in I\}\right\}, \end{aligned} \tag{2.3}$$

and a generating vector $\boldsymbol{z} \in \mathbb{Z}^d$ such that $\Lambda(\boldsymbol{z}, M) = \Lambda(\boldsymbol{z}, M, I)$ is a reconstructing rank-1 lattice for I.
For such a suitable rank-1 lattice size M, the generating vector $\boldsymbol{z} \in \mathbb{Z}^d$ can be constructed using a component-by-component approach, see [25], and the construction requires no more than $3\,d\,|I|M$ arithmetic operations.

Proof. The inequality (2.3) is a consequence of [25, Corollary 1] and [24, inequality (3.8)]. The lower bound for the rank-1 lattice size M is a consequence from [25, Theorem 1 and Lemma 2].
When searching for the component z_t, $t \in \{1, \ldots, d\}$, of the generating vector $\boldsymbol{z} := (z_1, \ldots, z_d)^\top$ in the component-by-component step t, the tests for the reconstruction property (2.1) for a given component z_t take no more than $|I|$ multiplications, $|I|$ additions as well as $|I|$ modulo operations, and this yields $3\,|I|$ many arithmetic operations. Due to this and since each component z_t, $t \in \{1, \ldots, d\}$, of the generating vector $\boldsymbol{z}$ can only have M different values modulo M, we obtain that the construction requires no more than $3\,d\,|I|M$ arithmetic operations in total.

■

The following Theorem is stated and proven in [25].

Theorem 2.2. *Let a dimension $d \in \mathbb{N}$, $d \geq 2$, and a frequency index set $I \subset \mathbb{Z}^d$ of finite cardinality $|I| \geq 2$ be given. We assume that $\Lambda(\boldsymbol{z}, M) = \Lambda(\boldsymbol{z}, M, I^{(1,\ldots,d-1)})$ with generating vector $\boldsymbol{z} := (z_1, \ldots, z_{d-1})^\top$ is a reconstructing rank-1 lattice for the frequency index set $I^{(1,\ldots,d-1)} := \{(k_s)_{s=1}^{d-1} \colon \boldsymbol{k} \in I\}$. Then, the rank-1 lattice $\Lambda((z_1, \ldots, z_{d-1}, M)^\top, MS)$ with*

$$S := \min\{m \in \mathbb{N} \colon |\{k_d \bmod m \colon \boldsymbol{k} \in I\}| = |\{k_d \colon \boldsymbol{k} \in I\}|\}$$

is a reconstructing rank-1 lattice for I.

Corollary 2.3. *Let a frequency set $I' \subset \hat{G}_N^d$, $|I'| = s \geq 2$, be given. Furthermore, let $I'' \subset \hat{G}_N^1$ be another non-empty frequency index set. Then, there exists a reconstructing rank-1 lattice for $I' \times I''$ of size $M \leq \max\{2s^2, 3N\}\,2(N+1)$.*

Proof. Due to (2.3) in Theorem 2.1, there exists a reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M', I')$ for I' with generating vector $\boldsymbol{z} := (z_1, \ldots, z_{d-1})^\top$ and size

$$M' \leq \max\left\{\frac{2}{3}(s^2 - s + 8), 3N\right\} \leq \max\{2s^2, 3N\}.$$

We apply Theorem 2.2 with $I := I' \times I''$. Consequently, $I^{(1,\ldots,d-1)} := \{(k_s)_{s=1}^{d-1} \colon \boldsymbol{k} \in I\} = I'$ and $\{k_d \colon \boldsymbol{k} \in I\} = I''$ in Theorem 2.2. Since we have $S \leq \max(I') - \min(I') + 1 \leq 2(N+1)$, the rank-1 lattice $\Lambda((z_1, \ldots, z_{d-1}, M')^\top, M'S)$ is a reconstructing rank-1 lattice for $I = I' \times I''$ of size $M'S \leq \max\{2s^2, 3N\}\, 2(N+1)$. ■

2.2 Dimension incremental reconstruction in the multidimensional case $d \geq 2$

In this subsection, we consider multi-dimensional trigonometric polynomials $p : \mathbb{T}^d \to \mathbb{C}$ with frequencies supported on a subset I of the index set $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, and we are going to determine the support $\operatorname{supp} \hat{p}$ of a trigonometric polynomial p in frequency domain as well as the non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp} \hat{p}$, of p from sampling values. For this, we repeatedly use reconstructing rank-1 lattices introduced in Section 2.1 as sampling nodes in order to find the support $\operatorname{supp} \hat{p}$ of the trigonometric polynomial p in frequency domain from sampling values in a dimension incremental way. We remark that if Γ is a small subset of the full grid $\hat{G}_N^d$, $|\Gamma| \ll |\hat{G}_N^d|$, and this fact is known, then this knowledge can be used to reduce the number of samples and the arithmetic complexity of our method which is described in the following sub-section.

2.2.1 The method

Recently, a dimension incremental method for anharmonic trigonometric polynomials based on Prony's method was presented in [38]. Here, we proceed similarly. We denote the projection of a frequency $\boldsymbol{k} := (k_1, \ldots, k_d)^\top \in \mathbb{Z}^d$ to the components $\boldsymbol{i} := (i_1, \ldots, i_m) \in \{1, \ldots, d\}^m$ by $\mathcal{P}_{\boldsymbol{i}}(\boldsymbol{k}) := (k_{i_1}, \ldots, k_{i_m})^\top \in \mathbb{Z}^m$. Correspondingly, we define the projection of a frequency index set $I \subset \mathbb{Z}^d$ to the components $\boldsymbol{i}$ by $\mathcal{P}_{\boldsymbol{i}}(I) := \{(k_{i_1}, \ldots, k_{i_m}) \colon \boldsymbol{k} \in I\}$. Using this notation, the general approach is the following:

1. Determine an index set $I^{(1)} \subseteq \mathcal{P}_1(\Gamma)$ which should be identical to the projection $\mathcal{P}_1(\operatorname{supp} \hat{p})$ or contain this projection, $I^{(1)} \supseteq \mathcal{P}_1(\operatorname{supp} \hat{p})$. If $\Gamma = \hat{G}_N^d$, then $\mathcal{P}_1(\Gamma) = \hat{G}_N^1$ and $I^{(1)} \subseteq \hat{G}_N^1$.

2. For dimension increment step $t = 2, \ldots, d$
   a) Determine an index set $I^{(t)} \subseteq \mathcal{P}_t(\Gamma)$ which should be identical to the projection $\mathcal{P}_t(\operatorname{supp} \hat{p})$ or contain this projection, $I^{(t)} \supseteq \mathcal{P}_t(\operatorname{supp} \hat{p})$. If $\Gamma = \hat{G}_N^d$, then $\mathcal{P}_t(\Gamma) = \hat{G}_N^1$ and $I^{(t)} \subseteq \hat{G}_N^1$.
   b) Determine a suitable sampling set $\mathcal{X}^{(1,\ldots,t)} \subset \mathbb{T}^d$, $|\mathcal{X}^{(1,\ldots,t)}| \ll |\Gamma|$, which allows to determine those frequencies from the index set $(I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ belonging to non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}}$.
   c) Sample the trigonometric polynomial p along the nodes of the sampling set $\mathcal{X}^{(1,\ldots,t)}$.
   d) Compute the Fourier coefficients $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$.
   e) Determine the non-zero Fourier coefficients from $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, and obtain the index set $I^{(1,\ldots,t)}$ of detected frequencies. Alternatively, determine those Fourier coefficients from $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$ which are larger than a certain threshold. The $I^{(1,\ldots,t)}$ index set should be equal to the projection $\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p})$.

3. Use the index set $I^{(1,\ldots,d)}$ and the computed Fourier coefficients $\tilde{\hat{p}}_{(1,\ldots,d),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,d)}$, as an approximation for the support $\operatorname{supp} \hat{p}$ and the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp} \hat{p}$.

There exist different methods for the realization of the steps 2b and 2d. In the following, we present two possible methods in detail as Algorithm 1 and 2.

Algorithm 1

Algorithm 1 is a realization for this method, which uses one-dimensional inverse fast Fourier transforms (1d iFFTs). Besides the search space $\Gamma \supset \operatorname{supp} \hat{p}$ and the trigonometric polynomial p (as black box), this algorithm has three additional input parameters, which are the relative threshold $\theta \in (0,1)$, the sparsity $s \in \mathbb{N}$ and the number of detection iterations $r \in \mathbb{N}$. The relative threshold parameter $\theta \in (0,1)$ is used to determine the "non-zero" Fourier coefficients from $\tilde{\hat{p}}_{1,k_1}$ for $k_1 \in \mathcal{P}_1(\Gamma)$ in step 1, $\tilde{\hat{p}}_{t,k_t}$ for $k_t \in \mathcal{P}_t(\Gamma)$ in step 2a, $t \in \{2,\dots,d\}$, as well as $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$ for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$ in step 2e. Since numerical algorithms are used to compute the Fourier coefficients $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$, the actual computed values of "zero" Fourier coefficients may be larger than zero but are smaller than a certain (relative) threshold. The sparsity input parameter $s \in \mathbb{N}$ may be used to truncate the number of detected frequencies and corresponding Fourier coefficients. Last, the input parameter $r \in \mathbb{N}$ for the number of detection iterations controls how many times the sampling and frequency detection in the step 2 is performed for each dimension increment step $t \in \{2,\dots,d\}$. Repetitions in these computations r times may be necessary to ensure a successful exact reconstruction of the trigonometric polynomial p, as we describe in this section and in Section 2.2.2.

First in step 1, we determine the index set of detected frequencies for the first component $I^{(1)} \subset \mathcal{P}_1(\operatorname{supp} \hat{p})$. For this, we set the last $d-1$ components in $\boldsymbol{x} := (x_1,\dots,x_d)^\top$ to fixed randomly chosen values $x'_2,\dots,x'_d \in \mathbb{T}$. We sample the trigonometric polynomial p at the nodes of the set $\mathcal{X}^{(1)} := \{(\frac{\ell}{L_1}, x'_2,\dots,x'_d)^\top : \ell = 0,\dots,L_1-1\}$, where $L_1 := \max(\mathcal{P}_1(\Gamma)) - \min(\mathcal{P}_1(\Gamma)) + 1$, and we compute one-dimensional projected Fourier coefficients for the first component

$$\tilde{\hat{p}}_{1,k_1} := \frac{1}{L_1} \sum_{\ell=0}^{L_1-1} p\left(\left(\frac{\ell}{L_1}, x'_2,\dots,x'_d\right)^\top\right) \mathrm{e}^{-2\pi \mathrm{i} \frac{\ell k_1}{L_1}}, \quad k_1 \in \mathcal{P}_1(\Gamma),$$

using a 1d iFFT of length L_1. Due to the definition of the trigonometric polynomial p, we obtain

$$\begin{aligned}
\tilde{\hat{p}}_{1,k_1} &= \frac{1}{L_1} \sum_{\ell=0}^{L_1-1} \sum_{\boldsymbol{h}:=(h_1,\dots,h_d)^\top \in \operatorname{supp} \hat{p}} \hat{p}_{\boldsymbol{h}} \, \mathrm{e}^{2\pi \mathrm{i}(h_2 x'_2 + \dots + h_d x'_d)} \, \mathrm{e}^{2\pi \mathrm{i} \frac{(h_1-k_1)\ell}{L_1}} \\
&= \sum_{\boldsymbol{h} \in \operatorname{supp} \hat{p}} \hat{p}_{\boldsymbol{h}} \, \mathrm{e}^{2\pi \mathrm{i}(h_2 x'_2 + \dots + h_d x'_d)} \, \frac{1}{L_1} \sum_{\ell=0}^{L_1-1} \mathrm{e}^{2\pi \mathrm{i} \frac{(h_1-k_1)\ell}{L_1}} \\
&= \sum_{\substack{(h_2,\dots,h_d) \in \mathcal{P}_{(2,\dots,d)}(\Gamma) \\ (k_1,h_2,\dots,h_d)^\top \in \operatorname{supp} \hat{p}}} \hat{p}_{(k_1,h_2,\dots,h_d)^\top} \, \mathrm{e}^{2\pi \mathrm{i}(h_2 x'_2 + \dots + h_d x'_d)}
\end{aligned}$$

for $k_1 \in \mathcal{P}_1(\Gamma)$. We define the index set of detected frequencies for the first component $I^{(1)} := \{k_1 \in \mathcal{P}_1(\Gamma) : \tilde{\hat{p}}_{1,k_1} \neq 0\}$. In practice, we do not test if the Fourier coefficients $\tilde{\hat{p}}_{1,k_1} \neq 0$, but use a threshold $\theta \in (0,1)$ relative to the largest absolute value of the computed Fourier coefficients $\tilde{\hat{p}}_{1,k_1}$ in numerical computations and we restrict the number of detected

Algorithm 1 Reconstruction of a trigonometric polynomial p from sampling values.

Input:		
	$\Gamma \subset \mathbb{Z}^d$	search space in frequency domain, superset for $\operatorname{supp} \hat{p}$
	$p(\circ)$	trigonometric polynomial p as black box (function handle)
	$\theta \in (0,1)$	relative threshold
	$s \in \mathbb{N}$	sparsity
	$r \in \mathbb{N}$	number of detection iterations

(step 1)

Set $L_1 := \max(\mathcal{P}_1(\Gamma)) - \min(\mathcal{P}_1(\Gamma)) + 1$, $I^{(1)} := \emptyset$.

for $i := 1, \dots, r$ ***do***

Choose $x'_2, \dots, x'_d \in \mathbb{T}$ uniformly at random.

Compute $\tilde{\hat{p}}_{1,k_1} := \frac{1}{L_1} \sum_{\ell=0}^{L_1-1} p\left((\frac{\ell}{L_1}, x'_2, \dots, x'_d)^\top \right) \mathrm{e}^{-2\pi \mathrm{i} \frac{\ell k_1}{L_1}}$, $k_1 \in \mathcal{P}_1(\Gamma)$, with 1d iFFT.

$I^{(1)} := I^{(1)} \cup \{k_1 \in \mathcal{P}_1(\Gamma)\colon$ (up to) s-largest values $|\tilde{\hat{p}}_{1,k_1}| \geq \theta \cdot \max_{\tilde{k}_1 \in \mathcal{P}_1(\Gamma)} |\tilde{\hat{p}}_{1,\tilde{k}_1}|\}$

end for i

Determine $S_1 := \min\left\{m \in \mathbb{N}\colon |\{k_1 \bmod m\colon k_1 \in I^{(1)}\}| = |I^{(1)}|\right\}$. Set $M_1 := S_1$, $z_1 := 1$.

(step 2) ***for*** $t := 2, \dots, d$ ***do***

(step 2a)

Set $L_t := \max(\mathcal{P}_t(\Gamma)) - \min(\mathcal{P}_t(\Gamma)) + 1$, $I^{(t)} := \emptyset$.

for $i := 1, \dots, r$ ***do***

Choose $x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d \in \mathbb{T}$ uniformly at random.

$\tilde{\hat{p}}_{t,k_t} := \sum_{\ell=0}^{L_t-1} p(x'_1, \dots, x'_{t-1}, \frac{\ell}{L_t}, x'_{t+1}, \dots, x'_d)^\top \mathrm{e}^{-2\pi \mathrm{i} \ell k_t / L_t}$, $k_t \in \mathcal{P}_t(\Gamma)$, using 1d iFFT.

Set $I^{(t)} := I^{(t)} \cup \{k_t \in \mathcal{P}_t(\Gamma)\colon$ (up to) s-largest values $|\tilde{\hat{p}}_{t,k_t}| \geq \theta \cdot \max_{\tilde{k}_t \in \mathcal{P}_t(\Gamma)} |\tilde{\hat{p}}_{t,\tilde{k}_t}|\}$.

end for i

(step 2b) Set $\tilde{r} := \begin{cases} r & \text{for } t < d, \\ 1 & \text{for } t = d. \end{cases}$

Determine $S_t := \min\left\{m \in \mathbb{N}\colon |\{k_t \bmod m\colon k_t \in I^{(t)}\}| = |I^{(t)}|\right\}$. Set $I^{(1,\dots,t)} := \emptyset$.

Search for reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M_t, (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma))$, $\boldsymbol{z} \in \mathbb{Z}^t$:

Set initial $M_t := M_{t-1} \cdot S_t$, cf. Theorem 2.2.

Search for $z_t \in \{0, \dots, M_t - 1\}$ such that reconstruction property (2.1) is fulfilled.

Reduce rank-1 lattice size M_t using [24, Algorithm 3.5].

for $i := 1, \dots, \tilde{r}$ ***do***

Choose $x'_{t+1}, \dots, x'_d \in \mathbb{T}$ uniformly at random.

Set $\mathcal{X}^{(1,\dots,t)} := \{\boldsymbol{x}_j := (\frac{j}{M_t} z_1, \dots, \frac{j}{M_t} z_t, x'_{t+1}, \dots, x'_d)^\top \bmod \mathbf{1}\colon j = 0, \dots, M_t - 1\}$.

(step 2c) Sample p along the nodes of the sampling set $\mathcal{X}^{(1,\dots,t)}$.

(step 2d)

Compute $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} := \frac{1}{M_t} \sum_{j=0}^{M_t-1} p(\boldsymbol{x}_j) \, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}_j}$ for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$ with inverse rank-1 lattice FFT based on a single 1d iFFT, see Section 2.1.

(step 2e)

absolute_threshold $:= \theta \cdot \max_{\tilde{\boldsymbol{k}} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)} |\tilde{\hat{p}}_{(1,\dots,t),\tilde{\boldsymbol{k}}}|$.

Set $I^{(1,\dots,t)} := I^{(1,\dots,t)} \cup \{\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)\colon$

(up to) s-largest values $|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| \geq$ absolute_threshold$\}$.

end for i

Algorithm 1 continued.

(additional step 2f)

If $t < d$, search for reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M_t, I^{(1,\dots,t)})$:

Search for new $z_t \in \{0, \dots, M_t - 1\}$ such that reconstruction property (2.1) is fulfilled.

Reduce rank-1 lattice size M_t.

end for t

(step 3) Set $I := I^{(1,\dots,d)}$ and $\tilde{\hat{\boldsymbol{p}}} := \left(\tilde{\hat{p}}_{(1,\dots,d),\boldsymbol{k}}\right)_{\boldsymbol{k}\in I}$.

Output:	$I \subset \mathbb{Z}^d$	index set of detected frequencies
	$\tilde{\hat{\boldsymbol{p}}} \in \mathbb{C}^{\lvert I\rvert}$	corresponding Fourier coefficients

frequencies to the sparsity s, i.e.,

$$I^{(1)} := I^{(1)} \cup \{k_1 \in \mathcal{P}_1(\Gamma)\colon \text{(up to) } s\text{-largest values } |\tilde{\hat{p}}_{1,k_1}| \geq \theta \cdot \max_{\tilde{k}_1 \in \mathcal{P}_1(\Gamma)} |\tilde{\hat{p}}_{1,\tilde{k}_1}|\}. \tag{2.4}$$

Since this frequency detection may fail, see Section 2.2.2 for details, we repeatedly perform the sampling, the computation of the projected Fourier coefficients $\tilde{\hat{p}}_{1,k_1}$, $k_1 \in \mathcal{P}_1(\Gamma)$, and the determination of the index set $I^{(1)}$ in totally $r \in \mathbb{N}$ detection iterations with different randomly chosen values $x'_2, \dots, x'_d \in \mathbb{T}$. Then, we use the union of the obtained index sets $I^{(1)}$. We determine $S_1 := \min\left\{m \in \mathbb{N}\colon |\{k_1 \bmod m\colon k_1 \in I^{(1)}\}| = |I^{(1)}|\right\}$ and obtain a reconstructing rank-1 lattice $\Lambda(z_1, M_1, I^{(1)})$ for the index set of detected frequencies for the first component $I^{(1)}$ by setting $z_1 := 1$ and $M_1 := S_1$.

Then, we continue with the dimension increment step 2 for $t = 2, \dots, d$. In step 2a, we randomly choose values $x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d \in \mathbb{T}$, we determine $L_t := \max(\mathcal{P}_t(\Gamma)) - \min(\mathcal{P}_t(\Gamma)) + 1$ and we compute the one-dimensional projected Fourier coefficients for the t-th component

$$\tilde{\hat{p}}_{t,k_t} := \frac{1}{L_t} \sum_{\ell=0}^{L_t-1} p\left(\left(x'_1, x'_{t-1}, \frac{\ell}{L_t}, x'_{t+1}, \dots, x'_d\right)^\top\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{\ell k_t}{L_t}} \tag{2.5}$$

$$= \sum_{\substack{(h_1,\dots,h_{t-1},h_{t+1},\dots,h_d)^\top \in \mathcal{P}_{(1,\dots,t-1,t+1,\dots,d)}(\Gamma) \\ (h_1,\dots,h_{t-1},k_t,h_{t+1},\dots,h_d)^\top \in \operatorname{supp}\hat{p}}} \hat{p}_{(h_1,\dots,h_{t-1},k_t,h_{t+1},\dots,h_d)^\top} \cdot \mathrm{e}^{2\pi\mathrm{i}(h_1x'_1+\dots+h_{t-1}x'_{t-1}+h_{t+1}x'_{t+1}+\dots+h_dx'_d)} \tag{2.6}$$

for $k_t \in \mathcal{P}_t(\Gamma)$, using a 1d iFFT of length L_t. Similarly as in step 1, we obtain $r \in \mathbb{N}$ many index sets of detected frequencies for the t-th component $\{k_t \in \mathcal{P}_t(\Gamma)\colon \text{(up to) } s\text{-largest values } |\tilde{\hat{p}}_{t,k_t}| \geq \theta \cdot \max_{\tilde{k}_t \in \mathcal{P}_t(\Gamma)} |\tilde{\hat{p}}_{t,\tilde{k}_t}|\}$ in $r \in \mathbb{N}$ detection iterations with different randomly chosen values $x'_1, \dots, x'_{t-1}, \dots, x'_{t+1}, \dots, x'_d \in \mathbb{T}$ and we set the union of these sets as the index set $I^{(t)}$.

Afterwards in step 2b, we determine $S_t := \min\left\{m \in \mathbb{N}\colon |\{k_t \bmod m\colon k_t \in I^{(t)}\}| = |I^{(t)}|\right\}$ and we search for a reconstructing rank-1 lattice for the index set $(I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$. For this, the initial rank-1 lattice size M_t is set to $M_{t-1} \cdot S_t$, cf. Theorem 2.2. The components $z_1, \dots, z_{t-1}$ of the generating vector $\boldsymbol{z}$ from the previous dimension increment steps $1, \dots, t-1$ are re-used and only one component $z_t \in \{0, \dots, M_t - 1\}$ is searched for, such that reconstruction property (2.1) is fulfilled. Next, the rank-1 lattice size

M_t is reduced using [24, Algorithm 3.5]. We set the sampling set $\mathcal{X}^{(1,\dots,t)} := \{\boldsymbol{x}_j := (\frac{j}{M_t} z_1, \dots, \frac{j}{M_t} z_t, x'_{t+1}, \dots, x'_d)^\top \bmod \mathbf{1} \colon j = 0, \dots, M_t - 1\}$ containing the sampling nodes $\boldsymbol{x}_j$ with fixed randomly chosen values $x'_{t+1}, \dots, x'_d \in \mathbb{T}$. Then, we sample the trigonometric polynomial p at these nodes $\boldsymbol{x}_j$ in step 2c. Next, we compute t-dimensional projected Fourier coefficients for the first t components

$$\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} := \frac{1}{M_t} \sum_{j=0}^{M_t-1} p(\boldsymbol{x}_j)\, \mathrm{e}^{-2\pi \mathrm{i} \frac{j \boldsymbol{k} \cdot \boldsymbol{z}}{M_t}} \tag{2.7}$$

for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$ in step 2d using an inverse rank-1 lattice FFTs, see Section 2.1. This means, we use only a single 1d iFFT. Note that we have

$$\begin{aligned} \tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} &= \sum_{\boldsymbol{h} \in \operatorname{supp} \hat{p}} \left(\hat{p}_{\boldsymbol{h}}\, \mathrm{e}^{2\pi \mathrm{i} (h_{t+1}, \dots, h_d)^\top \cdot (x'_{t+1}, \dots, x'_d)^\top} \right) \left(\frac{1}{M_t} \sum_{j=0}^{M_t-1} \mathrm{e}^{2\pi \mathrm{i} \frac{j (\boldsymbol{h} - \boldsymbol{k}) \cdot \boldsymbol{z}}{M_t}} \right) \\ &= \sum_{\substack{\boldsymbol{h} \in \operatorname{supp} \hat{p} \\ (\boldsymbol{h} - \boldsymbol{k}) \cdot \boldsymbol{z} \equiv 0 \pmod{M_t}}} \hat{p}_{\boldsymbol{h}}\, \mathrm{e}^{2\pi \mathrm{i} (h_{t+1}, \dots, h_d)^\top \cdot (x'_{t+1}, \dots, x'_d)^\top}. \end{aligned} \tag{2.8}$$

If the conditions $I^{(1,\dots,t-1)} = \mathcal{P}_{(1,\dots,t-1)}(\operatorname{supp} \hat{p})$ and $I^{(t)} = \mathcal{P}_t(\operatorname{supp} \hat{p})$ are fulfilled, then

$$\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = \begin{cases} \displaystyle\sum_{\substack{(h_{t+1}, \dots, h_d)^\top \in \mathcal{P}_{(t+1,\dots,d)}(\operatorname{supp} \hat{p}) \\ (k_1, \dots, k_t, h_{t+1}, \dots, h_d)^\top \in \operatorname{supp} \hat{p}}} \hat{p}_{(k_1, \dots, k_t, h_{t+1}, \dots, h_d)^\top}\, \mathrm{e}^{2\pi \mathrm{i} (h_{t+1} x'_{t+1} + \dots + h_d x'_d)}, & t < d, \\ \hat{p}_{\boldsymbol{k}}, & t = d, \end{cases}$$

for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$ and $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = 0$ for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \big(\mathcal{P}_{(1,\dots,t)}(\Gamma) \setminus \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})\big)$.
In step 2e, we determine the index set of detected frequencies for the first t components $\tilde{I}^{(1,\dots,t)} := \{\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma) \colon |\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| \geq \theta \cdot \max_{\tilde{\boldsymbol{k}} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)} |\tilde{\hat{p}}_{(1,\dots,t),\tilde{\boldsymbol{k}}}|\}$. If the cardinality $|\tilde{I}^{(1,\dots,t)}|$ is larger than the sparsity parameter s, $s \geq |\operatorname{supp} \hat{p}|$, we restrict the index set in $\tilde{I}^{(1,\dots,t)}$ to frequencies $\boldsymbol{k}$ belonging to the s-largest values $|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}|$. We repeatedly perform the sampling, the computation of the projected Fourier coefficients and the determination of the index sets $\tilde{I}^{(1,\dots,t)}$ in total for $r \in \mathbb{N}$ detection iterations if $t < d$ and $r = 1$ detection iteration if $t = d$. Afterwards, we use the union of the obtained index sets $\tilde{I}^{(1,\dots,t)}$ as $I^{(1,\dots,t)}$.
In the additional step 2f if $t < d$, we build reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M_t, I^{(1,\dots,t)})$ for the index set $I^{(1,\dots,t)}$. As initial rank-1 lattice size, we use the value M_t from step 2b. We only search for one component z_t of the generating vector $\boldsymbol{z}$ as in step 2b and then reduce the rank-1 lattice size M_t.

Finally in step 3, we obtain the index set $I = I^{(1,\dots,d)}$. If all frequency detections were successful, i.e.,

$$\begin{aligned} I^{(t)} &= \mathcal{P}_t(\operatorname{supp} \hat{p}) && \text{for } t = 1, \dots, d && \text{and} \\ I^{(1,\dots,t)} &= \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p}) && \text{for } t = 1, \dots, d-1, \end{aligned}$$

then we have

$$\begin{aligned} I = I^{(1,\dots,d)} &= \operatorname{supp} \hat{p}, \\ \tilde{\hat{p}}_{(1,\dots,d),\boldsymbol{k}} &= \hat{p}_{\boldsymbol{k}} \neq 0 \qquad \text{for all } \boldsymbol{k} \in I^{(1,\dots,d)} \qquad \text{and} \\ p &= \sum_{\boldsymbol{k} \in I^{(1,\dots,d)}} \tilde{\hat{p}}_{(1,\dots,d),\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \circ}. \end{aligned}$$

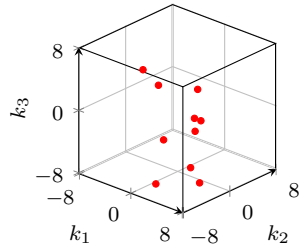
Note that we do not necessarily have $I^{(1,\dots,d)} \subset \operatorname{supp} \hat{p}$, i.e., the algorithm may wrongly yield frequencies where the corresponding Fourier coefficients are zero, see case iv in Section 2.2.2 and the discussion concerning this case.

In Figure 2.1, the sampling sets, the frequency index sets and the reconstructing rank-1 lattices for a three-dimensional example with a trigonometric polynomial p with $|\operatorname{supp} \hat{p}| = 10$ frequencies, $\operatorname{supp} \hat{p} \subset \Gamma := \hat{G}_8^3$, are depicted for $r = 1$ detection iteration. The support $\operatorname{supp} \hat{p}$ is shown in Figure 2.1a.

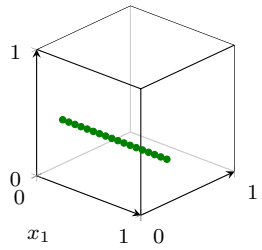
- Step 1: First in step 1, the parameter $L_1 = 17$ is determined, the sampling set $\mathcal{X}^{(1)} := \{(\frac{0}{17}, x_2', x_3')^\top, \dots, (\frac{16}{17}, x_2', x_3')^\top\}$ with randomly chosen points $x_2', x_3' \in \mathbb{T}$ is constructed, see Figure 2.1b, and the trigonometric polynomial p is sampled at the nodes of the set $\mathcal{X}^{(1)}$. Then, a 1d iFFT is applied and the index set of detected frequencies for the first component $I^{(1)}$, as depicted in Figure 2.1c, is determined from the resulting Fourier coefficients. In this example, this means obtaining $S_1 = 9$, $z_1 := 1$ as well as $M_1 := S_1 = 9$, which yields the reconstructing rank-1 lattice $\Lambda(z = 1, M = 9, I^{(1)})$.

- Step 2, $t = 2$: Correspondingly in step 2a for $t = 2$, the parameter $L_2 = 17$ is determined, the sampling set $\mathcal{X}^{(2)} := \{(x_1', \frac{0}{17}, x_3')^\top, \dots, (x_1', \frac{16}{17}, x_3')^\top\}$ is constructed with randomly chosen points $x_1', x_3' \in \mathbb{T}$, see Figure 2.1d, and the index set of detected frequencies for the second component $I^{(2)}$, as depicted in Figure 2.1e, is determined. Next in step 2b, the parameter $S_2 = 11$ is obtained and the index set of frequency candidates $I^{(1)} \times I^{(2)}$ is built, see Figure 2.1f, which has the cardinality $|I^{(1)} \times I^{(2)}| = 49$ for our example. We search for a reconstructing rank-1 lattice for the index set $I^{(1)} \times I^{(2)}$. For this, the initial rank-1 lattice size is set to the $M_2 := M_1 \cdot S_2$, in our example $M_2 := 9 \cdot 11 = 99$. For the generating vector $\boldsymbol{z}$, the component $z_1 = 1$ is used and the component $z_2 \in \{0, \dots, 98\}$ is searched for. In our example, we obtain $z_2 = 9$. Next, the algorithm tries to reduce the rank-1 lattice size M_2, which is not possible for this example, and we obtain $M_2 := 99$. Then, a random point $x_3' \in \mathbb{T}$ is chosen and the sampling set $\mathcal{X}^{(1,2)} := \Lambda(\boldsymbol{z} = (1, 9)^\top, M = 99, I^{(1)} \times I^{(2)}) \times \{x_3'\}$ is constructed as shown in Figure 2.1g, which consists of $|\mathcal{X}^{(1,2)}| = 99$ nodes, and then in step 2c, the trigonometric polynomial p is sampled at these nodes. The Fourier coefficients $\tilde{\hat{p}}_{(1,2),\boldsymbol{k}}$, $\boldsymbol{k} \in I^{(1)} \times I^{(2)}$, are computed from the sampling values in step 2d using a single 1d iFFT and the index set of detected frequencies for the first two components $I^{(1,2)}$ is determined, see Figure 2.1h, which yields $|I^{(1,2)}| = 10$ frequencies. Afterwards in the additional step 2f, a reconstructing rank-1 lattice for the index set $I^{(1,2)}$ is searched, i.e., a component z_2 as well as a new rank-1 lattice size M_2, and this yields the reconstructing rank-1 lattice $\Lambda(\boldsymbol{z} = (1, 4)^\top, M = 23, I^{(1,2)})$ in our example, i.e., $z_2 = 4$ and $M_2 = 23$.

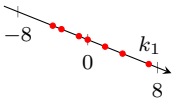
- Step 2, $t = 3$: Next, step 2a is started for $t = 3$, where the parameter $L_3 = 17$ is obtained, and the sampling set $\mathcal{X}^{(3)} := \{(x_1', x_2', \frac{0}{17})^\top, \dots, (x_1', x_2', \frac{16}{17})^\top\}$ is constructed



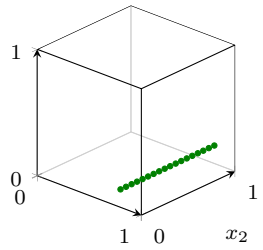
(a) $\operatorname{supp} \hat{p}$, $|\operatorname{supp} \hat{p}| = 10$.



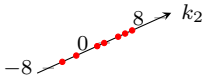
(b) $\mathcal{X}^{(1)}$, $|\mathcal{X}^{(1)}| = 17$ (step 1).



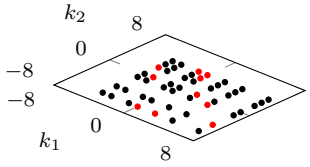
(c) $I^{(1)} \subseteq \mathcal{P}_1(\Gamma)$ (step 1).



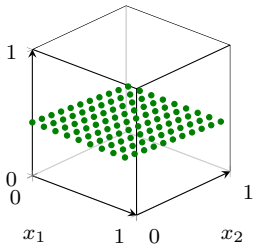
(d) $\mathcal{X}^{(2)}$, $|\mathcal{X}^{(2)}| = 17$ ($t = 2$, step 2a).



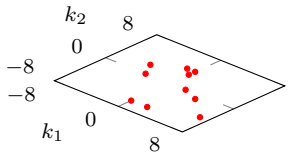
(e) $I^{(2)} \subseteq \mathcal{P}_2(\Gamma)$ ($t = 2$, step 2a).



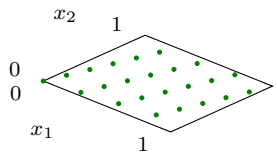
(f) $I^{(1)} \times I^{(2)}$, $|I^{(1)} \times I^{(2)}| = 49$ ($t = 2$, step 2b).



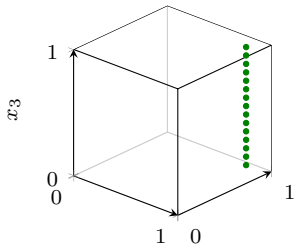
(g) $\mathcal{X}^{(1,2)}$, $|\mathcal{X}^{(1,2)}| = 99$ $\boldsymbol{z} = (1, 9)^\top$, $M = 99$ ($t = 2$, step 2b).



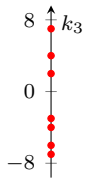
(h) $I^{(1,2)}$, $|I^{(1,2)}| = 10$ ($t = 2$, step 2e).



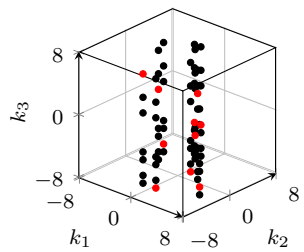
(i) $\Lambda(\boldsymbol{z}, M, I^{(1,2)})$, $\boldsymbol{z} = (1, 4)^\top$, $M = 23$ ($t = 2$, step 2f).



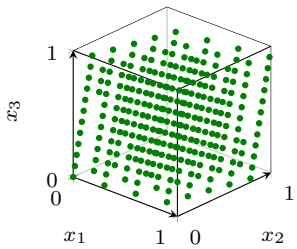
(j) $\mathcal{X}^{(3)}$, $|\mathcal{X}^{(3)}| = 17$ ($t = 3$, step 2a).



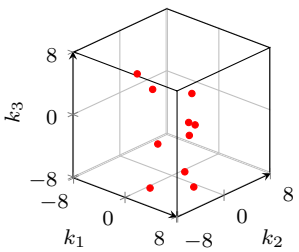
(k) $I^{(3)} \subseteq \mathcal{P}_3(\Gamma)$ ($t = 3$, step 2a).



(l) $I^{(1,2)} \times I^{(3)}$, $|I^{(1,2)} \times I^{(3)}| = 70$ ($t = 3$, step 2b).



(m) $\mathcal{X}^{(1,2,3)} = \Lambda(\boldsymbol{z}, M, I^{(1,2)} \times I^{(3)})$, $\boldsymbol{z} = (1, 4, 23)^\top$, $M = 276$ ($t = 3$, step 2b).



(n) $I^{(1,2,3)} = \operatorname{supp} \hat{p}$ ($t = 3$, step 2e, step 3).

Figure 2.1: Example of reconstructing a three-dimensional trigonometric polynomial p using Algorithm 1 with $\Gamma = \hat{G}_8^3$, $|\hat{G}_8^3| = 4\,913$.

with randomly chosen points $x'_1, x'_2 \in \mathbb{T}$, see Figure 2.1j. In step 2b, the index set of detected frequencies for the third component $I^{(3)}$ is determined, see Figure 2.1k, as well as the parameter $S_3 = 12$. As described for $t = 2$, the index set of frequency candidates for the first three components $I^{(1,2)} \times I^{(3)}$, $|I^{(1,2)} \times I^{(3)}| = 70$ is constructed. Similar to dimension increment step $t = 2$, we build a reconstructing rank-1 lattice for the index set $I^{(1,2)} \times I^{(3)}$ and we obtain the sampling set $\mathcal{X}^{(1,2,3)} := \Lambda(\boldsymbol{z} = (1, 4, 23)^\top, M = 276, I^{(1,2)} \times I^{(3)})$, $|\mathcal{X}^{(1,2,3)}| = 276$, as depicted in Figures 2.1l and 2.1m, respectively. The trigonometric polynomial p is sampled along this sampling set in step 2c and the Fourier coefficients $\tilde{\hat{p}}_{(1,2,3),\boldsymbol{k}}$ are determined in step 2d. Now, the frequency index set $I^{(1,2,3)}$ is determined in step 2e, see Figure 2.1n, using a single 1d iFFT.

- Step 3: Finally, this index set $I^{(1,2,3)}$ and the corresponding Fourier coefficients $\tilde{\hat{p}}_{(1,2,3),\boldsymbol{k}}$, $\boldsymbol{k} \in I^{(1,2,3)}$, are used as an approximation for the support $\operatorname{supp} \hat{p}$ and for the corresponding Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ of the trigonometric polynomial p, respectively.

Algorithm 2

Algorithm 2 Reconstruction of a trigonometric polynomial p from sampling values.

***Modifications of Algorithm* 1 :**

$\vdots$

(step 2b) Set $\tilde{r} := \begin{cases} r & \text{for } t < d, \\ 1 & \text{for } t = d. \end{cases}$

Determine $S_t := \min\left\{m \in \mathbb{N} \colon |\{k_t \bmod m \colon k_t \in I^{(t)}\}| = |I^{(t)}|\right\}$. Set $I^{(1,\dots,t)} := \emptyset$.

Build reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M_t, (I^{(1,\dots,t-1)} \cap \mathcal{P}_{(1,\dots,t-1)}(\Gamma)) \times (I^{(t)} \cap \mathcal{P}_{(t)}(\Gamma)))$:

Set $M_t := M_{t-1} \cdot S_t$ and $z_t := M_{t-1}$, i.e., $\boldsymbol{z} = (z_1, \dots, z_{t-1}, M_{t-1})^\top$, cf. Theorem 2.2.

for $i := 1, \dots, \tilde{r}$ ***do***

$\vdots$

Algorithm 2 is another realization for the dimension incremental method, which uses one-dimensional inverse fast Fourier transforms (1d iFFTs). The approach is based on Algorithm 1, but in step 2b, we do not search for a reconstructing rank-1 lattice for the frequency index set $I^{(1,\dots,t-1)} \times I^{(t)}$ but we explicitly build one using the construction from Theorem 2.2 in Section 2.1. The other steps of Algorithm 1 remain unchanged in Algorithm 2.

The arithmetic complexity for Algorithm 2 is distinctly lower than for Algorithm 1, see Section 2.2.3, and the upper bound for the number of samples is asymptotically the same for both algorithms. However, in practice, the number of samples when using Algorithm 2 may be larger because we do not search for a preferably small rank-1 lattice size. Especially, if the search space Γ is distinctly smaller than the full grid $\hat{G}_N^d$ and Γ is not a tensor product grid, then Algorithm 1 is better suited with respect to the number of samples. We observe this behavior in the numerical results in Section 3. For our small example from Figure 2.1, Algorithm 2 yields the identical index sets and rank-1 lattices as Algorithm 1.

Deterministic version of Algorithm 1 and 2

We remark that we do not need to use random sampling if the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp} \hat{p}$, of the trigonometric polynomial p fulfill the property that the signs of the real part

$\mathrm{Re}(\hat{p}_{\boldsymbol{k}})$ of all Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp}\hat{p}$, have to be the same as well as the signs of the imaginary part $\mathrm{Im}(\hat{p}_{\boldsymbol{k}})$. This means for all $\boldsymbol{k} \in \operatorname{supp}\hat{p}$, we have either

- $\mathrm{Re}(\hat{p}_{\boldsymbol{k}}) \geq 0$, $\mathrm{Im}(\hat{p}_{\boldsymbol{k}}) \geq 0$ or
- $\mathrm{Re}(\hat{p}_{\boldsymbol{k}}) \geq 0$, $\mathrm{Im}(\hat{p}_{\boldsymbol{k}}) \leq 0$ or
- $\mathrm{Re}(\hat{p}_{\boldsymbol{k}}) \leq 0$, $\mathrm{Im}(\hat{p}_{\boldsymbol{k}}) \geq 0$ or
- $\mathrm{Re}(\hat{p}_{\boldsymbol{k}}) \leq 0$, $\mathrm{Im}(\hat{p}_{\boldsymbol{k}}) \leq 0$.

Then, we may set the number of detection iterations $r := 1$, the sparsity parameter $s := \operatorname{supp}\hat{p}$ as well as the (random) components $x'_1, \ldots, x'_d$ of the sampling nodes always to zero in Algorithm 1 and 2, by which both algorithms become deterministic. However for arbitrary Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in \mathbb{C}$, we rely on random sampling in both algorithms.

2.2.2 Successful and failed detection

As mentioned in Section 2.2.1, the successful detection of all non-zero Fourier coefficients and the corresponding frequencies, i.e., obtaining $I^{(1,\ldots,d)} = \operatorname{supp}\hat{p}$, is not guaranteed. In the following, we discuss conditions for the successful detection and we address the question if it is possible to notice that not all frequencies were detected successfully during the incremental detection process in Section 2.2.1. We remark that the computations in (2.5) for Algorithm 1 and 2 are responsible for the correct frequency detection, which belong to the computation steps 1 and 2a, as well as the computations in (2.7), which belong to the computation step 2d.

For the computation of one-dimensional projected Fourier coefficients for the t-th component $\tilde{\hat{p}}_{t,k_t}$, $t \in \{1,\ldots,d\}$ and $k_t \in \mathcal{P}_t(\Gamma)$, in formula (2.5), the randomly chosen values $x'_1, \ldots, x'_{t-1}, x'_{t+1}, \ldots, x'_d \in \mathbb{T}$ directly influence the successful detection $I^{(t)} = \mathcal{P}_t(\operatorname{supp}\hat{p})$, see the aliasing formula (2.6). Note that the computation of the coefficient $\tilde{\hat{p}}_{t,k_t}$ for fixed $t \in \{1,\ldots,d\}$ and $k_t \in \mathcal{P}_t(\Gamma)$ may be regarded as the evaluation of the trigonometric polynomial

$$\tilde{p}_{t,k_t} \colon \mathbb{T}^{d-1} \to \mathbb{C}, \qquad \tilde{p}_{t,k_t} := \sum_{\substack{\tilde{\boldsymbol{h}} := (h_1,\ldots,h_{t-1},h_{t+1},\ldots,h_d)^\top \in \mathcal{P}_{(1,\ldots,t-1,t+1,\ldots,d)}(\Gamma) \\ \boldsymbol{h} := (h_1,\ldots,h_{t-1},k_t,h_{t+1},\ldots,h_d)^\top \in \operatorname{supp}\hat{p}}} \hat{p}_{\boldsymbol{h}}\, \mathrm{e}^{2\pi \mathrm{i} \tilde{\boldsymbol{h}} \cdot \circ}, \tag{2.9}$$

at the node $\tilde{\boldsymbol{x}}' := (x'_1, \ldots, x'_{t-1}, x'_{t+1}, \ldots, x'_d)^\top \in \mathbb{T}^{d-1}$, i.e., $\tilde{\hat{p}}_{t,k_t} = \tilde{p}_{t,k_t}(\tilde{\boldsymbol{x}}')$. Accordingly, for the computation of the t-dimensional projected Fourier coefficients for the first t components $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, the randomly chosen values $x'_{t+1}, \ldots, x'_d \in \mathbb{T}$ directly influence the successful frequency detection, see the aliasing formula (2.8). When we compute the coefficients $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, we apply one inverse rank-1 lattice FFT as described in Section 2.1. This means we compute, see step 2d of Algorithm 1,

$$\hat{g}_\ell := \frac{1}{M_t} \sum_{m=0}^{M_t-1} p\left(\left(\frac{m}{M_t}(z_1,\ldots,z_t) \bmod \mathbf{1}, x'_{t+1}, \ldots, x'_d \right)^\top \right) \mathrm{e}^{-2\pi \mathrm{i} m\ell/M_t} \tag{2.10}$$

for $\ell = 0, \ldots, M_t - 1$ using a single 1d iFFT and

$$\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}} := \hat{g}_{\boldsymbol{k}\cdot(z_1,\ldots,z_t)^\top \bmod M_t} \quad \text{for } \boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma).$$

We remark that the computation of the coefficient $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$ for fixed $t \in \{2,\dots,d-1\}$ and $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$ may be regarded as the evaluation of the trigonometric polynomial

$$\tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}} \colon \mathbb{T}^{d-t} \to \mathbb{C}, \qquad \tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}} := \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p} \\ \left((h_1,\dots,h_t)^\top - \boldsymbol{k}\right)\cdot\boldsymbol{z}\equiv 0 \pmod{M_t}}} \hat{p}_{\boldsymbol{h}}\, \mathrm{e}^{2\pi\mathrm{i}(h_{t+1},\dots,h_d)^\top\cdot\circ}, \quad (2.11)$$

at the node $\tilde{\boldsymbol{x}}' := (x'_{t+1},\dots,x'_d)^\top \in \mathbb{T}^{d-t}$, i.e., $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = \tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}}(\tilde{\boldsymbol{x}}')$.

In Theorem 2.5, we give an upper bound on the probability that the frequency detections in step 1, step 2a as well as step 2e do not recognize a frequency $k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p})$ and $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p})$, respectively. Prior to this, we require

Lemma 2.4. *Let $\delta \geq 0$ be a threshold value and let a trigonometric polynomial $g : \mathbb{T}^n \to \mathbb{C}$, $n \in \mathbb{N}$, $g(\boldsymbol{x}) := \sum_{\boldsymbol{k}\in\tilde{I}} \hat{g}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$, $\tilde{I} \subset \mathbb{Z}^n$, $|\tilde{I}| < \infty$, be given by its Fourier coefficients $\hat{g}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in \tilde{I}$, such that the property $\|g|L^1(\mathbb{T}^n)\| > \delta \geq 0$ or $\max_{\boldsymbol{k}\in\tilde{I}} |\hat{g}_{\boldsymbol{k}}| > \delta \geq 0$ is fulfilled. Moreover, let $X_1,\dots,X_n \in \mathbb{T}^d$ be independent, identical, uniformly distributed random variables and the random vector $\boldsymbol{X} := (X_1,\dots,X_n)^\top$. Then, the probability*

$$\mathbb{P}(|g(\boldsymbol{X})| \leq \delta) \leq \mathrm{e}^{-\frac{2(\|g|L^1(\mathbb{T}^n)\|-\delta)^2}{\|g|L^\infty(\mathbb{T}^n)\|^2}} < 1.$$

$$\text{If } \max_{\boldsymbol{k}\in\tilde{I}} |\hat{g}_{\boldsymbol{k}}| > \delta, \text{ then } \quad \mathbb{P}(|g(\boldsymbol{X})| \leq \delta) \leq \mathrm{e}^{-\frac{2(\|g|L^1(\mathbb{T}^n)\|-\delta)^2}{\|g|L^\infty(\mathbb{T}^n)\|^2}} \leq \mathrm{e}^{-\frac{2(\max_{\boldsymbol{k}\in\tilde{I}}|\hat{g}_{\boldsymbol{k}}|-\delta)^2}{(\sum_{\boldsymbol{k}\in\tilde{I}}|\hat{g}_{\boldsymbol{k}}|)^2}} < 1.$$

Proof. We define the random variable $Y_1 := -|g(\boldsymbol{X})|$. Formally, for the expectation value of Y_1, we have $\mathbb{E}(Y_1) = \int_{\mathbb{T}^n} -|g((x_1,\dots,x_n)^\top)|\, f_{X_1,\dots,X_n}(x_1,\dots,x_n)\,\mathrm{d}x_1\dots\mathrm{d}x_n$, where $f_{X_1,\dots,X_n}$ is the joint probability density function of the random variables $X_1,\dots,X_n$. Since the random variables $X_1,\dots,X_n$ are independent and uniformly distributed, we obtain $f_{X_1,\dots,X_n} \equiv 1$. This yields $\mathbb{E}(Y_1) = -\int_{\mathbb{T}^n} |g(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} = -\|g|L^1(\mathbb{T}^n)\|$. Next, we apply Hoeffdings inequality and obtain

$$\mathbb{P}\left(Y_1 - \mathbb{E}(Y_1) \geq \|g|L^1(\mathbb{T}^n)\| - \delta\right) \leq \mathrm{e}^{-\frac{2(\|g|L^1(\mathbb{T}^n)\|-\delta)^2}{\|g|L^\infty(\mathbb{T}^n)\|^2}},$$

for $\|g|L^1(\mathbb{T}^n)\| > \delta$ since $\mathbb{P}\left(g(Y_1) \in [-\|g|L^\infty(\mathbb{T}^n)\|, 0]\right) = 1$.
Due to $|\hat{g}_{\boldsymbol{k}}| = |\int_{\mathbb{T}^n} g(\boldsymbol{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}\mathrm{d}\boldsymbol{x}| \leq \int_{\mathbb{T}^n} |g(\boldsymbol{x})|\mathrm{d}\boldsymbol{x} = \|g|L^1(\mathbb{T}^n)\|$ for all $\boldsymbol{k} \in \tilde{I}$, we have $\max_{\boldsymbol{k}\in\tilde{I}} |\hat{g}_{\boldsymbol{k}}| \leq \|g|L^1(\mathbb{T}^n)\|$. Since $\|g|L^\infty(\mathbb{T}^n)\| = \operatorname{ess\,sup}_{\boldsymbol{x}\in\mathbb{T}^n} |g(\boldsymbol{x})| \leq \sum_{\boldsymbol{k}\in\tilde{I}} |\hat{g}_{\boldsymbol{k}}|$ and

$$\begin{aligned}\mathbb{P}\left(Y_1 - \mathbb{E}(Y_1) \geq \|g|L^1(\mathbb{T}^n)\| - \delta\right) &= \mathbb{P}\left(-|g(\boldsymbol{X})| + \|g|L^1(\mathbb{T}^n)\| \geq \|g|L^1(\mathbb{T}^n)\| - \delta\right) \\ &= \mathbb{P}\left(|g(\boldsymbol{X})| \leq \delta\right),\end{aligned}$$

we obtain the assertion. ∎

Theorem 2.5. *Let a threshold value $\delta \geq 0$, a trigonometric polynomial p of the form (1.1) with the property $\min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}| > \delta \geq 0$ and a search space $\Gamma \supset \operatorname{supp}\hat{p}$ of finite cardinality be given. For fixed $t \in \{1,\dots,d\}$ and $L_t := \max(\mathcal{P}_t(\Gamma)) - \min(\mathcal{P}_t(\Gamma)) + 1$, we compute the one-dimensional projected Fourier coefficients for the t-th component $\tilde{\hat{p}}_{t,k_t} = \tilde{p}_{t,k_t}(x'_1,\dots,x'_{t-1},x'_{t+1},\dots,x'_d)$, $k_t \in \mathcal{P}_t(\Gamma)$, by formula (2.5), where the values*

$x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d \in \mathbb{T}$ *are independently chosen uniformly at random. Then, the probability*

$$\mathbb{P}(|\tilde{\hat{p}}_{t,k_t}| \le \delta) \le \mathrm{e}^{-2\frac{(\min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|-\delta)^2}{(\sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|)^2}} < 1 \quad \text{for } k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p}).$$

Moreover, for fixed $t \in \{2, \dots, d-1\}$, we compute the t-dimensional projected Fourier coefficients for the first t components $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = \tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}}(x'_{t+1}, \dots, x'_d)$, $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, by formula (2.7), where the values $x'_{t+1}, \dots, x'_d \in \mathbb{T}$ are independently chosen uniformly at random. If the rank-1 lattice $\Lambda(\boldsymbol{z}, M_t)$ is a reconstructing rank-1 lattice for $\mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p})$, then the probability

$$\mathbb{P}(|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| \le \delta) \le \mathrm{e}^{-2\frac{(\min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|-\delta)^2}{(\sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|)^2}} < 1 \quad \text{for } \boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p}) \cap (I^{(1,\dots,t-1)} \times I^{(t)}).$$

Proof. For fixed $t \in \{1, \dots, d\}$ and for each $k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p})$, we regard the computation of the one-dimensional projected Fourier coefficient for the t-th component $\tilde{\hat{p}}_{t,k_t}$ as the evaluation of the trigonometric polynomial $\tilde{p}_{t,k_t}$ from (2.9) at the node $(x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d)^\top \in \mathbb{T}^{d-1}$. We apply Lemma 2.4 setting the index set $\tilde{I} := \mathcal{P}_{(1,\dots,t-1,t+1,\dots,d)}(\{\boldsymbol{h} \in \operatorname{supp}\hat{p} \colon h_t = k_t\})$ and the Fourier coefficients $\hat{g}_{\mathcal{P}_{(1,\dots,t-1,t+1,\dots,d)}(\boldsymbol{h})} := \hat{p}_{\boldsymbol{h}}$ for $\boldsymbol{h} \in \{\boldsymbol{h} \in \operatorname{supp}\hat{p} \colon h_t = k_t\}$. Since $0 \le \delta < \min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}| \le \max_{\boldsymbol{l}\in\tilde{I}}|\hat{g}_{\boldsymbol{l}}|$ and $\sum_{\boldsymbol{l}\in\tilde{I}}|\hat{g}_{\boldsymbol{l}}| \le \sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|$, we obtain the assertion for the coefficients $\tilde{\hat{p}}_{t,k_t}$, $k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p})$.
Similarly, for fixed $t \in \{2, \dots, d-1\}$ and for each $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p}) \cap (I^{(1,\dots,t-1)} \times I^{(t)})$, we regard the computation of the t-dimensional projected Fourier coefficients for the first t components $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$ as the evaluation of the trigonometric polynomial $\tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}}$ from (2.11) at the node $(x'_{t+1}, \dots, x'_d)^\top \in \mathbb{T}^{d-t}$. We apply Lemma 2.4 setting the index set $\tilde{I} := \mathcal{P}_{(t+1,\dots,d)}\big(\{\boldsymbol{h} \in \operatorname{supp}\hat{p} \colon \big((h_1, \dots, h_t)^\top - \boldsymbol{k}\big) \cdot \boldsymbol{z} \equiv 0 \pmod{M_t}\}\big)$ and the Fourier coefficients $\hat{g}_{\mathcal{P}_{(t+1,\dots,d)}(\boldsymbol{h})} := \hat{p}_{\boldsymbol{h}}$ for $\boldsymbol{h} \in \{\boldsymbol{h} \in \operatorname{supp}\hat{p} \colon \big((h_1, \dots, h_t)^\top - \boldsymbol{k}\big) \cdot \boldsymbol{z} \equiv 0 \pmod{M_t}\}$. This yields the assertion for the coefficients $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p}) \cap (I^{(1,\dots,t-1)} \times I^{(t)})$. ∎

In Algorithm 1 and 2, for a specified relative threshold parameter $\theta \in \mathbb{R}$, $0 < \theta < 1$, we use the threshold value $\delta := \theta \cdot \max_{k_t\in\mathcal{P}_t(\operatorname{supp}\hat{p})}|\tilde{\hat{p}}_{t,k_t}|$ in step 1 and 2a for $t \in \{1, \dots, d\}$ as well as $\delta := \theta \cdot \max_{\boldsymbol{k}\in\mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p})}|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}|$ in step 2e for $t \in \{2, \dots, d-1\}$. In the case $0 < \theta < (\min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|) \,/\, \sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|$, Theorem 2.5 yields

$$\mathbb{P}(|\tilde{\hat{p}}_{t,k_t}| \le \delta) \le C(p) < 1 \qquad \text{and} \qquad \mathbb{P}(|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| \le \delta) \le C(p) < 1,$$

where the constant $C(p) := \mathrm{e}^{-2\frac{(\min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|-\theta\cdot\sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|)^2}{(\sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}}|\hat{p}_{\boldsymbol{h}}|)^2}}$, $0 < C(p) < 1$. Since we use $r \in \mathbb{N}$ many detection iterations with new values $x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d \in \mathbb{T}$ independently chosen uniformly at random, the frequency detection for $k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p})$ succeeds if

$$|\tilde{\hat{p}}_{t,k_t}| = \Big|\tilde{p}_{t,k_t}((x'_1, \dots, x'_{t-1}, x'_{t+1}, \dots, x'_d)^\top)\Big| \ge \theta \cdot \max_{\tilde{k}_t\in\mathcal{P}_t(\operatorname{supp}\hat{p})}|\tilde{\hat{p}}_{t,\tilde{k}_t}|$$

in at least one detection iteration $i \in \{1, \dots, r\}$, and we obtain $\mathbb{P}(k_t \in I^{(t)}) \ge 1 - (C(p))^r$ for each $k_t \in \mathcal{P}_t(\operatorname{supp}\hat{p})$, see step 2a of Algorithm 1 for the index set $I^{(t)}$, assuming that

the sparsity input parameter $s \geq |\operatorname{supp} \hat{p}|$ and the search space $\Gamma \supset \operatorname{supp} \hat{p}$. Note that this probability can be arbitrarily close to 1 if the number r of detection iterations is sufficiently large. Similarly, the frequency detection for $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$ succeeds if

$$|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| = \left|\tilde{p}^{\Lambda(\boldsymbol{z},M_t)}_{(1,\dots,t),\boldsymbol{k}}((x'_{t+1},\dots,x'_d)^\top)\right| \geq \theta \cdot \max_{\tilde{\boldsymbol{k}} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})} |\tilde{\hat{p}}_{(1,\dots,t),\tilde{\boldsymbol{k}}}|$$

in at least one detection iteration $i \in \{1,\dots,r\}$, and this yields $\mathbb{P}(\boldsymbol{k} \in I^{(1,\dots,t)}) \geq 1 - (C(p))^r$ for each $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$, assuming that the sparsity input parameter $s \geq |\operatorname{supp} \hat{p}|$, the search space $\Gamma \supset \operatorname{supp} \hat{p}$, $I^{(\tau)} = \mathcal{P}_\tau(\operatorname{supp} \hat{p})$ for $\tau \in \{1,\dots,t\}$ and $I^{(1,\dots,\tau)} = \mathcal{P}_{(1,\dots,\tau)}(\operatorname{supp} \hat{p})$ for $\tau \in \{2,\dots,t-1\}$.

Finally, all non-zero Fourier coefficients and the corresponding frequencies are successfully detected if the frequency detections in the dimension increment steps $t \in \{1,\dots,d\}$ succeed.

During the computations in step 2 of Algorithm 1 and 2, the following cases may occur.

i. For a frequency $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, we have $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} \neq 0$ and $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$, i.e., the detection of the frequency $\boldsymbol{k}$ was successful.

ii. For a frequency $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, we have $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = 0$ but $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$, i.e., the frequency $\boldsymbol{k}$ was considered but not recognized, and the detection of the frequency $\boldsymbol{k}$ failed.

iii. For a frequency $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$, we have $\boldsymbol{k} \notin (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, i.e., the frequency $\boldsymbol{k}$ was not considered. This means the detection of the frequency $\boldsymbol{k}$ failed.
   a) For a frequency $\ell \in \{0,\dots,M_t-1\}$, we have $\hat{g}_\ell \neq 0$ in (2.10) but $\nexists\, \boldsymbol{k} \in I^{(1,\dots,t-1)} \times I^{(t)}$ such that $\boldsymbol{k} \cdot (z_1,\dots,z_t)^\top \equiv \ell \pmod{M_t}$.

iv. For a frequency $\boldsymbol{k} \in (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, we have $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} \neq 0$ but $\boldsymbol{k} \notin \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$, i.e., the frequency $\boldsymbol{k}$ was falsely detected.

As discussed in Section 2.2.1, we do not test the Fourier coefficients for zero/non-zero but if their absolute values are below/above a certain threshold. Correspondly, $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} \neq 0$ means $|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| \geq$ threshold_value and $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}} = 0$ means $|\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}| <$ threshold_value.

Case i is the optimal case where the frequency $\boldsymbol{k}$ was in the candidate list $(I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$ and detected correctly.

In contrast, case ii means that the frequency also was in the candidate list but was wrongly not included in the index set $I^{(1,\dots,t)}$ of detected frequencies. Similar to the discussion for the computation (2.5), the fixed values $x'_{t+1},\dots,x'_d \in \mathbb{T}$ influence the successful frequency detection, see the aliasing formula (2.8). Again, we suggest to repeatedly evaluate (2.7) with different randomly chosen values $x'_{t+1},\dots,x'_d \in \mathbb{T}$ and compare the obtained index sets $I^{(1,\dots,t)}$ of detected frequencies. If all of them coincide, it is very likely that the case ii did not occur. Otherwise, we suggest to use the union of the obtained index sets $I^{(1,\dots,t)}$ for the computations that follow in Section 2.2.1.

In case iii, at least one frequency $\boldsymbol{k} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$ is already missing in the candidate list $(I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$. Possibly, we will not be able to even notice this. If we encounter case iiia, which is a special variant of case iii, we know that there exists at least one frequency $\boldsymbol{k} := \binom{\boldsymbol{k}'}{k''} \in \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$ for which $\boldsymbol{k} \notin (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, but we do

not exactly know which and how many frequencies are affected by this. However, we know that these frequencies are from the set $\{\boldsymbol{h} \in \mathcal{P}_{(1,\dots,t)}(\Gamma) \colon \boldsymbol{h} \cdot (z_1, \dots, z_t)^\top \equiv \ell \pmod{M_t}\}$ for Algorithm 1.

Furthermore, case iv may occur and is a consequence of having $I^{(t)} \not\supset \mathcal{P}_t(\operatorname{supp} \hat{p})$ in the current (t-th) dimension increment step or case iii in one of the preceding dimension increment steps $1, \dots, t-1$. This means, in the current or one of the previous dimension increment steps, at least one frequency was not detected. Moreover, we have $I^{(1,\dots,t)} \not\subset \mathcal{P}_{(1,\dots,t)}(\operatorname{supp} \hat{p})$. However, we may not be able to notice that this case has occurred.

2.2.3 Number of samples and arithmetic complexity

In this section, we give upper bounds for the number of samples and for the arithmetic complexity of the methods described in Section 2.2.1 as Algorithm 1 and 2 in the case where the search space Γ is the full grid $\hat{G}_N^d$. For computing the index set of detected frequencies for the t-th component $I^{(t)}$ in the steps 1 and 2a of Algorithm 2, $(2N+1)$ function samples are taken and the used 1d iFFT requires at most $C_1 \, N \log N$ arithmetic operations in each detection iteration $i \in \{1, \dots, r\}$ for each $t \in \{1, \dots, d\}$, where the constant $C_1 \geq 1$ does not depend on N. This yields $r \, d \, |\hat{G}_N^1| = r \, d \, (2N+1)$ function samples and at most $\tilde{C} \, r \, d \, N \log N$ arithmetic operations for determining the index sets $I^{(1)}, \dots, I^{(d)}$, where $\tilde{C} \geq 1$ is an absolute constant.

In step 2 of Algorithm 1 and 2 for dimension increment step t, the index sets $I^{(1,\dots,t-1)}$ and $I^{(t)}$ consist of at most rs many frequencies. This yields that the index set $I^{(1,\dots,t-1)} \times I^{(t)}$ consists of $|I^{(1,\dots,t-1)} \times I^{(t)}| \leq r \, s \, |\hat{G}_N^1|$ frequency candidates. The sampling set $\mathcal{X}^{(1,\dots,t)}$ constructed in step 2b of Algorithm 1 and 2 has the size $|\mathcal{X}^{(1,\dots,t)}| = M_t$, where the rank-1 lattice size $M_t \leq \max\{2r^2s^2, 3N\}\,(2N+1)$ due to Corollary 2.3. The inverse rank-1 lattice FFT in step 2d requires no more than $C_1 M_t \log M_t + 2t|I^{(1,\dots,t-1)} \times I^{(t)}|$ arithmetic operations for each detection iteration $i \in \{1, \dots, r\}$ and each dimension increment step $t \in \{2, \dots, d\}$.
For each detection iteration $i \in \{1, \dots, r\}$ and each dimension increment step $t \in \{2, \dots, d\}$ when searching for the next component z_t of the generating vector $\boldsymbol{z}$ in step 2b in Algorithm 1, the number of arithmetic operations is bounded by $3\,|I^{(1,\dots,t-1)} \times I^{(t)}|\, M_t \leq 3 \, r \, s \, (2N+1) \max\{2r^2s^2, 3N\}(2N+1)$, see the proof of Theorem 2.1. Moreover, reducing the rank-1 lattice size M_t using [24, Algorithm 3.5] requires no more arithmetic operations.
At the end of step 2e, the index set $I^{(1,\dots,t)}$ consists of no more than $|I^{(1,\dots,t)}| \leq r \, s$ frequencies. Consequently, when searching for the new rank-1 lattice in the additional step 2f of Algorithm 1 and 2 for each $t \in \{2, \dots, d-1\}$, the new rank-1 lattice size M_t is bounded by $\max\{2r^2s^2, 3N\}$ due to Theorem 2.1. The number of arithmetic operations for the search of the next component z_t of the generating vector $\boldsymbol{z}$ is bounded by $3\,|I^{(1,\dots,t)}|\, M_t \leq 3 \, r \, s \, \max\{2r^2s^2, 3N\}$, see the proof of Theorem 2.1. Reducing the rank-1 lattice size M_t requires no more than $3 \, r \, s \, \max\{2r^2s^2, 3N\}$ arithmetic operations.

In total, this yields no more than

$$r \, (d-1) \max\{2r^2s^2, 3N\} \, 2(N+1) + r \, d \, (2N+1)$$

many samples for Algorithm 1 and 2 as well as

$$C \, d \cdot \big(\max\{r^3s^3N^2, r \, s \, N^3\} + \max\{r^3s^2N, r \, N^2\} \log(\max\{r^2s^2N, N^2\})\big)$$

arithmetic operations for Algorithm 1 and

$$C \, d \cdot \big(\max\{r^3s^3, r \, s \, N\} + \max\{r^3s^2N, r \, N^2\} \log(\max\{r^2s^2N, N^2\})\big)$$

arithmetic operations for Algorithm 2, where $C > 1$ is an absolute constant. We remark that a large contribution to the arithmetic complexity comes from the rank-1 lattice search, in particular for Algorithm 1 in step 2b. Moreover, we have no exponential dependence in the dimension d, neither for the number of samples nor the arithmetic complexity. Assuming $\sqrt{N} \lesssim s \lesssim N^d$, we require $\mathcal{O}(d\,s^2 N)$ many samples for both algorithms as well as $\mathcal{O}(d\,s^3 N^2)$ and $\mathcal{O}(d\,s^3 + d\,s^2 N \log(s\,N))$ arithmetic operations for Algorithm 1 and 2, respectively. In the case $s \lesssim \sqrt{N}$, we need $\mathcal{O}(d\,N^2)$ many samples for both algorithms as well as $\mathcal{O}(d\,s\,N^3)$ and $\mathcal{O}(d\,N^2 \log N)$ arithmetic operations for Algorithm 1 and 2, respectively.

2.3 Reducing the number of samples for the dimension incremental reconstruction of trigonometric polynomials using compressed sensing

The dimension incremental reconstruction method for trigonometric polynomials from samples presented in Section 2.2.1 and realized as Algorithm 1 and 2 may require $\mathcal{O}(d\,s^2 N)$ and $\mathcal{O}(d\,N^2)$ many samples in the case $\sqrt{N} \lesssim s \lesssim N^d$ and $s \lesssim \sqrt{N}$, respectively, see Section 2.2.3, if the search space Γ is the full grid $\hat{G}_N^d$. In this section, we discuss a possible approach to reduce this number of samples. In Section 1, we have already mentioned the possibility to recover trigonometric polynomials, i.e., solving the problem (1.2), using random sampling in compressed sensing, see [32] and the references therein. Then, $L \geq C\,|\operatorname{supp}\hat{p}|\,\log^4(|\Gamma|)\,\log(1/\eta)$ random samples suffice in order to reconstruct a trigonometric polynomial with frequencies supported on the search space $\Gamma \subset \mathbb{Z}^d$ with probability $1-\eta$, where $C > 0$ is an absolute constant. However, the arithmetic complexity contains in general the factor $|\Gamma|$ when we apply such a compressed sensing algorithm directly, since the computations usually involve multiplications with the Fourier matrix $\boldsymbol{A} := \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}_\ell}\right)_{\boldsymbol{x}_\ell\in\mathcal{X},\,\boldsymbol{k}\in\Gamma}$ and its adjoint $\boldsymbol{A}^*$, where $\mathcal{X}$ is the set of samples $\boldsymbol{x}_\ell$ of cardinality $|\mathcal{X}| = L$. Using our dimension incremental reconstruction approach from Section 2.2 in combination with compressed sensing methods, we can reduce this arithmetic complexity and still limit the number of random samples.

First, we consider the case where the sampling nodes $\boldsymbol{x}_\ell \in \mathcal{X} \subset \mathbb{T}^d$, $|\mathcal{X}| = L$, are chosen uniformly at random. For this, we remove the building of the reconstructing rank-1 lattices from the steps 2b and 2f in Algorithm 2. As mentioned before, the index set $I^{(1,\ldots,t-1)} \times I^{(t)}$ of frequency candidates in dimension increment step $t \in \{2,\ldots,d\}$ consists of at most $r\,s\,|\hat{G}_N^1| = r\,s\,(2N+1)$ frequency candidates if $\Gamma = \hat{G}_N^d$. Consequently, using $L = \lceil C\,|\operatorname{supp}\hat{p}|\,\log^4(r\,s\,|\hat{G}_N^1|)\,\log(1/\eta)\rceil$ many random samples from $\mathbb{T}^d$ is sufficient in each detection iteration $i \in \{1,\ldots,r\}$ and in each dimension increment step $t \in \{2,\ldots,d\}$ if we apply compressed sensing instead of the inverse rank-1 lattice FFT in step 2d. For this, we can use ℓ_1 minimization (Basis Pursuit) [4]. Concretely, we use the SPGL1 algorithm [48, 47], which is an iterative method for solving the ℓ_1 minimization problem that utilizes matrix-vector multiplications of the Fourier matrix $\boldsymbol{A} := \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}_\ell}\right)_{\boldsymbol{x}_\ell\in\mathcal{X},\,\boldsymbol{k}\in I}$ and its adjoint $\boldsymbol{A}^*$. The arithmetic complexity of the SPGL1 algorithm is dominated by the matrix-vector multiplications involving the Fourier matrix $\boldsymbol{A}$ and its adjoint $\boldsymbol{A}^*$ requiring each $\mathcal{O}(L\,|I|)$ arithmetic operations as well as a one-norm projection requiring $\mathcal{O}(|I|\log|I|)$ arithmetic operations in each of $R \in \mathbb{N}$ many iterations for SPGL1, cf. [48, Sec. 4.2]. For $s = \mathcal{O}(|\operatorname{supp}\hat{p}|)$, the ℓ_1 minimization using SPGL1 requires no more than $r\,L = \mathcal{O}(s\log^4(s\,N))$ many samples and $\mathcal{O}(R\,L\,|I^{(1,\ldots,t-1)} \times I^{(t)}|) = \mathcal{O}(R\,s^2 N \log^4(s\,N))$ arithmetic operations in each dimension increment step $t \in \{2,\ldots,d\}$.

Totally, this means $\mathcal{O}(d\,s\log^4(s\,N) + d\,N)$ many samples and $\mathcal{O}\left(d\,R\,s^2 N\log^4(s\,N)\right)$ arith-

metic operations for the dimension incremental reconstruction using ℓ_1 minimization if $\Gamma = \hat{G}_N^d$.

2.3.1 Sub-sampling on the rank-1 lattices

We can use also the SPGL1 ℓ_1 minimization in combination with reconstructing rank-1 lattices if we use partial derandomisation for the choice of the sampling nodes $\boldsymbol{x}_\ell$ by using a random subset of the rank-1 lattice $\mathcal{X}^{(1,\dots,t)}$ in step 2b of Algorithm 2 as the set of sampling nodes for the dimension increment step $t \in \{2,\dots,d\}$. Here, we apply one-dimensional fast Fourier transforms to compute the matrix-vector product of the Fourier matrix $\boldsymbol{A}$ and a vector from $\mathbb{C}^{|I|}$ as well as 1d iFFTs to compute the matrix-vector product of the adjoint Fourier matrix $\boldsymbol{A}^*$ and a vector from $\mathbb{C}^L$, see Section 2.1. The ℓ_1 minimization then requires $\mathcal{O}(R\,M_t \log M_t + |I^{(1,\dots,t-1)} \times I^{(t)}|)$ arithmetic operations in each dimension increment step $t \in \{2,\dots,d\}$, where $R \in \mathbb{N}$ is the number of iterations for SPGL1, $M_t \le \max\{2r^2s^2, 3N\}\,2(N+1)$ due to Corollary 2.3 and $|I^{(1,\dots,t-1)} \times I^{(t)}| \le r\,s\,(2N+1)$.

In total, this means $\mathcal{O}(d\,s\log^4(s\,N) + d\,N)$ many samples as well as $\mathcal{O}(d\,s^3 + d\,R\,s^2 N \log(sN))$ and $\mathcal{O}(d\,R\,N^2 \log N)$ arithmetic operations in the case $\sqrt{N} \lesssim s \lesssim N^d$ and $s \lesssim \sqrt{N}$, respectively, if $\Gamma = \hat{G}_N^d$. We successfully apply the sub-sampling on the rank-1 lattice in Section 3 in Example 3.3 and 3.8.

2.3.2 Sub-sampling using random generated sets

Instead of sub-sampling on the rank-1 lattices, we may also use so-called generated sets [23] as sampling set $\mathcal{X}$. This allows us to omit the rank-1 lattice search in the additional step 2f of Algorithm 2 and to potentially reduce the arithmetic complexity. A generated set, which is characterized by the number L of sampling nodes and a generating vector $\boldsymbol{r} \in \mathbb{R}^t$, $t \in \mathbb{N}$, is defined by $\mathcal{G}(\boldsymbol{r}, L) := \{\boldsymbol{x}_\ell = \ell\boldsymbol{r} \bmod \boldsymbol{1},\ \ell = 0,\dots,L-1\} \subset \mathbb{T}^t$. We remark that a rank-1 lattice $\Lambda(\boldsymbol{z}, M)$ as defined in Section 2.1 is a special case of a generated set, since we have $\Lambda(\boldsymbol{z}, M) \equiv \mathcal{G}(\boldsymbol{z}/M, M)$.

Here, we use shifted generated sets $\mathcal{G}(\boldsymbol{r}, L, \boldsymbol{\Delta}) := \{\boldsymbol{x}_\ell + \boldsymbol{\Delta} \colon \boldsymbol{x}_\ell \in \mathcal{G}(\boldsymbol{r}, L)\} \subset \mathbb{T}^t$, where $\boldsymbol{\Delta} \in \mathbb{T}^t$ is an arbitrary offset. For $\ell = 0,\dots,L-1$, the evaluation of a t-variate trigonometric polynomial p with frequencies supported on an arbitrary index set $I \subset \mathbb{Z}^t$ simplifies dramatically since

$$\begin{aligned} p(\boldsymbol{x}_\ell + \boldsymbol{\Delta}) &= \sum_{\boldsymbol{k}\in I} \hat{p}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{x}_\ell+\boldsymbol{\Delta})} = \sum_{\boldsymbol{k}\in I} \left(\hat{p}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\Delta}}\right) \mathrm{e}^{2\pi\mathrm{i}\ell\boldsymbol{k}\cdot\boldsymbol{r}} \\ &= \sum_{y\in\{\boldsymbol{k}\cdot\boldsymbol{r} \bmod 1 : \boldsymbol{k}\in I\}} \left(\sum_{\boldsymbol{k}\cdot\boldsymbol{r}\equiv y \,(\mathrm{mod}\,1)} \hat{p}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\Delta}} \right) \mathrm{e}^{2\pi\mathrm{i}\ell y}, \end{aligned}$$

cf. [23, 27]. Using the adjoint variant of the so-called nonequispaced fast Fourier transform (NFFT) [30], these function evaluations $p(\boldsymbol{x}_\ell + \boldsymbol{\Delta})$ for $\ell = 0,\dots,L-1$, which are equivalent to the matrix-vector multiplication of the Fourier matrix $\boldsymbol{A} := (\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}_\ell})_{\ell=0,\dots,L-1;\,\boldsymbol{k}\in I}$ and the vector $(\hat{p}_{\boldsymbol{k}}\,\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\Delta}})_{\boldsymbol{k}\in I}$, take $\mathcal{O}(L\log L + t\,|I|)$ arithmetic operations. Similarly, the matrix-vector multiplication of the adjoint Fourier matrix $\boldsymbol{A}^*$ and a vector from $\mathbb{C}^L$ can be computed in $\mathcal{O}(L\log L + t\,|I|)$ arithmetic operations using a one-dimensional NFFT. This means that R iterations of the SPGL1 algorithm require $\mathcal{O}\left(R\,L\log L + R\,|I^{(1,\dots,t-1)} \times I^{(t)}| \log |I^{(1,\dots,t-1)} \times I^{(t)}|\right)$ =

$\mathcal{O}\left(R\,s\log^4(s\,N)\,\log(s\log^4(s\,N)) + R\,s\,N\log(s\,N)\right) \quad = \quad \mathcal{O}\left(R\,s\log^5(s\,N) + R\,s\,N\log(s\,N)\right)$ arithmetic operations in each dimension increment step $t \in \{2,\dots,d\}$, since $L = \mathcal{O}(s\log^4(s\,N))$ and $|I^{(1,\dots,t-1)} \times I^{(t)}| = \mathcal{O}(s\,N)$.

Totally, this means $\mathcal{O}(d\,s\log^4(s\,N) + d\,N)$ many samples and

$$\mathcal{O}\left(d\,R\,s\log^5(s\,N) + d\,R\,s\,N\log(s\,N)\right)$$

arithmetic operations for the dimension incremental reconstruction using ℓ_1 minimization if $\Gamma = \hat{G}_N^d$.

In our heuristic approach for improving the condition number of the Fourier matrix $\boldsymbol{A}$, we choose more than one random shifted generated set $\mathcal{G}(\boldsymbol{r}, L, \boldsymbol{\Delta})$ with random generating vector $\boldsymbol{r} \in \mathbb{R}^t$ and random offset $\boldsymbol{\Delta} \in \mathbb{T}^t$ as sampling scheme $\mathcal{X}$. When we use $K \in \mathbb{N}$ many random shifted generated sets $\mathcal{G}(\boldsymbol{r}_1, L_1, \boldsymbol{\Delta}_1), \dots, \mathcal{G}(\boldsymbol{r}_K, L_K, \boldsymbol{\Delta}_K)$, $L = \sum_{p=1}^K L_p$, then the Fourier matrix is $\boldsymbol{A} := (\boldsymbol{A}_1, \dots, \boldsymbol{A}_K)^\top$, $\boldsymbol{A}_p := (\mathrm{e}^{2\pi\mathrm{i}\ell\boldsymbol{k}\cdot\boldsymbol{r}_p})_{\ell=0,\dots,L_p-1;\,\boldsymbol{k}\in I}$, and

$$\left(p(\boldsymbol{x}'+\boldsymbol{\Delta})\right)_{\boldsymbol{x}'\in\mathcal{G}(\boldsymbol{r}_1,L_1,\boldsymbol{\Delta}_1)\cup\ldots\cup\mathcal{G}(\boldsymbol{r}_K,L_K,\boldsymbol{\Delta}_K)} = \boldsymbol{A}\,\left(\hat{p}_{\boldsymbol{k}}\,\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\Delta}}\right)_{\boldsymbol{k}\in I}.$$

We remark that we can compute the matrix-vector-product of the Fourier matrix $\boldsymbol{A}$ and a vector $\hat{\boldsymbol{h}} \in \mathbb{C}^{|I|}$ as well as the matrix-vector-product of the adjoint Fourier matrix $\boldsymbol{A}^*$ and a vector $\boldsymbol{h} := (\boldsymbol{h}_1, \dots, \boldsymbol{h}_K)^\top$, $\boldsymbol{h}_p \in \mathbb{C}^{L_p}$ for $p = 1, \dots, K$, using K (adjoint) NFFTs due to

$$\boldsymbol{A}\hat{\boldsymbol{h}} = \begin{pmatrix} \boldsymbol{A}_1\hat{\boldsymbol{h}} \\ \vdots \\ \boldsymbol{A}_K\hat{\boldsymbol{h}} \end{pmatrix}, \quad \boldsymbol{A}^*\boldsymbol{h} = \begin{pmatrix} \boldsymbol{A}_1^* & \dots & \boldsymbol{A}_K^* \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_K \end{pmatrix} = \sum_{p=1}^K \boldsymbol{A}_p^*\boldsymbol{h}_p.$$

We successfully apply the sub-sampling on generated sets in Section 3 in Example 3.4 and 3.9.

2.4 Reducing the number of samples using Prony's method

The Prony method, see e.g. [39] and the references therein, allows the reconstruction of nonincreasing exponential sums $h\colon \mathbb{R} \to \mathbb{C}$ of order $s \in \mathbb{N}$,

$$h(x) := \sum_{m=1}^s c_m\,\mathrm{e}^{f_m\,x} \qquad (x \geq 0), \tag{2.12}$$

using $\mathcal{O}(s)$ suitable samples in a deterministic way, where $f_m \in (-\infty, 0) + \mathrm{i}[-\pi, \pi)$, $m = 1, \dots, s$, are distinct complex numbers and $c_m \in \mathbb{C} \setminus \{0\}$, $m = 1, \dots, s$, are coefficients. This reconstruction using a singular value decomposition has an arithmetic complexity of $\mathcal{O}(s^3)$.

For our special case, the idea is to use the Prony method in Algorithm 2 instead of the inverse rank-1 lattice FFTs and 1d iFFTs in step 2d, or instead of the Basis Pursuit (ℓ_1 minimization) approach from Section 2.3. In order to apply the Prony method, we need to transform our frequency detection task in the steps 2d and 2e of Algorithm 2 to the form (2.12). In step 2d for dimension increment step $t \in \{2, \dots, d\}$, we reconstruct the Fourier

coefficients $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$ of the trigonometric polynomial $g\colon \mathbb{T}^t \to \mathbb{C}$,

$$g(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})} \underbrace{\left(\sum_{\substack{\boldsymbol{h}':=(h_{t+1},\ldots,h_d)^\top\in\mathbb{Z}^d \\ (\boldsymbol{k}^\top,\boldsymbol{h}'^\top)^\top\in\operatorname{supp}\hat{p}}} \hat{p}\binom{\boldsymbol{k}}{\boldsymbol{h}'} \, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}'\cdot(x'_{t+1},\ldots,x'_d)^\top} \right)}_{=\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}} = p\left(\begin{pmatrix} \boldsymbol{x} \\ x'_{t+1} \\ \vdots \\ x'_d \end{pmatrix} \right).$$

Assuming $\mathcal{P}_{(1,\ldots,t-1)}(I^{(1,\ldots,t-1)}) \times \mathcal{P}_{(t)}(I^{(t)}) \supset \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$, we build a reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M_t, \mathcal{P}_{(1,\ldots,t-1)}(I^{(1,\ldots,t-1)}) \times \mathcal{P}_{(t)}(I^{(t)}))$ for the frequency index set $\mathcal{P}_{(1,\ldots,t-1)}(I^{(1,\ldots,t-1)}) \times \mathcal{P}_{(t)}(I^{(t)})$ according to Theorem 2.2 as in Algorithm 2, which is consequently also a reconstructing rank-1 lattice for $\mathcal{P}_{(1,\ldots,t)}(I^{(1,\ldots,t-1)} \times I^{(t)})$. Then, we choose a number $\sigma \in \{2,\ldots,M_t-1\}$ uniformly at random which is invertible modulo M_t and we use the generating vector $\tilde{\boldsymbol{z}} := \sigma\boldsymbol{z}$. Since σ is invertible module M_t, the rank-1 lattice $\Lambda(\tilde{\boldsymbol{z}}, M_t)$ is also a reconstructing rank-1 lattice for the index set $\mathcal{P}_{(1,\ldots,t-1)}(I^{(1,\ldots,t-1)}) \times \mathcal{P}_{(t)}(I^{(t)})$. We set the order $s \geq |\operatorname{supp}\hat{p}|$ as well as the vectors $\boldsymbol{c} = (c_1,\ldots,c_s) := (\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}})_{\boldsymbol{k}\in\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})}$ and $\boldsymbol{f} = (f_1,\ldots,f_s) := (\frac{2\pi\mathrm{i}}{M_t}(\boldsymbol{k}\cdot\tilde{\boldsymbol{z}} \bmod M_t))_{\boldsymbol{k}\in\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})}$. If the imaginary part of an entry f_m, $m=1,\ldots,s$, of the vector $\boldsymbol{f}$ is $\geq \pi$, we subtract $2\pi\mathrm{i}$. Finally, we obtain

$$g(\boldsymbol{x}_j) = \sum_{\boldsymbol{k}\in\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})} \tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}} \, \mathrm{e}^{\frac{2\pi\mathrm{i}}{M_t} j\,(\boldsymbol{k}\cdot\tilde{\boldsymbol{z}})} = \sum_{m=1}^{s} c_m \, \mathrm{e}^{f_m\, j} = h(j), \qquad j = 0,\ldots,M_t-1, \tag{2.13}$$

and this corresponds to (2.12) for $x = 0,\ldots,M_t-1$. Then, the Prony method uses the first $L = \mathcal{O}(s)$ samples of the exponential sum h at the nodes $x = 0,\ldots,L-1$, which is equivalent to the samples $g(\boldsymbol{x}_j) = p((\boldsymbol{x}_j^\top, x'_{t+1},\ldots,x'_d)^\top)$, $j=0,\ldots,L-1$, where the nodes $\boldsymbol{x}_j := \frac{j}{M}\boldsymbol{z} \bmod \boldsymbol{1}$ and the number of samples $L \ll M$.

The course of action for determining the frequency index set $I^{(1,\ldots,t)}$ is as follows. In step 2b, we determine $S_t := \min\left\{m\in\mathbb{N}\colon |\{k_t \bmod m\colon k_t \in I^{(t)}\}| = |I^{(t)}|\right\}$ and choose random values $x'_{t+1},\ldots,x'_d \in \mathbb{T}$ as before. Additionally, we draw a number $\sigma \in \{2,\ldots,M_t-1\}$ uniformly at random which is invertible modulo M_t. Then, we use the sampling set

$$\mathcal{X}^{(1,\ldots,t)} := (\tfrac{j}{M_{t-1}\cdot S_t}\sigma(z_1,\ldots,z_{t-1},M_{t-1})^\top)_{j=0}^{L-1} \times \{x'_{t+1}\} \times \ldots \times \{x'_d\},$$

where $L = \mathcal{O}(|\operatorname{supp}\hat{p}|)$, $L \leq M_{t-1}\cdot S_t$, is the number of samples. This means, we use the first L nodes of the reconstructing rank-1 lattice $\Lambda(\tilde{\boldsymbol{z}}, M_t, (I^{(1,\ldots,t-1)} \cap \mathcal{P}_{(1,\ldots,t-1)}(\Gamma)) \times (I^{(t)} \cap \mathcal{P}_{(t)}(\Gamma)))$, where $M_t := M_{t-1}\cdot S_t$ and $\tilde{\boldsymbol{z}} = \sigma(z_1,\ldots,z_{t-1},M_{t-1})^\top$, cf. Theorem 2.2. We sample the trigonometric polynomial p along this sampling set in step 2c and we apply the Prony method with order $s = |\operatorname{supp}\hat{p}|$ in step 2d which yields s distinct complex numbers $f_1,\ldots,f_s$. Next, we determine the frequency index $I^{(1,\ldots,t)}$ based on these distinct complex numbers f_m, $m=1,\ldots,s$. For this, we start with an empty index set $I^{(1,\ldots,t)}$, we compute $\tilde{f}_m := \operatorname{round}(\operatorname{Im}(\log(\frac{M_{t-1}\tilde{N}_t}{2\pi}f_m)))$ for each $m=1,\ldots,s$, and we try to determine the frequency $\boldsymbol{k} \in I^{(1,\ldots,t-1)} \times I^{(t)}$ for which $\boldsymbol{k}\cdot\tilde{\boldsymbol{z}} \equiv f_m \pmod{M_{t-1}\cdot\tilde{N}_t}$. If this frequency $\boldsymbol{k}$ exists, we add it to the index set $I^{(1,\ldots,t)}$. Note that the output of the Prony method has to be numerically correct up to an absolute error of $\mathrm{e}^{-\frac{M_{t-1}\tilde{N}_t}{\pi}}$ (besides other assumptions) in order to guarantee the correct frequency detection, i.e., $I^{(1,\ldots,t)} = \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$.

Assuming $\Gamma = \hat{G}_N^d$, we use $r\,2(N+1) + r\,\mathcal{O}(s)$ many samples in each dimension increment step $t \in \{2,\dots,d\}$ and $r\,2(N+1)$ many samples in the beginning. In each dimension increment step t, the Prony method is performed $r \in \mathbb{N}$ times and requires $\mathcal{O}(s^3)$ many arithmethic operations. Moreover, the rank-1 lattice search requires no more than $6\,r\,s\,\max\{2r^2s^2, 3N\}$ arithmetic operations for each dimension increment step $t \in \{2,\dots,d\}$, see Section 2.2.3. In total, we require $\mathcal{O}(d\,s + d\,N)$ many samples as well as $\mathcal{O}(d\,s^3)$ and $\mathcal{O}(d\,s\,N + d\,N\log N)$ arithmetic operations in the case $\sqrt{N} \lesssim s \lesssim N^d$ and $s \lesssim \sqrt{N}$, respectively, if $\Gamma = \hat{G}_N^d$.

We apply this version of Prony's method in Section 3 in Example 3.5.

3 Numerical results

In the following, we verify the methods from Section 2. In Section 3.1, we randomly generate s-sparse multivariate trigonometric polynomials $p\colon \mathbb{T}^d \to \mathbb{C}$, $s \ll |\hat{G}_N^d|$, and exactly reconstruct the frequencies $\boldsymbol{k} \in \operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_N^d$ belonging to the non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}} \neq 0$ with the methods described in Section 2.2, 2.3 and 2.4. Furthermore, we apply the methods from Section 2.2 and 2.3 on trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic crosses in Section 3.2 where we only assume $\operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_N^d$ during the dimension incremental reconstruction. We also test our reconstruction method from Section 2.2 on a 10-dimensional function in Section 3.3 and we test the robustness to noise in Section 3.4. For the tests, the algorithms described in Section 2 were implemented in MATLAB. The search for the reconstructing rank-1 lattice in step 2b of Algorithm 1 as well as in the additional step 2f of Algorithm 1 and 2 were implemented in C with OpenMP support and these C implementations are called from the MATLAB code using the MATLAB MEX interface. Instead of using [24, Algorithm 3.5] for the reduction of the rank-1 lattice size, we implemented a bisection method. All numerical computations were performed using double-precision floating-point arithmetic. Almost all numerical tests were run on a computer with 4x Intel Xeon CPU E5-4640 (in total 32 CPU cores) and 512 GB RAM. Time measurements were taken an a computer with an Intel i7-970 CPU (3.2 GHz) and 24 GB RAM while using only one thread. An implementation of Algorithm 1 and 2 is available online on the homepage of the authors.

The aim of the following subsections is a detailed investigation of the algorithms from Sections 2 with respect to different aspects such as number of sampling points, accuracy and computational time. In Subsection 3.1, we consider all algorithms from Sections 2.2, 2.3 and 2.4 on sparse trigonometric polynomials with random frequencies and corresponding random Fourier coefficients. Next, in Subsection 3.2, we apply the algorithms from Sections 2.2 and 2.3 on trigonometric polynomials with frequencies supported on weighted hyperbolic crosses and random Fourier coefficients. In Subsection 3.3, we demonstrate using Algorithm 1 and 2 from Section 2.2 to approximately reconstruct the largest Fourier coefficients of a 10-dimensional periodic tensor-product function of dominating mixed smoothness, which has infinitely many non-zero Fourier coefficients and we assume only $\operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_{32}^{10}$. Finally, in Section 3.4, we perturb sparse trigonometric polynomials with random frequencies using complex Gaussian noise with various signal-to-noise ratios.

3.1 Random sparse trigonometric polynomial

We set the refinement $N := 32$ and construct random multivariate trigonometric polynomials p with frequencies supported within the cube $\hat{G}_{32}^d = [-32, 32]^d \cap \mathbb{Z}^d$. This means, we randomly

choose $|\operatorname{supp}\hat{p}|$ frequencies $\boldsymbol{k} \in \hat{G}_{32}^d$ and corresponding Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in [-1,1) + [-1,1)\mathrm{i}$, $|\hat{p}_{\boldsymbol{k}}| \geq 10^{-6}$, $\boldsymbol{k} \in I = \operatorname{supp}\hat{p}$. For the reconstruction of the trigonometric polynomials p, we only assume $\operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_{32}^d$. Except for Example 3.5, we do not truncate the frequency index sets of detected frequencies $I^{(1,\ldots,t)}$, $t \in \{2,\ldots,d\}$, i.e., we set the sparsity parameter $s := |\Gamma|$. Moreover, we set the number of detection iterations $r := 1$. All tests are repeated 10 times with newly chosen frequencies and Fourier coefficients.

Example 3.1. *Sampling along reconstructing rank-1 lattices using Algorithm 1 ("A1-R1L").* We set the threshold parameter $\theta := 10^{-12}$. For the sparsities $|\operatorname{supp}\hat{p}| \in \{1\,000, 10\,000\}$, we applied Algorithm 1. In the cases $|\operatorname{supp}\hat{p}| = 1\,000$ and $|\operatorname{supp}\hat{p}| = 10\,000$, we ran the tests for dimensions $d \in \{3,4,\ldots,10,15,20,25,30\}$ and $d \in \{3,4,5,6\}$, respectively. In each test, all frequencies were successfully detected, $I^{(1,\ldots,d)} = \operatorname{supp}\hat{p}$. The used parameters and results are presented in Table 3.1. The column "max cand" shows the maximal number $\max_{t=2,\ldots,d} |I^{(1,\ldots,t-1)} \times I^{(t)}|$ of frequency candidates of all 10 repetitions and "max M" the maximal rank-1 lattice size used. Furthermore, the total number of samples for each repetition was computed and the maximum of these numbers for the 10 repetitions can be found in the column "#samples". The relative ℓ_2-error $\|(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k}\in I} - (\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k}\in I}\|_2 / \|(\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k}\in I}\|_2$ of the computed coefficients $(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k}\in I^{(1,\ldots,d)}}$ was determined for each repetition, where $I := \operatorname{supp}\hat{p} \cup I^{(1,\ldots,d)}$ and $\tilde{\hat{p}}_{\boldsymbol{k}} := 0$ for $\boldsymbol{k} \in I \setminus I^{(1,\ldots,d)}$, and the column "rel. ℓ_2-error" contains the maximal value of the 10 repetitions. In all tests, the relative ℓ_2-error is smaller than 10^{-14}. The numbers of used samples increase for increasing dimensions d and sparsities $|\operatorname{supp}\hat{p}|$ of the trigonometric polynomials p. Compared to the cardinality of the full grids $|\Gamma| = |\hat{G}_N^d|$, the observed numbers of samples are still moderate. □

N	d	$\lvert\operatorname{supp}\hat{p}\rvert$	$\lvert\Gamma\rvert = \lvert\hat{G}_N^d\rvert$	max cand	max M	#samples	rel. ℓ_2-error
32	3	1 000	274 625	53 365	142 870	145 275	4.5e-16
32	4	1 000	17 850 625	64 870	2 331 030	2 472 145	8.3e-16
32	5	1 000	1.16e+09	65 000	2 935 419	4 979 314	8.9e-16
32	6	1 000	7.54e+10	65 000	2 655 816	7 479 265	7.0e-16
32	7	1 000	4.90e+12	65 000	2 685 234	9 905 378	6.2e-16
32	8	1 000	3.19e+14	65 000	2 665 578	11 820 279	7.8e-16
32	9	1 000	2.07e+16	65 000	2 690 118	14 531 442	6.1e-16
32	10	1 000	1.35e+18	65 000	2 714 623	16 986 369	1.3e-15
32	15	1 000	1.56e+27	65 000	2 827 045	30 461 941	5.0e-16
32	20	1 000	1.81e+36	65 000	2 836 998	42 580 486	7.6e-16
32	25	1 000	2.10e+45	65 000	2 978 356	56 432 050	5.5e-16
32	30	1 000	2.44e+54	65 000	2 920 928	68 237 645	4.3e-16
32	3	10 000	274 625	143 585	147 810	150 280	5.0e-16
32	4	10 000	17 850 625	629 200	9 023 625	9 165 390	6.7e-16
32	5	10 000	1.16e+09	649 740	137 285 053	146 360 548	1.3e-15
32	6	10 000	7.54e+10	650 000	162 562 853	309 453 235	1.1e-15

Table 3.1: Results for random sparse trigonometric polynomials using reconstructing rank-1 lattices and Algorithm 1 when considering frequencies within $\Gamma = \hat{G}_{32}^d$.

Example 3.2. *Sampling along reconstructing rank-1 lattices using Algorithm 2 ("A2-R1L").* We set the threshold parameter $\theta := 10^{-12}$. For the sparsities $|\operatorname{supp}\hat{p}| \in \{1\,000, 10\,000\}$

and dimensions $d \in \{3, 4, \ldots, 10, 15, 20, 25, 30\}$, we applied Algorithm 2. In each test, all frequencies were successfully detected, $I^{(1,\ldots,d)} = \operatorname{supp} \hat{p}$. The numerical results are presented in Table 3.2, where the column names have the same meaning as described in Example 3.1. The relative ℓ_2 errors are similar to the ones for Algorithm 1 in Table 3.1. In this example, the maximal rank-1 lattices are larger compared to the results Algorithm 1 in Table 3.1, since the reconstructing rank-1 lattices are searched for in Algorithm 1 whereas they are explicitly constructed in Algorithm 2, cf. Theorem 2.2. Correspondingly, the numbers of samples are slightly higher in this example compared to the results when using Algorithm 1 for identical parameters N, d and sparsity $|\operatorname{supp} \hat{p}|$. However, the runtime of the algorithms can differ significantly, see Example 3.6. □

d	$\lvert\operatorname{supp} \hat{p}\rvert$	max cand	max M	#samples	rel. ℓ_2-error
3	1 000	58 695	272 155	276 575	4.8e-16
4	1 000	65 000	2 562 040	2 838 810	8.2e-16
5	1 000	65 000	2 735 720	5 262 140	5.0e-16
6	1 000	65 000	2 761 655	8 139 560	6.4e-16
7	1 000	65 000	2 795 390	10 953 150	4.8e-16
8	1 000	65 000	3 052 335	13 145 275	9.1e-16
9	1 000	65 000	2 932 085	16 339 115	8.0e-16
10	1 000	65 000	3 056 560	18 674 565	4.5e-16
15	1 000	65 000	3 007 095	31 954 910	5.2e-16
20	1 000	65 000	3 056 560	46 572 500	4.4e-16
25	1 000	65 000	3 149 055	58 568 770	7.3e-16
30	1 000	65 000	3 068 000	73 665 475	6.2e-16
3	10 000	251 030	274 625	279 045	2.1e-16
4	10 000	639 795	17 463 745	17 742 855	6.2e-16
5	10 000	649 935	181 940 460	199 581 915	1.1e-15
6	10 000	650 000	192 287 810	392 345 005	8.9e-16
7	10 000	650 000	194 595 570	572 814 190	6.8e-16
8	10 000	650 000	197 127 645	745 706 455	8.9e-16
9	10 000	650 000	203 536 385	967 031 390	5.8e-16
10	10 000	650 000	200 068 050	1 132 939 795	9.0e-16
15	10 000	650 000	197 036 775	2 050 649 770	5.8e-16
20	10 000	650 000	200 385 055	2 959 435 895	7.4e-16
25	10 000	650 000	206 296 415	3 959 584 980	6.5e-16
30	10 000	650 000	203 592 740	4 924 539 100	6.9e-16

Table 3.2: Results for random sparse trigonometric polynomials using reconstructing rank-1 lattices and Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_{32}^d$.

Next, we successfully applied the modifications described in Section 2.3 and used less samples to reconstruct the trigonometric polynomials.

Example 3.3. *Sub-sampling along reconstructing rank-1 lattices using ℓ_1 minimization ("A2-ℓ_1-sR1L").* We set the threshold parameter $\theta := 10^{-6}$. For the sparsity $|\operatorname{supp} \hat{p}| = 1\,000$ and dimensions $d \in \{3, 4, \ldots, 10, 15, 20, 25, 30\}$, we used sub-sampling on reconstructing rank-1 lattices as explained in Section 2.3.1. For this, we only considered $L = 10 \cdot |\operatorname{supp} \hat{p}|$ many

samples of the reconstructing rank-1 lattice generated in step 2c in each dimension increment step t of the dimension incremental method. We applied the ℓ_1 minimization algorithm SPGL1, where we set the parameter "optimality tolerance" to 10^{-7}, "Basis pursuit tolerance" to 10^{-8} as well as the maximal number of SPGL1 iterations to 2000, and used one-dimensional FFTs to compute the matrix-vector products of the corresponding Fourier matrices and vectors. The numerical results are presented in Table 3.3. We observe that, in most cases, the maximal number of frequency candidates "max cand" and the maximal rank-1 lattice sizes "max M" are similar to the ones from Table 3.2, where we used all rank-1 lattice samples. However, the total number of samples in the column "#samples" is more than two orders of magnitude smaller when we use the sub-sampling, while the relative ℓ_2-error is still less than 10^{-10}. We remark that in one test run out of the 10 runs of the numerical tests for $d = 7$, dimension increment step $t = 3$ of the dimension incremental algorithm returned a large index set of frequencies ($|I^{(1,2,3)}| = 5\,466$) and consequently many frequency candidates ($|I^{(1,\dots,3)} \times I^{(4)}| = 355\,290$) existed for dimension increment step $t = 4$, which was about 5 times larger than for the other nine runs, and this yielded a very large rank-1 lattice size ($M = 16\,229\,850$) for dimension increment step $t = 4$, which was about 6 times larger than for the other nine runs. Nevertheless, the resulting index set of detected frequencies at the end of dimension increment step $t = 4$ was again of the expected size ($|I^{(1,\dots,4)}| = |\operatorname{supp} \hat{p}| = 1\,000$) and the final results of the test run were correct. Moreover, the total number of samples was similar for all the 10 test runs of the case $d = 7$. □

d	max cand	max M	#samples	rel. ℓ_2-error
3	59 085	273 780	14 420	7.0e-11
4	65 000	2 678 520	24 485	8.9e-11
5	65 000	2 693 795	34 550	7.3e-11
6	65 000	2 730 780	44 615	8.5e-11
7	355 290	16 229 850	54 680	9.1e-11
8	65 000	2 897 180	64 745	7.8e-11
9	65 000	3 080 220	74 810	9.1e-11
10	65 000	2 949 180	84 875	9.0e-11
15	65 000	3 197 870	135 200	9.0e-11
20	65 000	3 138 265	185 525	7.6e-11
25	65 000	3 026 335	235 850	9.3e-11
30	65 000	3 113 110	286 175	8.8e-11

Table 3.3: Results for random sparse trigonometric polynomials with $|\operatorname{supp} \hat{p}| = 1\,000$ using ℓ_1 minimization with sub-sampling on reconstructing rank-1 lattices from Section 2.3.1 when considering frequencies within $\Gamma = \hat{G}^d_{32}$.

Example 3.4. *Sub-sampling along generated sets using ℓ_1 minimization ("ℓ_1-GS").* We set the threshold parameter $\theta := 10^{-6}$. For the sparsity $|\operatorname{supp} \hat{p}| = 1\,000$ and dimensions $d \in \{3, 4, \dots, 10, 15, 20, 25, 30\}$, we considered sampling on $K = 3$ random shifted generated sets as described in Section 2.3.2 using totally $L = 10 \cdot |\operatorname{supp} \hat{p}|$ many samples in step 2c in each dimension increment step t of the dimension incremental method. We applied the ℓ_1 minimization algorithm SPGL1 with the parameters from Example 3.3. All test runs were successful and the results are shown in Table 3.4. The total numbers of samples in the column

"#samples" and the relative ℓ_2-errors are slightly higher compared to the results in Table 3.3 when sub-sampling on the rank-1 lattices. □

d	max cand	#samples	rel. ℓ_2-error
3	274 105	20 195	2.4e-10
4	65 000	30 260	1.1e-10
5	65 000	40 325	1.0e-10
6	65 000	50 390	9.8e-11
7	65 000	60 455	9.2e-11
8	65 000	70 520	1.1e-10
9	65 000	80 585	9.3e-11
10	65 000	90 650	1.0e-10
15	65 000	140 975	9.5e-11
20	65 000	191 300	1.1e-10
25	65 000	241 625	1.2e-10
30	65 000	291 950	1.3e-10

Table 3.4: Results for random sparse trigonometric polynomials with $|\operatorname{supp}\hat{p}| = 1\,000$ using ℓ_1 minimization with random generated set samples from Section 2.3.2 when considering frequencies within $\Gamma = \hat{G}_{32}^d$.

Example 3.5. *Sub-sampling along reconstructing rank-1 lattices using Prony's method ("prony").* We set the threshold parameter $\theta := 10^{-9}$. For the sparsity $|\operatorname{supp}\hat{p}| = 1\,000$ and dimensions $d \in \{3, 4, \ldots, 10, 15, 20, 25, 30\}$, we applied Prony's method with sub-sampling on reconstructing rank-1 lattices as described in Section 2.4. We set the sparsity parameter $s := 1\,000$ and we used $L = 10 \cdot |\operatorname{supp}\hat{p}|$ many samples of the reconstructing rank-1 lattice generated in step 2c in each dimension increment step $t \in \{2, \ldots, d\}$ of the dimension incremental method. Moreover, in each dimension increment step t, we compute the coefficients $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in I^{(1,\ldots,t)}$, by solving the Vandermonde-like system (2.13) and we check if some of these values are less than $\theta \cdot \max_{\boldsymbol{k} \in I^{(1,\ldots,t)}} |\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}|$. In this case, the frequency detection failed in the current dimension increment step t and we repeatedly apply Prony's method with another randomly chosen number σ and consequently, we use additional samples. The numerical results are shown in Table 3.5. We observe that the numbers of samples are identical or slightly higher than for the ℓ_1 minimization in Table 3.3. Ideally, the numbers of samples should coincide in all cases for identical dimensions d. The higher numbers of samples are due to repeatedly applying Prony's method when the frequency detection failed in an intermediate step. The relative ℓ_2-errors in Table 3.5 are similar to the ones in Table 3.3. □

Example 3.6. *Computation times.* In Table 3.6, we compare the runtimes for the different methods considered above. We investigate the runtimes for refinement $N = 32$, dimensions $d \in \{6, 10\}$ and sparsity $|\operatorname{supp}\hat{p}| = 1\,000$ for all methods. For Algorithm 2 from Section 2.2.1, we additionally consider the sparsity $|\operatorname{supp}\hat{p}| = 10\,000$. The tests for each method and set of parameters were repeated 10 times. We present the results in Table 3.6. The "total runtime" was measured without the time required for sampling the trigonometric polynomials p. We observe that the total runtimes when using Algorithm 2 (method "A2-R1L") from Section

d	max cand	max M	#samples	rel. ℓ_2-error
3	58 240	272 610	14 420	6.0e-12
4	65 000	2 561 065	24 485	2.7e-11
5	65 000	2 660 775	44 615	9.6e-11
6	65 000	2 964 000	54 680	9.5e-12
7	65 000	2 737 540	64 745	7.7e-12
8	65 000	2 810 275	64 745	1.3e-11
9	65 000	2 920 255	84 875	3.6e-11
10	65 000	2 905 695	94 940	2.3e-11
15	65 000	3 121 365	145 265	1.9e-11
20	65 000	2 942 355	195 590	9.0e-11
25	65 000	3 048 305	255 980	1.3e-11
30	65 000	3 084 120	296 240	1.2e-11

Table 3.5: Results for random sparse trigonometric polynomials with $|\operatorname{supp} \hat{p}| = 1\,000$ using Prony's method with sub-sampling on reconstructing rank-1 lattices from Section 2.4 when considering frequencies within $\Gamma = \hat{G}_{32}^d$.

2.2.1 are dramatically smaller by about two orders of magnitude compared to the other methods. The reason for this behavior is that Algorithm 2 is a direct method which is mainly based on 1d iFFTs and only one reconstructing rank-1 lattice for the index set of detected frequencies $I^{(1,\dots,t)}$, $|I^{(1,\dots,t)}| \leq s$, is searched in the additional step 2f in each dimension increment step $t \in \{2,\dots,d\}$, whereas an additional reconstructing rank-1 lattice for the index set of frequency candidates $I^{(1,\dots,t-1)} \times I^{(t)}$ is searched for in step 2b of Algorithm 1 (method "A2-R1L"). This is apparent from the runtimes required for the rank-1 lattice constructions in column "time lattice search" in Table 3.6. Prony's method ("prony") from Section 2.4, which is based on Algorithm 2, is about six times slower than Algorithm 1 but uses distinctly less samples. The runtimes of the used Prony method are about 60 times higher compared to Algorithm 2, since internally a singular value decomposition is applied in the method "prony". The sub-sampling methods ("A2-ℓ_1-sR1L" and "ℓ_1-GS") based on ℓ_1 minimization from Sections 2.3.1 and 2.3.2 are iterative methods which require a certain number of iterations for a desired accuracy. The used numbers of iterations per dimension increment step t for our tests are shown in the column "#iterations per dim. increment step t" and the total runtimes are higher compared to Algorithm 1 and 2 but distinctly lower compared to the used Prony method. If we use more than the 10-times oversampling in Table 3.6 for the ℓ_1 minimization, then the number of iterations and the total runtime may decrease significantly. For instance, for the method "ℓ_1-GS" with $L = 30\,000$ samples using $K = 10$ generated sets of size 3 000 each, we obtained a maximal number of iterations of 98 and 112 in the cases $d = 6$ and $d = 10$, respectively, compared to 520 and 557 for $L = 10\,000$ samples in Table 3.6. Correspondingly, we observed a maximal total runtime of only 791 s and 1 655 s for $L = 30\,000$ samples in the cases $d = 6$ and $d = 10$, respectively, compared to 1 374 s and 2 678 s for $L = 10\,000$ samples in Table 3.6. □

In all our examples, the frequency detections succeeded and the Fourier coefficients were reconstructed exactly up to a small error caused by the used double-precision floating point arithmetic. We observe for the runtimes rt($\circ$) in Table 3.6 that

$$\text{rt(A2-R1L)} \ll \text{rt(A1-R1L)} < \text{rt(A2-}\ell_1\text{-sR1L)} < \text{rt(prony)} < \text{rt}(\ell_1\text{-GS}).$$

method	d	$\lvert\operatorname{supp}\hat{p}\rvert$	time lattice search (in s) min	max	avg	#iterations per dim. increment step t min	max	avg	total runtime (in s) min	max	avg
A1-R1L	6	1 000	191	247	215	1	1	1	193	249	217
A2-R1L	6	1 000	0.3	0.4	0.4	1	1	1	2.0	2.6	2.2
A2-ℓ_1-sR1L	6	1 000	0.3	0.4	0.3	1	200	120	419	632	491
ℓ_1-GS	6	1 000	-	-	-	79	520	363	1 196	1 374	1 268
prony	6	1 000	0.3	0.3	0.3	1	1	1	1 119	1 135	1 129
A1-R1L	10	1 000	608	746	662	1	1	1	612	751	667
A2-R1L	10	1 000	0.6	0.8	0.7	1	1	1	3.8	4.9	4.4
A2-ℓ_1-sR1L	10	1 000	0.6	0.9	0.7	1	159	134	959	1 197	1 117
ℓ_1-GS	10	1 000	-	-	-	91	557	392	2 445	2 678	2 583
prony	10	1 000	0.6	0.7	0.6	1	1	1	2 223	2 239	2 235
A2-R1L	6	10 000	63	215	133	1	1	1	168	324	231
A2-R1L	10	10 000	137	359	263	1	1	1	430	652	566

Table 3.6: Runtimes for random sparse trigonometric polynomial using different algorithms and methods. The methods "A1-R1L" and "A2-R1L" are Algorithm 1 and 2 from Section 2.2.1, respectively. "A2-ℓ_1-sR1L" and "ℓ_1-GS" mean ℓ_1 minimization with sub-sampling on rank-1 lattice and sampling on generated sets from Section 2.3.1 and 2.3.2, respectively. "prony" is Prony's method from Section 2.4. For "A2-ℓ_1-sR1L", "ℓ_1-GS" and "prony", $L = 10\,000$ samples were used.

Moreover, the numbers of samples #s($\circ$) behave like

$$\#\text{s(A2-}\ell_1\text{-sR1L)} < \#\text{s}(\ell_1\text{-GS}) < \#\text{s(prony)} \ll \#\text{s(A1-R1L)} < \#\text{s(A2-R1L)}$$

in most cases.

3.2 Symmetric weighted hyperbolic cross

In this test case, we reconstruct trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic crosses $H_N^{d,\gamma} := \{\boldsymbol{k} \in \mathbb{Z}^d \colon \prod_{t=1}^d \max(1, \gamma_t^{-1}|k_t|) \leq N\}$, where we only assume $\operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_N^d$ for our method from Section 2.2. As in the examples in Section 3.1, we do not truncate the frequency index sets of detected frequencies $I^{(1,\ldots,t)}$, $t \in \{2, \ldots, d\}$, i.e., we set the sparsity parameter $s := |\Gamma|$. Moreover, we set the threshold parameter $\theta := 10^{-12}$ and the number of detection iterations $r := 1$. All tests are repeated 10 times with different randomly chosen Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in [-1, 1) + [-1, 1)\mathrm{i}$, $|\hat{p}_{\boldsymbol{k}}| \geq 10^{-6}$, $\boldsymbol{k} \in I$. In each test case, all the frequencies were successfully detected, $I^{(1,\ldots,d)} = H_N^{d,\gamma}$, and coefficients $(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k} \in H_N^{d,\gamma}}$ were computed. Then, the relative ℓ_2-error $\|(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k} \in H_N^{d,\gamma}} - (\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k} \in H_N^{d,\gamma}}\|_2 / \|(\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k} \in H_N^{d,\gamma}}\|_2$ was computed.

Example 3.7. *Sampling along reconstructing rank-1 lattices ("A1-R1L" and "A2-R1L").* The used parameters and numerical results are shown in Table 3.7 for Algorithm 1 and in Table 3.8 for Algorithm 2, where the columns have the same meaning as in Section 3.1. We observe that the obtained relative ℓ_2-errors are comparable for both algorithms and the numbers of samples are slightly higher for Algorithm 2 compared to Algorithm 1, which is as expected. □

N	d	γ_2	$\lvert H_N^{d,\boldsymbol{\gamma}}\rvert$	max cand	max M	#samples	rel. ℓ_2-error
32	6	0.80	11 593	173 397	898 485	1 653 217	2.1e-16
32	8	0.80	15 477	197 081	1 349 994	4 180 523	2.9e-16
32	10	0.80	16 871	197 081	1 349 994	6 632 518	5.5e-16
16	10	0.87	22 953	200 541	1 358 148	5 039 519	3.0e-16
16	15	0.87	25 963	200 541	1 358 148	10 057 035	2.8e-16
16	20	0.87	26 185	200 541	1 358 148	12 555 880	3.3e-16
32	10	0.84	40 387	531 145	5 116 951	21 632 742	5.2e-16
32	15	0.84	44 201	531 145	5 116 951	38 955 122	2.6e-16
32	20	0.84	44 433	531 145	5 116 951	46 851 702	8.2e-16

Table 3.7: Results for trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic cross $H_N^{d,\boldsymbol{\gamma}}$ with weights $\boldsymbol{\gamma} = (1, \gamma_2, \gamma_2^2, \ldots, \gamma_2^{d-1})^\top$ using reconstructing rank-1 lattices with Algorithm 1 when considering frequencies within $\Gamma = \hat{G}_N^d$.

N	d	γ_2	$\lvert H_N^{d,\boldsymbol{\gamma}}\rvert$	max cand	max M	#samples	rel. ℓ_2-error
32	6	0.80	11 593	173 397	990 990	1745 779	2.1e-16
32	8	0.80	15 477	197 081	1 338 974	4 360 512	5.0e-16
32	10	0.80	16 871	197 081	1 430 231	6 961 062	5.5e-16
16	10	0.87	22 953	200 541	1 358 148	5 032 864	6.3e-16
16	15	0.87	25 963	200 541	1 358 148	10 175 387	7.3e-16
16	20	0.87	26 185	200 541	1 358 148	12 687 242	7.6e-16
32	10	0.84	40 387	531 145	5 337 879	23 712 165	3.4e-16
32	15	0.84	44 201	531 145	5 337 879	41 732 585	2.6e-16
32	20	0.84	44 433	531 145	5 337 879	49 777 589	2.7e-16

Table 3.8: Results for trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic cross $H_N^{d,\boldsymbol{\gamma}}$ with weights $\boldsymbol{\gamma} = (1, \gamma_2, \gamma_2^2, \ldots, \gamma_2^{d-1})^\top$ using reconstructing rank-1 lattices with Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_N^d$.

Again, we successfully applied the modifications described in Section 2.3.

Example 3.8. *Sub-sampling along reconstructing rank-1 lattices using ℓ_1 minimization ("A2-ℓ_1-sR1L").* We used sub-sampling on reconstructing rank-1 lattices as described in in Section 2.3.1. This time, we considered $L = \lvert I^{(1,\ldots,t-1)} \times I^{(t)}\rvert$ many samples of the reconstructing rank-1 lattices generated in step 2c in each dimension increment step t of the dimension incremental method. We applied the ℓ_1 minimization algorithm SPGL1, where we set the parameter "optimality tolerance" to 10^{-7}, "Basis pursuit tolerance" to 10^{-8} as well as the maximal number of SPGL1 iterations to 2000, and used one-dimensional FFTs to compute the matrix-vector products of the corresponding Fourier matrices and vectors. The numerical results are shown in Table 3.9. We observe that the maximal number of frequency candidates "max cand" and the maximal rank-1 lattice sizes "max M" are identical to the ones from Example 3.7 in Table 3.8, where we used all rank-1 lattice samples, except for one case. The total number of samples in the column "#samples" is about 5 times smaller when we use the sub-sampling, while the relative ℓ_2-error is still less than 10^{-11}. □

N	d	γ_2	$\lvert H_N^{d,\boldsymbol{\gamma}}\rvert$	max cand	max M	#samples	rel. ℓ_2-error
32	6	0.80	11 593	173 397	990 990	383 585	3.6e-12
32	8	0.80	15 477	197 081	1 338 974	763 225	3.6e-12
32	10	0.80	16 871	197 081	1 430 231	1 080 761	4.2e-12
16	10	0.87	22 953	200 541	1 358 148	867 879	3.5e-12
16	15	0.87	25 963	200 541	1 358 148	1 629 241	4.7e-12
16	20	0.87	26 185	2 061 664	12 707 266	3 882 026	9.4e-12
32	10	0.84	40 387	531 145	5 337 879	2 633 711	1.3e-12
32	15	0.84	44 201	531 145	5 337 879	4 283 003	3.3e-12
32	20	0.84	44 433	531 145	5 337 879	5 036 917	6.0e-12

Table 3.9: Results for trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic cross $H_N^{d,\boldsymbol{\gamma}}$ with weights $\boldsymbol{\gamma} = (1, \gamma_2, \gamma_2^2, \ldots, \gamma_2^{d-1})^\top$ using ℓ_1 minimization with sub-sampling on reconstructing rank-1 lattices with Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_N^d$.

Example 3.9. *Sub-sampling along generated sets using ℓ_1 minimization ("ℓ_1-GS").* We considered sampling on $K = 3$ random shifted generated sets as described in Section 2.3.2. and we used totally $L = \lvert I^{(1,\ldots,t-1)} \times I^{(t)}\rvert$ many samples in step 2c in each dimension increment step of the dimension incremental method. All test runs were successful and the results are presented in Table 3.10. The total numbers of samples in the column "#samples" is similar to the results from Example 3.9 in Table 3.9 when sub-sampling on the rank-1 lattices and the relative ℓ_2-errors are still less than 10^{-9} in Table 3.10. □

N	d	γ_2	$\lvert H_N^{d,\boldsymbol{\gamma}}\rvert$	max cand	#samples	rel. ℓ_2-error
32	6	0.80	11 593	173 397	383 585	1.2e-10
32	8	0.80	15 477	197 081	763 225	1.3e-10
32	10	0.80	16 871	197 081	1 098 746	1.1e-10
16	10	0.87	22 953	200 541	884 929	1.1e-10
16	15	0.87	25 963	200 541	1 630 891	8.7e-11
16	20	0.87	26 185	200 541	2 021 078	3.2e-11
32	10	0.84	40 387	531 145	2 633 711	3.2e-10
32	15	0.84	44 201	531 145	4 283 138	2.2e-10
32	20	0.84	44 433	531 145	5 164 672	1.5e-10

Table 3.10: Results for trigonometric polynomials with frequencies supported on symmetric weighted hyperbolic cross $H_N^{d,\boldsymbol{\gamma}}$ with weights $\boldsymbol{\gamma} = (1, \gamma_2, \gamma_2^2, \ldots, \gamma_2^{d-1})^\top$ using ℓ_1 minimization with random generated set samples with Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_N^d$.

3.3 Tensor-product function

Next, we apply our method from Section 2.2 to a multivariate periodic function $f\colon \mathbb{T}^d \to \mathbb{C}$, which is not sparse in frequency domain. For this, we perform all the steps as described in Section 2.2.1. However, we also have to take into consideration that the situation may

occur where the index set $I^{(t)} = [-N, N] \cap \mathbb{Z}$ after computing (2.5) and the index sets $I^{(1,\dots,t)} = [-N, N]^t \cap \mathbb{Z}^t$ after computing (2.7) for some or all $t \in \{2, \dots, d\}$. The resulting index set of detected frequencies $I^{(1,\dots,d)}$ could be the full cube $\hat{G}_N^d$ or a subset with cardinality of the same magnitude. Correspondingly, for a general search space $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, the index sets $I^{(1,\dots,t)}$ could be $\mathcal{P}_{(1,\dots,t)}(\Gamma)$ for some or all $t \in \{2, \dots, d\}$ and the resulting index set of detected frequencies $I^{(1,\dots,d)}$ could be the search space Γ itself. Therefore, we apply strategies to truncate the index sets $I^{(t)}$ and $I^{(1,\dots,t)}$, $t \in \{2, \dots, d\}$, in Section 3.3.1 and 3.3.2.

Here, we consider the function $f\colon \mathbb{T}^{10} \to \mathbb{R}$,

$$f((x_1, \dots, x_{10})^\top) := \prod_{t \in \{1,3,8\}} N_2(x_t) + \prod_{t \in \{2,5,6,10\}} N_4(x_t) + \prod_{t \in \{4,7,9\}} N_6(x_t), \tag{3.1}$$

where $N_m : \mathbb{T} \to \mathbb{R}$ is the B-Spline of order $m \in \mathbb{N}$,

$$N_m(x) := C_m \sum_{k \in \mathbb{Z}} \operatorname{sinc}\left(\frac{\pi}{m} k\right)^m \cos(\pi k)\, \mathrm{e}^{2\pi \mathrm{i} k x},$$

with a constant $C_m > 0$ such that $\|N_m | L^2(\mathbb{T})\| = 1$. We approximate the function f by trigonometric polynomials (1.1). For this, we determine a frequency index set $I = I^{(1,\dots,10)} \subset \Gamma = \hat{G}_N^{10}$ and compute approximated Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, from sampling values of f as described in Section 2.2.1. We expect the frequency index set to "consist of" three manifolds, a three-dimensional symmetric hyperbolic cross in the dimensions $1, 3, 8$, a four-dimensional symmetric hyperbolic cross in the dimensions $2, 5, 6, 10$ and a three-dimensional symmetric hyperbolic cross in the dimensions $4, 7, 9$. Furthermore, the cardinality $|I|$ should be $\mathcal{O}(N \log^3 N)$ and the largest rank-1 lattice of size $M = \mathcal{O}(N^3 \log^2 N)$. All tests were run 10 times and the relative $L^2(\mathbb{T}^{10})$ approximation error

$$\|f - \tilde{S}_I f | L^2(\mathbb{T}^{10})\| / \|f | L^2(\mathbb{T}^{10})\| = \sqrt{\|f | L^2(\mathbb{T}^{10})\|^2 - \sum_{\boldsymbol{k} \in I} |\hat{f}_{\boldsymbol{k}}|^2 + \sum_{\boldsymbol{k} \in I} |\tilde{\hat{p}}_{\boldsymbol{k}} - \hat{f}_{\boldsymbol{k}}|^2} / \|f | L^2(\mathbb{T}^{10})\|$$

was computed, where $\tilde{S}_I f := \sum_{\boldsymbol{k} \in I} \tilde{\hat{p}}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \circ}$.

3.3.1 s-sparse

One possibility is to use the sparsity input parameter $s \in \mathbb{N}$ of Algorithm 1 and 2. Consequently, the detected frequencies in each dimension increment step t are truncated and only those frequencies are used which belong to the $s \ll |\Gamma| < \infty$ largest Fourier coefficients $\tilde{\hat{p}}_{t,k_t}$ and $\tilde{\hat{p}}_{(1,\dots,t),\boldsymbol{k}}$. The relative threshold parameter $\theta \in (0, 1)$ should then be set to a very small value like $\theta := 10^{-12}$. Due to possible aliasing effects, see (2.6) and (2.8), it might be reasonable to use a larger value of s for the intermediate dimension increment steps $t \in \{2, \dots, d-1\}$ than for the final truncation of the index set $I^{(1,\dots,d)}$ in dimension increment step $t = d$.

Example 3.10. *s-sparse approximate reconstruction of a function ("A1-R1L" and "A2-R1L").* We set $N = 16, 32, 64$ and search for frequencies within the cube $\Gamma = \hat{G}_N^{10}$. We use $r := 5$ detection iterations and set the relative threshold parameter $\theta := 10^{-7}$. The used parameters and results are presented in Table 3.11 for Algorithm 1 and in Table 3.12 for Algorithm 2. In the column "sparsity", two parameters are found. The first one is the maximal number of frequencies belonging to the largest Fourier coefficients, which are used

for the approximate reconstruction. The second sparsity number is the maximal number $|I^{(1,\dots,t)}|$, $t = 1, \dots, d-1$, of frequencies and Fourier coefficients kept during the computation. Here, the column "rel. L^2-error" contains the relative $L^2(\mathbb{T}^{10})$ approximation error $\|f - \tilde{S}_I f | L^2(\mathbb{T}^{10})\| / \|f | L^2(\mathbb{T}^{10})\|$. The remaining columns have the same meaning as described in Section 3.1. We observe that for increasing sparsity parameter, the number of frequency candidates and samples increases while the relative $L^2(\mathbb{T}^{10})$ approximation error decreases. Furthermore, it is not sufficient to only increase the used sparsity s but the refinement parameter N also needs to be increased. Using a large refinement parameter N and a small target sparsity s results in the usage of distinctly more samples, e.g., about 41 million samples for $N = 16$ and sparsity $s = 1000$ compared to about 156 million samples for $N = 64$ in Table 3.11. The relative $L^2(\mathbb{T}^{10})$ approximation errors for Algorithm 1 and 2 are almost identical. □

N	sparsity	max cand	max M	#samples	rel. L^2-error
16	1 000/ 2 000	105 468	2 005 179	40 776 032	1.2e-02
16	2 000/ 4 000	213 345	7 400 212	109 229 485	4.3e-03
16	3 000/ 6 000	302 610	10 731 031	210 260 190	3.5e-03
16	4 000/ 8 000	402 468	16 554 352	306 647 729	3.3e-03
32	1 000/ 2 000	225 875	4 387 563	62 971 360	1.2e-02
32	2 000/ 4 000	392 795	15 793 929	232 962 422	3.4e-03
32	3 000/ 6 000	597 155	26 146 120	439 980 245	1.7e-03
32	4 000/ 8 000	830 375	40 409 497	686 588 714	1.4e-03
32	5 000/10 000	1 021 410	56 177 093	949 349 167	1.2e-03
64	1 000/ 2 000	483 105	5 541 810	155 887 998	1.2e-02
64	2 000/ 4 000	913 257	16 775 973	354 498 370	3.4e-03
64	3 000/ 6 000	1 167 321	28 981 586	591 378 719	1.6e-03
64	4 000/ 8 000	1 512 654	44 180 388	819 754 426	9.8e-04
64	5 000/10 000	1 982 214	136 551 319	2 170 526 041	7.1e-04
64	6 000/12 000	2 256 790	199 917 497	2 991 975 918	5.6e-04

Table 3.11: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 1 when considering frequencies within $\Gamma = \hat{G}_N^{10}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

Example 3.11. *s-sparse approximate reconstruction of a function restricting the search space ("A1-R1L" with $\Gamma = H_N^{d,\mathbf{1}}$).* In this example, we use the identical test sets and input parameters as in Example 3.10 except for the search space Γ. If we assume that the frequencies belonging to the largest Fourier coefficients lie within a hyperbolic cross $\Gamma = H_N^{10,\mathbf{1}}$, i.e., if we restrict the search space for the frequencies when using Algorithm 1, then the number of frequency candidates and the total number of samples can further be reduced while obtaining almost the same relative $L^2(\mathbb{T}^{10})$ approximation errors. The numerical results for this case can be found in Table 3.13. In the case $N = 64$ and sparsity $s = 4000$, we used only about 1/5 of total samples compared with the results from Example 3.10 in Table 3.11, where we assumed that the frequencies belonging to the largest Fourier coefficients lie within the cube $\Gamma = \hat{G}_N^{10}$. □

N	sparsity	max cand	max M	#samples	rel. L^2-error
16	1 000/ 2 000	104 016	2 275 218	41 440 344	1.2e-02
16	2 000/ 4 000	200 071	7 103 613	112 486 704	4.3e-03
16	3 000/ 6 000	293 766	11 447 502	211 976 106	4.2e-03
16	4 000/ 8 000	359 271	15 939 363	290 592 654	3.3e-03
32	1 000/ 2 000	217 230	5 867 095	76 456 418	1.2e-02
32	2 000/ 4 000	375 960	13 411 060	213 581 140	3.4e-03
32	3 000/ 6 000	609 895	36 135 125	362 485 290	1.7e-03
32	4 000/ 8 000	776 295	47 140 210	590 538 705	1.4e-03
32	5 000/10 000	922 025	67 627 755	788 170 875	1.2e-03
64	1 000/ 2 000	426 990	6 017 334	72 186 288	1.2e-02
64	2 000/ 4 000	766 002	21 572 928	278 124 358	3.4e-03
64	3 000/ 6 000	1 220 727	72 115 386	698 575 406	1.6e-03
64	4 000/ 8 000	1 423 386	88 214 715	870 806 143	9.8e-04
64	5 000/10 000	1 820 190	125 092 203	1 293 939 642	7.1e-04
64	6 000/12 000	2 180 487	164 668 113	1 660 790 581	5.6e-04

Table 3.12: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_N^{10}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

N	sparsity	$\lvert H_N^{10,\mathbf{1}}\rvert$	max cand	max M	#samples	rel. L^2-error
16	1 000/ 2 000	45 548 649	17 742	512 496	7 362 160	1.2e-02
16	2 000/ 4 000	45 548 649	27 112	1 190 389	15 922 974	4.4e-03
16	3 000/ 6 000	45 548 649	37 268	1 795 031	22 501 808	3.6e-03
16	4 000/ 8 000	45 548 649	45 662	2 449 317	31 680 101	3.6e-03
32	1 000/ 2 000	182 183 661	27 740	1 027 474	14 476 254	1.2e-02
32	2 000/ 4 000	182 183 661	43 898	2 308 048	33 723 808	3.4e-03
32	3 000/ 6 000	182 183 661	55 583	3 818 404	47 146 657	1.7e-03
32	4 000/ 8 000	182 183 661	65 510	5 413 888	65 824 783	1.4e-03
32	5 000/10 000	182 183 661	74 514	7 232 979	86 519 822	1.3e-03
64	1 000/ 2 000	696 036 321	50 014	1 745 803	28 427 634	1.2e-02
64	2 000/ 4 000	696 036 321	64 596	3 891 632	64 616 902	3.4e-03
64	3 000/ 6 000	696 036 321	87 599	6 779 467	101 748 629	1.6e-03
64	4 000/ 8 000	696 036 321	106 452	10 035 867	157 725 439	9.8e-04
64	5 000/10 000	696 036 321	122 236	15 199 650	186 959 406	7.2e-04
64	6 000/12 000	696 036 321	134 195	17 019 323	225 136 643	5.6e-04

Table 3.13: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 1 when only considering frequencies within $\Gamma = H_N^{10,\mathbf{1}}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

In the following example, we compare the numerical results of the algorithms presented in this paper with the ones when applying the non-incremental, single-step algorithm from [29]. We observe that the latter one has a drastically higher number of samples for similar relative $L^2(\mathbb{T}^{10})$ approximation errors.

Example 3.12. *s-sparse approximate reconstruction of a function using a single-step algorithm.* In this example, we do not use the dimension incremental method to approximately reconstruct the largest Fourier coefficients of the function f from (3.1) but we apply the direct single-step method described in [29]. This means we have to choose frequency index sets I which contain the largest Fourier coefficients of f. Due to the tensor product structure of our function f, we used hyperbolic cross index sets $I = H_N^{10,\mathbf{1}} := \{\boldsymbol{k} \in \mathbb{Z}^d \colon \prod_{t=1}^{10} \max(1, |k_t|) \leq N\}$, $N = 4, 8, 16$, and the corresponding reconstructing rank-1 lattices for $H_N^{10,\mathbf{1}}$ from [29, Table 6.2], We sampled the function f at the rank-1 lattice nodes and computed all approximated Fourier coefficients $\hat{\tilde{f}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in H_N^{10,\mathbf{1}}$. Then, we used sparsity $s = 1000, 2000, 3000, 4000$ many of the largest of these Fourier coefficients for the function approximation. The results are shown in Table 3.14. Comparing the number of samples and obtained relative $L^2(\mathbb{T}^{10})$ approximation errors with the results of our dimension incremental method in Table 3.11 and 3.12 for $N = 16$, we observe that the errors are almost the same and the numbers of samples are dramatically larger for the single-step algorithm. This means that the dimension incremental reconstruction method required distinctly less samples than the single-step algorithm while achieving similar approximation errors. □

N	sparsity	$\lvert H_N^{10,\mathbf{1}}\rvert$	M = #samples	rel. L^2-error
4	1 000	2 421 009	30 780 958	3.8e-02
4	2 000	2 421 009	30 780 958	3.8e-02
8	1 000	10 819 089	194 144 634	1.4e-02
8	2 000	10 819 089	194 144 634	1.1e-02
8	3 000	10 819 089	194 144 634	1.1e-02
16	1 000	45 548 649	2 040 484 044	1.2e-02
16	2 000	45 548 649	2 040 484 044	4.3e-03
16	3 000	45 548 649	2 040 484 044	3.6e-03
16	4 000	45 548 649	2 040 484 044	3.6e-03

Table 3.14: Results for function $f \colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for single-step algorithm from [29] when only considering frequencies within $\Gamma = H_N^{10,\mathbf{1}}$.

3.3.2 threshold-based

Another variant is to use the relative threshold parameter $\theta \in (0, 1)$ of Algorithm 1 and 2 for the truncation. The sparsity input parameter $s \in \mathbb{N}$ is set to $|\Gamma|$. We remark that due to the aliasing (2.6) and (2.8), smaller thresholds for the intermediate dimension increment steps $t \in \{2, \ldots, d-1\}$ should be used.

We search for frequencies within the cube $\Gamma = \hat{G}_N^{10}$ for various refinements $N \in \mathbb{N}$ belonging to those frequencies above a certain relative threshold and we use $r := 10$ detection iterations.

Example 3.13. *Threshold-based approximate reconstruction of a function ("A1-R1L" and "A2-R1L").* The parameters and results are shown in Table 3.15 for Algorithm 1 and in Table 3.16 for Algorithm 2. For the truncation of the one-dimensional index sets $I^{(t)}$ of frequency candidates for component t, $t \in \{1, \ldots, 10\}$, the relative threshold parameter $\theta := 10^{-12}$ is used. Moreover, for the truncation of the final index set $I^{(1,\ldots,10)}$, the relative threshold parameter $\theta \in (0, 1)$ with the value from the column "threshold" is used and $\theta :=$"threshold"$/10$

for all other truncations. We observe that the numbers of frequency candidates, the rank-1 lattice sizes and the total numbers of samples are dramatically smaller compared to the results from Section 3.3.1 while the relative $L^2(\mathbb{T}^{10})$ approximation errors are about the same for similar numbers $|I|$ of Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$ used for the approximation $\tilde{S}_I f$ of f. Moreover, the total numbers of samples are distinctly lower when using Algorithm 1 compared to Algorithm 2. □

N	threshold	$\lvert I\rvert$	max cand	max M	#samples	rel. L^2-error
64	1.0e-02	491	3 885	21 970	254 530	1.4e-01
64	1.0e-03	1 121	27 521	217 494	2 789 050	1.1e-02
64	1.0e-04	3 013	123 195	903 906	17 836 042	1.7e-03
64	1.0e-05	7 163	256 065	7 820 238	82 222 438	4.7e-04
64	1.0e-06	19 771	1 096 335	66 734 128	439 149 744	3.9e-04
2	1.0e-02	439	1 325	5 860	72 205	1.4e-01
4	1.0e-03	1 039	4 571	36 554	494 564	2.8e-02
8	1.0e-04	2 651	16 599	236 418	3 183 298	1.1e-02
16	1.0e-04	2 807	28 611	388 083	4 912 259	3.5e-03
32	3.0e-05	4 645	82 095	1 327 468	19 226 647	1.3e-03
64	1.0e-05	7 163	256 065	7 820 238	82 222 438	4.7e-04
128	3.0e-06	13 031	849 899	21 505 318	260 000 740	1.8e-04

Table 3.15: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 1 when considering frequencies within $\Gamma = \hat{G}_N^{10}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

N	threshold	$\lvert I\rvert$	max cand	max M	#samples	rel. L^2-error
64	1.0e-02	477	8041	68 055	890 640	6.5e-02
64	1.0e-03	1 111	77 015	768 840	11 836 434	1.0e-02
64	1.0e-04	2 991	251 679	4 599 792	60 717 348	1.6e-03
64	1.0e-05	7 371	661 059	39 765 728	338 464 342	4.7e-04
64	1.0e-06	20 371	3 091 527	312 897 648	1 989 191 578	4.0e-04
2	1.0e-02	453	2 375	11 955	196 055	9.7e-02
4	1.0e-03	1 041	11 511	147 915	2 057 733	2.8e-02
8	1.0e-04	2 683	55 607	1 183 370	20 299 628	9.2e-03
16	1.0e-04	2 799	96 575	2 255 498	22 864 862	3.4e-03
32	3.0e-05	4 623	297 185	14 475 426	77 830 316	1.2e-03
64	1.0e-05	7 369	665 861	36 740 738	322 531 170	4.7e-04
128	3.0e-06	13 381	2 110 227	166 717 512	1 344 032 822	1.7e-04

Table 3.16: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 2 when considering frequencies within $\Gamma = \hat{G}_N^{10}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

Example 3.14. *Threshold-based approximate reconstruction of a function restricting the search space ("A1-R1L" with $\Gamma = H_N^{d,\mathbf{1}}$).* Again, if we assume that the frequencies belonging

to the largest Fourier coefficients of f lie within the hyperbolic cross, $\Gamma = H_N^{10,\mathbf{1}}$, we can distinctly reduce the total number of function samples when using Algorithm 1 while obtaining similar relative $L_2(\mathbb{T}^d)$ approximation errors. We used the identical test parameters as in Example 3.13 except for the search space Γ. The numerical results for this assumption are presented in Table 3.17. □

N	threshold	$\lvert H_N^{10,\mathbf{1}}\rvert$	$\lvert I\rvert$	max cand	max M	#samples	rel. L^2-error
64	1.0e-02	696 036 321	479	3 613	20 033	257 096	7.1e-02
64	1.0e-03	696 036 321	1 101	10 623	107 837	1 662 533	1.1e-02
64	1.0e-04	696 036 321	3 009	27 561	617 400	7 740 420	1.7e-03
64	1.0e-05	696 036 321	6 923	46 373	2 015 127	19 288 758	5.1e-04
64	1.0e-06	696 036 321	14 085	66 987	5 408 176	42 893 192	4.3e-04
2	1.0e-02	452 709	399	869	3 828	49 469	1.3e-01
4	1.0e-03	2 421 009	927	2 427	19 150	267 538	3.9e-02
8	1.0e-04	10 819 089	2 301	6 407	85 592	1 101 462	1.1e-02
16	1.0e-04	45 548 649	2 655	9 881	164 590	2 071 588	3.6e-03
32	3.0e-05	182 183 661	4 301	20 991	601 490	5 679 603	2.7e-03
64	1.0e-05	696 036 321	6 937	46 335	2 198 884	20 968 600	5.1e-04
128	3.0e-06	2.53e+09	12 637	104 409	8 400 796	73 500 131	1.8e-04

Table 3.17: Results for function $f\colon \mathbb{T}^{10} \to \mathbb{R}$ from (3.1) for Algorithm 1 when only considering frequencies within $\Gamma = H_N^{10,\mathbf{1}}$. "#samples" means worst case number of function evaluations for 1 test run (out of the 10 runs).

3.4 Random sparse trigonometric polynomial with complex Gaussian noise

In this subsection, we test the robustness to noise of our method from Section 2.2. We construct random multivariate trigonometric polynomials p with frequencies supported within the cube $\hat{G}_N^d = [-N, N]^d \cap \mathbb{Z}^d$. In doing so, we randomly choose $|\operatorname{supp}\hat{p}|$ many frequencies $\boldsymbol{k} \in \hat{G}_N^d$ and we set the corresponding Fourier coefficients $\hat{p}_{\boldsymbol{k}} := \mathrm{e}^{2\pi\mathrm{i}\varphi_{\boldsymbol{k}}} \in \mathbb{C}$, $|\hat{p}_{\boldsymbol{k}}| = 1$, $\boldsymbol{k} \in I = \operatorname{supp}\hat{p}$, where the angles $\varphi_{\boldsymbol{k}} \in [0, 1)$ are chosen uniformly at random. For the reconstruction of the trigonometric polynomials p, we only assume $\operatorname{supp}\hat{p} \subset \Gamma = \hat{G}_N^d$. We perturb the samples $p(\boldsymbol{x}_j)$ taken at nodes $\boldsymbol{x}_j \in \mathbb{T}^d$, $j = 0, \dots, M-1$, of the trigonometric polynomial p by additive complex white Gaussian noise $\eta_j \in \mathbb{C}$ with zero mean and standard deviation σ, i.e., we have measurements $f(\boldsymbol{x}_j) = p(\boldsymbol{x}_j) + \eta_j$. Then, we may approximately compute the signal-to-noise ratio (SNR) in our case by

$$\mathrm{SNR} \approx \frac{\sum_{j=0}^{M-1} |p(\boldsymbol{x}_j)|^2/M}{\sum_{j=0}^{M-1} |\eta_j|^2/M} \approx \frac{\sum_{\boldsymbol{k}\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{k}}|^2}{\sigma^2} = \frac{|\operatorname{supp}\hat{p}|}{\sigma^2}.$$

Correspondingly, we choose $\sigma := \sqrt{|\operatorname{supp}\hat{p}|}/\sqrt{\mathrm{SNR}}$ for a targeted SNR value. For our numerical tests in MATLAB, we generate the noise by $\eta_j := \sigma/\sqrt{2}$ `* (randn + 1i*randn)`, $j = 0, \dots, M-1$. The SNR is often measured using the logarithmic decibel scale (dB), where $\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}\mathrm{SNR}$ and $\mathrm{SNR} = 10^{\mathrm{SNR}_{\mathrm{dB}}/10}$, i.e., a linear $\mathrm{SNR} = 10^8$ corresponds to a logarithmic $\mathrm{SNR}_{\mathrm{dB}} = 80\mathrm{dB}$ and $\mathrm{SNR} = 1$ corresponds to $\mathrm{SNR}_{\mathrm{dB}} = 0\mathrm{dB}$.

$\mathrm{SNR}_{\mathrm{dB}}$	noise σ	#detect. iter. r	#samples	min #freq. correct	success rate (all freq. correct)	rel. ℓ_2-error
80	3.2e-03	1	22 216 155	998	0.995	4.5e-02
70	1.0e-02	1	23 004 475	998	0.986	4.5e-02
60	3.2e-02	1	22 381 905	998	0.974	5.5e-02
50	1.0e-01	1	22 533 615	996	0.893	7.1e-02
40	3.2e-01	1	22 434 295	994	0.722	8.4e-02
30	1.0e+00	1	22 662 055	988	0.319	1.2e-01
20	3.2e+00	1	22 646 975	979	0.032	1.5e-01
10	1.0e+01	1	23 084 425	950	0.000	2.3e-01
0	3.2e+01	1	23 185 435	774	0.000	5.0e-01
80	3.2e-03	2	41 283 775	1 000	1.000	2.2e-06
70	1.0e-02	2	42 553 485	1 000	1.000	7.4e-06
60	3.2e-02	2	41 799 485	1 000	1.000	2.4e-05
50	1.0e-01	2	49 597 275	1 000	1.000	7.5e-05
40	3.2e-01	2	55 243 565	998	0.998	4.5e-02
30	1.0e+00	2	41 881 645	998	0.994	5.5e-02
20	3.2e+00	2	42 064 815	996	0.933	7.7e-02
10	1.0e+01	2	41 512 185	990	0.465	1.1e-01
0	3.2e+01	2	43 322 695	942	0.000	2.5e-01
40	3.2e-01	3	61 300 655	1 000	1.000	2.3e-04
30	1.0e+00	3	61 847 825	1 000	1.000	7.1e-04
20	3.2e+00	3	61 477 195	998	0.998	4.5e-02
10	1.0e+01	3	60 542 365	996	0.936	6.4e-02
0	3.2e+01	3	61 832 225	984	0.015	1.4e-01
20	3.2e+00	4	82 104 165	1 000	1.000	2.4e-03
10	1.0e+01	4	80 312 115	998	0.997	4.5e-02
0	3.2e+01	4	81 618 355	994	0.442	9.1e-02
20	3.2e+00	5	101 459 605	1 000	1.000	2.3e-03
10	1.0e+01	5	99 610 745	1 000	1.000	7.3e-03
0	3.2e+01	5	98 090 005	997	0.869	7.4e-02

Table 3.18: Results for random sparse trigonometric polynomials with sparsity $|\operatorname{supp} \hat{p}| = 1\,000$ perturbed by additive white Gaussian noise using reconstructing rank-1 lattices and Algorithm 2.

Example 3.15. *Sampling along reconstructing rank-1 lattices using Algorithm 2 ("A2-R1L"), where the samples are perturbed by additive complex Gaussian noise.* We choose the dimensionality $d := 10$, the refinement $N := 32$ and the sparsity $|\operatorname{supp} \hat{p}| := 1\,000$. We apply Algorithm 2 and we set the search space $\Gamma := \hat{G}_{32}^{10}$, the sparsity parameter $s := 1\,000$ as well as the threshold parameter $\theta := 10^{-12}$. The algorithm is run setting the parameter r for the number of detection iterations to $r := 1, 2, 3, 4, 5$ and using the SNR values $\mathrm{SNR}_{\mathrm{dB}} := 80, 70, \ldots, 10, 0$ (which corresponds to $\mathrm{SNR} = 10^8, 10^7, \ldots, 10, 1$). For each of these 45 test settings, Algorithm 2 is repeatedly run 1 000 times. In each of the total 45 000 test runs, new random frequencies and Fourier coefficients are drawn. The numerical results are presented in Ta-

ble 3.18. The total number of samples for each of the 1 000 repetitions was computed and the maximum of these numbers for each test setting can be found in the column "#samples". In the column "min #freq. correct", the minimal number of correctly detected frequencies $|I^{(1,\dots,10)} \cap \operatorname{supp} \hat{p}|$ for the 1 000 repetitions is shown, where $\operatorname{supp} \hat{p}$ denotes the set of true (input) frequencies of a trigonometric polynomial p and $I^{(1,\dots,10)}$ the frequencies returned by the detection algorithm. The column "success rate (all freq. correct)" represents the relative number of the 1 000 repetitions where all frequencies were successfully detected, $I^{(1,\dots,10)} = \operatorname{supp} \hat{p}$. Moreover, the relative ℓ_2-error $\|(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k}\in I} - (\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k}\in I}\|_2 / \|(\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k}\in I}\|_2$ of the computed coefficients $(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k}\in I^{(1,\dots,10)}}$ was determined for each repetition, where $I := \operatorname{supp} \hat{p} \cup I^{(1,\dots,10)}$ and $\tilde{\hat{p}}_{\boldsymbol{k}} := 0$ for $\boldsymbol{k} \in I \setminus I^{(1,\dots,10)}$, and the column "rel. ℓ_2-error" contains the maximal value of the 1 000 repetitions. For test settings which are not shown in Table 3.18 all frequencies in all 1 000 repetitions where correctly detected, i.e., the column "min #freq. correct"=1 000 and "success rate"=1.000. In general, we observe that for decreasing SNR values, the minimal number of correctly detected frequencies and the success rate decrease. When using $r = 1$ test iterations, there were always some (of the 1 000 test runs), where one or two frequencies were incorrect. However, in all test runs of all test settings, more than 77 percent of the frequencies were correctly detected, even for the case $\mathrm{SNR}_{\mathrm{dB}} = 0$ (SNR = 1) where the signal level equals the noise level. When we increased the number of detection iterations r, the SNR level at which all frequencies in all of the 1 000 test runs were correctly detected also decreased. For instance for $r = 5$ detection iterations, the success rate was at 100 percent including the case $\mathrm{SNR}_{\mathrm{dB}} = \mathrm{SNR} = 10$. However, we require about 5 times of the samples for $r = 5$ detection iterations compared to the test settings with $r = 1$. □

4 Conclusion

In this paper, we presented methods for the approximate reconstruction of the largest Fourier coefficients of high-dimensional multivariate period functions, which are sparse in frequency domain, from sampling values. In doing so, it is assumed that the exact location of these Fourier coefficients is unknown and only a (possibly) very large search space $\Gamma \subset \mathbb{Z}^d$ containing the corresponding frequencies is given.

Our method, presented in Section 2.2, is based on sampling such a function along the nodes of rank-1 lattices and on applying one-dimensional fast Fourier transforms on the obtained sampling values. Consequently, the performed numerical computations are fast and stable. In contrast to other methods, e.g., see [18, 21], we approximately reconstruct first the (projected) Fourier coefficients and select then the corresponding frequencies which belong to the largest or non-zero Fourier coefficients. In numerical tests in Section 3.1 and 3.2, we successfully applied our method. Additionally, we successfully tested the method on a 10-dimensional function which has infinitely many Fourier coefficients in Section 3.3 and obtained approximately the largest Fourier coefficients and the corresponding frequencies. Furthermore, we successfully reconstructed the frequencies and Fourier coefficients of trigonometric polynomials from sampling values which were perturbed by white Gaussian noise in Section 3.4.

Moreover, we discussed a possibility to reduce the number of samples by applying methods from compressed sensing in Section 2.3 with sub-sampling on rank-1 lattices and generated sets. The application of these sub-sampling methods on trigonometric polynomials in Section 3.1 and 3.2 also succeeded and we compared the numerical results with the ones of the sampling on (full) rank-1 lattices. Additionally, we discussed a variant of Prony's method

in Section 2.4 with sub-sampling on rank-1 lattices and successfully tested this method in Section 3.1.

method	samples	arithmetic complexity
A1-R1L	$\mathcal{O}(d\,s^2 N)$	$\mathcal{O}(d\,s^3 N^2)$
A2-R1L	$\mathcal{O}(d\,s^2 N)$	$\mathcal{O}(d\,s^3 + d\,s^2 N \log(s\,N))$
A2-ℓ_1-sR1L	$\mathcal{O}(d\,s \log^4(s\,N) + d\,N)$	$\mathcal{O}(d\,s^3 + d\,R\,s^2 N \log(sN))$
ℓ_1-GS	$\mathcal{O}(d\,s \log^4(s\,N) + d\,N)$	$\mathcal{O}\left(d\,R\,s(\log^5(s\,N) + N \log(s\,N))\right)$
prony	$\mathcal{O}(d\,s + d\,N)$	$\mathcal{O}(d\,s^3)$

Table 4.1: Sample and arithmetic complexity of the methods presented in this paper for the case $\sqrt{N} \lesssim s \lesssim N^d$ with sparsity s and search space $\Gamma = [-N, N]^d \cap \mathbb{Z}^d$.

method	samples	arithmetic complexity
A1-R1L	$\mathcal{O}(d\,N^2)$	$\mathcal{O}(d\,s\,N^3)$
A2-R1L	$\mathcal{O}(d\,N^2)$	$\mathcal{O}(d\,N^2 \log N)$
A2-ℓ_1-sR1L	$\mathcal{O}(d\,s \log^4(s\,N) + d\,N)$	$\mathcal{O}(d\,R\,N^2 \log N)$
ℓ_1-GS	$\mathcal{O}(d\,s \log^4(s\,N) + d\,N)$	$\mathcal{O}\left(d\,R\,s(\log^5(s\,N) + N \log(s\,N))\right)$
prony	$\mathcal{O}(d\,s + d\,N)$	$\mathcal{O}(d\,s\,N + d\,N \log N)$

Table 4.2: Sample and arithmetic complexity of the methods presented in this paper for the case $s \lesssim \sqrt{N}$ with sparsity s and search space $\Gamma = [-N, N]^d \cap \mathbb{Z}^d$.

Asymptotic upper bounds for the number of samples and arithmetic operations are given in Table 4.1 and 4.2 for the cases where the sparsity s is within the range $\sqrt{N} \lesssim s \lesssim N^d$ and $s \lesssim \sqrt{N}$, respectively. The methods "A1-R1L" and "A2-R1L" are Algorithm 1 and 2 from Section 2.2.1, respectively. "A2-ℓ_1-sR1L" and "ℓ_1-GS" mean ℓ_1 minimization with sub-sampling on rank-1 lattice and sampling on generated sets from Section 2.3.1 and 2.3.2, respectively. "prony" is Prony's method from Section 2.4. We stress on the fact that when comparing different approaches for the sparse reconstruction from a practical point of view, one should also consider the dependence on the dimension d, since algorithms having an exponential or super-exponential dependence on d may not be applicable in practice for higher dimensions d. Moreover, also constants independent of d, which may depend on the specific implementation, can heavily influence the number of arithmetic operations and consequently the computation times. For instance, the observed computation times in Table 3.6 of the implementation of Prony's method from Section 2.4 are distinctly higher compared to the implementation of Algorithm 2 from Section 2.2.1, whereas the arithmetic complexity is not higher for Prony's method.

Acknowledgements

We thank the referees for the valuable suggestions and we thank Lutz Kämmerer for numerous valuable discussions on the presented subject. Moreover, we gratefully acknowledge support by the German Research Foundation (DFG) within the Priority Program 1324, project PO 711/10-2.

References

[1] G. Baszenski and F.-J. Delvos. A discrete Fourier transform scheme for Boolean sums of trigonometric operators. In C. K. Chui, W. Schempp, and K. Zeller, editors, *Multivariate Approximation Theory IV*, ISNM 90, pages 15 – 24. Birkhäuser, Basel, 1989.

[2] E. J. Candès. Compressive sampling. In *International Congress of Mathematicians. Vol. III*, pages 1433 – 1452. Eur. Math. Soc., Zürich, 2006.

[3] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51:4203 – 4215, 2005.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20:33 – 61, 1998.

[5] A. Christlieb, D. Lawlor, and Wang. A multiscale sub-linear time fourier algorithm for noisy data. *Appl. Comput. Harmon. Anal.*, 2015. accepted.

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

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

[8] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289 – 1306, 2006.

[9] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

[10] V. Gradinaru. Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation. *Computing*, 80:1 – 22, 2007.

[11] V. Gradinaru. Strang splitting for the time-dependent Schrödinger equation on sparse grids. *SIAM J. Numer. Anal.*, 46:103 – 123, 2007.

[12] M. Griebel and J. Hamaekers. Sparse grids for the Schrödinger equation. *M2AN Math. Model. Numer. Anal.*, 41:215 – 247, 2007.

[13] M. Griebel and J. Hamaekers. Fast discrete Fourier transform on generalized sparse grids. In J. Garcke and D. Pflüger, editors, *Sparse Grids and Applications - Munich 2012*, volume 97 of *Lect. Notes Comput. Sci. Eng.*, pages 75 – 107. Springer International Publishing, 2014.

[14] M. Griebel and S. Knapek. Optimized general sparse grid approximation spaces for operator equations. *Math. Comp.*, 78:2223 – 2257, 2009.

[15] K. Hallatschek. Fouriertransformation auf dünnen Gittern mit hierarchischen Basen. *Numer. Math.*, 63:83 – 97, 1992.

[16] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pages 563 – 578. ACM, 2012.

[17] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse Fourier transform. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183 – 1194. SIAM, 2012.

[18] P. Indyk and M. Kapralov. Sample-Optimal Fourier Sampling in Any Constant Dimension – Part I. `http://arxiv.org/abs/1403.5804`, 2014.

[19] P. Indyk, M. Kapralov, and E. Price. (Nearly) sample-optimal sparse Fourier transform. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pages 563 – 578. ACM, 2014.

[20] M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Found. Comput. Math.*, 10:303 – 338, 2010.

[21] M. A. Iwen. Improved approximation guarantees for sublinear-time Fourier algorithms. *Appl. Comput. Harmon. Anal.*, 34:57–82, 2013.

[22] L. Kämmerer. Reconstructing hyperbolic cross trigonometric polynomials by sampling along rank-1 lattices. *SIAM J. Numer. Anal.*, 51:2773 – 2796, 2013.

[23] L. Kämmerer. Reconstructing multivariate trigonometric polynomials by sampling along generated sets. In J. Dick, F. Y. Kuo, G. W. Peters, and I. H. Sloan, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 439 – 454. Springer Berlin Heidelberg, 2013.

[24] L. Kämmerer. *High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling*. Dissertation. Universitätsverlag Chemnitz, 2014.

[25] L. Kämmerer. Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices. In G. E. Fasshauer and L. L. Schumaker, editors, *Approximation Theory XIV: San Antonio 2013*, pages 255 – 271. Springer International Publishing, 2014.

[26] L. Kämmerer and S. Kunis. On the stability of the hyperbolic cross discrete Fourier transform. *Numer. Math.*, 117:581 – 600, 2011.

[27] L. Kämmerer, S. Kunis, I. Melzer, D. Potts, and T. Volkmer. Computational Methods for the Fourier Analysis of Sparse High-Dimensional Functions. In S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, and H. Yserentant, editors, *Extraction of Quantifiable Information from Complex Systems*, 2014.

[28] L. Kämmerer, D. Potts, and T. Volkmer. Approximation of multivariate periodic functions by trigonometric polynomials based on sampling along rank-1 lattice with generating vector of Korobov form. *J. Complexity*, 2014.

[29] L. Kämmerer, D. Potts, and T. Volkmer. Approximation of multivariate functions by trigonometric polynomials based on rank-1 lattice sampling. *J. Complexity*, accepted, 2015.

[30] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, 36:Article 19, 1 – 30, 2009.

[31] S. Knapek. Approximation und Kompression mit Tensorprodukt-Multiskalenräumen. Dissertation, Universität Bonn, 2000.

[32] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials II, Orthogonal matching pursuit versus basis pursuit. *Found. Comput. Math.*, 8:737 – 763, 2008.

[33] F. Y. Kuo, I. H. Sloan, and H. Woźniakowski. Lattice rules for multivariate approximation in the worst case setting. In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 289 – 330. Springer Berlin Heidelberg, Berlin, 2006.

[34] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive sub-linear time Fourier algorithms. *Adv. Adapt. Data Anal.*, 5(1):1350003, 25, 2013.

[35] D. Li and F. J. Hickernell. Trigonometric spectral collocation methods on lattices. In S. Y. Cheng, C.-W. Shu, and T. Tang, editors, *Recent Advances in Scientific Computing and Partial Differential Equations*, volume 330 of *Contemp. Math.*, pages 121 – 132. AMS, 2003.

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

[37] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Found. Comput. Math.*, 9:317 – 334, 2009.

[38] D. Potts and M. Tasche. Parameter estimation for multivariate exponential sums. *Electron. Trans. Numer. Anal.*, 40:204 – 224, 2013.

[39] D. Potts and M. Tasche. Parameter estimation for nonincreasing exponential sums by Prony-like methods. *Linear Algebra Appl.*, 439:1024 – 1039, 2013.

[40] H. Rauhut. Random sampling of sparse trigonometric polynomials. *Appl. Comput. Harmon. Anal.*, 22:16 – 42, 2007.

[41] H. Rauhut. On the impossibility of uniform sparse reconstruction using greedy methods. *Sampl. Theory Signal Image Process.*, 7:197 – 215, 2008.

[42] H. Rauhut. Stability results for random sampling of sparse trigonometric polynomials. *IEEE Trans. Inform. Theory*, 54:5661 – 5670, 2008.

[43] J. Shen, T. Tang, and L.-L. Wang. *Spectral Methods*, volume 41 of *Springer Ser. Comput. Math.* Springer-Verlag Berlin Heidelberg, Berlin, 2011.

[44] I. H. Sloan and S. Joe. *Lattice methods for multiple integration.* Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1994.

[45] V. N. Temlyakov. Reconstruction of periodic functions of several variables from the values at the nodes of number-theoretic nets. *Anal. Math.*, 12:287 – 305, 1986. In Russian.

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

[47] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction, 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[48] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31:890 – 912, 2008.