# SPARSITY IN THE SPECTRUM: SPARSE FOURIER TRANSFORMS AND SPECTRAL METHODS FOR FUNCTIONS OF MANY DIMENSIONS

By

Craig Gross

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics—Doctor of Philosophy

2023

## ABSTRACT

The Fourier basis has been a cornerstone of numerical approximations due in part to its amenable algebraic properties resulting in efficient algorithmic approaches. Primary among these is the Fast Fourier Transform (FFT) which transforms a collection samples of a univariate function into that function's Fourier coefficients with computational complexity linear in the number of samples (with an extra logarithmic term). Extensions based on the FFT include algorithms that take advantage of sparsity in a function's Fourier coefficients (sparse Fourier transforms or SFTs) to lower this complexity even further as well as efficient approaches for approximating certain Fourier coefficients of multivariate functions, most often those indexed over computationally friendly hyperbolic cross structures. The ability to quickly compute a function's Fourier coefficients has additionally allowed for a variety of applications including fast algorithms for numerically solving partial differential equations (PDEs) via spectral methods. This dissertation considers improvements on these three applications of the FFT to produce (1) a high-dimensional Fourier transform over arbitrary index sets with reduced sampling complexity from current state of the art methods, (2) an accurate high-dimensional, sparse Fourier transform that can dramatically drive down the sampling and computational complexity so long as a sparsity assumption is satisfied, and (3) a high-dimensional, sparse spectral method which makes use of our sparse Fourier transform to solve PDEs with multiscale structure in extremely high dimensions.

All three of these applications rely on the method of rank-1 lattices for their flexibility. By using this quasi-Monte Carlo approach for sampling in high-dimensions, high-dimensional functions are converted into one-dimensional ones on which well-studied techniques can be used. We extend these approaches by first developing a fully deterministic construction of multiple, smaller, rank-1 lattices to sample over simultaneously which drive down the sampling complexity from traditional rank-1 lattice methods. Our improved technique depends only linearly on the size of the underlying set of frequencies that Fourier coefficients are computed over rather than the previously standard quadratic dependence (with additional logarithmic terms).

We can push further beyond this linear dependence on the frequency set of interest by making

use of univariate SFTs after the high-dimensional to one-dimensional conversion. However, to effectively integrate univariate SFT algorithms into the rank-1 lattice approach without ruining the derived computational speedups, we provide an alternative approach. Rather than employing multiple rank-1 lattice sampling sets, we need to employ multiple rank-1 lattice SFTs. The slightly inflated sampling cost allows for significant gains in coefficient reconstruction: we produce two methods whose dependence on the frequency set of interest is cast entirely into logarithmic terms. The complexity is then quadratically or linearly (depending on the chosen variation) dependent on an imposed sparsity parameter and linear in the dimension of the underlying function domain. The dependence on this sparsity is then fully characterized in near-optimal approximation guarantees for the function of interest.

And just as the FFT provided the foundation for fast spectral methods for numerically approximating solutions to PDE, so too does our high-dimensional, sparse Fourier transform provide the foundation for a high-dimensional, sparse spectral method. However, to be most effective, the underlying frequency set of interest should be primarily driven by the PDE itself rather than the user. As such, we provide a technique for efficiently converting sparse Fourier approximations of the PDE data into a Fourier basis in which the solution to the PDE will be guaranteed to have a good approximation. These ingredients combined with the rich literature on spectral methods allow for us to provide error estimates in the Sobolev norm for the solution which are fully characterized by properties of the PDE, namely the Fourier sparsity of its data and conditions related to its well-posedness.

Throughout the text, these proposed algorithms are accompanied with practical considerations and implementations. These implementations are then judged against a variety of numerical tests which demonstrate performance on par with the theoretical guarantees provided.

To Alan, my fellow scientist and my brother. I love you.

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Mark Iwen, for your incredible support throughout my time at Michigan State University. Your generosity in time, advice, ideas, and more is the reason that this work exists and would not have been possible without your guidance. I also owe to you the space that I had throughout my studies to fully explore my interests both mathematically and professionally, and find a path that was fulfilling. And I also have you to thank for allowing me to grow beyond the boundaries I came to MSU with, whether they be boundaries of perspective, opportunity, or geography.

In that vein, I would also like thank my collaborators at Technische Universität Chemnitz, Lutz Kämmerer and Toni Volkmer, who introduced me to the wonderful world of rank-1 lattices and whom I wrote Chapters 2 and 3 alongside. You helped me develop as a researcher and an applied mathematician through your invaluable mentorship, contributions, and conversations. And I have you and the rest of Daniel Potts' group to thank for your incredible hospitality in my unforgettable visit to Chemnitz.

And to one of my first mathematical mentors, Andrew Gillette, I thank you for showing me what it means to be a mathematician. From my first day of freshman year in the Cesar E. Chavez Building at the University of Arizona to our continued conversations at Lawrence Livermore National Laboratory, you have been there to foster my mathematical journey and afford me the opportunities to make it to this point.

I would also like to thank my fellow mathematicians who I had the pleasure of sharing thoughts with throughout my studies. In particular, I have Ben Adcock and Simone Brugiapaglia to thank for the inspiration and motivation resulting in the sparse spectral method presented in Chapter 4. I also thank the members of my committee, Yingda Cheng, Jun Kitagawa, and Rongrong Wang, for your instruction and guidance throughout my time at MSU.

To my friends in my cohort, thank you for the long nights of analysis homework, the HopCat happy hours, and the consistent cycle of commiseration and inspiration. And to those friends who came to MSU before or after me, thank you for making and keeping the math department bright,

welcoming, and growing.

But most of all, I owe my successes, my opportunities, and everything else to my family. My heroes, my mother and father, have provided the encouragement and continual support to reach where I am today. Your perpetual care, humor, creativity, and joy form the foundation for me every day, and it is only on that foundation that I am able to grow and push myself into places, ideas, and worlds previously unknown. And to my siblings, Katie, for your empathy, drive, and spirit that keeps me moving forward; Essa, for your conversations that bring me the perspective I need; and Alan, for the everlasting knowledge that I have your love and support behind me: I can't thank each of you enough.

And finally, to Sarina, I could write another 124 pages about how this, and every day, is due to you. But I'll keep it brief. Simply put, this, and so many more of the achievements in my life, wouldn't have been able to happen without you. You've kept me together in the bad and have been the celebration of the good. You've been by my side every day, my outlet, my reflection for thoughts, joys, and all the rest. You bring me everything. Thank you.

## TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

This dissertation is concerned with the efficient approximation of periodic functions of many variables by Fourier series and associated applications in solving partial differential equations. For a periodic function $g : \mathbb{T}^d \to \mathbb{C}$, where $\mathbb{T}$ is taken to be $\mathbb{R}/\mathbb{Z}$, we wish to compute its Fourier series, or at least an approximation, as quickly as possible. That is, we want to find the coefficients $\hat{g}$, a complex sequence indexed by multivariate frequencies $\mathbf{k} \in \mathbb{Z}^d$, of the Fourier series

$$g = \sum_{\mathbf{k} \in \mathbb{Z}^d} \hat{g}_{\mathbf{k}} \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ}.$$

Since the collection of multivariate trigonometric monomials $\{\mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ}\}_{\mathbf{k} \in \mathbb{Z}^d}$ forms an orthonormal basis for $L^2(\mathbb{T}^d)$ (cf. Theorem 1.1), the Fourier coefficients of $g$ can be computed by

$$\hat{g}_{\mathbf{k}} = \int_{\mathbb{T}^d} g(\mathbf{x}) \mathrm{e}^{-2\pi \mathrm{i} \mathbf{k} \cdot \mathbf{x}} \, d\mathbf{x}.$$

Of course, using this formulation would require full knowledge of $g$ to begin with, or at least enough information to approximate this integral. However, this is the problem we are attempting to solve in the first place.

The univariate formulation of this problem has been classically solved using the fast Fourier transform (FFT). Given a parameter $K \in \mathbb{N}$, the FFT computes approximate Fourier coefficients of a function $g^{1\mathrm{d}} : \mathbb{T} \to \mathbb{R}$ via a simple left Riemann sum over $K$ points:

$$\hat{g}^{1\mathrm{d}}_{\omega} \approx \frac{1}{K} \sum_{j=0}^{K} g\left(\frac{j}{K}\right) \mathrm{e}^{-2\pi \mathrm{i} \omega j / K}.$$

Computing all approximate Fourier coefficients with frequencies in $[K] := \{0, \ldots K-1\}$ at once can be performed by the matrix multiplication $\mathbf{F}_K \mathbf{g}^{1\mathrm{d}}$, where

$$\mathbf{F}_K := \left(\frac{1}{K} \mathrm{e}^{-2\pi \mathrm{i} \omega j / K}\right)_{\omega \in [K], j \in [K]} \quad \text{and} \quad \mathbf{g}^{1\mathrm{d}} := \left(g^{1\mathrm{d}}\left(\frac{j}{K}\right)\right)_{j \in [K]}.$$

Taking advantage of algebraic properties of the Fourier basis, the FFT algorithm performs this matrix multiplication in $O(K \log K)$ time and space, instead of the standard $O(K^2)$ computational complexity (see, e.g., [56] for a good survey of these techniques).

Returning to the multivariate setting, instead of using an equispaced sampling of the target function over an interval, we can take an equispaced sampling over the $d$-dimensional grid, denoted $(g(\mathbf{j}/K))_{\mathbf{j}\in[K]^d}$, and effectively apply $d$ FFTs along the sides of this now $d$-dimensional tensor. Thus, this multivariate FFT has a time/space-complexity of $O(K^d \log^d K)$. This exponential growth in $d$ characterizes the well-known *curse of dimensionality* and therefore, this multivariate FFT is only suitable for low dimensions.

Approaches to avoid this curse of dimensionality for Fourier approximation form a vast body of literature. The state of the art in the contexts we are interested in is discussed in the literature reviews of the subsequent chapters. However, we summarize a simple and effective approach upon which the remainder of this dissertation is based: using *rank-1 lattices*.

**Definition 1.1.** Given a natural number $M \in \mathbb{N}$ and a generating vector $\mathbf{z} \in \{1, \dots, M-1\}^d$, the *rank-1 lattice* $\Lambda(\mathbf{z}, M) \subset \mathbb{T}^d$ is defined as

$$\Lambda(\mathbf{z}, M) := \left\{ \frac{j}{M}\mathbf{z} \bmod \mathbf{1} \mid j \in [M] \right\}.$$

Intuitively, a rank-1 lattice gives a direction vector $\mathbf{z}$ to restrict the multivariate function $g$ into a univariate one $g^{1\mathrm{d}}$ defined by $t \mapsto g(t\mathbf{z})$. The $M$ sampling points in the rank-1 lattice are in fact an equispaced sampling over $\mathbb{T}$ of $g^{1\mathrm{d}}$. An FFT of these equispaced samples of $g^{1\mathrm{d}}$ is then able to give us information about the Fourier coefficients of the original, high-dimensional function $g$.

To see how the FFT relates to the Fourier coefficients of $g$, we consider the Fourier series of $g^{1\mathrm{d}}$ by way of the Fourier series of $g$,

$$g^{1\mathrm{d}}(t) = g(t\mathbf{z}) = \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{g}_{\mathbf{k}} \mathrm{e}^{2\pi \mathrm{i} \mathbf{k}\cdot\mathbf{z}t} = \sum_{\omega\in\mathbb{Z}} \left( \sum_{\substack{\mathbf{k}\in\mathbb{Z}^d \\ \mathbf{k}\cdot\mathbf{z}=\omega}} \hat{g}_{\mathbf{k}} \right) \mathrm{e}^{2\pi\mathrm{i}\omega t}.$$

Thus

$$\hat{g}^{1\mathrm{d}}_{\omega} = \sum_{\substack{\mathbf{k}\in\mathbb{Z}^d \\ \mathbf{k}\cdot\mathbf{z}=\omega}} \hat{g}_{\mathbf{k}}.$$

In light of the fact that we will be using an FFT approximation of $\hat{g}^{1\mathrm{d}}$, let us also note well-known

aliasing effect of the FFT. For all $\omega \in [M]$,

$$\left(\mathbf{F}_M \mathbf{g}^{\mathrm{1d}}\right)_\omega = \sum_{\substack{\omega' \in \mathbb{Z} \\ \omega' \equiv \omega \bmod M}} \hat{g}_{\omega'} \tag{1.1}$$

(see Lemma 1.3 for the proof and further explanation). We can then assert that

$$\left(\mathbf{F}_M \mathbf{g}^{\mathrm{1d}}\right)_\omega = \sum_{\substack{\mathbf{k} \in \mathbb{Z}^d \\ \mathbf{k}\cdot\mathbf{z} \equiv \omega \bmod M}} \hat{g}_{\mathbf{k}}.$$

To make the most effective use of the length-$M$ FFT, a rank-1 lattice should be chosen so that this sum contains at most one Fourier coefficient of the original function in $\hat{g}$. In order to accomplish this, we will restrict the scope of our Fourier coefficient approximation to some chosen frequency set of interest $\mathcal{I} \subset \mathbb{Z}^d$ and introduce the idea of the *modulus mapping* and a *reconstructing rank-1 lattice*.

**Definition 1.2.** Choose some $\mathcal{I} \subset \mathbb{Z}^d$. The *modulus mapping* $m_{\mathbf{z},M} : \mathcal{I} \to [M]$ is defined by $\mathbf{k} \mapsto \mathbf{k} \cdot \mathbf{z} \bmod M$. A rank-1 lattice $\Lambda(\mathbf{z}, M)$ is said to be *reconstructing for* $\mathcal{I}$ if the modulus mapping is injective. An equivalent condition is that

$$\mathbf{k} \cdot \mathbf{z} \not\equiv \mathbf{h} \cdot \mathbf{z} \bmod M \text{ for all } \mathbf{k} \neq \mathbf{h} \in \mathcal{I}.$$

We find then that for any trigonometric polynomial $g \in \Pi_{\mathcal{I}} := \operatorname{span}\{\mathrm{e}^{2\pi \mathrm{i} \mathbf{k}\cdot\circ} \mid \mathbf{k} \in \mathcal{I}\}$, (1.1) reduces to

$$\hat{g}_{\mathbf{k}} = \left(\mathbf{F}_M \mathbf{g}^{\mathrm{1d}}\right)_{m_{\mathbf{z},M}(\mathbf{k})} \text{ for all } \mathbf{k} \in \mathcal{I}$$

and for any $g$ with Fourier coefficients not necessarily supported on $\mathcal{I}$,

$$\hat{g}_{\mathbf{k}} = \left(\mathbf{F}_M \mathbf{g}^{\mathrm{1d}}\right)_{m_{\mathbf{z},M}(\mathbf{k})} + \sum_{\substack{\mathbf{h} \notin \mathcal{I} \\ \mathbf{h}\cdot\mathbf{z} \equiv \mathbf{k}\cdot\mathbf{z} \bmod M}} \hat{g}_{\mathbf{h}} \text{ for all } \mathbf{k} \in \mathcal{I}. \tag{1.2}$$

The upshot is that we are able to compute all Fourier coefficients in $\mathcal{I}$ of a periodic function $g$ up to errors related to restricting our attention to $\mathcal{I}$. The full rank-1 lattice FFT approach is summarized in Algorithm 1.1.

**Algorithm 1.1** Rank-1 lattice FFT

**Input:** A function $g : \mathbb{T}^d \to \mathbb{C}$, a frequency set of interest $\mathcal{I} \subset \mathbb{Z}^d$, and a reconstructing rank-1 lattice for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$

**Output:** Approximate Fourier coefficients $\hat{\mathbf{g}}^{\Lambda} \in \mathbb{C}^{\mathcal{I}}$

1: $\mathbf{g}^{1\mathrm{d}} \leftarrow (g(j\mathbf{z}/M))_{j\in[M]}$
2: Compute $\mathbf{F}_M \mathbf{g}^{1\mathrm{d}}$
3: **for** $\mathbf{k} \in \mathcal{I}$ **do**
4: $\quad \hat{g}^{\Lambda}_{\mathbf{k}} \leftarrow \left(\mathbf{F}_M \mathbf{g}^{1\mathrm{d}}\right)_{m_{\mathbf{z},M}(\mathbf{k})}$
5: **end for**

With the basic ideas behind the rank-1 lattice FFT in hand, we can motivate the remaining chapters of this dissertation.

### 1.1.1 Multiple rank-1 lattices and their construction

The most important ingredient for Algorithm 1.1 is a reconstructing rank-1 lattice for a chosen frequency set $\mathcal{I}$. The size of this rank-1 lattice also has major impacts on the computational complexity Algorithm 1.1, namely, the sampling step and the FFT. Thus, the goal should be to find as small a reconstructing rank-1 lattice as possible.

The most popular reconstructing rank-1 lattice construction is the component-by-component (CBC) approach [39, 56, 46, 48]. The idea is to start with the set of frequency differences $\mathcal{D}(\mathcal{I}) := \{\mathbf{k} - \mathbf{h} \mid \mathbf{k}, \mathbf{h} \in \mathcal{I}\}$ and consider these differences one component at a time. Each component of the generating vector is chosen by a brute-force scan through these differences to ensure that there are no collisions modulo $M$, where it suffices for $M$ to be a prime number between $|\mathcal{D}(\mathcal{I})|/2$ and $|\mathcal{D}(\mathcal{I})|$. But note also that $|\mathcal{I}| \lesssim |\mathcal{D}(\mathcal{I})| \lesssim |\mathcal{I}|^2$, and in fact, there exist specific frequency sets which require any associated reconstructing rank-1 lattice to have size $M = \Omega(|\mathcal{I}|^2)$ [12, Section 3]. See also [39, Section 5] for more information about rank-1 lattices and their often (seemingly unnecessarily) large sizes. The goal of Chapter 2 is to reduce this quadratic dependence on $|\mathcal{I}|$ (and therefore on the sampling and computational complexity of Algorithm 1.1) using a slight generalization of rank-1 lattices. Rather than restricting the high-dimensional function to just one lattice, we use *multiple rank-1 lattices* [40, 41], which can be smaller than a single reconstructing rank-1 lattice is required to be, to drive down the overall complexity.

In particular, Chapter 2 presents the first known deterministic algorithm for constructing a series of multiple rank-1 lattices for an arbitrary frequency set. As input, it takes a reconstructing single reconstructing rank-1 lattice and returns $O(\log|\mathcal{I}|)$ lattices each of size at $O\left(|I|\log^2(K_{\mathcal{I}}|\mathcal{I}|)\right)$ where $K_{\mathcal{I}}$ is the sidelength of the smallest hypercube containing $\mathcal{I}$. Each lattice handles a portion of the frequencies in $\mathcal{I}$ so that performing FFTs over all smaller lattices will exactly recover the Fourier coefficients of trigonometric polynomials in $\Pi_{\mathcal{I}}$ Approximation guarantees are also provided for general periodic functions similar to (1.2). Due to the size of the full multiple rank-1 lattice returned, any quadratic dependence on a single rank-1 lattice in $|\mathcal{I}|$ can therefore be reduced to a linear (with polylogarithmic terms) dependence without incurring significant additional errors.

### 1.1.2 Sparse Fourier transforms and rank-1 lattices

Though the efforts of Chapter 2 are able to reduce the amount of work necessary in a rank-1 lattice FFT approach, a linear dependence on $|\mathcal{I}|$ in the complexity may still be intolerable. For large search spaces of multivariate frequencies $\mathcal{I}$ such as the full hypercube of sidelength $K$, $\mathcal{I} = \left((-\lceil\frac{K}{2}\rceil, \lfloor\frac{K}{2}\rfloor] \cap \mathbb{Z}\right)^d$, these methods still suffer from the curse of dimensionality.

Rather than a more general multiple rank-1 lattice approach, Chapter 3 considers the case of functions whose Fourier series are sparse or compressible. Since the rank-1 lattice procedure reduces high-dimensional functions into one-dimensional ones, one-dimensional sparse Fourier transform (SFT) techniques [25, 27, 36, 35, 51, 62, 37, 26, 18, 45, 57, 58, 53, 3, 2, 1] become particularly appealing. SFTs are compressive sensing algorithms which are highly specialized to take advantage of the number theoretic and algebraic structure of the Fourier basis as much as possible. As a result, SFTs rarely have to consider Fourier basis functions individually during the reconstruction process, and so can simultaneously reduce both their measurement needs and computational complexity to effectively depend only on the number of important Fourier series coefficients in the function one aims to approximate. Thus, SFTs can sidestep runtimes which are polynomially dependent on the bandwidth (in the case of a rank-1 lattice FFT, $M$), and instead run sublinearly in the magnitude of the underlying frequency space under consideration. If one desires to capture only the largest $s$ Fourier coefficients of a function, the SFT discussed in Theorem 3.1, for example, runs

in $O(s^2 \log^4 M)$-time/space (with a randomized version cutting the quadratic factor of $s$ down to linear). Additionally, these techniques often furnish recovery guarantees for Fourier compressible functions in terms of best $s$-term approximations in the same vein as compressed sensing results [19, 24].

However, simply replacing the FFT $\mathbf{F}_M \mathbf{g}^{1\mathrm{d}}$ in Line 2 of Algorithm 1.1 with a suitable SFT $\mathcal{A}_{s,M} g^{1\mathrm{d}}$ is not enough to relieve linear dependence on $|\mathcal{I}|$. The for loop from Line 3 to Line 5 which matches $d$-dimensional and one-dimensional frequencies requires a linear scan through $\mathcal{I}$. A simple optimization is to swap the order this process, and match the $s$-many entries of $\mathcal{A}_{s,M} g^{1\mathrm{d}}$ to the corresponding Fourier coefficients indexed over $\mathcal{I}$. But even this is not enough, as it requires complete knowledge of the inverse modulus mapping $m_{\mathbf{z},M}^{-1}$ which is either built up through the rank-1 lattice construction and stored or computed through a $O(d|\mathcal{I}|)$-computation. All benefit in swapping the FFT along the lattice with an SFT is then lost.

The methods given in Chapter 3 instead use samples along possibly larger lattices to produce a sparse approximation of the Fourier transform of $g$ without directly inverting $m_{\mathbf{z},M}$. Two algorithms are presented which operate on SFTs of manipulations of $g^{1\mathrm{d}}$ in order to relate the univariate coefficients to their multivariate counterparts in $o(|\mathcal{I}|)$-time. This allows the methods to run faster and with less memory than it takes to simply enumerate the frequency set $\mathcal{I}$ and/or store $m_{\mathbf{z},M}(\mathcal{I})$ whenever $g$ has a sufficiently accurate sparse approximation.

The result is a series of curse-of-dimensionality breaking, high-dimensional SFTs with proven compressive-sensing type guarantees for arbitrary periodic functions. The approaches are linear in $d$ in the sampling and run-time complexities which succeed deterministically in quadratic in $s$ time. As with the univariate SFT discussed above, this can be reduced to linear in $s$ time via randomization. We defer to Section 3.1 for a fuller discussion in the context of the provided literature review.

Finally, though these results are able to sidestep the necessity of the inverse of the modulus mapping, $m_{\mathbf{z},M}(\mathcal{I})$, an existing reconstructing rank-1 lattice for $\mathcal{I}$ is still required. As discussed above, CBC constructions, though only necessary to perform once, are still relatively expensive

in the context of SFT complexities. This requirement is dropped via a randomized approach to constructing rank-1 lattices in Section 4.5, resulting in an algorithm with complexity fully sublinear in $|\mathcal{I}|$.

### 1.1.3 Applications to PDE

The fast, high-dimensional SFT techniques of Chapter 3 are applied in Chapter 4 to construct an efficient, numerical PDE solver. For this exposition, we consider as a model problem an elliptic PDE with periodic boundary conditions

$$-\nabla \cdot (a \nabla u) = f \tag{1.3}$$

where $a, f : \mathbb{T}^d \to \mathbb{R}$ are the PDE data, and $u : \mathbb{T}^d \to \mathbb{R}$ is the solution. Solving (1.3) using a traditional Fourier spectral method amounts to replacing the data and the solution with their Fourier series, simplifying the left-hand side into a single Fourier series, matching the Fourier coefficients of both sides, and solving the resulting system of equations for the Fourier coefficients of $u$.

Two main sources of approximation error arise when implementing this technique computationally. The first is due to truncating the Fourier series involved to a finite number of terms. The second is due to numerically approximating the Fourier coefficients of the PDE data. Due to the rich theory of traditional spectral methods, these two sources of error can directly quantify the error of the resulting approximation of $u$.

**Lemma 1.1** (Strang's lemma, [13])**.** *Let $u^{\text{truncation}}$ be the function which has the same Fourier series as $u$ but truncated in some manner, and $a^{\text{approximate}}$ and $f^{\text{approximate}}$ be computed using approximations of the Fourier series of $a$ and $f$ truncated in the same way as $u^{\text{truncation}}$. Then the procedure outlined above produces a solution $u^{\text{spectral}}$ which satisfies*

$$\left\|u - u^{\text{spectral}}\right\|_{H^1} \lesssim_{a,f} \left\|u - u^{\text{truncation}}\right\|_{H^1} + \left\|a - a^{\text{approximate}}\right\|_{L^\infty} + \left\|f - f^{\text{approximate}}\right\|_{L^2}$$

*where $\lesssim_{a,f}$ denotes an upper bound with constants that depend on the PDE data.*

This is a rough simplification of *Strang's lemma* [13], which is itself a generalization of the well-known *Céa's lemma* (the specific version of the lemma that we use is presented and proven in Lemma 4.6 below). Effectively, it states that the spectral method solution is optimal up to its Fourier

series truncation and the approximation of the PDE data $a$ and $f$. Thus, analyzing convergence reduces to estimating these two errors.

Using $d$-dimensional FFTs to compute $a^{\text{approximate}}$ and $f^{\text{approximate}}$ in the procedure suggested in Lemma 1.1 naturally enforces a Fourier series truncation. A $d$-dimensional FFT using a tensorized grid of $K$ uniformly spaced points in each dimension will produce approximate Fourier coefficients indexed by frequencies in the $d$-dimensional hypercube on the integer lattice $\mathbb{Z}^d$ of sidelength $K$ (note that when we refer to "bandwidth" in a multidimensional sense, we are still referring to the sidelength $K$ of the hypercube containing these integer frequencies). As discussed above, the cost of each $d$-dimensional FFT in general requires more than $K^d$ operations, as does the linear-system solve (in the absence of any sparsity or other tricks). Thus, not only do traditional Fourier spectral methods suffer from the curse of dimensionality, but even in moderate dimensions, multiscale problems (i.e., PDE data which require very high bandwidth to be fully resolved) can result in intractable computations.

This is a prime opportunity to take advantage of our high-dimensional SFT algorithms to compute $a^{\text{approximate}}$ and $f^{\text{approximate}}$. This allows for the data terms in Strang's Lemma above to converge near-optimally in terms of their compressibility in the Fourier basis. However, these SFTs only provide us with truncation information useful for $a$ and $f$, not necessarily $u$. One of the more significant contributions of Chapter 4 is resolving this truncation gap. By analyzing in detail the effect of the differential operator discretized using SFT approximations in frequency, we provide a technique for computing the most important Fourier coefficients of $u$ using knowledge of the most important Fourier coefficients of $a$ and $f$. We can then prove truncation estimates which allow for a sparse spectral method with $H^1$ error guarantees fully characterized by the Fourier compressibility of the data and terms relating to the ellipticity properties of the original PDE. Note that though we only consider a diffusion term in (1.3) for the simplicity of this overview, the analysis in Chapter 4 is actually that of a full multiscale, high-dimensional advection-diffusion-reaction equation, similar to, e.g., the governing equations for flow dynamics in a porous medium used in hydrological modeling [61].

### 1.1.4 A note on previous publication of this work

The three chapters following this introduction are each comprised of the results presented in three previously available manuscripts. With some exceptions, Chapter 2 is published as [34], Chapter 3 is published as [33], and (at the time of submission of this dissertation) Chapter 4 is publicly available at [32] and has been submitted for publication. Thus, the contents of Chapters 2 and 3 were developed collectively with Lutz Kämmerer and Toni Volkmer and Chapters 2 to 4 with Mark Iwen. Additionally, portions of this introduction were adapted from the introductions of the original three manuscripts.

That being said, there are changes in the results given in this dissertation from their original presentations. In Chapter 2, the main modification is clarifying the Fourier recovery mechanism and the error guarantees for approximation in Corollary 2.1. Chapter 3 includes the $L^\infty$ error guarantees for the phase-encoding SFT originally provided in [32] and extends these guarantees to all algorithms analyzed. Finally, Chapter 4 provides a complete analysis of advection-diffusion-reaction equations rather than solely the diffusion equations of the original text.

### 1.1.5 Organization

The remainder of this chapter is comprised of a section setting the notation and a section collecting some useful Fourier series related lemmas that are used throughout the text. The three following chapters respectively present the three main results summarized above. Each chapter gives a short overview, followed by the theory, and finally a numerics section with implementation details and tests demonstrating that theory in practice.

## 1.2 Notation

We let $d$ be the ambient dimension of function domains under consideration. The torus $\mathbb{T}$ is defined as $\mathbb{R}/\mathbb{Z}$, i.e., $[0, 1]$ with the endpoints identified. Given a natural number $M \in \mathbb{N}$, we let $[M] := \{0, \dots, M-1\}$.

Finite length vectors are defined using boldface. For example, we often use $\mathbf{x} \in \mathbb{T}^d$ as a point in the spatial domain of a function and $\mathbf{k} \in \mathbb{Z}^d$ as a $d$-dimensional frequency to index Fourier coefficients. This also extends to multiindexed finite vectors. For example, if $\mathcal{I} \subset \mathbb{Z}^d$ with $|\mathcal{I}| < \infty$, then

we would refer to a vector indexed over $\mathcal{I}$ as, e.g., $\hat{\mathbf{g}} = (\hat{g}_{\mathbf{k}})_{\mathbf{k}\in\mathcal{I}}$. Infinite length sequences remain in standard roman font, e.g., $\hat{g} = (\hat{g}_{\mathbf{k}})_{\mathbf{k}\in\mathbb{Z}^d}$. All finite length vectors will be implicitly extended to larger index sets by taking on the value zero wherever they are not originally defined. Additionally, the set of all complex-valued, finite length vectors or infinite length sequences supported on an index set $\mathcal{D}$ is denoted as $\mathbb{C}^{\mathcal{D}}$. Our convention is to use zero-based indexing, i.e., $\mathbb{C}^M = \mathbb{C}^{[M]}$.

In general, a multivariate function to be recovered is $g : \mathbb{T}^d \to \mathbb{C}$. Specific functions of used in the context of elliptic PDE are

- $a : \mathbb{T}^d \to \mathbb{R}$, the diffusion coefficient;
- $\mathbf{b} : \mathbb{T}^d \to \mathbb{R}^d$, the advection field;
- $c : \mathbb{T}^d \to \mathbb{R}$, the reaction coefficient;
- $f : \mathbb{T}^d \to \mathbb{R}$, the forcing function; and
- $u : \mathbb{T}^d \to \mathbb{R}$, the solution to the PDE.

Unless otherwise stated, we assume all functions are complex-valued and defined on the torus $\mathbb{T}^d$. For example, we take the inner product for $u, v \in L^2 := L^2(\mathbb{T}^d; \mathbb{C})$ to be

$$\langle u, v\rangle_{L^2} := \int_{\mathbb{T}^d} u(\mathbf{x})\overline{v}(\mathbf{x})\, d\mathbf{x}$$

where $\overline{v}$ is taken to be the complex conjugate of $v$. Additionally, we assume all vectors and sequences are complex-valued and defined on $\mathbb{Z}^d$ unless otherwise stated. For example, we take the inner product for $\hat{u}, \hat{v} \in \ell^2 := \ell^2(\mathbb{Z}^d; \mathbb{C})$ to be

$$\langle \hat{u}, \hat{v}\rangle_{\ell^2} := \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{u}_{\mathbf{k}}\overline{\hat{v}}_{\mathbf{k}}.$$

The domains and ranges for the function spaces $L^1$, $L^\infty$, $C$ (the space of continuous functions), and $C^\infty$ (the space of infinitely differentiable functions) are inferred similarly, as is the index set of the spaces of sequences $\ell^1$ and $\ell^\infty$.

We now define our specific notion of periodic Sobolev spaces (see also [8, Section 2.1] and [47, Appendix A.2.2]).

**Definition 1.3.** For $u \in L^2$ and $\alpha \in \mathbb{N}_0^d$ a multiindex, if there exists a $v \in L^2$ such that

$$\langle v, \phi\rangle_{L^2} = (-1)^{|\alpha|}\langle u, \partial^\alpha \phi\rangle_{L^2} \qquad \text{for all } \phi \in C^\infty \subset L^2,$$

we call $v$ the *weak $\alpha$ derivative of $u$*, and write $\partial^\alpha u := v$. We define the inner product

$$\langle u, v \rangle_{H^1} := \sum_{\alpha \in \{0,1\}^d, \|\alpha\|_1 \leq 1} \int_{\mathbb{T}^d} \partial^\alpha u(\mathbf{x}) \overline{\partial^\alpha v}(\mathbf{x}) \, d\mathbf{x},$$

(where all derivatives are considered in the weak sense) and have the associated norm $\|u\|_{H^1} := \sqrt{\langle u, u \rangle_{H^1}}$. The *periodic Sobolev space* $H^1$ is defined as $H^1 := \{u \in L^2 \mid \|u\|_{H^1} < \infty\}$.

For any $g \in L^1$, and any $\mathbf{k} \in \mathbb{Z}^d$, we define the $\mathbf{k}$th Fourier coefficient

$$\hat{g}_\mathbf{k} := \left\langle g, \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ} \right\rangle_{L^2} = \int_{\mathbb{T}^d} g(\mathbf{x}) \mathrm{e}^{-2\pi \mathrm{i} \mathbf{k} \cdot \mathbf{x}} \, d\mathbf{x}.$$

The Wiener algebra $W := W(\mathbb{T}^d; \mathbb{C})$ is defined as the set of all functions with absolutely summable Fourier coefficients, $W := \left\{ g \in L^1 \mid \hat{g} \in \ell^1 \right\}$. For any function $g \in W$, its Fourier series is written as

$$g = \sum_{\mathbf{k} \in \mathbb{Z}^d} \hat{g}_\mathbf{k} \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ}$$

(see also Theorem 1.1 below). Given a hatted sequence $\hat{g} \in \mathbb{C}^{\mathbb{Z}^d}$ without having previously defined $g$, the function $g$ is then implicitly defined as the Fourier series with Fourier coefficients $\hat{g}$. In examples where sequences of Fourier coefficients are known to be finite length, e.g., the output of sparse Fourier transform algorithms, these coefficients are written in boldface, e.g., $\hat{\mathbf{g}}^s$. Note also that for notational aesthetic, Fourier coefficients for functions with super or subscripts will not include the super or subscript under the hat, e.g., the Fourier coefficients of $G_3^d$ are $\hat{G}_3^d$. There are some occasions where super or subscripts will refer to modifications of Fourier coefficients rather than referring to the Fourier coefficients of a super or subscripted function (e.g., $\hat{g}_s^{\mathrm{opt}}$ is the best $s$-term approximation of $\hat{g}$, not the Fourier coefficients of a function $g_s^{\mathrm{opt}}$), but these will be made clear from context. For univariate functions $g^{1\mathrm{d}} : \mathbb{T} \to \mathbb{C}$, we usually use $\omega$ to index the Fourier coefficients, e.g., $\hat{g}^{1\mathrm{d}} = \left(\hat{g}_\omega^{1\mathrm{d}}\right)_{\omega \in \mathbb{Z}}$.

A $d$-dimensional frequency set of interest is usually taken to be $\mathcal{I} \subset \mathbb{Z}^d$. In general, most $d$-dimensional frequency sets are labeled using calligraphic font. For example, Chapter 4 introduces a particularly important class of frequency sets, the stamping sets denoted $\mathcal{S}^N \subset \mathbb{Z}^d$ for $N \in \mathbb{N}_0$ which are implicitly parameterized by the set of all active frequencies in a PDE's data $\mathcal{A}$. The space

of all trigonometric polynomials with frequencies in $\mathcal{I}$ is denoted by $\Pi_{\mathcal{I}} := \operatorname{span}\{e^{2\pi i \mathbf{k}\cdot\circ} \mid \mathbf{k} \in \mathcal{I}\}$. The expansion $K_{\mathcal{I}}$ of a frequency set $\mathcal{I} \subset \mathbb{Z}^d$ is defined as

$$K_{\mathcal{I}} := \max_{j\in[d]} \left( \max_{\mathbf{k}\in\mathcal{I}} k_j - \min_{\mathbf{l}\in\mathcal{I}} l_j \right) + 1.$$

Note that this can be interpreted as the sidelength of the smallest hypercube containing $\mathcal{I}$.

For a sequence $\hat{g} \in \mathbb{C}^{\mathbb{Z}^d}$, its restriction to an index set $\mathcal{I}$ is denoted by $\hat{g}|_{\mathcal{I}}$. The same is true for vectors. This can be interpreted as either a vector in $\mathbb{C}^{\mathcal{I}}$ or a sequence in $\mathbb{C}^{\mathbb{Z}^d}$ which is set to zero outside of $\mathcal{I}$. When $\hat{g}$ refers to the Fourier coefficients of the function $g$, restrictions of $g$ to index sets refer to the Fourier series with Fourier coefficients restricted in the same way, i.e.,

$$g|_{\mathcal{I}} := \sum_{\mathbf{k}\in\mathbb{Z}^d} (\hat{g}|_{\mathcal{I}})_{\mathbf{k}}\, e^{2\pi i \mathbf{k}\cdot\circ} = \sum_{\mathbf{k}\in\mathcal{I}} \hat{g}_{\mathbf{k}} e^{2\pi i \mathbf{k}\cdot\circ}.$$

We will also often consider restricting a multiindexed sequence to the hypercube with a fixed sidelength $K$. We will denote this set by $\mathcal{B}_K^d$, where the one-dimensional frequency band of length $K$, $\mathcal{B}_K$, is defined by $\mathcal{B}_K := \left(-\left\lceil \frac{K}{2} \right\rceil, \left\lfloor \frac{K}{2} \right\rfloor\right] \cap \mathbb{Z}$. Rather than subscript with this set, we use the shorthand $\hat{g}|_K := \hat{g}|_{\mathcal{B}_K^d}$. The best $s$-term approximation of a sequence $\hat{g}$ is defined as the restriction $\hat{g}$ to its $s$-largest magnitude entries, denoted by $\hat{g}_s^{\text{opt}}$. The same applies to vectors.

Given a univariate function $g^{1\text{d}} : \mathbb{T} \to \mathbb{C}$, we define the vector $\mathbf{g}^{1\text{d}} \in \mathbb{C}^M$ as the vector of $M$ equispaced samples of $g^{1\text{d}}$ on $\mathbb{T}$, that is,

$$\mathbf{g}^{1\text{d}} := \left( g^{1\text{d}}\left(\frac{j}{M}\right)\right)_{j\in[M]}.$$

If not explicitly stated, the length of this sampled vector will be clear from context. The length-$M$ discrete Fourier transform (DFT) of a vector $\mathbf{g}^{1\text{d}}$ is defined by

$$\left(\mathbf{F}_M \mathbf{g}^{1\text{d}}\right)_\omega := \frac{1}{M} \sum_{j\in[M]} g_j^{1\text{d}} e^{-2\pi i \omega j/M} = \frac{1}{M} \sum_{j\in[M]} g^{1\text{d}}\left(\frac{j}{M}\right) e^{-2\pi i \omega j/M} \text{ for all } \omega \in [M],$$

where the matrix

$$\mathbf{F}_M := \left(\frac{1}{M} e^{-2\pi i \omega j/M}\right)_{\omega\in[M], j\in[M]}$$

is the discrete Fourier transform matrix. In the context of discrete Fourier transforms, without loss of generality, frequencies $\omega$ are always taken implicitly modulo the length of the DFT, e.g., $\left(\mathbf{F}_M \mathbf{g}^{1\text{d}}\right)_{-1} = \left(\mathbf{F}_M \mathbf{g}^{1\text{d}}\right)_{M-1}$. The same applies to the columns of the DFT matrix.

Given a natural number $M \in \mathbb{N}$ (often prime) and a generating vector $\mathbf{z} \in \{1, \dots, M-1\}^d$, the associated rank-1 lattice is denoted

$$\Lambda(\mathbf{z}, M) := \left\{ \frac{j}{M}\mathbf{z} \bmod \mathbf{1} \mid j \in [M] \right\}.$$

For any $d$-variate function $g$, we define its restriction to a rank-1 lattice as $g^{1\mathrm{d}}(t) := g(t\mathbf{z})$. Notice then that by combining our previous conventions, given $g : \mathbb{T}^d \to \mathbb{C}$, $\mathbf{g}^{1\mathrm{d}}$ is the vector of samples of $g$ on the rank-1 lattice $\Lambda(\mathbf{z}, M)$. The modulus function for a rank-1 lattice $m_{\mathbf{z},M} : \mathcal{I} \to [M]$ is defined by $\mathbf{k} \mapsto \mathbf{k} \cdot \mathbf{z} \bmod M$

### 1.3 Fourier preliminaries

In the sequel, we will make use of various well-known results on Fourier series and discrete Fourier transforms. We provide their statements adapted to our setting here.

**Theorem 1.1.** *The space of all infinitely differentiable periodic functions $C^\infty$ is dense in $L^2$ and $H^1$. In particular, space of trigonometric monomials $\{\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ} \in C^\infty \mid k \in \mathbb{Z}^d\}$ is a basis for $C^\infty$, an orthonormal basis for $L^2$, and an orthogonal basis for $H^1$.*

**Proposition 1.1** (Plancherel's identity)**.** *If $u \in L^2$, then $\hat{u} \in \ell^2$ with $\|u\|_{L^2} = \|\hat{u}\|_{\ell^2}$. If $v \in L^2$, then $\langle u, v\rangle_{L^2} = \langle \hat{u}, \hat{v}\rangle_{\ell^2}$.*

*Proof.* Consider

$$\begin{aligned}
\langle u, v\rangle_{L^2} &= \left\langle \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{u}_{\mathbf{k}} \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}, \sum_{\mathbf{l}\in\mathbb{Z}^d} \hat{v}_{\mathbf{l}} \mathrm{e}^{\pi\mathrm{i}\mathbf{l}\cdot\circ} \right\rangle_{L^2} \\
&= \sum_{\mathbf{k},\mathbf{l}\in\mathbb{Z}^d} \hat{u}_{\mathbf{k}} \overline{\hat{v}}_{\mathbf{l}} \left\langle \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}, \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}\cdot\circ} \right\rangle_{L^2} \\
&= \sum_{\mathbf{k},\mathbf{l}\in\mathbb{Z}^d} \hat{u}_{\mathbf{k}} \overline{\hat{v}}_{\mathbf{l}} \delta_{\mathbf{k},\mathbf{l}} \\
&= \sum_{\mathbf{k}\in\mathbb{Z}^d} \hat{u}_{\mathbf{k}} \overline{\hat{v}}_{\mathbf{k}} \\
&= \langle \hat{u}, \hat{v}\rangle_{\ell^2}
\end{aligned}$$

where we have used the orthonormality of the basis of trigonometric monomials in $L^2$. The norm result comes from taking $v = u$. □

**Lemma 1.2.** *Let $g^{1\mathrm{d}} \in C(\mathbb{T})$ be bandlimited, that is,* $\operatorname{supp}(\hat{g}^{1\mathrm{d}}) \subset \mathcal{B}_M$. *Then $\hat{g}^{1\mathrm{d}} = \mathbf{F}_M \mathbf{g}^{1\mathrm{d}}$.*

*Proof.* Writing $g^{1\mathrm{d}}(t) = \sum_{\omega\in\mathcal{B}_M} \hat{g}^{1\mathrm{d}}_\omega \mathrm{e}^{2\pi\mathrm{i}\omega t}$, for any $\omega \in \mathcal{B}_M$, we calculate

$$\begin{aligned}
\left(\mathbf{F}_M \mathbf{g}^{1\mathrm{d}}\right)_\omega &= \frac{1}{M} \sum_{j\in[M]} g^{1\mathrm{d}}\left(\frac{j}{M}\right) \mathrm{e}^{-2\pi\mathrm{i}\omega j/M} \\
&= \frac{1}{M} \sum_{j\in[M]} \left( \sum_{\omega'\in\mathcal{B}_M} \hat{g}^{1\mathrm{d}}_{\omega'} \mathrm{e}^{2\pi\mathrm{i}\omega' j/M} \right) \mathrm{e}^{-2\pi\mathrm{i}\omega j/M} \\
&= \frac{1}{M} \sum_{\omega'\in\mathcal{B}_M} \hat{g}^{1\mathrm{d}}_{\omega'} \sum_{j\in[M]} \mathrm{e}^{2\pi\mathrm{i}(\omega'-\omega)j/M} \\
&= \sum_{\omega'\in\mathcal{B}_M} \hat{g}^{1\mathrm{d}}_{\omega'} \delta_{0,(\omega'-\omega \bmod M)} \\
&= \hat{g}^{1\mathrm{d}}_\omega,
\end{aligned}$$

as desired. □

**Lemma 1.3.** *For any function $g^{1\mathrm{d}} : \mathbb{T} \to \mathbb{C}$ with Fourier series $g^{1\mathrm{d}}(t) = \sum_{\omega\in\mathbb{Z}} \hat{g}^{1\mathrm{d}}_\omega \mathrm{e}^{2\pi\mathrm{i}\omega t}$, define the aliased polynomial*

$$g^{1\mathrm{d}}_{\mathrm{alias}}(t) = \sum_{\omega\in\mathcal{B}_M} \underbrace{\left( \sum_{\omega'\equiv\omega \bmod M} \hat{g}^{1\mathrm{d}}_{\omega'} \right)}_{=:\left(\hat{g}^{1\mathrm{d}}_{\mathrm{alias}}\right)_\omega} \mathrm{e}^{2\pi\mathrm{i}\omega t}.$$

*Then the equispaced samples coincide, giving $\mathbf{g}^{1\mathrm{d}} = \mathbf{g}^{1\mathrm{d}}_{\mathrm{alias}} \in \mathbb{C}^M$ and $\hat{g}^{1\mathrm{d}}_{\mathrm{alias}} = \mathbf{F}_M \mathbf{g}^{1\mathrm{d}}$.*

*Proof.* We group frequencies in the Fourier series of $g^{1\mathrm{d}}$ by their residues in $\mathcal{B}_M$, giving

$$\begin{aligned}
\left(\mathbf{g}^{1\mathrm{d}}\right)_j &= \sum_{\omega'\in\mathbb{Z}} \hat{g}^{1\mathrm{d}}_{\omega'} \mathrm{e}^{2\pi\mathrm{i}\omega' j/M} = \sum_{\omega\in\mathcal{B}_M} \sum_{n\in\mathbb{Z}} \hat{g}^{1\mathrm{d}}_{\omega+nM} \mathrm{e}^{2\pi\mathrm{i}(\omega+nM)j/M} \\
&= \sum_{\omega\in\mathcal{B}_M} \left( \sum_{\omega'\equiv\omega \bmod M} \hat{g}^{1\mathrm{d}}_{\omega'} \right) \mathrm{e}^{2\pi\mathrm{i}\omega j/M} = \left(\mathbf{g}^{1\mathrm{d}}_{\mathrm{alias}}\right)_j \text{ for all } j \in [M].
\end{aligned}$$

Now, since $\mathrm{supp}(\hat{g}^{1\mathrm{d}}_{\mathrm{alias}}) \subset \mathcal{B}_M$, Lemma 1.2 implies $\hat{g}^{1\mathrm{d}}_{\mathrm{alias}} = \mathbf{F}_M \mathbf{g}^{1\mathrm{d}}_{\mathrm{alias}} = \mathbf{F}_M \mathbf{g}^{1\mathrm{d}}$. □

# CHAPTER 2

# CONSTRUCTING MULTIPLE RANK-1 LATTICES DETERMINISTICALLY

As discussed in Section 1.1.1, this chapter focuses on computing Fourier series representations of high-dimensional functions using multiple rank-1 lattices. We begin with a short overview of the lattice construction and associated Fourier recovery methods in Section 2.1 and present the main result in Theorem 2.1. Section 2.2 builds up the proof of Theorem 2.1 with some additional algorithmic comments. Section 2.3 provides numerical tests of our multiple rank-1 lattice construction and Fourier recovery algorithm.

## 2.1 Overview of results

We provide the first known deterministic algorithm for constructing multiple rank-1 lattices [40] for any given index set $\mathcal{I} \subset \mathbb{Z}^d$ with expansion $K_{\mathcal{I}} := \max_{j\in[d]} \left(\max_{\mathbf{k}\in\mathcal{I}} k_j - \min_{\mathbf{l}\in\mathcal{I}}\right) + 1$. The proposed algorithm takes a given generating vector $\mathbf{z} \in [M]^d$ of a reconstructing rank-1 lattice for $\mathcal{I}$ as input and uses it to deterministically generate $L$ smaller lattice sizes $P_0, \ldots, P_{L-1}$. Rather than using the single set $\Lambda(\mathbf{z}, M)$ of $M$ equispaced sampling points along the lattice generating vector $\mathbf{z}$ as in Algorithm 1.1, we use the $L$ sampling sets $\Lambda(\mathbf{z}, P_0), \ldots, \Lambda(\mathbf{z}, P_{L-1})$ which are each still equispaced points in the direction of $\mathbf{z}$ but are spaced out at different intervals. The frequencies in $\mathcal{I}$ are then partitioned into $L$ groups, each associated with one of the smaller lattices This partitioning is tracked by a function $\nu : \mathcal{I} \to [L]$ with the defining property that

$$\mathbf{k} \cdot \mathbf{z} \not\equiv \mathbf{h} \cdot \mathbf{z} \bmod P_{\nu(\mathbf{k})} \text{ for all } \mathbf{h} \neq \mathbf{k} \in \mathcal{I}, \tag{2.1}$$

that is, for each lattice size $P_\ell$, the frequencies in $\nu^{-1}(\ell)$ do not collide with any of the other frequencies in $\mathcal{I}$ modulo $P_\ell$.

This is similar to the reconstructing property underlying the standard rank-1 lattice FFT approach Algorithm 1.1. However, to effectively use these $L$ sampling sets, we must take one FFT along each smaller lattice and match only the frequencies associated to this lattice. Note though that in total, these smaller lattices require only[1] $O(|\mathcal{I}| \log^2(K_{\mathcal{I}}|\mathcal{I}|))$ function evaluations as opposed to the $O(|\mathcal{I}|^2)$ function evaluations generally required by a single rank-1 lattice approach (cf.

Section 1.1.1). This process is outlined Algorithm 2.1.

**Algorithm 2.1** Multiple rank-1 lattice FFT

**Input:** A function $g : \mathbb{T}^d \to \mathbb{C}$, a frequency set of interest $I \subset \mathbb{Z}^d$, multiple rank-1 lattices $\Lambda(\mathbf{z}, P_0), \ldots \Lambda(\mathbf{z}, P_{L-1})$ and mapping $\nu : I \to [L]$ satisfying (2.1)
**Output:** Approximate Fourier coefficients $\hat{\mathbf{g}}^L \in \mathbb{C}^I$

1: **for** $\ell \in [L]$ **do**
2: $\quad \mathbf{g}^{1\mathrm{d},\ell} \leftarrow (g(j\mathbf{z}/P_\ell))_{j\in[P_\ell]}$
3: $\quad \hat{\mathbf{g}}^{1\mathrm{d},\ell} \leftarrow \mathbf{F}_{P_\ell}\mathbf{g}^{1\mathrm{d},\ell}$
4: **end for**
5: **for** $\mathbf{k} \in I$ **do**
6: $\quad \hat{g}^L_{\mathbf{k}} \leftarrow \left(\hat{\mathbf{g}}^{1\mathrm{d},\nu(\mathbf{k})}\right)_{m_{\mathbf{z},P_{\nu(\mathbf{k})}}(\mathbf{k})}$ // recall $m_{\mathbf{z},P_{\nu(\mathbf{k})}}(\mathbf{k}) := \mathbf{k}\cdot\mathbf{z} \bmod P_{\nu(\mathbf{k})}$
7: **end for**

In detail, this chapter is devoted to proving this main theorem concerning the proposed Fourier coefficient reconstruction algorithm on multiple rank-1 lattices.

**Theorem 2.1.** *Let $I \subset \mathbb{Z}^d$ be some frequency set with expansion $K_I$. If $\Lambda(\mathbf{z}, M)$ is a reconstructing single rank-1 lattice for $I$, then one can deterministically construct multiple rank-1 lattices $\Lambda(\mathbf{z}, P_0), \ldots, \Lambda(\mathbf{z}, P_{L-1})$ such that the Fourier coefficients $\{\hat{g}_{\mathbf{k}} \mid \mathbf{k} \in I\}$ of any trigonometric polynomial $g \in \Pi_I$ can be exactly reconstructed using only samples of $g$ on these lattices by Algorithm 2.1. Moreover, the total number of function evaluations on these lattice points is bounded by*

$$\sum_{\ell\in[L]} P_\ell \le \begin{cases} 2 & \text{for } |I| = 1, \\ 6\,|I|\,\log_2(dK_I M)\,\log\left(3\frac{|I|}{\log_2(|I|)}\log_2(dK_I M)\right) & \text{for } |I| \ge 2. \end{cases}$$

*The total computational complexity for the construction of these rank-1 lattices can be bounded by*

$$O\left(|I|^2\,\log(|I|)\,\log(dK_I M) + |I|\,(d + \log(dK_I M)\log(\log(dK_I M)))\right),$$

*and the total computational complexity for reconstructing the Fourier coefficients can be bounded by*

$$O\left(|I|\,\left(d + \log(dK_I M)\log^2(|I|\log_{|I|}(dK_I M))\right)\right). \tag{2.2}$$

[1]These bounds are simplifications of those in Lemma 2.2 and Theorem 2.2 under the mild assumptions that the dimension $d$ and size of the original single rank-1 lattice $M$ are bounded polynomially by $\max\{|I|, K\}$. The latter assumption holds for single rank-1 lattices constructed by CBC methods, cf. Section 2.2.1.

*Proof.* The bounds on the total number of samples from the rank-1 lattices follow from Theorem 2.2, and the bound on the computational complexity for lattice construction follows from Section 2.2.1. The exactness of the Fourier coefficient recovery is a result of Corollary 2.1. Since Algorithm 2.1 requires an FFT of length $\ell$ for each $\ell \in [L]$, the total complexity of Line 1 to Line 4 requires $O(\log\left(\max_{\ell\in[L]} P_\ell\right) \sum_{\ell\in[L]} P_\ell)$ complexity, where the maximum is bounded in Lemma 2.2 and the sum is bounded above. The remaining lines are $O(d|\mathcal{I}|)$ (assuming the modulus functions have not been precomputed, in which case the complexity would reduce to $O(|I|)$). Simplifying these complexities results in (2.2). □

Note that Algorithm 2.1 exactly reconstructs all Fourier coefficients of multivariate trigonometric polynomials with frequencies in a specific frequency set $\mathcal{I}$ which is assumed to be given. One can also apply these rules in order to compute approximations of the Fourier coefficients of more general periodic functions. The resulting trigonometric polynomial can be used as an approximant. For specific approximation settings, the worst case error of this approximation is almost as good as the approximation one achieves when approximating the Fourier coefficients using the lattice rule that uses all samples of the reconstructing single rank-1 lattice from which we start the construction of our rules, cf. [44] for details. From that point of view, the strategy we present here even yields a general approach for significantly reducing the number of sampling values used while only slightly increasing approximation errors. We refer to Corollary 2.1 for more details and to the numerical example in section 2.3.2 that yields Figure 2.5 illustrating this assertion.

## 2.2 The proof of Theorem 2.1

We denote the $q$th prime number by $p_q$, $q \in \mathbb{N}$. For technical reasons, we define $p_0 := 1$.

**Lemma 2.1.** *Let $\mathcal{J} := \{k_1, \ldots, k_J\} \subset \mathbb{N}$ with $k_1 < \ldots < k_J$ and $\tilde{M} \geq k_J - k_1$. Also let $q \in \mathbb{N}$ be such that $p_{q-1} < J \leq p_q$, and $Q := \max\left(1, 2(J-1)\left\lceil \log_{p_q}(\tilde{M}) - 1 \right\rceil\right)$. Then, there exist primes $P_0, \ldots, P_{L-1} \in \mathcal{P}_J := \{p_{q+\ell}\}_{\ell\in[Q]}$ with $L \leq \log_2(J) + 1$ such that*

$$\mathcal{J} = \bigcup_{\ell\in[L]} \{k \in \mathcal{J} \mid k \not\equiv h \bmod P_\ell \text{ for all } h \in \mathcal{J} \setminus \{k\}\}$$

*holds.*

*Proof.* We assume $J \geq 2$ and $\tilde{M} > p_q$, otherwise the statement is trivial. Without loss of generality, we can also assume $\mathcal{J} \subset [\tilde{M}]$ by considering the residues of each $k_j \in \mathcal{J}$ modulo $\tilde{M}$. Note that these residues are all unique due to $\tilde{M} > k_J - k_1$, and therefore, any modulo $P_\ell$ collision of the residues is equivalent to a collision of their original values.

Let $\mathcal{P}_J = \{p_{q+\ell}\}_{\ell \in [Q]}$ be the set of the $Q$ smallest prime numbers not smaller than $p_q$ and $Y_{i,j} := \{p \in \mathcal{P}_J \mid k_i \equiv k_j \bmod p\}$ a subset which collects all primes $p$ in $\mathcal{P}_J$ where the frequencies $k_i \in \mathcal{J}$ and $k_j \in \mathcal{J}$ collide modulo $p$. Since $|k_i - k_j|$ is divisible by each prime $p$ in $Y_{i,j}$, the Chinese Remainder Theorem implies that $\prod_{p \in Y_{i,j}} p$ divides $|k_i - k_j| < \tilde{M}$. Therefore, we observe

$$p_q^{|Y_{i,j}|} \leq \prod_{p \in Y_{i,j}} p < \tilde{M}$$

for all $i \neq j \in \{1, \ldots, J\} =: S_0$, i.e., $k_i \neq k_j$, and this implies $|Y_{i,j}| \leq \left\lceil -1 + \log_{p_q}(\tilde{M}) \right\rceil$.

Moreover, we collect all primes for which $k_i$ collides with any other $k_j$ in the sets

$$\begin{aligned} Y_i :&= \{p \in \mathcal{P}_J \mid k_i \equiv k_j \bmod p \text{ for at least one } k_j \in \mathcal{J} \setminus \{k_i\}\} \\ &= \bigcup_{k_j \in \mathcal{J} \setminus \{k_i\}} Y_{i,j}. \end{aligned}$$

The cardinality of each $Y_i$ is bounded by

$$|Y_i| \leq \sum_{k_j \in \mathcal{J} \setminus \{k_i\}} |Y_{i,j}| \leq (J-1) \left\lceil -1 + \log_{p_q}(\tilde{M}) \right\rceil.$$

Accordingly, we count

$$|\mathcal{P}_J \setminus Y_i| = |\mathcal{P}_J| - |Y_i| \geq Q - (J-1) \left\lceil -1 + \log_{p_q}(\tilde{M}) \right\rceil \geq |\mathcal{P}_J|/2.$$

We define the indicator variables

$$Z_{i,q+\ell} := \begin{cases} 1 & p_{q+\ell} \in \mathcal{P}_J \setminus Y_i, \\ 0 & p_{q+\ell} \in Y_i, \end{cases}$$

for all $k_i \in \mathcal{J}$ and $p_{q+\ell} \in \mathcal{P}_J$. Summing up these indicator variables and using the estimates from above yields

$$\sum_{i \in S_0} \sum_{\ell \in [Q]} Z_{i,q+\ell} = \sum_{i \in S_0} |\mathcal{P}_J \setminus Y_i| \geq |S_0||\mathcal{P}_J|/2 = J|\mathcal{P}_J|/2. \tag{2.3}$$

We will now show that $\sum_{i\in S_0} Z_{i,q+\ell} \geq J/2$ holds for at least one $p_\ell \in \mathcal{P}_J$ by contradiction. To this end, suppose that $\sum_{i\in S_0} Z_{i,q+\ell} < J/2$ for all $p_{q+\ell} \in \mathcal{P}_J$. Accordingly, we estimate

$$\sum_{\ell\in[Q]} \sum_{i\in S_0} Z_{i,q+\ell} < |S_0||\mathcal{P}_J|/2 = J|\mathcal{P}_J|/2$$

which is in contradiction to (2.3). Thus, there exists at least one prime $p_{q+\ell_0} \in \mathcal{P}_J$ such that

$$\sum_{i\in S_0} Z_{i,q+\ell_0} = |\underbrace{\{k_i \in \mathcal{J} \mid k_i \not\equiv k_j \bmod p_{q+\ell_0} \text{ for all } k_j \in \mathcal{J}\setminus\{k_i\}\}}_{=:\mathcal{J}_1}| \geq J/2.$$

We set $P_0 := p_{q+\ell_0}$, and then apply the strategy iteratively.

For $r \in \mathbb{N}$, we define $S_r := \{i \in S_{r-1} \mid \exists k_j \in \mathcal{J}\setminus\{k_i\} \text{ with } k_i \equiv k_j \bmod P_{r-1}\}$ and obtain $s_r := |S_r| \leq 2^{-r}J$. Obviously, we have

$$\mathcal{J}_r' := \{k_i \mid i \in S_r\} = \mathcal{J} \setminus \bigcup_{t=1}^{r} \mathcal{J}_t, \tag{2.4}$$

which are the frequencies that collide modulo each of $P_0, \ldots, P_{r-1}$ to some other frequency in $\mathcal{J}$. We reconsider the variables defined above, but now we restrict the indices to $i \in S_r$. For instance, we observe $\{P_0, \ldots, P_{r-1}\} \subset Y_i$ for all $i \in S_r$. We estimate

$$\sum_{i\in S_r} \sum_{\ell\in[Q]} Z_{i,q+\ell} = \sum_{i\in S_r} |\mathcal{P}_J \setminus Y_i| \geq s_r|\mathcal{P}_J|/2.$$

Using the same contradiction as above, we observe that for at least one $p_{q+\ell_r} \in \mathcal{P}_J \setminus \{P_0, \ldots, P_{r-1}\}$ we have

$$\sum_{i\in S_r} Z_{i,q+\ell_r} = |\underbrace{\{k_i \in \mathcal{J}_r' \mid k_i \not\equiv k_j \bmod p_{q+\ell_r} \text{ for all } k_j \in \mathcal{J}\setminus\{k_i\}\}}_{=:\mathcal{J}_{r+1}}| \geq s_r/2.$$

We now set $P_r := p_{q+\ell_r}$ and increase $r$ up to the point where $0 = |S_{r+1}| = s_{r+1}$ holds. In order to estimate the largest possible step number $r_{\max} \geq r$, we require that $s_{r_{\max}+1} \leq 2^{-(r_{\max}+1)}J < 1$. This is satisfied in particular when $r_{\max} = \lfloor \log_2(J) \rfloor$, and thus we bound the total number of primes as $L \leq r_{\max} + 1 \leq \log_2(J) + 1$. □

*Remark* 2.1. In the proof of Lemma 2.1 we determined that there exist primes in the candidate set $\mathcal{P}_J$ fulfilling the assertion. This set contains the first $Q := \max\left(1, 2(J-1)\left\lceil \log_{p_q}(\tilde{M}) - 1 \right\rceil\right)$

prime numbers not smaller than $p_q$, $p_{q-1} < J \leq p_q$, which only depends on $J$. However, from a theoretical point of view, any prime number $p$ larger than $\lceil J/2 \rceil$ may fulfill $|\mathcal{J}_1| \geq J/2$. Thus, one also could start the set of prime candidates at that point, which would result in a slightly increased cardinality of the candidate set, due to the fact that $Q$ depends on the logarithm to the base of the smallest prime in the candidate set. In spite of that increased cardinality, the maximal prime number in the candidate set $p_{q+Q-1}$, which is estimated in the next lemma, may be decreased. Analyzing this approach leads to similar statements as in the previous and the following lemmas with slightly changed constants. In more detail, both constants $C_1$ and $C_2$ can be bounded less than 3. However, the proof requires more effort and we could not bound the resulting constants lower than those stated in Lemma 2.2.

**Lemma 2.2.** *Assume $J, \tilde{M} \in \mathbb{N}$, $J \leq \tilde{M}$, $p_q$ is the smallest prime not smaller than $J$, and let $Q := \max\left(1, 2(J-1)\left\lceil \log_{p_q}(\tilde{M}) - 1 \right\rceil\right)$. Then, we estimate*

$$p_{q+Q-1} \leq \begin{cases} 2 & \text{for } J = 1, \\ C_1 J \, \log_J(\tilde{M}) \, \log\left(C_2 J \log_J(\tilde{M})\right) & \text{for } J \geq 2, \end{cases}$$

*with absolute constants $C_1 < 2.3\,(1 + \mathrm{e}^{-3/2}) \leq 2.832$ and $C_2 \leq 2.3$.*

*Proof.* For $J = 1$, we observe $p_{q+Q-1} = p_q = 2$.

When $J \geq 2$ and $p_q \geq \tilde{M}$ we have $Q = 1$ and $p_q < 2J$ as a result of Bertrand's postulate.

We then consider $J \geq 2$ and $p_q < \tilde{M}$ which yields

$$q + Q - 1 = q - 1 + 2(J-1)\left\lceil \log_{p_q}(\tilde{M}) - 1 \right\rceil \leq q - 1 + 2(J-1)\log_{p_q}(\tilde{M}).$$

We distinguish two cases, where the final constants from the lemma are determined by the second case. In the first, we restrict to the finite range where $2 \leq J \leq 8$ with $p_q < \tilde{M} < p_q^{\lceil 10/(J-1) \rceil}$, and numerically check that the upper bound

$$p_{q+Q-1} < 2.831\, J \, \log_J(\tilde{M}) \log\left(2.3\, J \, \log_J(\tilde{M})\right)$$

is satisfied. In the second case, where $2 \leq J \leq 8$ with $\tilde{M} \geq p_q^{\lceil 10/(J-1) \rceil}$ or $J \geq 9$, we have

$q + Q - 1 \geq 20$. We then estimate this quantity from above as

$$\begin{aligned} q + Q - 1 &\leq q - 1 + 2(J-1)\log_J(\tilde{M}) = \left(\frac{q-1}{J\log_J(\tilde{M})} + 2\frac{J-1}{J}\right) J\log_J(\tilde{M}) \\ &\leq \left(\frac{q-1}{J} + 2\frac{J-1}{J}\right) J\log_J(\tilde{M}) \leq 2.3\, J\, \log_J(\tilde{M}) \end{aligned}$$

where one achieves the last estimate by computing $\frac{q-1}{J} + 2\frac{J-1}{J}$ for $2 \leq J < 66$ and for $J \geq 66$, one obtains

$$\frac{q-1}{J} + 2\frac{J-1}{J} \overset{[60,\,\text{Eq. (3.6)}]}{\leq} \frac{1.25506}{\log J} + 2 \leq \frac{1.25506}{\log 66} + 2 < 2.3\,.$$

By the estimate

$$e^{-1/2} x \log(x) \leq x^{1+\mathrm{e}^{-3/2}},$$

implying

$$\log\left(e^{-1/2} x \log x\right) = \log(x) + \log\log(x) - \frac{1}{2} \leq (1 + \mathrm{e}^{-3/2}) \log x$$

for $x > 1$, an application of [60, Eq. (3.11)] gives

$$\begin{aligned} p_{q+Q-1} &< (q+Q-1)\left(\log(q+Q-1) + \log\log(q+Q-1) - 1/2\right) \\ &\leq (1+\mathrm{e}^{-3/2})\,(q+Q-1)\,\log(q+Q-1) \\ &\leq (1+\mathrm{e}^{-3/2})\, 2.3\, J\, (\log_J(\tilde{M}))\, \log\left(2.3\, J \log_J(\tilde{M})\right), \end{aligned}$$

as desired. □

Lemma 2.1 ensures the existence of a set of primes $P_0, \ldots, P_{L-1}$ such that each single element of a given set of integers will not collide modulo at least one $P_\ell$ with any other of these integers. We can now use these primes to convert the large reconstructing single rank-1 lattice $\Lambda(z, M, \mathcal{I})$ for some frequency set $\mathcal{I}$ into smaller rank-1 lattices which, based on their ability to avoid collisions in the frequency domain, will provide a sampling set to exactly reconstruct the Fourier coefficients of all multivariate trigonometric polynomials in $\Pi_{\mathcal{I}}$.

**Theorem 2.2.** *Let $\mathcal{I} \subset \mathbb{Z}^d$, $|\mathcal{I}| \geq 2$, and a generating vector $\mathbf{z} \in [M]^d$ of $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathcal{I}$, be given. We determine $\tilde{M} := \max\{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k} \in \mathcal{I}\} - \min\{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k} \in \mathcal{I}\} + 1$.*

*Then there exists a set of prime numbers $P_0, \ldots, P_{L-1}$, $L \leq \log_2(|\mathcal{I}|) + 1$, such that*

$$\mathcal{I} = \bigcup_{\ell \in [L]} \{\mathbf{k} \in \mathcal{I} \mid \mathbf{k} \cdot \mathbf{z} \not\equiv \mathbf{h} \cdot \mathbf{z} \bmod P_\ell \text{ for all } \mathbf{h} \in \mathcal{I} \setminus \{\mathbf{k}\}\}, \tag{2.5}$$

*Thus, the multiple rank-1 lattices $\Lambda(\mathbf{z}, P_0)$, ..., $\Lambda(\mathbf{z}, P_{L-1})$ can be used as input for the multiple rank-1 lattice Fourier transform Algorithm 2.1. The total number of sampling values in these multiple rank-1 lattices can be bounded by*

$$\sum_{\ell \in [L]} P_\ell \leq 2\, C_1 |\mathcal{I}|\, (\log_2(\tilde{M}))\, \log\left(C_2\, |\mathcal{I}| \log_{|\mathcal{I}|}(\tilde{M})\right), \tag{2.6}$$

*with constants $C_1$, $C_2$ from Lemma 2.2.*

*Proof.* Define the set of hashed multivariate frequencies in $\mathcal{I}$ as $\mathcal{I}^{1\mathrm{d}} := \{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathcal{I}\}$. Applying Lemma 2.1 with $\mathcal{J} = \mathcal{I}^{1\mathrm{d}}$ and $\tilde{M} = \max \mathcal{I}^{1\mathrm{d}} - \min \mathcal{I}^{1\mathrm{d}} + 1$ as above, we find a set of prime numbers $\{P_0, \ldots, P_{L-1}\}$ with $\max_{\ell \in [L]} P_\ell \leq p_{q+Q-1}$ and respective rank-1 lattices $\Lambda(\mathbf{z}, P_0), \ldots \Lambda(\mathbf{z}, P_{L-1})$ such that (2.5) holds. We estimate

$$\begin{aligned}\left|\bigcup_{\ell \in [L]} \Lambda(\mathbf{z}, P_\ell)\right| &\leq \sum_{\ell \in [L]} P_\ell \leq (\log_2(|\mathcal{I}|) + 1) p_{q-1+Q} \\ &\overset{\text{Lem. 2.2}}{\leq} 2C_1 |\mathcal{I}| \log_2(\tilde{M}) \log\left(C_2 |\mathcal{I}| \log_{|\mathcal{I}|}(\tilde{M})\right).\end{aligned}$$

□

*Remark* 2.2. We consider two crucial estimates on $\tilde{M}$ in Theorem 2.2

$$\tilde{M} = 1 + \max_{\mathbf{k} \in \mathcal{I}} \left\{\sum_{i \in [d]} k_i z_i\right\} + \max_{\mathbf{h} \in \mathcal{I}} \left\{\sum_{i \in [d]} -h_i z_i\right\} \leq 1 + \sum_{i \in [d]} z_i \left(\max_{\mathbf{k} \in \mathcal{I}} k_i - \min_{\mathbf{h} \in \mathcal{I}} h_i\right) \leq d K_{\mathcal{I}} M \tag{2.7}$$

$$\tilde{M} = 1 + \max_{\mathbf{k} \in \mathcal{I}} \left\{\sum_{i \in [d]} k_i z_i\right\} - \min_{\mathbf{h} \in \mathcal{I}} \left\{\sum_{i \in [d]} h_i z_i\right\} \leq 2\|\mathbf{z}\|_\infty \max_{\mathbf{k} \in \mathcal{I}} \|\mathbf{k}\|_1 + 1 \leq 2M \max_{\mathbf{k} \in \mathcal{I}} \|\mathbf{k}\|_1 \tag{2.8}$$

where $K_{\mathcal{I}}$ is the expansion of $\mathcal{I}$.

The estimate in (2.7) is a rough but universal upper bound on $\tilde{M}$ that depends on the dimension $d$. The inequality in (2.8) provides a dimension independent upper bound on $\tilde{M}$ in cases where the frequency set $\mathcal{I}$ is contained in an $\ell_1$-ball of a specific size $R$, i.e., $\mathcal{I} \subset \{\mathbf{k} \in \mathbb{Z}^d \mid \|\mathbf{k}\|_1 \leq R\}$, which yields $\tilde{M} \leq 2MR$. We refer to Section 2.2.1, where we present and analyze the computational costs and discuss the advantages of the latter estimate.

The Fourier coefficient reconstruction process in Algorithm 2.1 allows for us to prove theoretical error guarantees for approximation of functions that are not necessarily Fourier polynomials supported on a known $\mathcal{I}$. In particular, we are able provide $L^\infty$ and $L^2$ bounds for the approximation error in terms of the error in truncating a function's Fourier coefficients to a chosen $\mathcal{I}$. The proof relies on the fact that the aliasing error in a DFT is comparable to the truncation error. See, e.g., [44, Lemma 3.1] for similar results and further details. For the following we define the Wiener algebra $W := \{g \in L^1 \mid \|\hat{g}\|_{\ell^1} < \infty\}$.

**Corollary 2.1.** *Let $g \in W$ and fix a frequency set $\mathcal{I} \subset \mathbb{Z}^d$ with $|\mathcal{I}| < \infty$. Use the multiple rank-1 lattices for $\mathcal{I}$ in Theorem 2.2 with Algorithm 2.1 to produce $\hat{\mathbf{g}}^L$ and $g^L := \sum_{\mathbf{k}\in\mathcal{I}} \hat{g}^L_{\mathbf{k}} \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ} \in \Pi_{\mathcal{I}}$. Then $g^L$ approximates $g$ with the error bounds*

$$\begin{aligned}\left\|g - g^L\right\|_{L^\infty} &\leq (1+L)\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1}\\ \left\|g - g^L\right\|_{L^2} &\leq \left(1+\sqrt{L}\right)\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2}.\end{aligned}$$

*Proof.* By the triangle inequality

$$\begin{aligned}\left\|g - g^L\right\|_{L^\infty} &\leq \sum_{\mathbf{k}\in\mathcal{I}} \left|\hat{g}_{\mathbf{k}} - \hat{g}^L_{\mathbf{k}}\right| + \sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\mathcal{I}} |\hat{g}_{\mathbf{k}}|\\ &= \left\|\hat{g}|_{\mathcal{I}} - \hat{g}^L\right\|_{\ell^1} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1}.\end{aligned}$$

Now, note that by partitioning the frequencies $\mathbf{k} \in \mathcal{I}$ by their values of $\nu(\mathbf{k})$, for $\hat{\mathbf{g}}^{1\mathrm{d},\ell}$ as in Line 3 of Algorithm 2.1, we obtain

$$\begin{aligned}\left\|\hat{g}|_{\mathcal{I}} - \hat{g}^L\right\|_{\ell^1} &= \sum_{\ell\in[L]} \sum_{\mathbf{k}\in\nu^{-1}(\ell)} \left|\hat{g}_{\mathbf{k}} - \hat{g}^L_{\mathbf{k}}\right|\\ &= \sum_{\ell\in[L]} \sum_{\mathbf{k}\in\nu^{-1}(\ell)} \left|\hat{g}_{\mathbf{k}} - \hat{\mathbf{g}}^{1\mathrm{d},\ell}_{\mathbf{k}\cdot\mathbf{z} \bmod P_\ell}\right|\\ &= \sum_{\ell\in[L]} \sum_{\mathbf{k}\in\nu^{-1}(\ell)} \left|\hat{g}_{\mathbf{k}} - \sum_{\omega\equiv\mathbf{k}\cdot\mathbf{z} \bmod P_\ell} \hat{g}^{1\mathrm{d}}_{\omega}\right|,\end{aligned}$$

where the final line follows from Lemma 1.3. Since the multidimensional frequencies $\mathbf{k} \in \mathbb{Z}^d$ of $\hat{g}$ map to the frequencies of $\hat{g}^{1\mathrm{d}}$ by $\mathbf{k} \mapsto \mathbf{k}\cdot\mathbf{z}$ and for any $\mathbf{k} \in \nu^{-1}(\ell)$, there are no such $\mathbf{h} \in \mathcal{I} \setminus \{\mathbf{k}\}$

such that $\mathbf{h} \cdot \mathbf{z} \equiv \mathbf{k} \cdot \mathbf{z} \bmod P_\ell$, we know that

$$\left| \hat{g}_{\mathbf{k}} - \sum_{\omega \equiv \mathbf{k}\cdot\mathbf{z} \bmod P_\ell} \hat{g}^{1\mathrm{d}}_{\omega} \right| \leq \sum_{\substack{\mathbf{h} \in \mathbb{Z}^d \setminus \mathcal{I} \\ \mathbf{h}\cdot\mathbf{z} \equiv \mathbf{k}\cdot\mathbf{z} \bmod P_\ell}} |\hat{g}_{\mathbf{h}}|.$$

Thus

$$\begin{aligned}
\left\| \hat{g}|_{\mathcal{I}} - \hat{g}^L \right\|_{\ell^1} &= \sum_{\ell \in [L]} \sum_{\mathbf{k} \in \nu^{-1}(\ell)} \left| \hat{g}_{\mathbf{k}} - \sum_{\omega \equiv \mathbf{k}\cdot\mathbf{z} \bmod P_\ell} \hat{g}^{1\mathrm{d}}_{\omega} \right| \\
&\leq \sum_{\ell \in [L]} \sum_{\mathbf{k} \in \nu^{-1}(\ell)} \sum_{\substack{\mathbf{h} \in \mathbb{Z}^d \setminus \mathcal{I} \\ \mathbf{h}\cdot\mathbf{z} \equiv \mathbf{k}\cdot\mathbf{z} \bmod P_\ell}} |\hat{g}_{\mathbf{h}}| \\
&\leq \sum_{\ell \in [L]} \sum_{\mathbf{h} \in \mathbb{Z}^d \setminus \mathcal{I}} |\hat{g}_{\mathbf{h}}| \\
&= L \| \hat{g} - \hat{g}|_{\mathcal{I}} \|_{\ell^1},
\end{aligned}$$

finishing the proof of the $L^\infty/\ell^1$ result. The $L^2/\ell^2$ result follows by replacing the $L^\infty$ norm by the $L^2$ norm, taking squares of all terms, and taking a final square root. □

As considered in [40, Subsection 4.2] for randomized lattice constructions, we can take an alternative approach to Theorem 2.2 which requires fewer samples at the cost of having only theoretical reconstruction guarantees for trigonometric polynomials (i.e., the results concerning approximation discussed in Corollary 2.1 do not apply in a straightforward manner). Rather than require that at each step of the lattice construction, a prime $p$ is chosen so that a set of frequencies can be obtained which do not collide with *any* other frequency in the original frequency set modulo $p$, we instead recursively reduce the size of the set that the resulting rank-1 lattice has the reconstruction property over without concern for other frequencies.

**Theorem 2.3.** *Let $\mathcal{I} \subset \mathbb{Z}^d$, $|\mathcal{I}| \geq 1$, $d \geq 2$, $\tilde{M} := \max\{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathcal{I}\} - \min\{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathcal{I}|\} + 1$. For $\Lambda(\mathbf{z}, M)$ a reconstructing single rank-1 lattice for $\mathcal{I}$ , there exist primes $P_0, \ldots, P_{L-1}$, $L \leq \log_2(|\mathcal{I}|) + 1$, with*

$$\sum_{\ell \in [L]} P_\ell \leq \begin{cases} 2 & \text{for } |\mathcal{I}| = 1, \\ 8\,|\mathcal{I}|\,\log_2(\tilde{M})\,\log\left(2\,\log_2(\tilde{M})\right) & \text{for } |\mathcal{I}| \geq 2, \end{cases} \tag{2.9}$$

*such that for every $g \in \Pi_{\mathcal{I}}$, the formula*

$$\hat{g}_{\mathbf{k}} = \frac{1}{P_{\nu(\mathbf{k})}} \sum_{j=0}^{P_{\nu(\mathbf{k})}-1} g_{\nu(\mathbf{k})-1}\left(\frac{(j\mathbf{z}) \bmod P_{\nu(\mathbf{k})}}{P_{\nu(\mathbf{k})}}\right) \mathrm{e}^{\frac{-2\pi \mathrm{i} j \mathbf{k}\cdot\mathbf{z}}{P_{\nu(\mathbf{k})}}} \tag{2.10}$$

$$\textit{with} \qquad g_{\nu(\mathbf{k})-1}(\mathbf{x}) := g(\mathbf{x}) - \sum_{\mathbf{h}\in\{\mathbf{l}|\nu(\mathbf{l})<\nu(\mathbf{k})\}} \hat{g}_{\mathbf{h}}\, \mathrm{e}^{2\pi\mathrm{i}\mathbf{h}\cdot\mathbf{x}}$$

*holds where $\nu : \mathcal{I} \to [L]$ maps frequencies to the lattice used to reconstruct the corresponding Fourier coefficient, i.e., we can uniquely reconstruct each multivariate trigonometric polynomial with frequencies in $\mathcal{I}$ using samples along the rank-1 lattices $\Lambda(\mathbf{z}, P_0), \ldots, \Lambda(\mathbf{z}, P_{L-1})$.*

*Proof.* The proof is simply a recursive application of part of the previously discussed approach, so we only provide a sketch.

We use only the first prime $P_0$ from Lemma 2.1 to determine a set of frequencies $\mathcal{I}_0 \subset \mathcal{I}$ such that $\Lambda(\mathbf{z}, P_0)$ is a reconstructing single rank-1 lattice for $\mathcal{I}_0$ with $|\mathcal{I}_0| \geq |\mathcal{I}|/2$. Performing the reconstruction process in Theorem 2.2 for only frequencies in $\mathcal{I}_0$ using samples from $\Lambda(\mathbf{z}, P_0)$ recovers the corresponding Fourier coefficients exactly. This then defines the correspondence $\nu(\mathbf{k}) = 0$ for all $\mathbf{k} \in \mathcal{I}_0$. Subtracting off the recovered polynomial terms and recursively repeating the process with the frequency set $\mathcal{I} \setminus \mathcal{I}_0$ gives (2.10).

The upper bound on the number of samples is a result of Lemma 2.2, noting that at each step, the cardinality of the frequency set is reduced by half. Splitting the dependence on $|\mathcal{I}|$ and $\tilde{M}$ in the second logarithm using the inequality $\log(xy) \leq 2(\log x)(\log y)$ for $x, y \geq \mathrm{e}$ and estimating the resulting geometric series gives (2.9). □

### 2.2.1 Analysis of lattice construction

The approach analyzed in Theorem 2.2 provides a constructive, deterministic method for building reconstructing multiple rank-1 lattices from reconstructing single rank-1 lattices. Algorithm 2.2 summarizes the suggested approach in detail. In the following, we analyze the runtime complexity.

We start by analyzing Line 1 which is $O(d\,|\mathcal{I}|)$. The arithmetic complexity of Lines 2 and 4 are dominated by determining the set of primes $\mathcal{P}_{|\mathcal{I}|}$, which can be done in linear time with respect to $p_{q+Q-1} \leq C_1|\mathcal{I}|(\log_{|\mathcal{I}|}(\tilde{M}))\log\Big(C_2|\mathcal{I}|\log_{|\mathcal{I}|}(\tilde{M})\Big)$ estimated in Lemma 2.2, therefore requiring

**Algorithm 2.2** Deterministic construction of multiple rank-1 lattice suitable for reconstruction and approximation, according to Theorem 2.2 and Lemma 2.1

**Input:** frequency set $\mathcal{I} \subset \mathbb{Z}^d$, generating vector $\mathbf{z} \in \mathbb{N}_0^d$ of a reconstructing single rank-1 lattice for $\mathcal{I}$

**Output:** number of lattices $L$, lattice sizes $P_0, \ldots, P_{L-1}$, and mapping $\nu : \mathcal{I} \to [L]$ for which coefficients are computed by which lattice

1: $\mathcal{J}_0' \leftarrow \{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathcal{I}\}$
2: Determine $q \in \mathbb{N}$ s.t. $p_{q-1} < |\mathcal{I}| \leq p_q$ // recall $p_\ell$ is the $\ell$th prime
3: $Q \leftarrow \max\left(0, 2(|\mathcal{I}|-1)\left\lceil \log_{p_q}(\tilde{M}) - 1 \right\rceil\right)$ // recall $\tilde{M} := \max \mathcal{J}_0' - \min \mathcal{J}_0' + 1$
4: $\mathcal{P}_{|\mathcal{I}|} \leftarrow \left\{p_{q+\ell}\right\}_{\ell \in [Q]}$
5: Initialize $r \leftarrow 0$ and $\nu : \mathcal{I} \to \mathbb{N}$ with $\nu(\mathbf{k}) = 0$ for all $\mathbf{k} \in \mathcal{I}$
6: **repeat**
7: **for all** $\ell \in [Q]$ **do**
8: $\mathcal{J}_{r+1}' = \emptyset$
9: **for all** $\mathbf{k} \cdot \mathbf{z} \in \mathcal{J}_r'$ **do**
10: $\nu(\mathbf{k}) \leftarrow r$
11: **if** $\mathbf{k} \cdot \mathbf{z} \equiv h' \bmod p_{q+\ell}$ for any $h' \in \mathcal{J}_0' \setminus \{h\}$ **then**
12: $\mathcal{J}_{r+1}' \leftarrow \mathcal{J}_{r+1}' \cup \{\mathbf{k} \cdot \mathbf{z}\}$
13: **end if**
14: **end for**
15: **if** $\left|\mathcal{J}_{r+1}'\right| \leq \left|\mathcal{J}_r'\right|/2$ **then**
16: $P_r \leftarrow p_{q+\ell}$
17: **break**
18: **end if**
19: **end for**
20: $r \leftarrow r + 1$
21: **until** $\mathcal{J}_r' = \emptyset$
22: $L \leftarrow r$

$O\left(|\mathcal{I}| \log(\tilde{M}) \log(\log(\tilde{M}))\right)$ arithmetic operations.

The goal of the loop from Lines 6 to 21 is to separate the frequencies in $\mathcal{I}$ into $L$ groups. Each of these $\ell \in [L]$ groups is assigned a prime $P_\ell$ so that the frequencies do not collide with any others in $\mathcal{I}$ modulo $P_\ell$. In the worst case, there will be at most $L = O(\log(|\mathcal{I}|))$ (cf. Theorem 2.2) repetitions of this loop. The first inner loop requires, at most, a scan through each of the $Q = O(|\mathcal{I}| \log_{p_q}(\tilde{M}))$ primes in $\mathcal{P}_{|I|}$. The body of this inner loop can be accomplished in $O(|\mathcal{I}| \log(|\mathcal{I}|))$ time. Indeed, this requires the computation of $k \bmod p_{q+\ell}$ for all $k \in \mathcal{J}_0'$ (where we make sure to track the association between $k \bmod p_{q+\ell}$ and the original frequency $\mathbf{k} \in \mathcal{I}$ with $k = \mathbf{k} \cdot \mathbf{z}$), a sort of these residues, and a linear scan to determine duplicates of the residues of elements originally in $\mathcal{J}_r'$

(where we can rely on our function $\nu$ and the aforementioned association between $k \bmod p_{q_\ell}$ and $\mathbf{k}$). This is dominated by the sort complexity, $O(|\mathcal{I}|\log(|\mathcal{I}|))$. Thus, the total complexity for Lines 6 to 21 is $O\left(|I|^2 \log(|\mathcal{I}|)\log(\tilde{M})\right)$ (noting that $\log(|\mathcal{I}|) < \log(p_q)$). Altogether, we observe a runtime complexity of

$$O\left(|\mathcal{I}|^2\ \log(|\mathcal{I}|)\ \log(\tilde{M}) +\ |\mathcal{I}|\left(d + \log(\tilde{M})\log(\log(\tilde{M}))\right)\right).$$

In the following, we comment on practical issues of Algorithm 2.2. Line 1 might suffer from overflowing integers which can be avoided by using higher precision integer representations. An alternative is to skip this precomputation and instead compute the inner products modulo $p_{q+\ell}$ on the fly in Line 11 which will increase the runtime complexity by a factor of $d$ in the first summand. Note also that one does not necessarily need to compute $\tilde{M}$ in advance. For the loop over primes starting in Line 7, one might just start with the prime $p_q$ and increase the prime number using some "nextprime" function, which would increase the second summand in the runtime complexity.

Finally, we discuss the range of the numbers $\tilde{M}$ as well as the influence of the original single rank-1 lattice on the estimates herein. In general, there are two different suitable approaches for finding a single reconstructing rank-1 lattice for a given frequency index set $\mathcal{I}$. A simple approach is to just pick a rank-1 lattice $\Lambda(\mathbf{z}, M)$ that provides the reconstruction property from a simple number-theoretic point of view. For instance one can choose generating vectors $\mathbf{z}$ and lattice sizes $M$ that fulfill

$$z_0 \in \mathbb{N}, \quad z_i \geq (1 + \max_{\mathbf{k}\in\mathcal{I}} k_{i-1} - \min_{\mathbf{h}\in\mathcal{I}} h_{i-1}) z_{i-1}, \quad i = 1, \ldots, d-1,$$
$$M \geq (1 + \max_{\mathbf{k}\in\mathcal{I}} k_{d-1} - \min_{\mathbf{h}\in\mathcal{I}} h_{d-1}) z_{d-1}.$$

Clearly, even for extremely sparse frequency sets and moderate expansions of $\mathcal{I}$ this approach will lead to exponentially increasing $d-1$ components $z_{d-1} \geq 2^{d-1}$ and lattice sizes $M \geq 2^d$.

As in Remark 2.2, this approach will lead to exponential increase in $\tilde{M}$ and thus a linear dependence of the dimension $d$ in all $\log(\tilde{M})$ terms. From a theoretical point of view, this turns out to be disadvantageous for higher dimensions $d$ due to the fact that the runtime complexity of Algo-

rithm 2.2 as well as the estimates of the total number of sampling values in Theorems 2.2 and 2.3 will be affected by this factor.

A more costly way of determining reconstructing single rank-1 lattices is a suitable CBC construction as suggested in [46], which requires a computational complexity in $O\left(d|\mathcal{I}|^2\right)$. The additional computational effort pays off when applying the theoretical bounds on the resulting lattice size $M$. In more detail, the CBC approach offers reconstructing rank-1 lattices with prime lattice sizes $M$ bounded from above by $M \leq \max(|\mathcal{I}|^2, 2(K_{\mathcal{I}} + 1))$, cf. [39, 46]. As a consequence, the estimates in Remark 2.2 give $\tilde{M} \leq CdK_{\mathcal{I}}^2|\mathcal{I}|^2$ or even $\tilde{M} \leq C'RK_{\mathcal{I}}|\mathcal{I}|^2$ for $\mathcal{I}$ a subset of an $\ell_1$-ball of radius $R$. Thus, the estimates on the required number of sampling values for unique reconstruction of multivariate trigonometric polynomials in $\Pi_{\mathcal{I}}$ estimated in (2.6) are respectively only either logarithmically dependent on $d$ or even independent of $d$.

### 2.3 Numerics

In this section, we investigate the statements of Theorems 2.2 and 2.3 numerically[2]. We consider different types of frequency sets $\mathcal{I}$. In particular, we use symmetric hyperbolic cross type frequency sets

$$\mathcal{I} = H^d_{R,\text{even}} := \left\{ \mathbf{k} := (k_0, \ldots, k_{d-1})^\top \in (2\mathbb{Z})^d \mid \prod_{t\in[d]} \max(1, |k_t|) \leq R \right\} \tag{2.11}$$

with expansion parameter $R \in \mathbb{N}$, which results in $K_{\mathcal{I}} \leq 2R$, in up to $d = 9$ spatial dimensions. These frequency sets $H^d_{R,\text{even}}$ have the property that in each frequency component only even indices occur. This matches the behavior of the Fourier support of the test function $G^d_3$ introduced below in Section 2.3.2 which we approximate using samples on multiple rank-1 lattices, see also [50, 44] and [65, section 2.3.5].

In addition, we use random frequency sets $\mathcal{I} \subset ([-R, R] \cap \mathbb{Z})^d$, which yield $K_{\mathcal{I}} \leq 2R$, and we consider these in up to $d = 10\,000$ spatial dimensions.

[2]All code is available at https://www.math.msu.edu/~markiwen/Code.html

### 2.3.1 Deterministic multiple rank-1 lattices generated by Algorithm 2.2 suitable for reconstruction and approximation

#### 2.3.1.1 Resulting numbers of samples and oversampling factors

In the beginning, we determine the overall number of samples in the multiple rank-1 lattices output from Algorithm 2.2. Up to an additive term of $1 - L$, this corresponds to $\sum_{\ell \in [L]} P_\ell$ in Theorem 2.2, since the node $\mathbf{0}$ (point of origin) is contained in each of the resulting rank-1 lattices $\Lambda(\mathbf{z}, P_\ell)$. We start with symmetric hyperbolic cross sets $\mathcal{I} = H^d_{R,\text{even}}$ as defined in (2.11) and consider three different types of reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, as input for Algorithm 2.2.

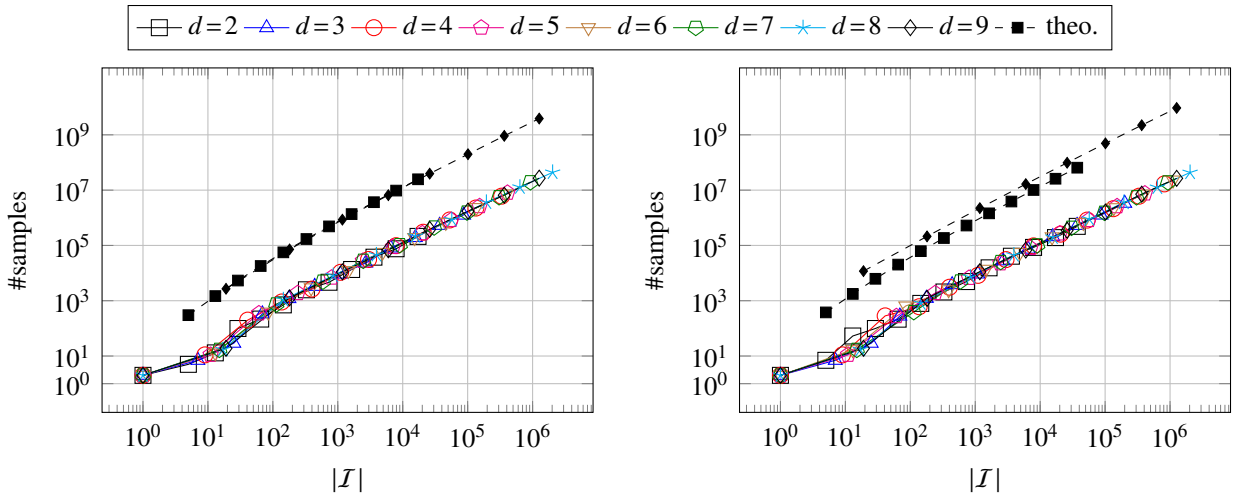
First, we use the rank-1 lattices from [50, Table 6.1], which were generated by the CBC method [38, Algorithm 3.7], as input for Algorithm 2.2. We plot the results in Figure 2.1a for spatial dimensions $d \in \{2, 3, \ldots, 9\}$ and with various refinements $R \in \mathbb{N}$ of $\mathcal{I} = H^d_{R,\text{even}}$. The observed numbers of samples seem to behave slightly worse than linear with respect to the cardinality of the frequency set $\mathcal{I}$. The corresponding theoretical upper bounds according to Theorem 2.2 using (2.8) for $\tilde{M}$ are also shown as filled markers with dashed lines for spatial dimensions $d \in \{2, 9\}$ in Figure 2.1a. The plotted upper bounds are distinctly larger and their slopes seem to be slightly higher than those observed by plotting the numerical tests.

Second, we consider single reconstructing rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, with

$$\mathbf{z} := (1, K_\mathcal{I} + 1, (K_\mathcal{I} + 1)^2, \ldots, (K_\mathcal{I} + 1)^{d-1})^\top \text{ and } M := (K_\mathcal{I} + 1)^d = (2R + 1)^d, \tag{2.12}$$

where $K_\mathcal{I} = 2R$ in our case, and we show the results in Figure 2.1b. We observe that the obtained numbers of samples are similar to the ones in Figure 2.1a, and the theoretical upper bounds according to Theorem 2.2 using (2.8) for $\tilde{M}$ are slightly higher due components of the generating vector $\mathbf{z}$ being larger.

Third, we apply Algorithm 2.2 to the reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, as




(a) $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7]

(b) $\mathbf{z}$ and $M$ according to (2.12)

Figure 2.1 Overall #samples $= 1 - L + \sum_{\ell \in [L]} P_\ell$ for symmetric hyperbolic cross index sets $\mathcal{I} = H^d_{R,\text{even}}$. Filled markers with dashed lines represent theoretical upper bounds from Theorem 2.2 for $d \in \{2, 9\}$ calculated using (2.8).

considered in [37, section 6]. In detail, we choose

$$M := \prod_{t \in [d]} q_t \text{ and } \mathbf{z} := (M/q_0, M/q_1, \ldots, M/q_{d-1})^\top, \tag{2.13}$$

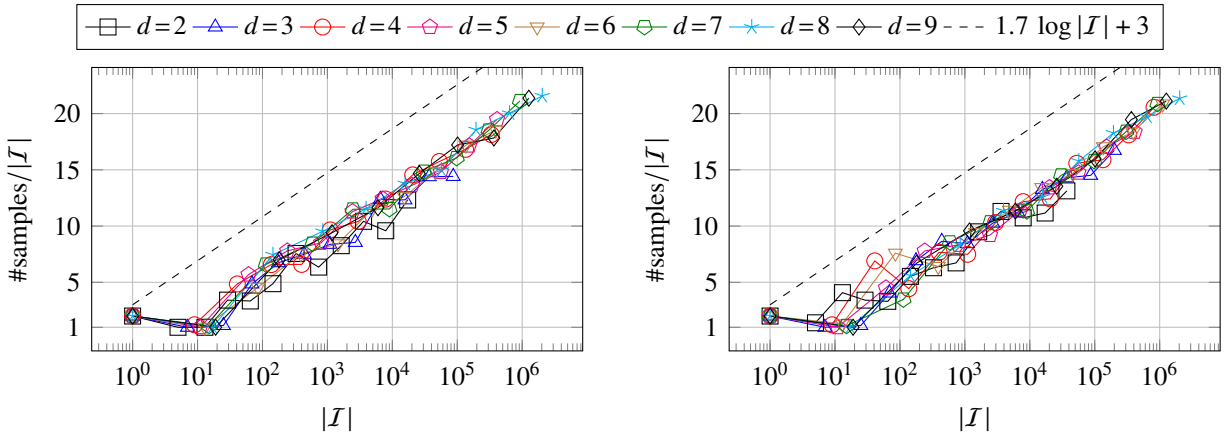
where $q_0 := dK_{\mathcal{I}} + d + 1$ and $q_{t+1} := \min\{p \in \mathbb{N} \mid p > q_t \text{ and } p \text{ prime}\}$.

Here, the observed numerical results yield results that do not differ recognizably from Figure 2.1b, and we therefore omit these plots. We would like to point out, that the theoretical upper bounds for that kind of reconstructing single rank-1 lattices are slightly worse than those plotted in Figure 2.1b, cf. Remark 2.2.

Note that when running Algorithm 2.2 using single rank-1 lattices $\Lambda(\mathbf{z}, M)$ of type (2.12) and (2.13) in practice, one may need to deal with limited numeric precision in the computer arithmetic. For instance, for higher spatial dimensions, some components $z_t$ of the generating vector $\mathbf{z}$ may become larger than 64-bit integers. This means that the sets $\mathcal{J}_r'$ may have to be computed carefully and repeatedly modulo each considered prime $p \in \mathcal{P}_{|\mathcal{I}|}$ when searching for the primes $P_0, \ldots, P_{L-1}$ in Lines 6 to 21 of Algorithm 2.2.

In order to have a closer look at the number of samples, we visualize the oversampling factor #samples $/\,|\mathcal{I}| = (1 - L + \sum_{\ell \in [L]} P_\ell)/|\mathcal{I}|$ in Figure 2.2. For the considered test cases and the

three different types of lattices, we observe that the oversampling factors are below $1.7 \log |\mathcal{I}| + 3$ for $|\mathcal{I}| > 1$. This is distinctly smaller than the theoretical upper bounds in Theorem 2.2 suggest, which have a constant of $\approx 5.7$ and additional logarithmic factors depending on $\tilde{M}$. For instance in Figure 2.2a, for $\mathcal{I} = H^9_{256,\text{even}}$ (cardinality $|\mathcal{I}| = 1\,264\,513$ and #samples $= 27\,025\,383$), the oversampling factor is $\approx 21.37$ whereas the corresponding upper bound for the oversampling factor is $\approx 3\,069$ according to Theorem 2.2 using (2.8) for $\tilde{M}$. The plots for reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, according to (2.13) look similar to the ones according to (2.12), where the latter are shown in Figure 2.2b. Moreover, we only observe a relatively small difference compared to Figure 2.2a.



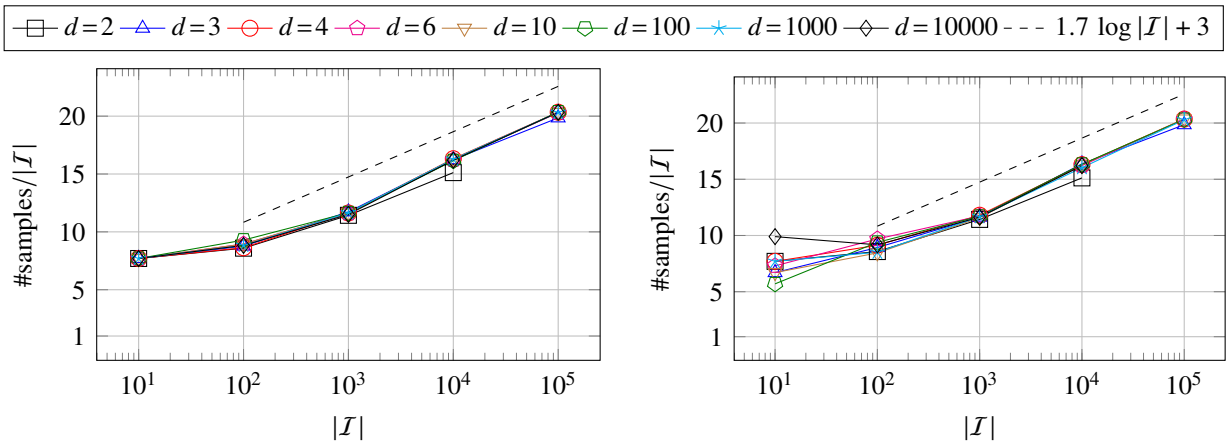

(a) $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7]

(b) $\mathbf{z}$ and $M$ according to (2.12)

Figure 2.2 Oversampling factors for deterministic reconstructing multiple rank-1 lattices for symmetric hyperbolic cross index sets $H^d_{R,\text{even}}$.

Next, we change the setting and use frequency sets $\mathcal{I}$ drawn uniformly randomly from cubes $[-R, R]^d \cap \mathbb{Z}^d$. We generate reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, using [38, Algorithm 3.7]. Then, we apply Algorithm 2.2 in order to deterministically generate reconstructing multiple rank-1 lattices. We repeat the test 10 times for each setting with newly randomly chosen frequency sets $\mathcal{I}$ and determine the maximum number of samples over the 10 repetitions. For frequency set sizes $|\mathcal{I}| \in \{10, 100, 1\,000, 10\,000\}$ in $d \in \{2, 3, 4, 6, 10, 100, 1\,000, 10\,000\}$ spatial dimensions and $|\mathcal{I}| = 100\,000$ for only some of the aforementioned spatial dimensions $d$, we visualize the resulting oversampling factors in Figure 2.3 for expansion parameter $R = 64$ ($K_\mathcal{I} \leq 128$).

Using different reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, as in Figure 2.2, changes the oversampling factors only slightly, and the oversampling factors are still well below $1.7 \log |\mathcal{I}| + 3$, compare Figures 2.3a and 2.3b. The plots for reconstructing single rank-1 lattices $\Lambda(\mathbf{z}, M)$ according to (2.13) are omitted since they look very similar to Figure 2.3b. As mentioned before, we have to take care of possible issues with numeric precision when running Algorithm 2.2 on reconstructing single rank-1 lattices of type (2.12) and (2.13) in practice.




(a) $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7] (b) $\mathbf{z}$ and $M$ according to (2.12)

Figure 2.3 Oversampling factors for deterministic reconstructing multiple rank-1 lattices for random frequency sets $\mathcal{I} \subset \{-64, -63, \ldots, 64\}^d$.

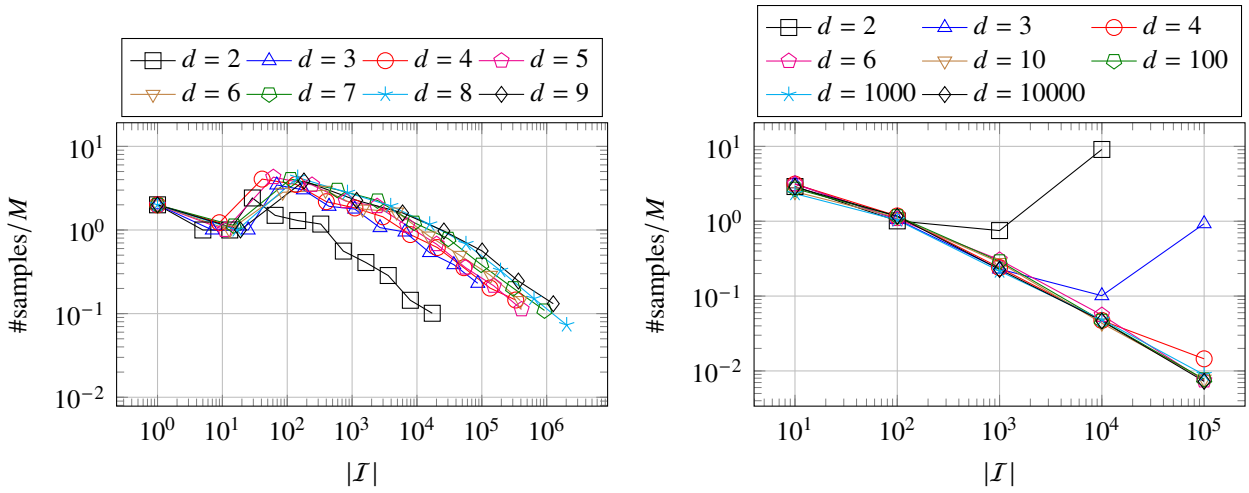
#### 2.3.1.2 Improvement of numbers of samples compared to single rank-1 lattices constructed component-by-component

For the deterministic reconstructing multiple rank-1 lattices generated by Algorithm 2.2 in the previous subsection, one aspect of particular interest is the total number of nodes compared to the reconstructing single rank-1 lattices, which are given as an input to the algorithm. We investigate this in more detail for the case of lattices generated component-by-component by [38, Algorithm 3.7]. These reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, are specifically tailored to the structure of the corresponding frequency sets $\mathcal{I}$. We do not consider the case when Algorithm 2.2 is applied to single rank-1 lattices of type (2.12) or (2.13) as these ones are typically extremely large compared to the cardinality $|\mathcal{I}|$ of the frequency sets $\mathcal{I}$.

First, we start with symmetric hyperbolic cross index sets $\mathcal{I} = H^d_{R,\text{even}}$ and reconstructing sin-

gle rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7]. In Figure 2.4a, the obtained #samples from Figure 2.1a is divided by the size $M$ of the single rank-1 lattice. We observe that for smaller expansion parameters $R$ and consequently smaller cardinalities $|\mathcal{I}|$, the generated multiple rank-1 lattices still consist of more nodes than the corresponding single rank-1 lattices and therefore the ratio is larger than one. One main reason for this behavior is that for the component-by-component constructed single rank-1 lattices, the number of nodes is initially much less than the worst case upper bounds of almost $O(|\mathcal{I}|^2)$ suggest, cf. [38, section 3.8.2] for a detailed discussion. Once a certain expansion $K_{\mathcal{I}}$ and cardinality $|\mathcal{I}|$ have been reached, the multiple rank-1 lattices outperform the single rank-1 lattices, yielding ratios around 0.1 in Figure 2.4a, i.e., Algorithm 2.2 reduces the number of sampling nodes by 9/10.

Second, we consider randomly generated frequency sets as in Figure 2.3a. In Figure 2.4b, we visualize the ratios of the number of nodes of the deterministic reconstructing multiple rank-1 lattices generated by Algorithm 2.2 over the lattice sizes $M$ of the reconstructing single rank-1 lattices generated by [38, Algorithm 3.7]. For the spatial dimensions $d \geq 4$ considered in Figure 2.3a, the ratios decrease rapidly for increasing cardinality $|\mathcal{I}|$, and we do not observe any noticeable dependence on the spatial dimension $d$. Note that in the case $d = 2$, the ratios are close to or above one since the cube $\{-64, -63, \ldots, 64\}^2$ of possible frequencies only has cardinality 16 641 and the single rank-1 lattices already have small oversampling factors $M/|\mathcal{I}| < 16$. Similarly, in the case $d = 3$ for cardinality $|\mathcal{I}| = 10^5$, the frequency set $\mathcal{I}$ fills approximately 1/20 of the cube $\{-64, -63, \ldots, 64\}^3$ and again the low oversampling factors $M/|\mathcal{I}| < 22$ of the single rank-1 lattices are hard to beat for multiple rank-1 lattices.




(a) using symmetric hyperbolic cross index sets $\mathcal{I} = H^d_{R,\text{even}}$

(b) $\mathcal{I} \subset \{-64, -63, \ldots, 64\}^d$ random frequency sets

Figure 2.4 Ratio #samples for deterministic reconstructing multiple rank-1 lattices suitable for approximation over lattice size $M$ of reconstructing single rank-1 lattice $\Lambda(\mathbf{z}, M)$, where $\Lambda(\mathbf{z}, M)$ was generated by [38, Algorithm 3.7].

### 2.3.2 Comparison of reconstructing multiple and single rank-1 lattices for function approximation

As mentioned in Corollary 2.1, we can use Algorithm 2.1 to compute approximations of functions from samples along multiple rank-1 lattices. We consider the tensor-product test functions $G_3^d \colon \mathbb{T}^d \to \mathbb{C}$ from [50], $G_3^d(\mathbf{x}) := \prod_{j\in[d]} g_3(x_j)$, where the one-dimensional function $g_3 \colon \mathbb{T} \to \mathbb{C}$ is defined by

$$g_3(x) := 4\sqrt{\frac{3\pi}{207\pi - 256}}\left(2 + \operatorname{sgn}((x \bmod 1) - 1/2)\,\sin(2\pi x)^3\right)$$

and $\|G_3^d\|_{L^2(\mathbb{T}^d)} = 1$. The function $G_3^d$ lies in a so-called Sobolev space of dominating mixed smoothness with smoothness almost 3.5 such that its Fourier coefficients $\hat{G}_3^d$ decay fast with respect to hyperbolic cross structures. In addition, $(\hat{G}_3^d)_{\mathbf{k}} = 0$ if at least one component of $\mathbf{k}$ is odd. Therefore, we approximate the function $G_3^d$ by multivariate trigonometric polynomials $G_3^{d,L} := \sum_{\mathbf{k}\in\mathcal{I}} (\hat{\mathbf{G}}_3^{d,L})_{\mathbf{k}}\, \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}$ with Fourier coefficients supported on modified hyperbolic cross index sets $\mathcal{I} = H^d_{R,\text{even}}$ as defined in (2.11). We compute the Fourier coefficients $\hat{\mathbf{G}}_3^{d,L}$ based on samples of $G_3^d$ and determine the relative $L^2(\mathbb{T}^d)$ sampling errors $\|G_3^d - G_3^{d,L}\|_{L^2(\mathbb{T}^d)} / \|G_3^d\|_{L^2(\mathbb{T}^d)}$, where

$$\|G_3^d - G_3^{d,L}\|_{L^2(\mathbb{T}^d)} = \sqrt{\|G_3^d\|^2_{L^2(\mathbb{T}^d)} - \sum_{\mathbf{k}\in\mathcal{I}} \left|\left(\hat{\mathbf{G}}_3^{d,L}\right)_{\mathbf{k}}\right|^2 + \sum_{\mathbf{k}\in\mathcal{I}} \left|\left(\hat{G}_3^d\right)_{\mathbf{k}} - \left(\hat{\mathbf{G}}_3^{d,L}\right)_{\mathbf{k}}\right|^2}.$$

We compare the numerical results from [44, Figure 4.3b], where reconstructing single rank-1 lattices and reconstructing random multiple rank-1 lattices were used, with new results using deterministic multiple rank-1 lattices returned by Algorithm 2.2.

As input for Algorithm 2.2, we use reconstructing single rank-1 lattices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, with generating vectors chosen according to (2.12). Instead of computing the Fourier coefficients $\hat{G}_3^{d,L}$ of the multivariate trigonometric polynomial $G_3^{d,L}$ by Algorithm 2.1, we use [43, Algorithm 2], which averages over all single rank-1 lattices $\Lambda(\mathbf{z}, P_\ell)$ that are able to reconstruct a Fourier coefficient $\hat{g}_\mathbf{k}$ of any multivariate trigonometric polynomial $g$ for a given frequency $\mathbf{k} \in \mathcal{I}$, whereas Algorithm 2.1 uses only one single rank-1 lattice $\Lambda(\mathbf{z}, P_{\nu(\mathbf{k})})$. Note that both computation methods are based on the same samples of $G_3^d$ along the obtained deterministic multiple rank-1 lattices. The resulting relative $L^2(\mathbb{T}^d)$ sampling errors are visualized for spatial dimensions $d \in \{2, 3, 5, 8\}$ in Figure 2.5 as solid lines and filled markers. We observe that the errors decrease rapidly for increasing expansion parameters $R$ of the hyperbolic cross $\mathcal{I} = H_{R,\text{even}}^d$ and correspondingly increasing number of samples. In addition, we consider reconstructing single rank-1 lattices generated by [38, Algorithm 3.7] as input for Algorithm 2.2 and obtain results which are very close and therefore omit their plots.

Moreover, the relative errors from [44, Figure 4.3b] when using reconstructing random multiple rank-1 lattices are shown in Figure 2.5 as dotted lines and filled markers. We observe that the obtained number of samples and errors are similar to the deterministic ones. The results for the deterministic multiple rank-1 lattice seem to be slightly better for $d \in \{3, 5, 8\}$. In addition, the relative errors from [44, Figure 4.3b] when directly sampling along reconstructing single rank-1 lattices are drawn as dashed lines and unfilled markers. It has already been observed in [44] that in the beginning for smaller expansion parameters $R$ and consequently smaller number of samples, the single rank-1 lattices perform better until a certain expansion parameter $R$ has been reached. Afterwards, the multiple rank-1 lattices clearly outperform the single ones.

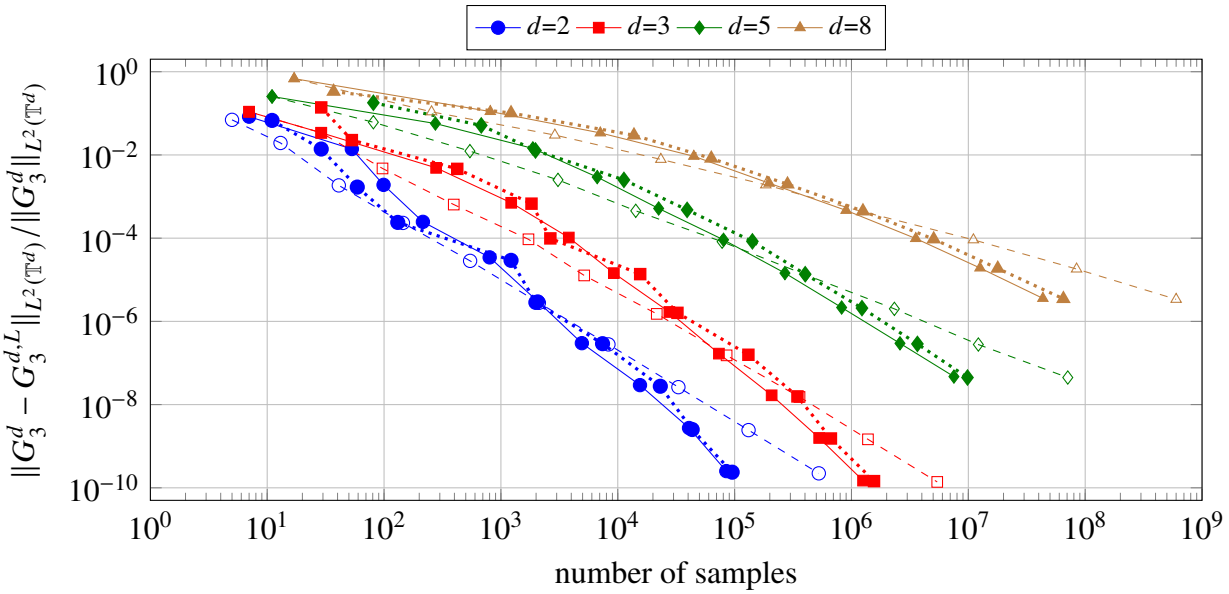
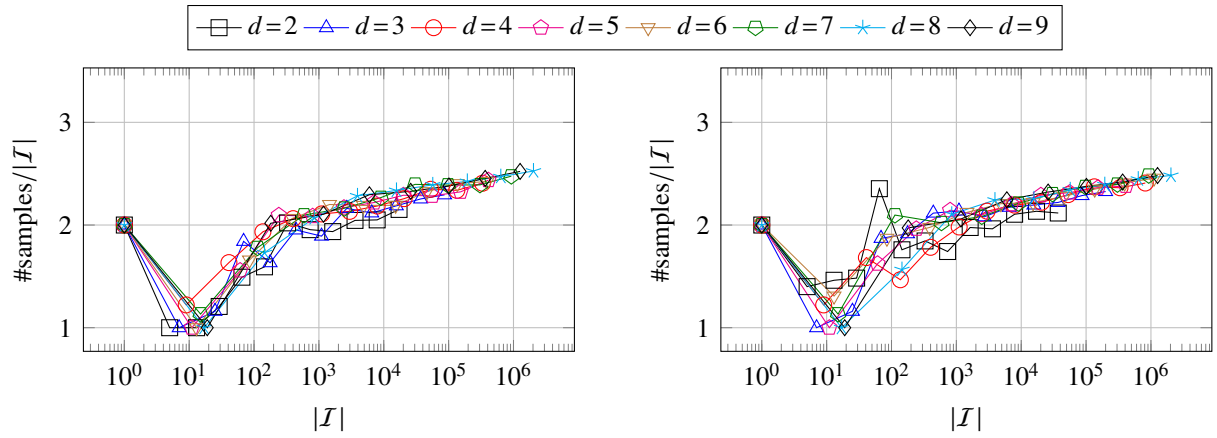



Figure 2.5 Relative $L^2(\mathbb{T}^d)$ sampling errors for $G_3^d$ with respect to the number of samples for reconstructing single rank-1 lattices (dashed lines, unfilled markers), reconstructing random multiple rank-1 lattices (dotted lines, filled markers), and reconstructing deterministic multiple rank-1 lattices (solid lines, filled markers), when using the frequency index sets $I := H^d_{R,\text{even}}$. Results for single rank-1 lattices from [65, Figure 2.14] and for reconstructing random multiple rank-1 lattices from [44, Figure 4.3].

### 2.3.3 Deterministic multiple rank-1 lattices with decreasing lattice size for reconstruction of trigonometric polynomials

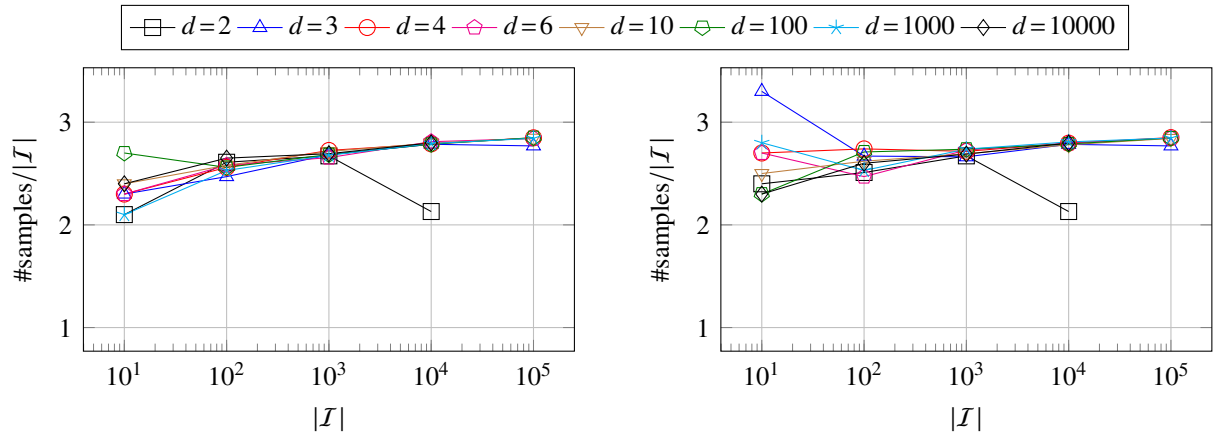
Besides generating deterministic multiple rank-1 lattices according to Theorem 2.2 and Algorithm 2.2, we have also discussed the alternate approach of Theorem 2.3, where the theoretical results for function approximation, as mentioned in Corollary 2.1, cannot be applied directly, but the number of required samples for the reconstruction of multivariate trigonometric polynomials may be distinctly smaller.

We start with symmetric hyperbolic cross type index sets $I = H^d_{R,\text{even}}$ and apply the generation strategy of Theorem 2.3 on reconstructing single rank-1 lattices for $I$, $\Lambda(\mathbf{z}, M)$, generated by [38, Algorithm 3.7]. We visualize the resulting oversampling factors $\#\text{samples} / |I| = (1 - L + \sum_{\ell \in [L]} P_\ell)/|I|$ in Figure 2.6a for spatial dimensions $d \in \{2, 3, \ldots, 9\}$ and various expansion parameters $R$. For the considered test cases, we observe that the oversampling factors are well below 3. When starting with single rank-1 lattices according to (2.12), the observed oversampling factors only differ slightly, cf. Figure 2.6b.




(a) $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7] for symmetric hyperbolic cross index sets $I = H^d_{R,\text{even}}$

(b) $\mathbf{z}$ and $M$ according to (2.12) for symmetric hyperbolic cross index sets $I = H^d_{R,\text{even}}$




(c) $\Lambda(\mathbf{z}, M)$ generated by [38, Algorithm 3.7] for random frequency sets $I \subset \{-64, -63, \ldots, 64\}^d$

(d) $\mathbf{z}$ and $M$ according to (2.12) for random frequency sets $I \subset \{-64, -63, \ldots, 64\}^d$

Figure 2.6 Oversampling factors for deterministic reconstructing multiple rank-1 lattices constructed according to Theorem 2.3.

The reason for these very low oversampling factors is that during the generation process according to the proof of Theorem 2.3 the prime $P_0$ is relatively close to $|I|$, the next prime $P_1$ is relatively close to $|I \setminus I_0|$, $P_2$ is relatively close to $|I \setminus (I_0 \cup I_1)|$, and so on, where $I_0$ contains the frequencies of $I$ which can be reconstructed by the lattice $\Lambda(\mathbf{z}, P_0)$ and where $I_1$ contains the frequencies of $I \setminus I_0$ which can be reconstructed by $\Lambda(\mathbf{z}, P_1)$. In particular, we do not have the fixed lower bound $|I| \leq P_\ell$ for all $\ell$ as in Algorithm 2.2.

Next, we change the setting and use the frequency sets $I$ drawn uniformly randomly from cubes $[-R, R]^d \cap \mathbb{Z}^d$, see Section 2.3.1. As before, we generate reconstructing single rank-1 lat-

tices for $\mathcal{I}$, $\Lambda(\mathbf{z}, M)$, using [38, Algorithm 3.7]. Then, we apply the strategy of Theorem 2.3 in order to deterministically generate reconstructing multiple rank-1 lattices. We repeat the test 10 times for each setting with newly randomly chosen frequency sets $\mathcal{I}$ and determine the maximum number of samples over the 10 repetitions. For cardinalities $|\mathcal{I}| \in \{10, 100, 1\,000, 10\,000\}$ in $d \in \{2, 3, 4, 6, 10, 100, 1\,000, 10\,000\}$ spatial dimensions, we visualize the resulting oversampling factors in Figure 2.6c for expansion parameter $R = 64$ ($K_{\mathcal{I}} \leq 128$). Starting with reconstructing single rank-1 lattices $\Lambda(\mathbf{z}, M)$ according to (2.12) as in Figure 2.3b changes the oversampling factors only slightly, and the oversampling factors are still well below 4, cf. Figure 2.6d.

# CHAPTER 3

# HIGH-DIMENSIONAL SPARSE FOURIER TRANSFORMS

As discussed in Section 1.1.2, this chapter focuses on efficient sparse Fourier transforms (SFTs) for high-dimensional functions. We begin with a review of the prior work against which we compare our techniques as well as provide a more in-depth discussion of the methods in Section 3.1. Section 3.2 reviews and further refines the univariate SFTs from [37, 53] which we will use in our multivariate techniques. Section 3.3 presents our main multivariate approximation algorithms and their analysis. Finally, we implement these two algorithms numerically and present the empirical results in Section 3.4.

## 3.1 Overview of results and prior work

Much recent work has considered the problem of quickly recovering both exactly sparse multivariate trigonometric polynomials as well as approximating more general functions by sparse trigonometric polynomials using dimension-incremental approaches [65, 59, 17, 16]. These methods recover multivariate frequencies adaptively by searching lower-dimensional projections of $\mathcal{I} \subset \left(\left(-\left\lceil \frac{K}{2} \right\rceil, \left\lfloor \frac{K}{2} \right\rfloor\right] \cap \mathbb{Z}\right)^d$ for energetic frequencies. These lower dimensional candidate sets are then paired together to build up a fully $d$-dimensional search space smaller than the original one, which is expected to support the most energetic frequencies (see e.g., [42, Section 3] and the references within for a general overview).

In the context of Fourier methods, lattice-based techniques do a good job of support identification on the intermediary, lower-dimensional candidate sets, and especially recently, techniques based on multiple rank-1 lattices have shown success [43, 42] (see also Chapter 2). Though the total complexity in each of these steps is manageable and can be kept linear in the sparsity $s$ of the Fourier series to be computed, these steps must be repeated in general to ensure that no potential frequencies have been left out. In particular, this results in at least $O(ds^2K)$ operations (up to logarithmic factors) for functions supported on arbitrary frequency sets in order to obtain approximations that are guaranteed to be accurate with high probability. Though from an implementational perspective, this runtime can be mitigated by completing many of the repetitions and initial one-dimensional

searches in parallel, once pairing begins, the results of previous iterations must be synchronized and communicated to future steps, necessitating serial interruptions.

Other earlier works include [37] in which previously existing univariate SFT results [36, 62] are refined and adapted to the multivariate setting. Though the resulting complexity on the dimension is well above the dimension-incremental approaches, deterministic guarantees are given for multivariate Fourier approximation in $O(d^4s^2)$ (up to logarithmic factors) time and memory, as well as a random variant which drops to linear scaling in $s$, leading to a runtime on the order of $O(d^4s)$ with respect to $s$ and $d$. Additionally, the compressed sensing type guarantees in terms of Fourier compressibility of the function under consideration carry over from the univariate SFT analysis. The scheme essentially makes use of a reconstructing rank-1 lattice on a superset of the full integer cube $\mathcal{I} = \left(\left(-\lceil \frac{dK}{2} \rceil, \lfloor \frac{dK}{2} \rfloor\right] \cap \mathbb{Z}\right)^d$ with certain number theoretic properties that allow for fast inversion of the resulting one-dimensional coefficients by the Chinese Remainder Theorem. We note that this necessarily inflated frequency domain accounts for the suboptimal scaling in $d$ above.

In [54], another fully deterministic sampling strategy and reconstruction algorithm is given. Like [37] though, the method can only be applied to Fourier approximations over an ambient frequency space $\mathcal{I}$ which is a full $d$-dimensional cube. Moreover, the vector space structure exploited to construct the sampling sets necessitates that the side length $K$ of this cube is the power of a prime. However, the benefits to this construction are among the best considered so far: the method is entirely deterministic, has noise-robust recovery guarantees in terms of best $s$-term estimates, the sampling sets used are on the order of $O(d^3s^2K)$, and the reconstruction algorithm's runtime complexity is on the order of $O(d^3s^2K^2)$ both up to logarithmic factors. On the other hand, this algorithm still does not scale linearly in $s$.

Finally, we discuss [15, 14], a pair of papers detailing high-dimensional Fourier recovery algorithms which offer a simplified (and therefore faster) approach to lattice transforms and dimension-incremental methods. These algorithms make heavy use of a one-dimensional SFT [51, 18] based on a phase modulation approach to discover energetic frequencies in a fashion similar to our Algorithm 3.1 below. The main idea is to recover entries of multivariate frequencies by using equis-

paced evaluations of the function along a coordinate axis as well as samples of the function at the same points slightly shifted (the remaining dimensions are generally ignored). This shift in space produces a modulation in frequency from which frequency data can be recovered (cf. (3.6) and Algorithm 3.1 below). By supplementing this approach with simple reconstructing rank-1 lattice analysis for repetitions of the full integer cube, the runtime and number of samples are given on average as $O(ds)$ up to logarithmic factors.

However, due to the possibility of collisions of multivariate frequencies under the hashing algorithms employed, these results hold only for random signal models. In particular, theoretical results are only stated for functions with randomly generated Fourier coefficients on the unit circle with randomly chosen frequencies from a given frequency set. Additionally, the analysis of these techniques assumes that the algorithm applied to the randomly generated signal does not encounter certain low probability (with respect to the random signal model considered therein) energetic frequency configurations. Furthermore, the method is restricted in stability, allowing for spatial shifts in sampling bounded by at most the reciprocal of the side length of the multivariate frequency cube under consideration, and only exact recovery is considered (or recovery up to factors related to sample corruption by Gaussian noise in [14]). In addition, no results given are proven concerning the approximation of more general periodic functions, e.g., compressible functions.

### 3.1.1 Main contributions

We begin with a brief summary of the benefits provided by our approach in comparison to the methods discussed above. Below, we ignore logarithmic factors in our summary of the runtime/sampling complexities.

- All variants, deterministic and random, of both algorithms presented in this paper have runtime and sampling complexities **linear in** $d$ with best $s$-term estimates for **arbitrary signals**. This is in contrast to the complexities of dimension-incremental approaches [16, 17, 43, 42] and the number theoretic approaches [37, 54] while still achieving similarly strong best $s$-term guarantees.
- Both algorithms proposed herein have randomized variants with runtime and sampling com-

plexities **linear in** $s$ with best $s$-term estimates on **arbitrary signals** that hold **with high probability**. Thus, the randomized methods proposed in this paper achieve the efficient runtime complexities of [15, 14] while simultaneously exhibiting best $s$-term approximation guarantees for general periodic functions thereby improving on the non-deterministic dimension incremental approaches [16, 17, 43, 42].

- Both algorithms proposed herein have a deterministic variant with runtime and sampling complexities **quadratic in** $s$ with best $s$-term estimates on **arbitrary signals** that also hold **deterministically**. This is in contrast to all previously discussed methods without deterministic guarantees, [16, 17, 43, 42, 14, 15], as well as improving on prior deterministic results [37, 54] for functions whose energetic frequency support sets $\mathcal{I}$ are smaller than the full cube.

**Overview of the methods and related theory**

We will build on the fast and potentially deterministic one-dimensional SFT from [37] and its discrete variant from [53] by applying those techniques along rank-1 lattices. As previously discussed, the primary difficulty in doing so is matching energetic one-dimensional Fourier coefficients with their $d$-dimensional counterparts. We are especially interested in doing this in an efficient and provably accurate way. We propose and analyze two different methods for solving this problem herein.

The first frequency identification approach, Algorithm 3.1, involves modifications of the phase shifting technique from [51, 18, 15, 14]. We make use of the translation to modulation property of the Fourier transform (cf. (3.6) below) observed in these works to extract out frequency data. Combining this with SFTs on rank-1 lattices gives a new class of fast methods with several benefits. Notably, we are able to maintain error guarantees for any function (not just random signals) in terms of best Fourier $s$-term approximations. Additionally, we factor the instability and potential for collisions from [15, 14] into these best $s$-term approximations. The only downside in our estimates is an additional linear factor of $K$ multiplying the terms commonly seen in standard error bounds (cf. Corollaries 3.1 and 3.2). However, we are able to maintain deterministic results with runtime and sampling complexities that are quadratic in $s$, as well as results for random variants

with complexities that are linear in $s$. Additionally, the dependence on the dimension $d$ is reduced from $O(d^4)$ in [37] to only $O(d)$.

Our second technique in Algorithm 3.2 uses a different approach to applying SFTs to modifications of the multivariate function along a reconstructing rank-1 lattice. Effectively, we reduce $g$ to a two-dimensional function. This is done by mapping all but one dimension, say $\ell$, down to one using a rank-1 lattice, and leaving the $\ell$ dimension free. From here, we take a two-dimensional DFT (taking care to use SFTs where possible). The locations of Fourier coefficients in this two-dimensional DFT can then be used to determine the $\ell$th coordinate of the frequency data. This is then repeated for each dimension $\ell \in [d]$.

This process is slower but more stable than Algorithm 3.1. In particular this produces more accurate best Fourier $s$-term approximation guarantees without the extraneous factor of $K$ (cf. Corollaries 3.3 and 3.4). The deterministic results still have a complexity quadratic in $s$ with random extensions that are linear in $s$. However, we incur an extra quadratic factor of $K$ in the complexity bounds (cf. Lemma 3.4).

We stress here that by compartmentalizing the translation from multivariate analysis to univariate analysis into the theory of rank-1 lattices, our techniques are suitable for any frequency set of interest $\mathcal{I}$. The only constraint is the necessity for a reconstructing rank-1 lattice for $\mathcal{I}$ (and potentially projections of $\mathcal{I}$ in the case of Algorithm 3.2). This flexibility improves the results from [37], primarily with respect to the polynomial factor of $d$ in our runtime and sampling complexities. We remark that though the existence of the necessary reconstructing rank-1 lattice is a nontrivial requirement, there exist efficient construction algorithms for arbitrary frequency sets via deterministic component by component methods, see e.g., [39, 46, 56].

In terms of implementation, we note that the multivariate techniques we employ are entirely modular with respect to the univariate SFT used. As such, the complexity estimates and error bounds for our approaches in Section 3.3 are directly derived from the chosen SFT.

Finally, the methods we present are trivially parallelizable so that in particular, a large majority of these univariate SFTs in Algorithm 3.1 or Algorithm 3.2 can occur in parallel.

### 3.2 One-dimensional sparse Fourier transform results

Below, we summarize some of the previous work on one-dimensional sparse Fourier transforms which will be used in our multivariate algorithms. Rather than focus on the inner workings of these SFTs, we highlight five main properties concerning their recovery guarantees and computational complexity. This compartmentalization allows for any SFT satisfying these properties to be easily extended for multivariate Fourier recovery simply by plugging into Algorithm 3.1 and 3.2.

We first review the sublinear-time algorithm from [37] which uses fewer than $M$ nonequispaced samples of a function to compute Fourier coefficients in $\mathcal{B}_M$. We refer the reader interested in its implementation and mathematical explanation to [37] as well as [36, 62]. Below, we will use slightly improved error bounds over those in its original presentation. The proof of these improvements necessitates the following lemma.

**Lemma 3.1.** *For* $\mathbf{x} \in \mathbb{C}^K$ *and* $\mathcal{S}_\tau := \{k \in [K] \mid |x_k| \geq \tau\}$*, if* $\tau \geq \frac{\|\mathbf{x}-\mathbf{x}_s^{\text{opt}}\|_1}{s}$*, then* $|\mathcal{S}_\tau| \leq 2s$ *and*

$$\|\mathbf{x} - \mathbf{x}|_{\mathcal{S}_\tau}\|_2 \leq \|\mathbf{x} - \mathbf{x}_{2s}^{\text{opt}}\|_2 + \tau\sqrt{2s},$$
$$\|\mathbf{x} - \mathbf{x}|_{\mathcal{S}_\tau}\|_1 \leq \|\mathbf{x} - \mathbf{x}_{2s}^{\text{opt}}\|_1 + \tau \cdot 2s.$$

*Proof.* Ordering the entries of $\mathbf{x}$ in descending order (with ties broken arbitrarily) as $|x_{k_1}| \geq |x_{k_2}| \geq \ldots$, we first note that

$$\|\mathbf{x} - \mathbf{x}_s^{\text{opt}}\|_1 \geq \sum_{j=s+1}^{2s} |x_{k_j}| \geq s|x_{k_{2s}}|.$$

By assumption then, $\tau \geq |x_{k_{2s}}|$, and since $\mathcal{S}_\tau$ contains the $|\mathcal{S}_\tau|$-many largest entries of $\mathbf{x}$, we must have $\mathcal{S}_\tau \subset \text{supp}(\mathbf{x}_{2s}^{\text{opt}})$. Note then that $|\mathcal{S}_\tau| \leq 2s$. Finally, we calculate

$$\begin{aligned}
\|\mathbf{x} - \mathbf{x}|_{\mathcal{S}_\tau}\|_2 &\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\text{opt}}\|_2 + \|\mathbf{x}_{2s}^{\text{opt}} - \mathbf{x}_{\mathcal{S}_\tau}\|_2 \\
&\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\text{opt}}\|_2 + \sqrt{\sum_{k \in \text{supp}(\mathbf{x}_{2s}^{\text{opt}}) \setminus \mathcal{S}_\tau} |x_k|^2} \\
&\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\text{opt}}\|_2 + \tau\sqrt{2s}.
\end{aligned}$$

The $\ell^1$ estimate is proved by the same procedure, where the $2s$ many terms are bounded by $\tau$ in the last line without a square root. □

**Theorem 3.1** (Robust sublinear-time, nonequispaced SFT: [37], Theorem 7/[53], Lemma 4)**.** *For a signal $g^{1\mathrm{d}} \in W(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, Algorithm 3 of [37], denoted $\mathcal{A}^{\mathrm{sub}}_{2s,M}$, will output a $2s$-sparse coefficient vector $\hat{\mathbf{g}}^{1\mathrm{d},s} \in \mathbb{C}^{\mathcal{B}_M}$ which*

1. *reconstructs every frequency of $\hat{g}^{1\mathrm{d}}|_M \in \mathbb{C}^{\mathcal{B}_M}$, $\omega \in \mathcal{B}_M$, with corresponding Fourier coefficients meeting the tolerance*

$$|\hat{g}^{1\mathrm{d}}_\omega| > (4+2\sqrt{2})\left(\frac{\|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_s\|_1}{s} + \|\hat{g}^{1\mathrm{d}} - \hat{g}^{1\mathrm{d}}|_M\|_1 + \|\mu\|_\infty\right),$$

2. *satisfies the $\ell^\infty$ error estimate for recovered coefficients*

$$\left\|(\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s})|_{\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\right\|_\infty \leq \sqrt{2}\left(\frac{\left\|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_s\right\|_1}{s} + \left\|\hat{g}^{1\mathrm{d}} - \hat{g}^{1\mathrm{d}}|_M\right\|_1 + \|\mu\|_\infty\right),$$

3. *satisfies the $\ell^2$ error estimate*

$$\begin{aligned}\|\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_2 \leq \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_2 + \frac{(8\sqrt{2}+6)\|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_s\|_1}{\sqrt{s}} \\ + (8\sqrt{2}+6)\sqrt{s}\left(\|\hat{g}^{1\mathrm{d}} - \hat{g}^{1\mathrm{d}}|_M\|_1 + \|\mu\|_\infty\right),\end{aligned}$$

4. *satisfies the $\ell^1$ error estimate*

$$\begin{aligned}\|\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_1 \leq \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_1 + (6\sqrt{2}+16)\|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_s\|_1 \\ + (6\sqrt{2}+16)s\left(\|\hat{g}^{1\mathrm{d}} - \hat{g}^{1\mathrm{d}}|_M\|_1 + \|\mu\|_\infty\right),\end{aligned}$$

5. *and the number of required samples of $g^{1\mathrm{d}}$ and the operation count for $\mathcal{A}^{\mathrm{sub}}_{2s,M}$ are*

$$O\left(\frac{s^2 \log^4 M}{\log s}\right).$$

*The Monte Carlo variant of $\mathcal{A}^{\mathrm{sub}}_{2s,M}$, denoted $\mathcal{A}^{\mathrm{sub,MC}}_{2s,M}$, referred to by Corollary 4 of [37] satisfies all of the conditions (1) – (4) simultaneously with probability $(1-\sigma) \in [2/3, 1)$ and has number of required samples and operation count*

$$O\left(s \log^3(M) \log\left(\frac{M}{\sigma}\right)\right).$$

*The samples required by $\mathcal{A}^{\mathrm{sub,MC}}_{2s,M}$ are a subset of those required by $\mathcal{A}^{\mathrm{sub}}_{2s,M}$.*

*Proof.* We refer to [37, Theorem 7] and its modification for noise robustness in [53, Lemma 4] for the proofs of properties (2) and (5). As for (1), [37, Lemma 6] and its modification in [53, Lemma 4] imply that any $\omega \in \mathcal{B}_M$ with $|\hat{g}^{1\mathrm{d}}_\omega| > 4(\|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_s\|_1/s + \|\hat{g}^{1\mathrm{d}} - \hat{g}^{1\mathrm{d}}|_M\|_1 + \|\mu\|_\infty) =: 4\delta$ will be identified in [37, Algorithm 3]. An approximate Fourier coefficient for these and any other recovered frequencies is stored in the vector $\mathbf{x}$ which satisfies the same estimate in property (2) by the proof of [37, Theorem 7] and [53, Lemma 4]. However, only the $2s$ largest magnitude values of $\mathbf{x}$ will be returned in $\hat{\mathbf{g}}^{1\mathrm{d},s}$. We therefore analyze what happens when some of the potentially large Fourier coefficients corresponding to frequencies in $\mathcal{S}_{4\delta}$ do not have their approximations assigned to $\hat{\mathbf{g}}^{1\mathrm{d},s}$.

Using the definition of $\mathcal{S}_\tau$ given in Lemma 3.1 applied to $\hat{g}^{1\mathrm{d}}|_M$, we must have $|\mathcal{S}_{4\delta}| \leq 2s = |\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})|$. If $\omega \in \mathcal{S}_{4\delta} \setminus \operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})$, there must then exist some other $\omega' \in \operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s}) \setminus \mathcal{S}_{4\delta}$ which was identified and took the place of $\omega$ in $\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})$. For this to happen, $|\hat{g}^{1\mathrm{d}}_{\omega'}| \leq 4\delta$ and $|x_{\omega'}| \geq |x_\omega|$. But by property (2) (extended to all coefficients in $\mathbf{x}$), we know

$$4\delta + \sqrt{2}\delta \geq |\hat{g}^{1\mathrm{d}}_{\omega'}| + \sqrt{2}\delta \geq |x_{\omega'}| \geq |x_\omega| \geq |\hat{g}^{1\mathrm{d}}_\omega| - \sqrt{2}\delta.$$

Thus, any frequency in $\mathcal{S}_{4\delta}$ not chosen satisfies $|\hat{g}^{1\mathrm{d}}_\omega| \leq (4 + 2\sqrt{2})\delta$, and so every frequency in $\mathcal{S}_{(4+2\sqrt{2})\delta}$ is in fact identified in $\hat{\mathbf{g}}^{1\mathrm{d},s}$ verifying property (1).

As for property (3), we estimate the $\ell^2$ error using property (2), Lemma 3.1, and the above argument as

$$\begin{aligned}
\|\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_2 &\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_2 + \|(\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s})|_{\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_2 \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}\cap\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_2 + \sqrt{2}\delta\sqrt{2s} \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}}\|_2 + \|\hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}\setminus\operatorname{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_2 + 2\delta\sqrt{s} \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_2 + 4\delta\sqrt{2s} + (4 + 2\sqrt{2})\delta\sqrt{2s} + 2\delta\sqrt{s} \\
&= \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_2 + (8\sqrt{2} + 6)\sqrt{s}\delta.
\end{aligned}$$

The proof of property (4) is very similar. We estimate the $\ell^1$ error using the same techniques as

$$\begin{aligned}
\|\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_1 &\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_1 + \|(\hat{g}^{1\mathrm{d}}|_M - \hat{\mathbf{g}}^{1\mathrm{d},s})|_{\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_1 \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}\cap\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_1 + \sqrt{2}\delta\cdot 2s \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - \hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}}\|_1 + \|\hat{g}^{1\mathrm{d}}|_{\mathcal{S}_{4\delta}\setminus\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_1 + 2\sqrt{2}\delta s \\
&\leq \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_1 + 4\delta\cdot 2s + (4+2\sqrt{2})\delta\cdot 2s + 2\sqrt{2}\delta s \\
&= \|\hat{g}^{1\mathrm{d}}|_M - (\hat{g}^{1\mathrm{d}}|_M)^{\mathrm{opt}}_{2s}\|_1 + (6\sqrt{2}+16)s\delta. \qquad \square
\end{aligned}$$

*Remark* 3.1. In the noiseless case, if the univariate function $g^{1\mathrm{d}}$ is Fourier $s$-sparse, i.e., is a trigonometric polynomial and $M$ is large enough such that $\mathrm{supp}(\hat{g}^{1\mathrm{d}}) \subset \mathcal{B}_M$, both $\mathcal{A}^{\mathrm{sub}}_{2s,M}$ and $\mathcal{A}^{\mathrm{sub,MC}}_{2s,M}$ will exactly recover $\hat{g}^{1\mathrm{d}}|_M$ (the latter with probability $1-\sigma$), and therefore $\hat{g}^{1\mathrm{d}}$. In particular, note that the output of either algorithm will then actually be $s$-sparse.

Using the above SFT algorithm with the discretization process outlined in [53] leads to a fully *discrete* sparse Fourier transform, requiring only equispaced samples of $g^{1\mathrm{d}}$, denoted $\mathbf{g}^{1\mathrm{d}}$. However, rather than separately accounting for the truncation to the frequency band $\mathcal{B}_M$ as above, the equispaced samples allow us to take advantage of aliasing, which is particularly important when we apply the algorithm along reconstructing rank-1 lattices. Thus, instead of approximating $\hat{g}^{1\mathrm{d}}|_M \in \mathbb{C}^{\mathcal{B}_M}$, we prefer to approximate the discrete Fourier transform of $\mathbf{g}^{1\mathrm{d}}$ given by $\mathbf{F}_M\mathbf{g}^{1\mathrm{d}}$.

Eventually, we will consider techniques for approximation of arbitrary periodic functions rather than simply polynomials. For this reason, we require noise-robust recovery results for the method in [53]. The necessary modifications to account for this robustness as well as the improved guarantees carried over from the previous algorithm are given below. The upshot is that we are able to state five properties of this SFT analogous to those in Theorem 3.1 which allow for modular proofs of the multivariate results later on.

**Theorem 3.2** (Robust discrete sublinear-time SFT: see [53], Theorem 5)**.** *For a signal $g^{1\mathrm{d}} \in W(\mathbb{T})\cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, and $1 \leq r \leq \frac{M}{36}$, Algorithm 1 of [53], denoted $\mathcal{A}^{\mathrm{disc}}_{2s,M}$, will output a $2s$-sparse coefficient vector $\hat{\mathbf{g}}^{1\mathrm{d},s} \in \mathbb{C}^{\mathcal{B}_M}$ which*

1. *reconstructs every frequency of* $\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} \in \mathbb{C}^{\mathcal{B}_M}$, $\omega \in \mathcal{B}_M$, *with corresponding aliased Fourier coefficient meeting the tolerance*

$$|(\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_\omega| > 12(1+\sqrt{2})\left(\frac{\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|\mathbf{g}^{1\mathrm{d}}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

2. *satisfies the* $\ell^\infty$ *error estimate for recovered coefficients*

$$\|(\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s})|_{\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})}\|_\infty \leq 3\sqrt{2}\left(\frac{\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|\mathbf{g}^{1\mathrm{d}}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

3. *satisfies the* $\ell^2$ *error estimate*

$$\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_2 \leq \|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_{2s}^{\mathrm{opt}}\|_2 + 38\frac{\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}}$$
$$+ 152\sqrt{s}(\|\mathbf{g}^{1\mathrm{d}}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty),$$

4. *satisfies the* $\ell^1$ *error estimate*

$$\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s}\|_1 \leq \|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_{2s}^{\mathrm{opt}}\|_1 + 54\|\mathbf{F}_M \mathbf{g}^{1\mathrm{d}} - (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_s^{\mathrm{opt}}\|_1$$
$$+ 215s(\|\mathbf{g}^{1\mathrm{d}}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty),$$

5. *and the number of required samples of* $\mathbf{g}^{1\mathrm{d}}$ *and the operation count for* $\mathcal{A}_{2s,M}^{\mathrm{disc}}$ *are*

$$O\left(\frac{s^2 r^{3/2} \log^{11/2} M}{\log s}\right).$$

*The Monte Carlo variant of* $\mathcal{A}_{2s,M}^{\mathrm{disc}}$, *denoted* $\mathcal{A}_{2s,M}^{\mathrm{disc,MC}}$, *satisfies all of the conditions (1) – (4) simultaneously with probability* $(1-\sigma) \in [2/3, 1)$ *and has number of required samples and operation count*

$$O\left(sr^{3/2} \log^{9/2}(M) \log\left(\frac{M}{\sigma}\right)\right).$$

*Proof.* All notation in this proof matches that in [53] (in particular, we use $f$ to denote the one-dimensional function in place of $g^{1\mathrm{d}}$ in the theorem statement and $N = 2M+1$). We begin by substituting the $2\pi$-periodic Gaussian filter given in (3) on page 756 with the 1-periodic Gaussian and associated Fourier transform

$$g(x) = \frac{1}{c_1}\sum_{n=-\infty}^{\infty} \mathrm{e}^{-\frac{(2\pi)^2(x-n)^2}{2c_1^2}}, \quad \hat{g}_\omega = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{c_1^2\omega^2}{2}}.$$

Note then that all results regarding the Fourier transform remain unchanged, and since this 1-periodic Gaussian is a just a rescaling of the $2\pi$-periodic one used in [53], the bound in [53, Lemma 1] holds with a similarly compressed Gaussian, that is, for all $x \in \left[-\frac{1}{2}, \frac{1}{2}\right]$

$$g(x) \leq \left(\frac{3}{c_1} + \frac{1}{\sqrt{2\pi}}\right) \mathrm{e}^{-\frac{(2\pi x)^2}{2c_1^2}}. \tag{3.1}$$

Analogous results up to and including [53, Lemma 10] for 1-periodic functions then hold straightforwardly.

Assuming that our signal measurements $\mathbf{f} = (f(y_j))_{j=0}^{2M} = (f(\frac{j}{N}))_{j=0}^{2M}$ are corrupted by some discrete noise $\boldsymbol{\mu} = (\mu_j)_{j=0}^{2M}$, we consider for any $x \in \mathbb{T}$ a similar bound to [53, Lemma 10]. Here, $j' := \arg\min_j |x - y_j|$ and $\kappa := \lceil \gamma \ln N \rceil + 1$ for some $\gamma \in \mathbb{R}^+$ to be determined. Then,

$$\begin{aligned}
&\left|\frac{1}{N}\sum_{j=0}^{2M} f(y_j)g(x-y_j) - \frac{1}{N}\sum_{j=j'-\kappa}^{j'+\kappa} (f(y_j)+\mu_j)g(x-y_j)\right| \\
&\quad\leq \frac{1}{N}\left|\sum_{j=0}^{2M} f(y_j)g(x-y_j) - \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j)g(x-y_j)\right| + \frac{1}{N}\left|\sum_{j=j'-\kappa}^{j'+\kappa} \mu_j g(x-y_j)\right| \\
&\quad\leq \frac{1}{N}\left|\sum_{j=0}^{2M} f(y_j)g(x-y_j) - \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j)g(x-y_j)\right| + \frac{\|\boldsymbol{\mu}\|_\infty}{N}\sum_{k=-\kappa}^{\kappa} g(x-y_{j'+k})
\end{aligned}$$

We bound the first term in this sum by a direct application of [53, Lemma 10]; however, we take this opportunity to reduce the constant in the bound given there. In particular, bounding this term by the final expression in the proof of [53, Lemma 10] and using our implicit assumption that $36 \leq N$, we have

$$\begin{aligned}
&\left|\frac{1}{N}\sum_{j=0}^{2M} f(y_j)g(x-y_j) - \frac{1}{N}\sum_{j=j'-\kappa}^{j'+\kappa} (f(y_j)+\mu_j)g(x-y_j)\right| \\
&\qquad\leq \left(\frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi}\sqrt{\frac{\ln 36}{36}}\right)\|\mathbf{f}\|_\infty N^{-r} + \frac{1}{N}\|\boldsymbol{\mu}\|_\infty \sum_{k=-\kappa}^{\kappa} g(x-y_{j'+k}).
\end{aligned} \tag{3.2}$$

We now work on bounding the second term. First note that for all $k \in [-\kappa, \kappa] \cap \mathbb{Z}$,

$$g(x - y_{j'\pm k}) = g\left(x - y_{j'} \pm \frac{k}{N}\right).$$

Assuming without loss of generality that $0 \le x - y_{j'}$, we can bound the nonnegatively indexed summands by (3.1) as

$$g\left(x - y_{j'} + \frac{k}{N}\right) \le \left(\frac{3}{c_1} + \frac{2}{\sqrt{2\pi}}\right) \mathrm{e}^{-\frac{(2\pi)^2 k^2}{2c_1^2 N^2}}. \tag{3.3}$$

For the negatively indexed summands, the definition of $j' = \arg\min_j |x - y_j|$ implies that $x - y_{j'} \le \frac{1}{2N}$. In particular,

$$x - y_{j'} - \frac{k}{N} \le \frac{1 - 2k}{2N} < 0$$

implies

$$\left(x - y_{j'} - \frac{k}{N}\right)^2 \ge \frac{1 - 2k}{2N}\left(x - y_{j'} - \frac{k}{N}\right) \ge \frac{2k - 1}{2N} \cdot \frac{k}{N},$$

giving

$$g\left(x - y_{j'} - \frac{k}{N}\right) \le \left(\frac{3}{c_1} + \frac{2}{\sqrt{2\pi}}\right) \mathrm{e}^{-\frac{(2\pi)^2 k^2}{2c_1^2 N^2}} \mathrm{e}^{\frac{(2\pi)^2 k}{4c_1^2 N^2}}. \tag{3.4}$$

We now bound the final exponential. We first recall from [53] the choices of parameters

$$c_1 = \frac{\beta\sqrt{\ln N}}{N}, \quad \kappa = \lceil \gamma \ln N \rceil + 1, \quad \gamma = \frac{6r}{\sqrt{2}\pi} = \frac{\beta\sqrt{r}}{2\sqrt{\pi}}, \quad \beta = 6\sqrt{r}$$

with $1 \le r \le \frac{N}{36}$. For $k \in [1, \kappa] \cap \mathbb{Z}$ then,

$$\begin{aligned}
\exp\left(\frac{(2\pi)^2 k}{4c_1^2 N^2}\right) &\le \exp\left(\frac{(2\pi)^2 \kappa}{4c_1^2 N^2}\right) \\
&\le \exp\left(\frac{\pi^2\left(\frac{6r\ln N}{\sqrt{2}\pi} + 2\right)}{36r\ln N}\right) \\
&\le \exp\left(\frac{\pi}{6\sqrt{2}} + \frac{\pi^2}{18r\ln N}\right) \\
&\le \exp\left(\frac{\pi}{6\sqrt{2}} + \frac{\pi^2}{18\ln 36}\right) =: A.
\end{aligned}$$

Combining this with our bounds for the nonnegatively indexed summands (3.3) and the negatively indexed summands (3.4), we have

$$\frac{1}{N}\sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k}) \le \left(\frac{3}{\beta\sqrt{\ln N}} + \frac{1}{N\sqrt{2\pi}}\right)\left(1 + (1 + A)\sum_{k=1}^{\kappa} \mathrm{e}^{-\frac{(2\pi)^2 k^2}{2\beta^2 \ln N}}\right)$$

Expressing the final sum as a truncated lower Riemann sum and applying a change of variables on the resulting integral, we have

$$\sum_{k=1}^{\kappa} \mathrm{e}^{-\frac{(2\pi)^2 k^2}{2\beta^2 \ln N}} \leq \frac{\beta\sqrt{\ln N}}{\sqrt{2}\pi} \int_0^{\infty} e^{-x^2}\, dx = \frac{\beta\sqrt{\ln N}}{2\sqrt{2\pi}}.$$

Making use of our parameter values from [53], and the fact that $1 \leq r \leq \frac{N}{36}$,

$$\begin{aligned} \frac{1}{N} \sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k}) &\leq \left(\frac{3}{\beta\sqrt{\ln N}} + \frac{1}{N\sqrt{2\pi}}\right)\left(1 + \frac{1+A}{2\sqrt{2\pi}}\beta\sqrt{\ln N}\right) \\ &\leq \frac{3}{6\sqrt{\ln 36}} + \frac{3(1+A)}{2\sqrt{2\pi}} + \frac{1}{36\sqrt{2\pi}} + \frac{1+A}{4\pi}\sqrt{\frac{\ln 36}{36}} \qquad (3.5) \\ &< 2. \end{aligned}$$

With our revised bound for (3.2) above, we reprove [53, Theorem 4] to estimate $g * f$ by the truncated discrete convolution with noisy samples. In particular, we apply [53, Theorem 3], (3.2), (3.1), and finally our same assumption that $1 \leq r \leq \frac{N}{36}$ to obtain

$$\begin{aligned} &\left| (g * f)(x) - \frac{1}{N} \sum_{j=j'-\left\lceil \frac{6r}{\sqrt{2}\pi} \ln N \right\rceil - 1}^{j'+\left\lceil \frac{6r}{\sqrt{2}\pi} \ln N \right\rceil + 1} (f(y_j) + \mu_j) g(x - y_j) \right| \\ &\quad \leq \frac{N^{1-r}}{6\sqrt{r}\sqrt{\ln N}} \|\mathbf{f}\|_\infty N^{-r} + \left(\frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi}\sqrt{\frac{\ln 36}{36}}\right) \|\mathbf{f}\|_\infty N^{-r} + 2\|\boldsymbol{\mu}\|_\infty \\ &\quad \leq \left(\frac{1}{6\sqrt{\ln 36}} + \frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi}\sqrt{\frac{\ln 36}{36}}\right) \frac{\|\mathbf{f}\|_\infty}{N^r} + 2\|\boldsymbol{\mu}\|_\infty < 2\left(\frac{\|\mathbf{f}\|_\infty}{N^r} + \|\boldsymbol{\mu}\|_\infty\right). \end{aligned}$$

Replacing all references of $3\|\mathbf{f}\|_\infty N^{-r}$ by $2(\|\mathbf{f}\|_\infty N^{-r} + \|\boldsymbol{\mu}\|_\infty)$ in the remainder of the steps up to proving [53, Theorem 5] gives the desired noise robustness (with a slightly improved constant).

Using the revised error estimates of the nonequispaced algorithm from Theorem 3.1 and redefining $\delta = 3(\|\hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\mathrm{opt}}\|_1/2s + 2(\|\mathbf{f}\|_\infty N^{-r} + \|\boldsymbol{\mu}\|_\infty))$ as in the proof of [53, Theorem 5] (which also contains the proof of property (2)), the discretization algorithm [53, Algorithm 1] will produce candidate Fourier coefficient approximations in lines 9 and 12 corresponding to every $|\hat{f}_\omega| \geq (4+2\sqrt{2})\delta$ in place of $4\delta$ in Theorem 3.1. The exact same argument as in the proof of Theorem 3.1 then applies to the selection of the $2s$-largest entries of this approximation with the revised threshold values and error bounds to give properties (1), (3) and (4).

In detail, [53, Lemma 13] and the discussion right after its statement gives that property (2) holds for any approximate coefficient with frequency recovered throughout the algorithm (which, for the purposes of the following discussion, we will store in $\mathbf{x}$ rather than $\hat{R}$ defined in [53, Algorithm 1]), not just those in the final output $\mathbf{v} := \mathbf{x}_s^{\text{opt}}$. Additionally, by the same lemma and our revised bounds from Theorem 3.1, any frequency $\omega \in [N]$ satisfying $|f_\omega| > (4 + 2\sqrt{2})\delta$ will have an associated coefficient estimate in $\mathbf{x}$.

By Lemma 3.1, $|\mathcal{S}_{(4+2\sqrt{2})\delta}| \leq 2s = |\operatorname{supp}(\mathbf{v})|$, and so if $\omega \in \mathcal{S}_{(4+2\sqrt{2})\delta} \setminus \operatorname{supp}(\mathbf{v})$, there exists some $\omega' \in \operatorname{supp}(\mathbf{v}) \setminus \mathcal{S}_{(4+2\sqrt{2})\delta}$ such that $v_{\omega'}$ took the place of $v_\omega$ in $\mathcal{S}$. In particular, this means that $|x_{\omega'}| \geq |x_\omega|$, $|\hat{f}_{\omega'}| \leq (4 + 2\sqrt{2})\delta$, and $|\hat{f}_\omega| > (4 + 2\sqrt{2}\delta)$. Thus,

$$(4 + 2\sqrt{2})\delta + \sqrt{2}\delta > |\hat{f}_{\omega'}| + \sqrt{2}\delta \geq |x_{\omega'}| \geq |x_\omega| \geq |\hat{f}_\omega| - \sqrt{2}\delta,$$

implying that $|\hat{f}_\omega| \leq 4(1 + \sqrt{2})\delta$ and therefore proving (1).

To prove (3), we use Lemma 3.1, and consider

$$\begin{aligned}
\|\hat{\mathbf{f}} - \mathbf{v}\|_2 &\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\operatorname{supp}(\mathbf{v})}\|_2 - \|(\hat{\mathbf{f}} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_2 \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta} \cap \operatorname{supp}(\mathbf{v})}\|_2 + \sqrt{2}\delta\sqrt{2s} \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta}}\|_2 + \|\hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta} \setminus \operatorname{supp}(\mathbf{v})}\|_2 + 2\delta\sqrt{s} \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\text{opt}}\|_2 + (4 + 2\sqrt{2})\delta\sqrt{2s} + 4(1 + \sqrt{2})\delta\sqrt{2s} + 2\delta\sqrt{s} \\
&= \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\text{opt}}\|_2 + (14 + 8\sqrt{2})\delta\sqrt{s}.
\end{aligned}$$

The proof of (4) is similar, bounding the $\ell^1$ error as

$$\begin{aligned}
\|\hat{\mathbf{f}} - \mathbf{v}\|_1 &\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\operatorname{supp}(\mathbf{v})}\|_1 - \|(\hat{\mathbf{f}} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_1 \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta} \cap \operatorname{supp}(\mathbf{v})}\|_1 + \sqrt{2}\delta \cdot 2s \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta}}\|_1 + \|\hat{\mathbf{f}}|_{\mathcal{S}_{(4+2\sqrt{2})\delta} \setminus \operatorname{supp}(\mathbf{v})}\|_1 + 2\sqrt{2}\delta s \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\text{opt}}\|_1 + (4 + 2\sqrt{2})\delta \cdot 2s + 4(1 + \sqrt{2})\delta \cdot 2s + 2\sqrt{2}\delta s \\
&= \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\text{opt}}\|_1 + (16 + 14\sqrt{2})\delta s.
\end{aligned}$$

□

## 3.3 Fast multivariate sparse Fourier transforms

Having detailed two sublinear-time, one-dimensional SFT algorithms, we are now prepared to extend these to the multivariate setting. The general approach will be to apply the one-dimensional methods to transformations of our multivariate function of interest with samples taken along rank-1 lattices. The particular approaches for transforming our multivariate function will then allow for the efficient extraction of multidimensional frequency information for the most energetic coefficients identified by univarate SFTs. In particular, our first approach considered in Section 3.3.1 successively shifts the function in each dimension, whereas our second approach considered in Section 3.3.2 successively collapses all but one dimension along a rank-1 lattice and samples the resulting two-dimensional function.

Since the two approaches in Algorithms 1 and 2 below can make use of any univariate SFT algorithm, their analysis will be presented in a modular fashion below. Each algorithm is followed by a lemma (Lemma 3.2 and Lemma 3.4 respectively) which provides associated error guarantees when any sufficiently accurate univariate SFT $\mathcal{A}_{s,M}$ is employed. The lemmas are then each followed by two corollaries (Corollaries 3.1 and 3.2 and Corollaries 3.3 and 3.4 respectively) where we apply the lemma to the two example univariate SFTs reviewed in Section 3.2 specified by Theorems 3.2 and 3.1.

### 3.3.1 Phase encoding

We begin by noting that this section makes significant use of the property of the Fourier transform that translation of a function modulates its Fourier coefficients. We denote the shift operator $S_{\ell,\alpha}$ in the $\ell$th coordinate with shift $\alpha \in \mathbb{R}$ defined by its action on the multivariate periodic function $g : \mathbb{T}^d \to \mathbb{C}$ as

$$S_{\ell,\alpha}(g)(x_1, \ldots, x_d) := g(x_1, \ldots, x_{\ell-1}, (x_\ell + \alpha) \bmod 1, x_{\ell+1}, \ldots, x_d).$$

By a change of coordinates in the integral defining a Fourier coefficient, we see that translating will modulate the Fourier coefficients of $g : \mathbb{T}^d \to \mathbb{C}$ as

$$\widehat{(S_{\ell,\alpha} g)}_{\mathbf{k}} = \mathrm{e}^{2\pi \mathrm{i} k_\ell \alpha} \hat{g}_{\mathbf{k}}. \tag{3.6}$$

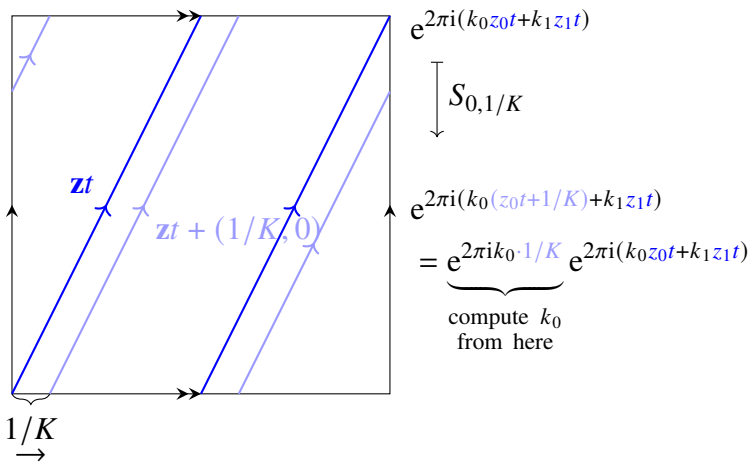



Figure 3.1 The basic procedure for the phase encoding algorithm applied to the trigonometric monomial $g(\mathbf{x}) = e^{2\pi i \mathbf{k}\cdot\mathbf{x}}$.

The main idea of our phase encoding approach in Algorithm 3.1 is that by exploiting this spatial translation property, we can separate out the components of recovered frequencies in modulations of the function's Fourier coefficients. Before stating the algorithm in detail, we begin with a simple example.

**Example 3.1** (Phase encoding on a trigonometric monomial)**.** Let $d = 2$. Suppose that $g(\mathbf{x}) = e^{2\pi i \mathbf{k}\cdot\mathbf{x}}$ is a trigonometric monomial with single frequency $\mathbf{k} \in \mathcal{I} \subset \mathbb{Z}^2$ for some known, potentially large $\mathcal{I}$. Given $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathcal{I}$, we consider the one-dimensional restriction of $g$ to the lattice $g^{1\mathrm{d}}(t) := g(t\mathbf{z})$. Since $g$ is Fourier-sparse, a lattice FFT (cf. Algorithm 1.1) on $g^{1\mathrm{d}}$ is unnecessarily expensive. Thus, applying a much faster SFT to $g^{1\mathrm{d}}$ returns $\hat{g}^{1\mathrm{d}}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = 1$. Our goal is to match this coefficient of $g^{1\mathrm{d}}$ to the correct Fourier coefficient of $g$ without having to search all of $\mathcal{I}$.

Figure 3.1 depicts the phase encoding method we use in Algorithm 3.1 below. In order to compute $g^{1\mathrm{d}}$, we restrict $g$ to the dark blue line in this figure, $\mathbf{z}t$. However, to get extra information about $\mathbf{k}$, we also consider $S_{0,1/K}g$, a shift of $g$ in the first coordinate by $1/K$, restricted to the same lattice. The shifted lattice that we effectively restrict $g$ to, $\mathbf{z}t + (1/K, 0)$, is depicted in light blue.

The resulting modulation of $g$ induced by this spatial shift (as described by (3.6)) is detailed in the remainder of Figure 3.1. Thus, defining $g^{1\mathrm{d},1}(t) := S_{0,1/K}g(\mathbf{z}t)$, an SFT would discover

$\hat{g}^{1\mathrm{d},1}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = \mathrm{e}^{2\pi i k_0/K}$. We then can extract $k_0$ from this modulation.

Repeating this process in the $\ell = 1$ coordinate will recover $k_1$, and therefore, the entirety of $\mathbf{k}$ is recovered by using $d = 2$ SFTs. From here, we can then match $\hat{g}^{1\mathrm{d}}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = 1$ to $\hat{g}_{\mathbf{k}}$ in faster than $O(|\mathcal{I}|)$ time and memory as desired.

In the language of Algorithm 3.1, the original SFT of $g^{1\mathrm{d}}$ occurs on Line 1. The SFTs of the shifts of $g$, denoted $g^{1\mathrm{d},0}, \ldots, g^{1\mathrm{d},d-1}$, occur on Line 3. In this example, we considered a function with only one significant Fourier mode, however, we will generally recover $s$ significant Fourier modes from the SFT algorithm. Thus, the for loop from Lines 6 to 14 considers each of these recovered one-dimensional frequencies separately. Line 9 computes the modulation induced by each of the $d$ shifts, then extracts each coordinate of the $d$-dimensional frequency. The remaining check on Line 11 is useful for the theoretical analysis to ensure that spuriously recovered frequencies are ignored.

**Algorithm 3.1** Simple Frequency Index Recovery by Phase Encoding

**Input:** A multivariate periodic function $g \in W(\mathbb{T}^d) \cap C(\mathbb{T}^d)$ (from which we are able to obtain potentially noisy samples), a multivariate frequency set $\mathcal{I} \subset \mathcal{B}^d_K$, a reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathcal{I}$, and an SFT algorithm $\mathcal{A}_{s,M}$.

**Output:** Sparse coefficient vector $\hat{\mathbf{g}}^s = (\hat{g}^s_{\mathbf{k}})_{\mathbf{k}\in\mathcal{B}^d_K}$ (optionally supported on $\mathcal{I}$, see Line 11), an approximation to $(\hat{g}|_{\mathcal{I}})^{\mathrm{opt}}_s$.

1: Apply $\mathcal{A}_{s,M}$ to the univariate restriction of $g$ to the lattice, $g^{1\mathrm{d}}(t) = g(t\mathbf{z})$, to produce $\hat{\mathbf{g}}^{1\mathrm{d},s} = \mathcal{A}_{s,M} g^{1\mathrm{d}}$, a sparse approximation of $\mathbf{F}_M\, \mathbf{g}^{1\mathrm{d}} \in \mathbb{C}^{\mathcal{B}_M}$.
2: **for all** $\ell \in [d]$ **do**
3: Apply $\mathcal{A}_{s,M}$ to $g^{1\mathrm{d},\ell}(t) = S_{\ell,1/K} g(t\mathbf{z})$ to produce $\hat{\mathbf{g}}^{1\mathrm{d},\ell,s} = \mathcal{A}_{s,M} g^{1\mathrm{d},\ell}$, a sparse approximation of $\mathbf{F}_M\, \mathbf{g}^{1\mathrm{d},\ell} \in \mathbb{C}^{\mathcal{B}_M}$.
4: **end for**
5: $\hat{\mathbf{g}}^s \leftarrow \mathbf{0}$
6: **for all** $\omega \in \mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s}) \subset \mathcal{B}_M$ **do**
7: $\mathbf{k}_\omega \leftarrow \mathbf{0}$
8: **for all** $\ell \in [d]$ **do**
9: $(k_\omega)_\ell \leftarrow \mathrm{round}(K \arg(\hat{g}^{1\mathrm{d},\ell,s}_\omega / \hat{g}^{1\mathrm{d},s}_\omega)/2\pi)$
10: **end for**
11: **if** $\mathbf{k}_\omega \cdot \mathbf{z} \equiv \omega \pmod M$ (and optionally $\mathbf{k}_\omega \in \mathcal{I}$; see Remark 3.2) **then**
12: $\hat{g}^s_{\mathbf{k}_\omega} \leftarrow \hat{g}^s_{\mathbf{k}_\omega} + \hat{g}^{1\mathrm{d},s}_\omega$
13: **end if**
14: **end for**

#### 3.3.1.1 Analysis of Algorithm 3.1

Having seen the phase encoding approach of Algorithm 3.1 in action, we now provide an error guarantee for its output. Notice that the assumptions on the SFT necessary for this theoretical analysis are exactly those provided by Theorems 3.1 and 3.2. When we use the complex argument function in Algorithm 3.1 and below, we use the principal branch, so that $\arg : \mathbb{C} \to (-\pi, \pi]$.

**Lemma 3.2** (General recovery result for Algorithm 3.1)**.** *Let $\mathcal{A}_{s,M}$ in the input to Algorithm 3.1 be a noise-robust SFT algorithm which, for a function $g^{1\mathrm{d}} \in W(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, constructs an $s$-sparse Fourier approximation $\mathcal{A}_{s,M}(g^{1\mathrm{d}} + \mu) =: \hat{\mathbf{g}}^{1\mathrm{d},s} \in \mathbb{C}^{\mathcal{B}_M}$ which*

1. *reconstructs every frequency (up to $s$ many) of $\mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}} \in \mathbb{C}^M$, $\omega \in \mathcal{B}_M$, with corresponding Fourier coefficient meeting the tolerance $|(\mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}})_\omega| > \tau$,*
2. *satisfies the $\ell^\infty$ error estimate for recovered coefficients*

$$\left\| (\mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s})|_{\mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})} \right\|_\infty \leq \eta_\infty < \tau,$$

3. *satisfies the $\ell^2$ error estimate*

$$\left\| \mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s} \right\|_2 \leq \eta_2,$$

4. *satisfies the $\ell^1$ error estimate*

$$\left\| \mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}} - \hat{\mathbf{g}}^{1\mathrm{d},s} \right\|_1 \leq \eta_1,$$

5. *and requires $O(P(s, M))$ total evaluations of $g^{1\mathrm{d}}$, operating with computational complexity $O(R(s, M))$.*

*Additionally, assume that the parameters $\tau$ and $\eta_\infty$ hold uniformly for each SFT performed in Algorithm 3.1.*

*Let $g$, $\mathcal{I}$, and $\Lambda(\mathbf{z}, M)$ be as specified in the input to Algorithm 3.1. Collecting the $\tau$-significant frequencies of $g$ into the set $\mathcal{S}_\tau := \{\mathbf{k} \in \mathcal{I} \mid |\hat{g}_\mathbf{k}| > \tau\}$, assume that $|\mathcal{S}_\tau| \leq s$, and set*

$$\beta = \max\left(\tau, \eta_\infty \left(1 + \frac{2}{\sin\left(\frac{\pi}{K}\right)}\right)\right).$$

*Then Algorithm 3.1 (ignoring the optional check on Line 11) will produce an $s$-sparse approximation $\hat{\mathbf{g}}^s$ of the Fourier coefficients of $g$ satisfying the error estimates*

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} \leq \eta_2 + (\beta + \eta_\infty)\sqrt{\max(s - |\mathcal{S}_\beta|, 0)} \\ + \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)}\end{aligned}$$

*and*

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} \leq \eta_1 + (\beta + \eta_\infty)\max(s - |\mathcal{S}_\beta|, 0) \\ + \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)}\end{aligned}$$

*requiring $O\left(d \cdot P(s, M)\right)$ total evaluations of $g$, in $O\left(d \cdot (R(s, M) + s)\right)$ total operations.*

*Proof.* We begin by assuming that $g$ is a trigonometric polynomial with $\operatorname{supp}(\hat{g}) \subset \mathcal{I}$. Since $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\mathcal{I}$, there are no collisions among the one-dimensional frequencies $\{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathcal{I}\}$ modulo $M$. Setting $g(t\mathbf{z}) = g^{1\mathrm{d}}(t)$ then ensures that for each $\mathbf{k} \in \mathcal{I}$, $\hat{g}_{\mathbf{k}} = \hat{g}^{1\mathrm{d}}_{\mathbf{k}\cdot\mathbf{z}}$. Since there are no frequency collisions in the lattice FFT, Lemma 1.3 implies that $\hat{g}_{\mathbf{k}} = (\mathbf{F}_M \mathbf{g}^{1\mathrm{d}})_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Thus, by assumption 1 on the SFT algorithm $\mathcal{A}_{s,M}$, Lines 1 and 3 of Algorithm 3.1 will produce coefficient estimates of $\hat{g}_{\mathbf{k}}$ for every $\mathbf{k} \in \mathcal{S}_\tau$. We then write these SFT approximations as $\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = \hat{g}_{\mathbf{k}} + \eta_{\mathbf{k}}$ and $\hat{g}^{1\mathrm{d},\ell,s}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = \mathrm{e}^{2\pi \mathrm{i} k_\ell / K}(\hat{g}_{\mathbf{k}} + \eta^\ell_{\mathbf{k}})$ respectively, where we have made use of (3.6). Note that $|\eta_{\mathbf{k}}|, |\eta^\ell_{\mathbf{k}}| \leq \eta_\infty$. Now, considering the estimate for $k_\ell$, we have

$$\begin{aligned}\frac{K}{2\pi}\arg\left(\frac{\hat{g}^{1\mathrm{d},\ell,s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) &= \frac{K}{2\pi}\arg\left(\mathrm{e}^{2\pi\mathrm{i}k_\ell/K}\frac{\hat{g}_{\mathbf{k}} + \eta^\ell_{\mathbf{k}}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) \\ &= k_\ell + \frac{K}{2\pi}\arg\left(\frac{\hat{g}_{\mathbf{k}} + \eta^\ell_{\mathbf{k}}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) \\ &= k_\ell + \frac{K}{2\pi}\arg\left(1 + \frac{\eta^\ell_{\mathbf{k}} - \eta_{\mathbf{k}}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right).\end{aligned}$$

We now only consider $|\hat{g}_{\mathbf{k}}| > \beta \geq \max(\tau, 3\eta_\infty)$, that is $\mathbf{k} \in \mathcal{S}_\beta \subset \mathcal{S}_\tau$, and therefore, the corresponding approximate coefficient satisfies $|\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}| > \beta - \eta_\infty$. Thus, the magnitude of the fraction in the argument must be strictly less than $\frac{2\eta_\infty}{\beta - \eta_\infty} \leq 1$. Therefore, we consider the argument of a point lying in the right half of the complex plane, in the open disc of radius $\frac{2\eta_\infty}{\beta-\eta_\infty}$ centered at 1. The

maximal absolute argument of a point in this disc will be that of a point lying on a tangent line passing through the origin. This point, the origin, and 1 then form a right triangle from which we deduce that

$$\left|\arg\left(1+\frac{\eta_{\mathbf{k}}^{\ell}-\eta_{\mathbf{k}}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right)\right| < \arcsin\left(\frac{2\eta_\infty}{\beta-\eta_\infty}\right),$$

and our choice of $\beta \geq \eta_\infty(1+2/\sin(\pi/K))$ then implies that

$$\left|\arg\left(1+\frac{\eta_{\mathbf{k}}^{\ell}-\eta_{\mathbf{k}}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot z \bmod M}}\right)\right| < \frac{\pi}{K}.$$

Thus,

$$\left|\frac{K}{2\pi}\arg\left(\frac{\hat{g}^{1\mathrm{d},\ell,s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}{\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right)-k_\ell\right| < \frac{1}{2},$$

and so after rounding to the nearest integer, Algorithm 3.1 will recover $k_\ell$ for all $\ell \in [d]$ and $\mathbf{k} \in \mathcal{S}_\beta$.

We now know that the final loop of Algorithm 3.1 will properly map the one-dimensional frequency $\omega = \mathbf{k}\cdot\mathbf{z} \bmod M$ to $\mathbf{k}$ for all $\mathbf{k} \in \mathcal{S}_\beta$. Thus, for these same $\mathbf{k} \in \mathcal{S}_\beta$, Line 12 ensures that we set $\hat{g}^s_{\mathbf{k}} := \hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Additionally, the $\max(s-|\mathcal{S}_\beta|,0)$ many coefficients $\hat{g}^{1\mathrm{d},s}_\omega$ for which $\omega \neq \mathbf{k}\cdot\mathbf{z} \bmod M$ for any $\mathbf{k} \in \mathcal{S}_\beta$ are still available for potential assignment. If any multivariate frequency $\mathbf{k}_\omega \in \mathcal{I}$ is reconstructed and passes the mandatory check in Line 11 then the approximate Fourier coefficient $\hat{g}^{1\mathrm{d},s}_\omega$ properly corresponds to $(\mathbf{F}_M\,\mathbf{g}^{1\mathrm{d}})_{\mathbf{k}_\omega\cdot\mathbf{z} \bmod M} = \hat{g}_{\mathbf{k}_\omega}$.

On the other hand, if some error introduced in the SFTs reconstructs a multivariate frequency $\mathbf{k}_\omega \notin \mathcal{I}$, the reconstructing property does not allow us to conclude anything about a $(k_\omega,\omega)$ pair passing the check in Line 11. Thus, it is possible that $\hat{g}^{1\mathrm{d},s}_\omega$ will contribute to some component of $\hat{\mathbf{g}}^s$ not corresponding to any frequency in $\mathcal{I}$. At the least however, since we know that all entries of $\hat{\mathbf{g}}^{1\mathrm{d},s}$ corresponding to frequencies in $\mathcal{S}_\beta$ are correctly assigned, the remaining ones satisfy $|\hat{g}^{1\mathrm{d},s}_\omega| \leq \beta+\eta_\infty$. Using these facts allows us to estimate the $\ell^2$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s-\hat{g}\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\hat{\mathbf{g}}^s|_{\mathbb{Z}^d\setminus\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{\mathbf{g}}^s|_{\mathcal{I}}-\hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^s)\cap\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g}-\hat{g}|_{\mathcal{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} \\ &\leq (\beta+\eta_\infty)\sqrt{\max(s-|\mathcal{S}_\beta|,0)} + \eta_2 + \|\hat{g}-\hat{g}|_{\mathcal{S}_\beta}\|_{\ell^2(\mathcal{I})}\end{aligned} \tag{3.7}$$

and the $\ell^1$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s-\hat{g}\|_{\ell^1(\mathbb{Z}^d)} &\leq \|\hat{\mathbf{g}}^s|_{\mathbb{Z}^d\setminus\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{\mathbf{g}}^s|_{\mathcal{I}}-\hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^s)\cap\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g}-\hat{g}|_{\mathcal{S}_\beta}\|_{\ell^1(\mathbb{Z}^d)} \\ &\leq (\beta+\eta_\infty)\max(s-|\mathcal{S}_\beta|,0) + \eta_1 + \|\hat{g}-\hat{g}|_{\mathcal{S}_\beta}\|_{\ell^1(\mathcal{I})}\end{aligned} \tag{3.8}$$

where we have additionally used the accuracy of the initial one-dimensional SFT and the assumption that $\hat{g}$ is supported on $\mathcal{I}$.

We now handle the case when $g$ is not necessarily a polynomial with Fourier support contained in $\mathcal{I}$. Rather than aiming to approximate $\hat{g}_{\mathbf{k}}$ for every $\mathbf{k} \in \mathbb{Z}^d$, we restrict attention to only frequencies in $\mathcal{I}$, instead attempting to approximate the Fourier coefficients of $g|_{\mathcal{I}} = \sum_{\mathbf{k}\in\mathcal{I}} \hat{g}_{\mathbf{k}} e^{2\pi i \mathbf{k}\cdot\circ}$. We then have that $g =: g|_{\mathcal{I}} + g|_{\mathbb{Z}^d\setminus\mathcal{I}}$ and view potentially noisy input $g+\mu$ to our algorithm as

$$g + \mu = g|_{\mathcal{I}} + \underbrace{g|_{\mathbb{Z}^d\setminus\mathcal{I}} + \mu}_{\mu'}.$$

Algorithm 3.1 applied to $g+\mu$ is then equivalent to applying it to $g|_{\mathcal{I}}+\mu'$, where now $\tau, \eta_\infty, \eta_2$, and $\eta_1$ depend on $\mu'$, and the output is an approximation of $\hat{g}|_{\mathcal{I}}$. Since $\mu'$ represents noise on the input to $\mathcal{A}_{s,M}$ in its applications to $g|_{\mathcal{I}}(t\mathbf{z})$ and $S_{\ell,1/K}g|_{\mathcal{I}}(t\mathbf{z})$ we remark here that

$$\|\mu'\|_\infty \le \|g|_{\mathbb{Z}^d\setminus\mathcal{I}}\|_\infty + \|\mu\|_\infty \le \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\mu\|_\infty \tag{3.9}$$

so as to help us estimate $\tau, \eta_\infty, \eta_2$, and $\eta_1$ in future applications of the lemma. Accounting for the truncation to $\mathcal{I}$ in the $\ell^2$ error bound and using (3.7) applied to $\hat{g}|_{\mathcal{I}}$, we estimate the $\ell^2$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} &\le \|\hat{\mathbf{g}}^s - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} \\ &\le (\beta+\eta_\infty)\sqrt{\max(s - |\mathcal{S}_\beta|, 0)} + \eta_2 + \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} \\ &\quad + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)}\end{aligned}$$

and the $\ell^1$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} &\le \|\hat{\mathbf{g}}^s - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} \\ &\le (\beta+\eta_\infty)\max(s - |\mathcal{S}_\beta|, 0) + \eta_1 + \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_{\ell^1(\mathbb{Z}^d)} \\ &\quad + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)}\end{aligned}$$

Recalling that $P(s,M)$ and $R(s,M)$ are the sampling and runtime complexity of $\mathcal{A}_{s,M}$ respectively, since $1+d$ SFTs are required, the number of $g$ evaluations is $O(d\cdot P(s,M))$ and the associated computational complexity is $O(d\cdot R(s,M))$. The complexity of Lines 6 to 14 is $O(sd)$. □

*Remark* 3.2. Since the only possible misassigned values of $\hat{g}_\omega^{1\mathrm{d},s}$ contribute to coefficients in $\hat{\mathbf{g}}^s$ outside the chosen frequency set $\mathcal{I}$ for which $\Lambda(\mathbf{z}, M)$ is reconstructing, if it is possible to quickly (e.g., in $O(d)$ time) check a multivariate frequency's inclusion in $\mathcal{I}$ (e.g., a hyperbolic cross), entries outside of $\mathcal{I}$ in $\hat{\mathbf{g}}^s$ can be identified in the optional check on Line 11 and remain (correctly) unassigned. This has the effect of removing the $\max(s-|\mathcal{S}_\beta|, 0)$ terms in the error bounds while not increasing the computational complexity. Additionally, this outputs an approximation to $(\hat{g}|_\mathcal{I})_s^{\mathrm{opt}}$ which is supported only on our supplied frequency set $\mathcal{I}$ as we may expect or prefer.

We now apply Lemma 3.2 with the discrete sublinear-time SFT from Theorem 3.2 to give specific error bounds in terms of best $s$-term approximation errors as well as detailed runtime and sampling complexities.

**Corollary 3.1** (Algorithm 3.1 with discrete sublinear-time SFT)**.** *Let $K \geq 9$. For $\mathcal{I} \subset \mathcal{B}_K^d$ with reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ and the function $g \in W(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 3.1 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude. Using the discrete sublinear-time SFT algorithm $\mathcal{A}_{2s,M}^{\mathrm{disc}}$ or $\mathcal{A}_{2s,M}^{\mathrm{disc,MC}}$ with parameter $1 \leq r \leq \frac{M}{36}$, Algorithm 3.1 will produce $\hat{\mathbf{g}}^s = (\hat{g}_\mathbf{k}^s)_{\mathbf{k}\in\mathcal{B}_K^d}$ a $2s$-sparse approximation of $\hat{g}$ satisfying the error estimates*

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_2 \leq (48+4K)\,\frac{\|\hat{g}|_\mathcal{I} - (\hat{g}|_\mathcal{I})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + (189+16K)\,\sqrt{s}(\|g\|_\infty M^{-r} + \|\hat{g}-\hat{g}|_\mathcal{I}\|_1 + e_\infty)$$

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_1 \leq (69+6K)\left\|\hat{g}|_\mathcal{I} - (\hat{g}|_\mathcal{I})_s^{\mathrm{opt}}\right\|_1 + (267+23K)s\,(\|g\|_\infty M^{-r} + \|\hat{g}-\hat{g}|_\mathcal{I}\|_1 + e_\infty)\,,$$

*albeit with probability $1-\sigma \in [0,1)$ for the Monte Carlo version. The total number of evaluations of $g$ and computational complexity will be*

$$O\left(\frac{ds^2r^{3/2}\log^{11/2}M}{\log s}\right) \text{ or } O\left(dsr^{3/2}\log^{9/2}M\log\left(\frac{dM}{\sigma}\right)\right)$$

*for $\mathcal{A}_{2s,M}^{\mathrm{disc}}$ or $\mathcal{A}_{2s,M}^{\mathrm{disc,MC}}$ respectively.*

*Proof.* For the definitions of $\tau$ and $\beta$ in Lemma 3.2 with associated values given by Theorem 3.2,

Lemma 3.1 applied with $\mathbf{x} = \hat{g}|_{\mathcal{I}}$ implies that $\mathcal{S}_\beta$ can contain at most $2s$ elements and the bound

$$\begin{aligned} \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_{2s}^{\mathrm{opt}}\|_{\ell^2(\mathbb{Z}^d)} + \beta\sqrt{2s} \\ &\leq \frac{\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_{s}^{\mathrm{opt}}\|_{\ell^1(\mathbb{Z}^d)}}{2\sqrt{s}} + \beta\sqrt{2s} \end{aligned} \tag{3.10}$$

holds. Note that the last inequality follows from [24, Theorem 2.5] applied to $\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\mathrm{opt}}$. Lemma 3.2 then holds with $s$ replaced by $2s$ for the $2s$-sparse approximations given by $\mathcal{A}_{2s,M}^{\mathrm{disc}}$ or $\mathcal{A}_{2s,M}^{\mathrm{disc,MC}}$ in Algorithm 3.1.

After treating the truncation error as measurement noise as well as accounting for any noise in the input bounded by $e_\infty$, Theorem 3.2 gives the values

$$\eta_\infty = 3\sqrt{2}\left(\frac{\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|g\|_\infty M^{-r} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + e_\infty)\right),$$

$$\tau = \frac{12(1+\sqrt{2})}{3\sqrt{2}}\eta_\infty.$$

Assuming $K \geq 9$,

$$\beta = \max\left(\tau, \eta_\infty\left(1 + \frac{2}{\sin\left(\frac{\pi}{K}\right)}\right)\right) = \eta_\infty\left(1 + \frac{2}{\sin\left(\frac{\pi}{K}\right)}\right) \leq \eta_\infty\left(1 + \frac{2}{9\sin\left(\frac{\pi}{9}\right)}K\right).$$

Inserting the estimate for $\|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_\beta}\|_2$ from (3.10), this bound for $\beta$, and the values for $\eta_2$ (where again we use [24, Theorem 2.5]) and $\eta_1$ from Theorem 3.2

$$\eta_2 \leq \frac{77\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\mathrm{opt}}\|_1}{2\sqrt{s}} + 152\sqrt{s}(\|g\|_\infty M^{-r} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + e_\infty)$$

$$\eta_1 \leq 55\left\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\mathrm{opt}}\right\|_1 + 215s(\|g\|_\infty M^{-r} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + e_\infty)$$

into the recovery bound in Lemma 3.2 and upper bounding $\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_2$ by $\sqrt{s}\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1$ gives the final error estimate.

The change to the complexity of the randomized algorithm arises from distributing the probability of failure $\sigma$ over the $d+1$ SFTs in a union bound. □

Because the nonequispaced SFTs discussed in Theorem 3.1 do not approximate the discrete Fourier transform and therefore do not alias the one-dimensional frequencies $\mathbf{k}\cdot\mathbf{z}$ into frequencies in $\mathcal{B}_M$, slightly modifying Algorithm 3.1 to use SFTs with a larger bandwidth allows for the following recovery result.

**Corollary 3.2** (Algorithm 3.1 with nonequispaced sublinear-time SFT)**.** *For $\mathcal{I} \subset \mathcal{B}_K^d$ with $K \geq 6$, fix the new bandwidth parameter $\tilde{M} := 2\max_{\mathbf{k}\in\mathcal{I}} |\mathbf{k}\cdot\mathbf{z}| + 1$. For $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathcal{I}$ with $M \leq \tilde{M}$, and the function $g \in W(\mathbb{T}^d)\cap C(\mathbb{T}^d)$, we consider applying Algorithm 3.1 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude with the following modifications:*

1. *use the sublinear-time SFT algorithm $\mathcal{A}^{\text{sub}}_{2s,\tilde{M}}$ or $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$*
2. *and only check equality against $\omega$ in Line 11 (rather than equivalence modulo $M$),*

*to produce $\hat{\mathbf{g}}^s = (\hat{g}^s_{\mathbf{k}})_{\mathbf{k}\in\mathcal{B}_K^d}$ a $2s$-sparse approximation of $\hat{g}$ satisfying the error estimates*

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} \leq (25+3K)\left[\frac{\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})^{\text{opt}}_s\|_1}{\sqrt{s}} + \sqrt{s}\|\hat{g}-\hat{g}|_{\mathcal{I}}\|_1 + \sqrt{s}e_\infty\right],$$
$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} \leq (35+3K)\left[\left\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})^{\text{opt}}_s\right\|_1 + s\|\hat{g}-\hat{g}|_{\mathcal{I}}\|_1 + se_\infty\right]$$

*albeit with probability $1-\sigma \in [0,1)$ for the Monte Carlo version. For $\mathcal{A}^{\text{sub}}_{2s,\tilde{M}}$ and $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$ respectively, the total number of evaluations of $g$ and computational complexity will be*

$$O\left(\frac{ds^2\log^4\tilde{M}}{\log s}\right) \text{ or } O\left(ds\log^3(\tilde{M})\log\left(\frac{d\tilde{M}}{\sigma}\right)\right).$$

*Proof.* The bandwidth specified ensures that $\mathcal{B}_{\tilde{M}} \supset \{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k}\in\mathcal{I}\}$. In the case where $g$ is a trigonometric polynomial with $\text{supp}(\hat{g}) \subset \mathcal{I}$, so long as there exists some $M \leq \tilde{M}$ such that $\Lambda(\mathbf{z}, M)$ is reconstructing for $\mathcal{I}$, we are guaranteed that a length-$\tilde{M}$ DFT on a polynomial supported on $\{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k}\in\mathcal{I}\}$ will not suffer from aliasing collisions. Thus, by Lemma 1.2, the one-dimensional Fourier transforms truncated to $\mathcal{B}_{\tilde{M}}$ coincide with length $\tilde{M}$ DFTs. We can therefore view an approximation from the algorithm in Theorem 3.1 as one of a length $\tilde{M}$ DFT. The reasoning in the proofs of Lemma 3.2 and Corollary 3.1 then holds with the SFT algorithms, parameters, numbers of samples, and complexities of Theorem 3.1. □

*Remark* 3.3. As in Chapter 2, (2.7) and (2.8), we can estimate $\tilde{M}$ above with two different tech-

niques:

$$\tilde{M} = 1 + 2 \max_{\mathbf{k} \in \mathcal{I}} \left| \sum_{\ell \in [d]} k_\ell z_\ell \right| \leq 1 + 2 \sum_{\ell \in [d]} |z_\ell| \max_{\mathbf{k} \in \mathcal{I}} |k_\ell| = O(dK_{\mathcal{I}} M),$$

$$\tilde{M} = 1 + 2 \max_{\mathbf{k} \in \mathcal{I}} \left| \sum_{\ell \in [d]} k_\ell z_\ell \right| \leq 1 + 2 \|\mathbf{z}\|_\infty \max_{\mathbf{k} \in \mathcal{I}} \|\mathbf{k}\|_1 = O\left( M \max_{\mathbf{k} \in \mathcal{I}} \|\mathbf{k}\|_1 \right).$$

The latter case is especially useful when $\mathcal{I}$ is a subset of a known $\ell^1$ ball as it will provide a dimension independent upper bound on $\tilde{M}$. Either of these upper bounds may then be used in practice to avoid having to estimate $\tilde{M}$.

That being said however, if one is willing to perform the one-time search through the frequency set $\mathcal{I}$ to more accurately calculate $\tilde{M}$, one can go even further to use the minimal bandwidth $\tilde{M}' = \max_{\mathbf{k} \in \mathcal{I}} (\mathbf{k} \cdot \mathbf{z}) - \min_{\mathbf{k} \in \mathcal{I}} (\mathbf{k} \cdot \mathbf{z}) + 1$ so long as the function samples are properly modulated to shift the one-dimensional frequencies into $\mathcal{B}_{\tilde{M}'}$. For example, running $\mathcal{A}^{\text{sub}}_{2s,\tilde{M}'}$ or $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}'}$ on $g^{\text{1d}}(t) = \mathrm{e}^{2\pi \mathrm{i} \phi t} g(t\mathbf{z})$ and $g^{\text{1d},\ell}(t) = \mathrm{e}^{2\pi \mathrm{i} \phi t} S_{\ell,1/K} g(t\mathbf{z})$ with $\phi = \left\lfloor \frac{\tilde{M}'}{2} \right\rfloor - \max_{\mathbf{k} \in \mathcal{I}} (\mathbf{k} \cdot \mathbf{z})$ is acceptable so long as this shift is accounted for in the frequency check on Line 11. Note though that these improvements will only have the effect of reducing the logarithmic factors in the computational complexity.

### 3.3.2 Two-dimensional DFT technique

Below, we will consider a method for recovering frequencies which, rather than shifting one dimension of the multivariate periodic function $g$ at a time, leaves one dimension of $g$ out at a time. We will fix one dimension $\ell \in [d]$ of $g$ at equispaced nodes over $\mathbb{T}$ and apply a lattice SFT to the other $d - 1$ components. Applying a standard FFT to the results will produce a two-dimensional DFT. The indices corresponding to the standard FFT will represent frequency components in dimension $\ell$ while the indices corresponding to the lattice SFT will be used to synchronize with known one-dimensional frequencies $\mathbf{k} \cdot \mathbf{z} \bmod M$.

Note that below, we will separate out coordinate $\ell$ of a multivariate point $\mathbf{x} \in \mathbb{T}^d$ or frequency $\mathbf{k} \in \mathbb{Z}^d$, denoting the remaining coordinates as $\mathbf{x}'_\ell \in \mathbb{T}^{d-1}$ or $\mathbf{k}'_\ell \in \mathbb{Z}^{d-1}$. With a slight abuse of notation, we can rewrite the original point or frequency as $\mathbf{x} = (x_\ell, \mathbf{x}'_\ell)$ or $\mathbf{k} = (k_\ell, \mathbf{k}'_\ell)$.

Again, before stating Algorithm 3.2 in detail, we present an example.

**Example 3.2** (Two-dimensional DFT technique on a trigonometric monomial)**.** As in Example 3.1, we let $g$ be the trigonometric monomial $g(\mathbf{x}) := \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \mathbf{x}}$. However, in this example, we let $d = 3$, so $\mathbf{k} \in \mathcal{I} \subset \mathbb{Z}^3$ and the domain of $g$ is $\mathbb{T}^3$ depicted in Figure 3.2. We will consider the procedure to compute the $\ell = 0$ component of $\mathbf{k}$.

First, we take a reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathcal{I}$ and restrict all but the first component of $g$ to the lattice. This produces a two-dimensional function of the form

$$(x_0, t) \mapsto \mathrm{e}^{2\pi \mathrm{i}(k_0 x_0 + k_1 z_1 t + k_2 z_2 t)}.$$

We then sample this function at $K$ equispaced points over $\mathbb{T}$ in the $x_0$ variable. This produces $K$ projected lattices spaced $1/K$ apart in the $x_0$ direction on which we sample $g$, depicted in Figure 3.2. Fixing $x_0$ at each equispaced point produces the $K$ univariate functions which are organized into the top array of Figure 3.3. Notice that colors of the entries in this array correspond to the lattices in Figure 3.2 over which we sample $g$ to produce that entry.

The next step is to apply an SFT to each of the univariate functions in this array. Each function has exactly one active frequency, $k_1 z_1 + k_2 z_2$, with corresponding Fourier coefficient $\mathrm{e}^{2\pi \mathrm{i} k_0 j / K}$. Thus, collecting the results into a matrix produces the left-most matrix in Figure 3.3 with only the $k_1 z_1 + k_2 z_2 \bmod M$ column filled. This column contains $K$ equispaced samples of the function $\mathrm{e}^{2\pi \mathrm{i} k_0 x_0}$, and so finally applying a DFT to the matrix will produce the right-most matrix in Figure 3.3. We find only one entry in row $k_0 \bmod M$ corresponding to the only active frequency of $\mathrm{e}^{2\pi \mathrm{i} k_0 x_0}$. Thus, we can read off the $\ell = 0$ entry of $\mathbf{k}$ by determining which row contains the Fourier coefficient of $g$ of interest. Repeating this process for all $\ell = 0, \ldots, d-1$ we will be able to recover $\mathbf{k}$.

We now generalize the procedure demonstrated in Example 3.2 in a lemma. In particular, we must account for functions which have more than one significant frequency. For theoretical simplicity, we use a length $M$-DFT in the first step rather than an SFT.

**Lemma 3.3.** *Fix some finite multivariate frequency set $\mathcal{I} \subset \mathcal{B}_K^d$, let $\Lambda(\mathbf{z}, M)$ be a reconstructing rank-1 lattice for $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathcal{I}\}$ (where $\mathbf{e}_\ell \in \mathbb{Z}^d$ is the canonical basis vector which has $(\mathbf{e}_\ell)_\ell = 1$ and zeros in all other entries) for all $\ell \in [d]$, and assume that $g$ has Fourier support $\operatorname{supp}(\hat{g}) \subset \mathcal{I}$. Fixing one dimension $\ell \in [d]$, and writing the generating vector as $\mathbf{z} = (z_\ell, \mathbf{z}'_\ell) \in \mathbb{Z}^d$,*

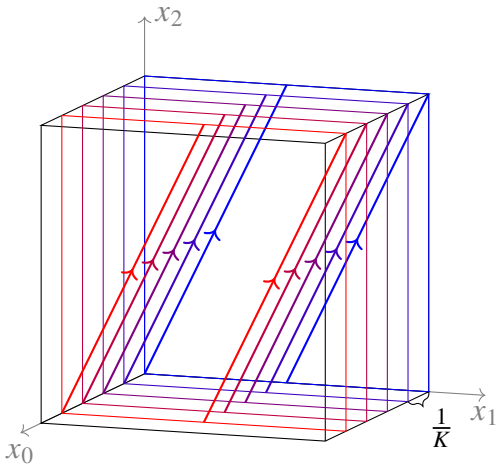



Figure 3.2 An example of $\mathbb{T}^3$ depicting the $K$ projected rank-1 lattices on which $g(\mathbf{x})$ is sampled to compute the $\ell = 0$ component of each $d$-dimensional frequency.

$$\begin{pmatrix} e^{2\pi i(k_0\frac{0}{K}+k_1z_1t+k_2z_2t)} \\ e^{2\pi i(k_0\frac{1}{K}+k_1z_1t+k_2z_2t)} \\ \vdots \\ e^{2\pi i(k_0\frac{K-2}{K}+k_1z_1t+k_2z_2t)} \\ e^{2\pi i(k_0\frac{K-1}{K}+k_1z_1t+k_2z_2t)} \end{pmatrix}$$

Apply SFT $\mathcal{A}_{s,M}$ to rows

$$\begin{pmatrix} 0 & \cdots & 0 & e^{2\pi ik_0\frac{0}{K}} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & e^{2\pi ik_0\frac{1}{K}} & 0 & \cdots & 0 \\ & & & \vdots & & & \\ 0 & \cdots & 0 & e^{2\pi ik_0\frac{K-2}{K}} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & e^{2\pi ik_0\frac{K-1}{K}} & 0 & \cdots & 0 \end{pmatrix}$$

column $k_1z_1 + k_2z_2 \bmod M$

Apply $\mathbf{F}_K$ to columns

$$\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots & & \ddots & \vdots \\ 0 & & 0 & 0 & 0 & & 0 \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & & 0 & 0 & 0 & & 0 \\ \vdots & \ddots & & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \leftarrow \text{row } k_0 \bmod K$$

column $k_1z_1 + k_2z_2 \bmod M$

Figure 3.3 One round of the basic procedure for the two dimensional DFT algorithm applied to the trigonometric monomial $g(\mathbf{x}) = e^{2\pi i\mathbf{k}\cdot\mathbf{x}}$ sampled over the sets depicted in Figure 3.2. Notice that each row corresponds to samples of $g(\mathbf{x})$ on the shifted lattice of the corresponding color.

*define the polynomials*

$$g_j^{1\mathrm{d},\ell}(t) := g\left(\frac{j}{K}, t\mathbf{z}'_\ell\right) \text{ for all } j \in [K],$$

*that is, fix coordinate $\ell$ at $j/K$ and restrict the remaining coordinates to dimensions $[d] \setminus \{\ell\}$ of the rank-1 lattice. Then for all one-dimensional frequencies $\omega \in [M]$,*

$$\left(\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}\right)_\omega = \begin{cases} \sum\limits_{\substack{h_\ell \in \mathcal{B}_K \text{ s.t.} \\ (h_\ell, \mathbf{k}'_\ell) \in \mathcal{I}}} \mathrm{e}^{2\pi \mathrm{i} j h_\ell / K} \, \hat{g}_{(h_\ell, \mathbf{k}'_\ell)} & \text{if } \exists \mathbf{k} \in \mathcal{I} \text{ with } \omega \equiv \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \pmod{M}, \\ 0 & \text{otherwise.} \end{cases}$$

*Moreover, defining the matrix $\mathbf{G}^\ell = \left(\left(\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}\right)_\omega\right)_{j \in [K], \omega \in [M]}$, we have*

$$\left(\mathbf{F}_K \, \mathbf{G}^\ell\right)_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = \hat{g}_{\mathbf{k}} \text{ for all } \mathbf{k} \in \mathcal{I},$$

*and the remaining entries of the matrix $\mathbf{F}_K \, \mathbf{G}^\ell \in \mathbb{C}^{K \times M}$ are zero.*

*Proof.* Using the Fourier series representation of $g$, we have

$$g_j^{1\mathrm{d},\ell}(t) := \sum_{\mathbf{k} \in \mathcal{I}} \hat{g}_{\mathbf{k}} \, \mathrm{e}^{2\pi \mathrm{i} \left(\frac{j k_\ell}{K} + \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell t\right)}.$$

We calculate for $\omega \in [M]$

$$\begin{aligned} \left(\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}\right)_\omega &= \frac{1}{M} \sum_{i \in [M]} \sum_{\mathbf{h} \in \mathcal{I}} \mathrm{e}^{\frac{2\pi \mathrm{i} j h_\ell}{K}} \, \hat{g}_{\mathbf{h}} \, \mathrm{e}^{\frac{2\pi \mathrm{i} (\mathbf{h}'_\ell \cdot \mathbf{z}'_\ell - \omega) i}{M}} \\ &= \sum_{\mathbf{h} \in \mathcal{I}} \mathrm{e}^{\frac{2\pi \mathrm{i} j h_\ell}{K}} \, \hat{g}_{\mathbf{h}} \, \delta_{0, (\mathbf{h}'_\ell \cdot \mathbf{z}'_\ell - \omega \bmod M)}. \end{aligned}$$

When there exists some $\mathbf{k} \in \mathcal{I}$ such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \equiv \omega \bmod M$, the fact that $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathcal{I}\}$ ensures that such $\mathbf{k}'_\ell$ satisfying this equivalence is unique. Then, we can simplify this sum to

$$\left(\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}\right)_\omega = \sum_{\substack{h_\ell \in \mathcal{B}_K \text{ s.t.} \\ (h_\ell, \mathbf{k}'_\ell) \in \mathcal{I}}} \mathrm{e}^{\frac{2\pi \mathrm{i} j h_\ell}{K}} \, \hat{g}_{(h_\ell, \mathbf{k}'_\ell)}.$$

When no $\mathbf{k} \in \mathcal{I}$ exists such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \equiv \omega \pmod{M}$, this sum is instead zero as desired. Applying $\mathbf{F}_K$ to $\mathbf{G}^\ell$ then allows us to compute

$$\left(\mathbf{F}_K \, \mathbf{G}^\ell\right)_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = \frac{1}{K} \sum_{j \in [K]} \sum_{h_\ell \in \mathcal{B}_K} \hat{g}_{(h_\ell, \mathbf{k}'_\ell)} \, \mathrm{e}^{\frac{2\pi \mathrm{i} (h_\ell - k_\ell \bmod K) j}{K}} = \hat{g}_{\mathbf{k}}. \qquad \square$$

**Algorithm 3.2** Frequency Index Recovery by Two Dimensional DFT

**Input:** A multivariate periodic function $g \in W(\mathbb{T}^d) \cap C(\mathbb{T}^d)$ (from which we are able to obtain potentially noisy samples), a multivariate frequency set $\mathcal{I} \subset \mathcal{B}_K^d$, a rank-1 lattice $\Lambda(\mathbf{z}, M)$ which is reconstructing for $\mathcal{I}$ and $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathcal{I}\}$ for all $\ell \in [d]$, and an SFT algorithm $\mathcal{A}_{s,M}$.

**Output:** Sparse coefficient vector $\hat{\mathbf{g}}^s = (\hat{g}^s_{\mathbf{k}})_{\mathbf{k} \in \mathcal{B}_K^d}$ (optionally supported on $\mathcal{I}$, see Line 16), an approximation to $(\hat{g}|_{\mathcal{I}})_s^{\mathrm{opt}}$.

1: Apply $\mathcal{A}_{s,M}$ to the univariate restriction of $g$ to the lattice, $g^{1\mathrm{d}}(t) := g(t\mathbf{z})$, to produce $\hat{\mathbf{g}}^{1\mathrm{d},s} := \mathcal{A}_{s,M} g^{1\mathrm{d}}$, a sparse approximation of $\mathbf{F}_M \, \mathbf{g}^{1\mathrm{d}} \in \mathbb{C}^M$.
2: **for all** $\ell \in [d]$ **do**
3:  **for all** $j \in [K]$ **do**
4:  Apply $\mathcal{A}_{s,M}$ to $g_j^{1\mathrm{d},\ell}(t) := g\left(\frac{j}{K}, t\mathbf{z}'_\ell\right)$ to produce $\hat{\mathbf{g}}_j^{1\mathrm{d},\ell,s} := \mathcal{A}_{s,M} g_j^{1\mathrm{d},\ell}$, a sparse approximation of $\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}$.
5:  Row $j$ of $\mathbf{G}^{\ell,s} \leftarrow \hat{\mathbf{g}}_j^{1\mathrm{d},\ell,s}$.
6:  **end for**
7:  **for all** nonzero columns $\omega$ of $\mathbf{G}^{\ell,s}$ **do**
8:  Apply $\mathbf{F}_K$ to column $\omega$ of $\mathbf{G}^{\ell,s}$ to produce $\mathbf{F}_K \, \mathbf{G}^{\ell,s}$.
9:  **end for**
10: **end for**
11: $\hat{\mathbf{g}}^s \leftarrow \mathbf{0}$
12: **for all** $\omega \in \mathrm{supp}(\hat{\mathbf{g}}^{1\mathrm{d},s})$ **do**
13:  **for all** $\ell \in [d]$ **do**
14:  $((k_\omega)_\ell, \sim) \leftarrow \arg\min\{|\hat{g}_\omega^{1\mathrm{d},s} - (\mathbf{F}_K \, \mathbf{G}^{\ell,s})_{h,\omega'}| \mid (h, \omega') \in \mathcal{B}_K \times [M] \text{ with } hz_\ell + \omega' \equiv \omega \bmod M\}$
15:  **end for**
16:  **if** $\mathbf{k}_\omega \cdot \mathbf{z} \equiv \omega \bmod M$ (and optionally $\mathbf{k}_\omega \in \mathcal{I}$) **then**
17:  $\hat{g}^s_{\mathbf{k}_\omega} \leftarrow \hat{g}^s_{\mathbf{k}_\omega} + \hat{g}_\omega^{1\mathrm{d},s}$
18:  **end if**
19: **end for**

Example 3.2 and Lemma 3.3 explain the procedure in Lines 1 through 10 of Algorithm 3.2. However, some care must be taken when we assign rows of nonzero entries in the resulting matrix to coordinates of significant frequencies. The solution is the minimization problem in Line 14. This step uses column information as well as the values of the entries in the matrix to ensure that we are properly matching frequency components with the correct Fourier coefficient $\hat{g}_\omega^{1\mathrm{d},s}$.

The remainder of the algorithm is the same as Algorithm 3.1. Line 16 consists of the same check to ensure that recovered frequencies are correct, and if this check passes, the one-dimensional Fourier coefficient is assigned to its matched $d$-dimensional frequency.

*Remark* 3.4. We bring special attention to the fact that Algorithm 3.2 requires as input a rank-1

lattice $\Lambda(\mathbf{z}, M)$ which is reconstructing for not only $\mathcal{I}$, but also the projections of $\mathcal{I}$ of the form $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathcal{I}\}$ for any $\ell \in [d]$. For frequency sets $\mathcal{I}$ which are downward closed, that is, if $\mathcal{I}$ is such that for any $\mathbf{k} \in \mathcal{I}$ and $\mathbf{h} \in \mathbb{Z}^d$, $|\mathbf{h}| \leq |\mathbf{k}|$ component-wise implies that $\mathbf{h} \in \mathcal{I}$, any reconstructing rank-1 lattice for $\mathcal{I}$ is necessarily one for the considered projections as well. Thus, for many frequency spaces of interest, e.g., hyperbolic crosses (cf. Remarks 3.2 and 3.3 as well as Section 3.4 below), any reconstructing rank-1 lattice for $\mathcal{I}$ will suffice as input to Algorithm 3.2.

#### 3.3.2.1 Analysis of Algorithm 3.2

With the conceptual explanation of Algorithm 3.2 complete, we now provide error guarantees for its output.

**Lemma 3.4** (General recovery result for Algorithm 3.2.)**.** *Let $g$, $\mathcal{I}$, and $\Lambda(\mathbf{z}, M)$ be as specified in the input to Algorithm 3.2. Additionally, let $\mathcal{A}_{s,M}$ be a noise-robust SFT algorithm satisfying the same constraints as in Lemma 3.2 with parameters $\tau$ and $\eta_\infty$ holding uniformly for each SFT performed in Algorithm 3.2.*

*Collect the $\tau$-significant frequencies of $g$ into the set $\mathcal{S}_\tau := \{\mathbf{k} \in \mathcal{I} \mid |\hat{g}_\mathbf{k}| > \tau\}$ and assume that $|\mathcal{S}_\tau| \leq s$. Then Algorithm 3.2 (ignoring the optional check on Line 16) will produce an $s$-sparse approximation of the Fourier coefficients of $g$ satisfying the error estimates*

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} \leq \eta_2 + (4\tau + \eta_\infty)\sqrt{\max(s - |\mathcal{S}_{4\tau}|, 0)} + \|\hat{g}|_\mathcal{I} - \hat{g}|_{\mathcal{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_\mathcal{I}\|_{\ell^2(\mathbb{Z}^d)}$$

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} \leq \eta_1 + (4\tau + \eta_\infty)\max(s - |\mathcal{S}_{4\tau}|, 0) + \left\|\hat{g}|_\mathcal{I} - \hat{g}|_{\mathcal{S}_{4\tau}}\right\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_\mathcal{I}\|_{\ell^1(\mathbb{Z}^d)}.$$

*requiring $O\left(dK \cdot P(s, M)\right)$ total evaluations of $g$, in $O\left(dK(R(s, M) + sK\log K)\right)$ total operations.*

*Proof.* We begin by assuming that $g$ is a trigonometric polynomial with $\operatorname{supp}(\hat{g}) \subset \mathcal{I}$. Since $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\mathcal{I}$, the DFT-aliasing ensures that Line 1 of Algorithm 3.2 will return approximate coefficients uniquely corresponding to all $\tau$-significant frequencies $\mathbf{k} \in \mathcal{S}_\tau$ which we can label $\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Additionally, Line 4 recovers approximations to all $\tau$-significant frequencies of $\mathbf{F}_M \mathbf{g}_j^{1\mathrm{d},\ell}$ which have the form given in Lemma 3.3. In particular, if

$\mathbf{k} \in \mathcal{S}_\tau$, we have

$$\begin{aligned}
\tau < |\hat{g}_{\mathbf{k}}| &= \left| \left( \mathbf{F}_K \, \mathbf{G}^{\ell} \right)_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \\
&= \left| \frac{1}{K} \sum_{j \in [K]} \left( \mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \mathrm{e}^{\frac{-2\pi \mathrm{i} j k_\ell \bmod K}{K}} \right| \\
&\leq \frac{1}{K} \sum_{j \in [K]} \left| \left( \mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \\
&\leq \max_{j \in [K]} \left| (\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell})_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| .
\end{aligned}$$

Thus, there exists at least one $\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}$ with $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ recovered as a $\tau$-significant frequency in the SFT of Line 4, and $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ will be a nonzero column in $\mathbf{G}^{\ell,s}$ for all $\mathbf{k} \in \mathcal{S}_\tau$.

Analyzing these SFTs in more detail for any $\mathbf{k} \in \mathcal{I}$ such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ is a nonzero column of $\mathbf{G}^{\ell,s}$, we write

$$\left( \hat{\mathbf{g}}_j^{1\mathrm{d},\ell,s} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = \left( \mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} + \left( \eta_j^{\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M}$$

where, by the $\ell^\infty$ and recovery guarantees for $\mathcal{A}_{s,M}$, the error satisfies

$$\left| \left( \eta_j^{\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \leq \begin{cases} \eta_\infty & \text{if } \left( \hat{\mathbf{g}}_j^{1\mathrm{d},\ell,s} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \neq 0 \\ \tau & \text{if } \left( \hat{\mathbf{g}}_j^{1\mathrm{d},\ell,s} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = 0 \end{cases} \leq \tau.$$

Thus, in the application of $\mathbf{F}_K$ to column $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ of $\mathbf{G}^{\ell,s}$, we have

$$\begin{aligned}
&\left( \mathbf{F}_K \, \mathbf{G}^{\ell,s} \right)_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \\
&\quad = \left( \mathbf{F}_K \, \mathbf{G}^{\ell} \right)_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} + \left( \mathbf{F}_K \left( \left( \eta_j^{\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right)_{j \in [K]} \right)_{k_\ell \bmod K} \\
&\quad =: \hat{g}_{\mathbf{k}} + \eta_{\mathbf{k}}^{\ell}
\end{aligned}$$

with

$$|\eta_{\mathbf{k}}^{\ell}| = \left| \frac{1}{K} \sum_{j \in [K]} \left( \eta_j^{\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \mathrm{e}^{\frac{-2\pi \mathrm{i} j k_\ell \bmod K}{K}} \right| \leq \max_{j \in [K]} \left| \left( \eta_j^{\ell} \right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \leq \tau.$$

These same calculations apply to the computed columns of $\mathbf{F}_K \mathbf{G}^{\ell,s}$ which do not correspond to values of $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ for some $\mathbf{k} \in \mathcal{I}$ since we assume $\mathrm{supp}(\hat{g}) \subset \mathcal{I}$, and so at worst, these columns are filled with noise bounded in magnitude by $\tau$.

Restricting our attention to $\mathbf{k} \in \mathcal{S}_{4\tau} \subset \mathcal{S}_\tau$, we know that Line 14 will be run with $\omega = \mathbf{k} \cdot \mathbf{z} \bmod M$ and $(k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M)$ as an admissible index in the minimization. By the reconstructing property of $\Lambda(\mathbf{z}, M)$, no other $\mathbf{h} \in \mathcal{I}$ will correspond to an admissible index $(h_\ell \bmod K, \mathbf{h}'_\ell \cdot \mathbf{z}'_\ell \bmod M)$, and so the only remaining values of $(\mathbf{F}_K \mathbf{G}^{\ell,s})_{h,\omega'}$ in the minimization correspond to pure noise $\eta$ bounded in magnitude by $\tau$. Analyzing the objective at $(k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M)$, we find

$$|\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M} - (\mathbf{F}_K \mathbf{G}^{\ell,s})_{k_\ell \bmod K, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M}| \leq 2\tau < |\hat{g}_{\mathbf{k}}| - 2\tau \leq |\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M} - \eta|,$$

and so the value for $(k_\omega)_\ell$ will in fact be assigned $k_\ell$. Thus, after all $d$ components of $\mathbf{k}_\omega = \mathbf{k}$ have been recovered, $\hat{g}^s_{\mathbf{k}}$ will be assigned $\hat{g}^{1\mathrm{d},s}_{\mathbf{k}\cdot\mathbf{z} \bmod M}$.

The remaining $\max(s - |\mathcal{S}_{4\tau}|, 0)$ nonzero entries of $\hat{\mathbf{g}}^{1\mathrm{d},s}$ can be distributed to entries of $\hat{\mathbf{g}}^s$ possibly correctly but with no guarantee; at the very least however, these values must be at most $4\tau + \eta_\infty$ in magnitude. We split $\hat{\mathbf{g}}^s$ as $\hat{\mathbf{g}}^s = \hat{\mathbf{g}}^{s,\mathrm{correct}} + \hat{\mathbf{g}}^{s,\mathrm{incorrect}}$ to account for the values of $\hat{\mathbf{g}}^s$ respectively assigned correctly and incorrectly and note that $\mathrm{supp}(\hat{\mathbf{g}}^{s,\mathrm{correct}}) \supset \mathcal{S}_{4\tau}$. We then estimate the $\ell^2$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\hat{\mathbf{g}}^{s,\mathrm{correct}} - \hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^{s,\mathrm{correct}})}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{\mathbf{g}}^{s,\mathrm{incorrect}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^{s,\mathrm{correct}})}\|_{\ell^2(\mathbb{Z}^d)} \\ &\leq \eta_2 + (4\tau + \eta_\infty)\sqrt{\max(s - |\mathcal{S}_{4\tau}|, 0)} + \|\hat{g} - \hat{g}|_{\mathcal{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)}\end{aligned}$$

and the $\ell^1$ error as

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} &\leq \left\|\hat{\mathbf{g}}^{s,\mathrm{correct}} - \hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^{s,\mathrm{correct}})}\right\|_{\ell^1(\mathbb{Z}^d)} + \left\|\hat{\mathbf{g}}^{s,\mathrm{incorrect}}\right\|_{\ell^1(\mathbb{Z}^d)} + \left\|\hat{g} - \hat{g}|_{\mathrm{supp}(\hat{\mathbf{g}}^{s,\mathrm{correct}})}\right\|_{\ell^1(\mathbb{Z}^d)} \\ &\leq \eta_1 + (4\tau + \eta_\infty)\max(s - |\mathcal{S}_{4\tau}|, 0) + \left\|\hat{g} - \hat{g}|_{\mathcal{S}_{4\tau}}\right\|_{\ell^1(\mathbb{Z}^d)}.\end{aligned}$$

As in the proof of Lemma 3.2, we note that the mandatory check in Line 16 helps ensure that all misassigned values $\hat{g}^{1\mathrm{d},s}_\omega$ which contribute to $\hat{\mathbf{g}}^{s,\mathrm{incorrect}}$ correspond to reconstructed $\mathbf{k}_\omega$ *outside* of $\mathcal{I}$, with the optional check in this line (see Remark 3.2) eliminating $\hat{\mathbf{g}}^{s,\mathrm{incorrect}}$ and the corresponding term in the error estimate entirely.

Now, supposing that the Fourier support of $g$ is not limited to only $\mathcal{I}$, just as in the analysis for Algorithm 3.1, we treat $g$ as a perturbation of $g|_{\mathcal{I}}$, and use the robust SFT algorithm and the previous argument to approximate $\hat{g}|_{\mathcal{I}}$. Note again that in each SFT, the noise added when using measurements of $g$ as proxies for those of $g|_{\mathcal{I}}$ is compounded by $\|g|_{\mathbb{Z}^d \setminus \mathcal{I}}\|_\infty$ and is bounded by $\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)}$. Applying the guarantees above gives the $\ell^2$ estimate

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\hat{\mathbf{g}}^s - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)} \\ &\leq \eta_2 + (4\tau + \eta_\infty)\sqrt{\max(s - |\mathcal{S}_{4\tau}|, 0)} + \|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^2(\mathbb{Z}^d)}\end{aligned}$$

and the $\ell^1$ estimate

$$\begin{aligned}\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} &\leq \|\hat{\mathbf{g}}^s - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)} \\ &\leq \eta_1 + (4\tau + \eta_\infty)\max(s - |\mathcal{S}_{4\tau}|, 0) + \left\|\hat{g}|_{\mathcal{I}} - \hat{g}|_{\mathcal{S}_{4\tau}}\right\|_{\ell^1(\mathbb{Z}^d)} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_{\ell^1(\mathbb{Z}^d)}.\end{aligned}$$

Employing fast Fourier transforms for the at most $dsK$ DFTs, the computational complexity of Lines 2 to 10 is $O\left(d(K \cdot R(s, M) + sK^2 \log K)\right)$ (which dominates the complexity of the remainder of the algorithm). Since $1 + dK$ SFTs are required, the number of $g$ evaluations is $O(dK \cdot P(s, M))$.

□

*Remark* 3.5. Though the number of nonzero columns of $\mathbf{G}^{\ell,s}$ can be theoretically at most $sK$, in practice with a high quality algorithm, each of the $K$ SFTs should recover nearly the same frequencies, meaning that there are actually $O(s)$ columns. This would remove a power of $K$ in the second term of the runtime estimate.

Note however, that even with near exact SFT algorithms, recovering exactly $s$ total frequencies is not a certainty. There can be cancellations for certain terms in $\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}$ depending interactions between the coefficients sharing the same values on their $[d] \setminus \{\ell\}$ entries, which makes it possible that an SFT on $\mathbf{F}_M \, \mathbf{g}_j^{1\mathrm{d},\ell}$ will miss coefficients. If required to output $s$-entries, an SFT algorithm could favor some noisy value corresponding to a frequency outside the support.

*Remark* 3.6. Though we perform an exact FFT of the nonzero columns of $\mathbf{G}^{1\mathrm{d},\ell}$ in Line 8 of Algorithm 3.2, Lemma 3.3 implies that the resulting matrix will be as sparse as the original function's Fourier transform. Thus, for a truly compressible function, an SFT down the columns of $\mathbf{G}^{1\mathrm{d},\ell}$

would be feasible as well. However, in especially higher dimensions, even small $K$ can allow for large frequency spaces $\mathcal{I}$. In these large frequency spaces, what is perceived as relatively sparse can therefore quickly surpass $K$, rendering an $s$-sparse, length $K$ SFT useless.

As a simple example, consider $\mathcal{I}$ to be the cube of side length $K = s$, $\mathcal{B}_s^d$. For $d$ large enough, any frequency support of size $s$ will be small in comparison to $|\mathcal{I}| = s^d$. However, using an $s$-sparse SFT instead of a length-$s$ DFT in Algorithm 3.2 will actually be more expensive.

Applying the discrete sublinear-time SFT from Theorem 3.2 to Lemma 3.4 analogously to the derivation of Corollary 3.1 from Lemma 3.2 allows for the following recovery bound for Algorithm 3.2. In particular, we observe asymptotically improved error guarantees over Corollary 3.1 at the cost of a slight increase in runtime.

**Corollary 3.3** (Algorithm 3.2 with discrete sublinear-time SFT)**.** *For $\mathcal{I} \subset \mathbb{Z}^d$ with reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ and the function $g \in W(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 3.2 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude. Using the discrete sublinear-time SFT algorithm $\mathcal{A}_{2s,M}^{\text{disc}}$ or $\mathcal{A}_{2s,M}^{\text{disc,MC}}$ with parameter $1 \leq r \leq \frac{M}{36}$ will produce $\hat{\mathbf{g}}^s = (\hat{g}_{\mathbf{k}}^s)_{\mathbf{k}\in\mathcal{B}_K^d}$ a $2s$-sparse approximation of $\hat{g}$ satisfying the error estimates*

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} \leq 206\frac{\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\text{opt}}\|_1}{\sqrt{s}} + 821\sqrt{s}(\|g\|_\infty M^{-r} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + e_\infty)$$

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} \leq 293\left\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})_s^{\text{opt}}\right\|_1 + 1161s\,(\|g\|_\infty M^{-r} + \|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + e_\infty)$$

*albeit with probability $1 - \sigma \in [0, 1)$ for the Monte Carlo version.*

*The total number of evaluations of $g$ and the computational complexity will be*

$$O\left(dsK\left(\frac{sr^{3/2}\log^{11/2} M}{\log s} + K\log K\right)\right)$$

$$\text{or} \quad O\left(dsK\left(r^{3/2}\log^{9/2}(M)\log\left(\frac{dKM}{\sigma}\right) + K\log K\right)\right)$$

*for $\mathcal{A}_{2s,M}^{\text{disc}}$ or $\mathcal{A}_{2s,M}^{\text{disc,MC}}$ respectively.*

Again, the same strategy from Corollary 3.2 of widening the frequency band and shifting the one-dimensional transforms accordingly allows us to use the nonequispaced SFT algorithm from Theorem 3.1 in Algorithm 3.2. Note here that the widening and shifting occurs on a dimension by

dimension basis so as to account for the differing one-dimensional frequencies of the form $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell$ for $\mathbf{k} \in \mathcal{I}$.

**Corollary 3.4** (Algorithm 3.2 with nonequispaced sublinear-time SFT)**.** *For $\mathcal{I} \subset \mathcal{B}_K^d$, let $\tilde{M}$ be the larger one-dimensional bandwidth parameter from Corollary 3.2, and additionally define $\tilde{M}^\ell := 2\max_{\mathbf{k}\in\mathcal{I}} |\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell| + 1$. For $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathcal{I}$ and where $M$ is such that $M \leq \min\{\tilde{M}, \min_{\ell\in[d]} \tilde{M}^\ell\}$, for the function $g \in W(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 3.2 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude with the following modifications:*

1. *use the sublinear-time SFT algorithm $\mathcal{A}^{\text{sub}}_{2s,\tilde{M}}$ or $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$ in Line 1 and $\mathcal{A}^{\text{sub}}_{2s,\tilde{M}^\ell}$ or $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}^\ell}$ in Line 4*
2. *and only check equality against $\omega$ in Line 14 (rather than equivalence modulo $M$),*

*to produce $\hat{\mathbf{g}}^s = (\hat{g}^s_{\mathbf{k}})_{\mathbf{k}\in\mathcal{B}_K^d}$ a $2s$-sparse approximation of $\hat{g}$ satisfying the error estimates*

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^2(\mathbb{Z}^d)} \leq 98\left(\frac{\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})^{\text{opt}}_s\|_1}{\sqrt{s}} + \sqrt{s}\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + \sqrt{s}e_\infty\right)$$

$$\|\hat{\mathbf{g}}^s - \hat{g}\|_{\ell^1(\mathbb{Z}^d)} \leq 139\left(\left\|\hat{g}|_{\mathcal{I}} - (\hat{g}|_{\mathcal{I}})^{\text{opt}}_s\right\|_1 + s\|\hat{g} - \hat{g}|_{\mathcal{I}}\|_1 + se_\infty\right),$$

*albeit with probability $1 - \sigma \in [0, 1)$ for the Monte Carlo version.*

*Letting $\bar{M} = \max(\tilde{M}, \max_{\ell\in[d]} \tilde{M}^\ell)$, the total number of evaluations of $g$ will be*

$$O\left(\frac{dKs^2 \log^4 \bar{M}}{\log s}\right) \text{ or } O\left(dKs \log^3 \bar{M} \log\left(\frac{dK\bar{M}}{\sigma}\right)\right)$$

*with associated computational complexities*

$$O\left(dKs\left(\frac{s\log^4 \bar{M}}{\log s} + K\log K\right)\right) \text{ or } O\left(dKs\left(\log^3 \bar{M}\log\left(\frac{dK\bar{M}}{\sigma}\right) + K\log K\right)\right)$$

*for $\mathcal{A}^{\text{sub}}_{2s,\cdot}$ and $\mathcal{A}^{\text{sub,MC}}_{2s,\cdot}$ respectively.*

*Remark* 3.7. The bounds in Remark 3.3 will still hold for $\tilde{M}^\ell$ as well; thus one of these upper bounds can be used as the effective bandwidth parameter for every SFT without having to calculate the $d + 1$ bandwidths by scanning $\mathcal{I}$. Again however, if this scan is tolerable, one can reduce the overall complexity by using analogous minimal bandwidths discussed in Remark 3.3 along with corresponding frequency shifts.

### 3.4 Numerics

We now demonstrate the effectiveness of our phase encoding and two-dimensional DFT algorithms for computing Fourier coefficients of multivariate functions in a series of empirical tests. The two techniques are implemented in MATLAB, with the code for the algorithms and tests in this section publicly available[1]. The results below use a MATLAB implementation[2] of the randomized univariate sublinear-time nonequispaced algorithm $\mathcal{A}_{2s,M}^{\text{sub,MC}}$ (cf. Theorem 3.1) as the underlying SFT for both multivariate approaches as this allows for the fastest runtime and most sample efficient implementations.

In the univariate code, all parameters but one are qualitatively tuned below theoretical upper bounds to increase efficiency while maintaining accuracy and are kept constant between tests below. In particular, we fix the values C := 1, sigma := 2/3, and primeShift := 0 (see the documentation and the original paper [37] for more detail). The only parameter we vary is "randomScale" which affects the rate at which the deterministic algorithm $\mathcal{A}_{2s,M}^{\text{sub}}$ is randomly sampled to produce the Monte Carlo version $\mathcal{A}_{2s,M}^{\text{sub,MC}}$. This parameter represents a multiplicative scaling on logarithmic factors of the bandwidth which determines how many prime numbers are randomly selected from those used in the deterministic SFT implementation. Therefore, lower values of "randomScale" will result in using fewer prime numbers, decreasing the number of function samples and overall runtime at the risk of a higher probability of failure. We consider values well below the code default and theoretical upper bound of 21 given in [37].

#### 3.4.1 Exactly sparse case

In the beginning, we consider the case of multivariate trigonometric polynomials with frequencies supported within hyperbolic cross index sets. We define the $d$-dimensional hyperbolic cross frequency set

$$\mathcal{H}_K^d := \left\{ \mathbf{k} \in \mathbb{Z}^d : \prod_{\ell \in [d]} \max(1, |k_\ell|) \leq \frac{K}{2} \quad \text{and} \quad \max_{\ell \in [d]} k_\ell < \frac{K}{2} \right\} \subset \mathcal{B}_K^d$$

[1] available at https://gitlab.com/grosscra/Rank1LatticeSparseFourier

[2] available at https://gitlab.com/grosscra/SublinearSparseFourierMATLAB

where the second condition ensures that $\mathcal{H}_K^d$ is of expansion $K \in \mathbb{N}$. For a given sparsity $s$, we choose $s$ many frequencies uniformly at random from $\mathcal{H}_K^d$, and we randomly draw corresponding Fourier coefficients $\hat{g}_\mathbf{k}$ from $[-1,1]+\mathrm{i}\,[-1,1]$, $|\hat{g}_\mathbf{k}| \geq 10^{-3}$. For each parameter setting, we perform the tests 100 times.

Over these tests, we will determine the success rate as the percentage of times that all frequencies were correctly identified in the output. We focus on frequency identification since this is the core issue that Algorithms 3.1 and 3.2 solve, with the coefficient estimates carrying over directly from the SFT algorithm. Moreover, with the $s$ most significant frequencies identified, any alternative method for quickly computing the corresponding Fourier coefficients (if those from $\mathcal{A}_{s,M}$ are not tolerable) can be performed. Nevertheless, see the experiments following those in Section 3.4.1.1 for examples where we compute $\ell^2$ errors in the coefficient vectors rather than just comparing frequencies.

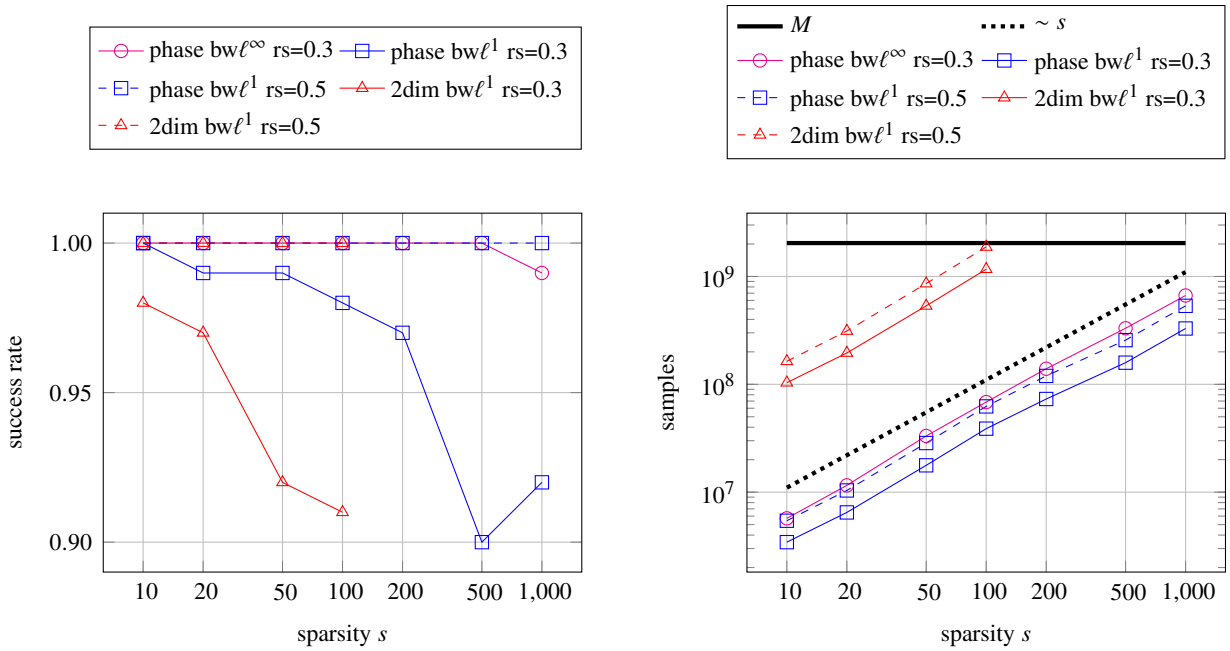
#### 3.4.1.1 Randomized frequency sets within the 10-dimensional hyperbolic cross and high-dimensional full cuboids

We set the spatial dimension $d := 10$, the expansion $K := 33$, and use $\mathcal{I} := \mathcal{H}_{33}^{10}$ as set of possible frequencies with cardinality $|\mathcal{I}| = 45\,548\,649$. Then, the rank-1 lattice with generating vector

$$\begin{aligned}\mathbf{z} :=(&1,\ 33,\ 579,\ 3\,628,\ 21\,944,\ 169\,230,\\ &1\,105\,193,\ 7\,798\,320,\ 49\,768\,670,\ 320\,144\,128)^\top\end{aligned} \tag{3.11}$$

and lattice size $M := 2\,040\,484\,044$ is a reconstructing one. We apply Algorithm 3.1 and Algorithm 3.2 with the SFT algorithm $\mathcal{A}_{2s,\tilde{M}}^{\mathrm{sub,MC}}$.

In Figure 3.4a, the success rate over 100 test runs is plotted against the sparsity values $s \in \{10, 20, 50, 100, 200, 500, 1000\}$ for Algorithm 3.1 and $s \in \{10, 20, 50, 100\}$ for Algorithm 3.2. In Figure 3.4b, the average numbers of samples over 100 tests are reported. The magenta line with circles corresponds to Algorithm 3.1 with bandwidth parameter $\tilde{M} = dKM \approx 6.7 \cdot 10^{11}$ and randomScale = 0.3. We observe that the number of samples grow nearly linearly with respect to the sparsity $s$. Moreover, the success rate is at least 0.99 (99 out of 100 test runs), where we define success such that the support of output (sparse coefficient vector) contains the true frequencies. Next, we reduce the bandwidth $\tilde{M}$ to $1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathcal{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$ (see also Remark 3.3)

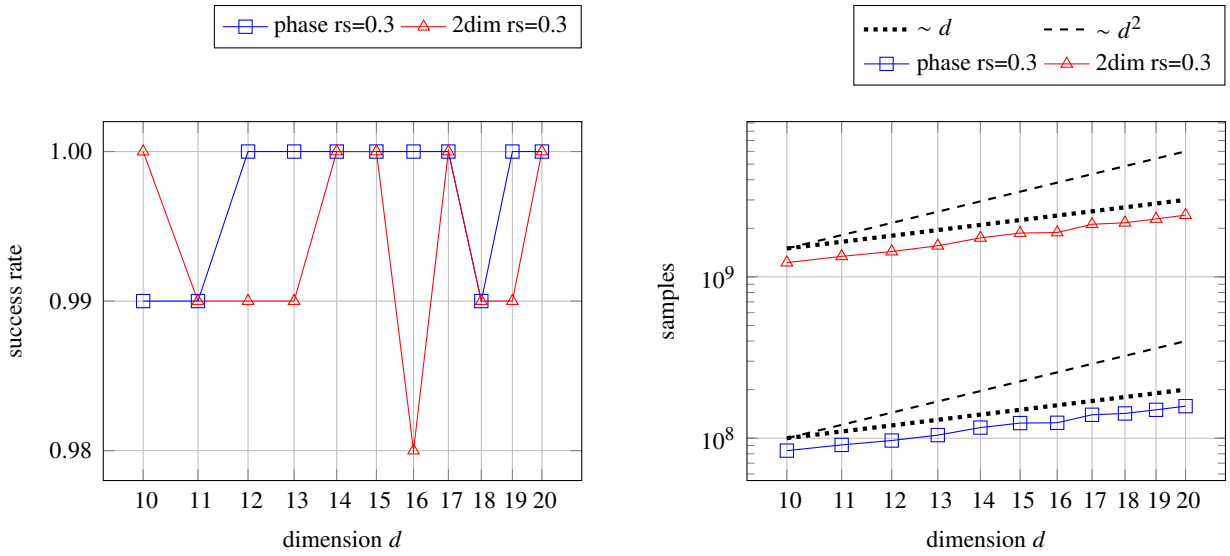



(a) Success rates vs. sparsity $s$.

(b) Samples vs. sparsity $s$.

Figure 3.4 Success rates and average number of samples over 100 test runs for Algorithm 3.1 with $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 3.2 with $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials, setting randomScale := rs. Random frequencies are chosen from hyperbolic cross $I := \mathcal{H}^{10}_{33}$. "bw$\ell^\infty$" and "bw$\ell^1$" respectively correspond to the bandwidth parameters $\tilde{M} = dKM$ with approximate value $6.7 \cdot 10^{11}$ and $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in I} \|\mathbf{k}\|_1$ with approximate value $1.6 \cdot 10^{10}$.

and visualize this as solid blue line with squares. This smaller bandwidth causes a decrease in the number of samples of up to 50 percent while only mildly decreasing the success rates to values not below 0.90. Increasing the randomScale parameter to 0.5, denoted by dashed blue line with squares, raises the success rate to 1.00 while achieving still fewer samples than bandwidth parameter $\tilde{M} = dKM \approx 6.7 \cdot 10^{11}$ and randomScale = 0.3 (solid magenta line with circles). The numbers of samples for Algorithm 3.2 are plotted as solid and dashed red lines with triangles for randomScale = 0.3 and 0.5, respectively, choosing the bandwidth $\tilde{M} := 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in I} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$. We observe that Algorithm 3.2 requires a much larger number of samples, more than one order of magnitude, compared to Algorithm 3.1, while achieving similar success rates. For comparison, in the case of sparsity $s = 100$ and randomScale = 0.5, Algorithm 3.2 takes almost $M = 2\,040\,484\,044$ samples, the number to use a non-SFT, standard rank-1 lattice FFT.




(a) Success rate vs. spatial dimension $d$.

(b) Samples vs. spatial dimension $d$.

Figure 3.5 Average number of samples over 100 test runs for Algorithm 3.1 with SFT algorithm $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 3.2 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials, setting randomScale := rs. Random frequencies are chosen from **full cuboid** of cardinality $|\mathcal{I}| \approx 10^{12}$ with lattice size $M = |\mathcal{I}|$ and bandwidth parameter $\tilde{M} = M$.

In Figure 3.5b, we investigate the dependence of the required number of samples of Algorithm 3.1 and 3.2 on the spatial dimension $d$, where we consider the values $d \in \{10, 11, \ldots, 20\}$. As before, the success rates are reported in Figure 3.5a. For this, we use a slightly different setting, where we choose $s = 100$ random frequencies from a full cuboid of cardinality $\approx 10^{12}$. Note that a cuboid with edge lengths $K_1, K_2, \ldots, K_d$ has the rank-1 lattice construction

$$\mathbf{z} = (1, K_1, K_1 \cdot K_2, \ldots, K_1 \cdot K_2 \cdots K_{d-1}) = \left( \prod_{j=[\ell]} K_j \right)_{\ell=[d]}$$

with lattice size $M = \prod_{\ell \in [d]} K_\ell = |\mathcal{I}|$. The main benefit of this construction is that the map $\mathbf{k} \mapsto \mathbf{k} \cdot \mathbf{z}$ is a bijection between $\mathcal{I}$ and $\mathcal{B}_M$. Thus, the one-dimensional bandwidth parameter $\tilde{M} = 2 \max_{\mathbf{k} \in \mathcal{I}} |\mathbf{k} \cdot \mathbf{z}| + 1$ (which is usually larger than $M$) in this case coincides with $M = |\mathcal{I}|$. By choosing cuboids in this experiment which have approximately the same cardinality, we remove any dependence on $\tilde{M}$ in our experiments, allowing us to focus on the dependence on $d$.

In our examples, the cuboids are constructed by manually tuning the edge lengths for each dimension so that the total cardinality is $\approx 10^{12}$. One way to start this procedure is by computing

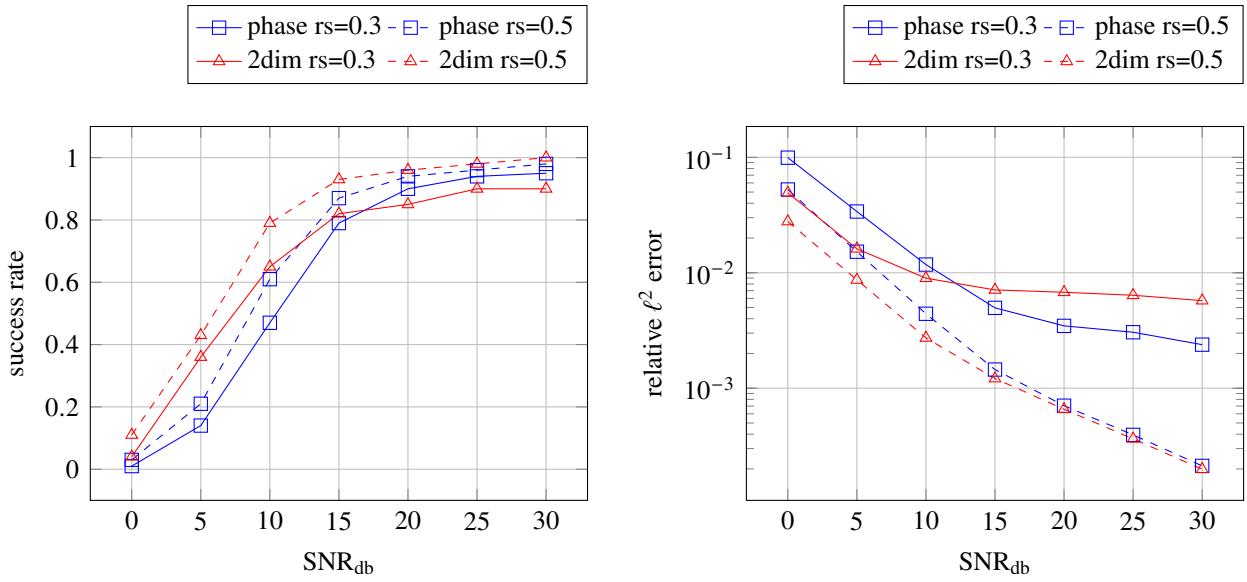
$(10^{12})^{1/d}$ and then choosing $d$ edge lengths that approximately average to this value. From here, the edge lengths can be qualitatively tweaked to arrive at a cuboid of the desired size. For instance, we utilize the cuboid $\mathcal{I} := \{-8, -7, \ldots, 7\}^9 \times \{-7, -6, \ldots, 7\}$, $|\mathcal{I}| \approx 1.03 \cdot 10^{12}$, in the case $d = 10$ and $\mathcal{I} := \{-2, -1, \ldots, 2\} \times \{-2, -1, 0, 1\}^{18} \times \{-1, 0, 1\}$, $|\mathcal{I}| \approx 1.03 \cdot 10^{12}$, for $d = 20$. Since the expansion $K$ is a factor in the number of samples of Algorithm 3.2 (cf. Corollary 3.4) and we want to concentrate on the dependence on the spatial dimension $d$, we now fix this parameter to $K := 16$ independent of $d$. Moreover, the randomScale parameter is set to 0.3. The plots indicate that the numbers of samples grow approximately linearly with respect to the dimension $d$ as stated by Corollaries 3.2 and 3.4 for Algorithms 3.1 and 3.2, respectively. The success rates are slightly better compared to the tests from Figure 3.4a.

#### 3.4.1.2 Random frequency sets within 10-dimensional hyperbolic cross and noisy samples

In this section, we again consider random multivariate trigonometric polynomials with frequencies supported within the hyperbolic cross index set $\mathcal{H}_{33}^{10}$ of expansion $K = 33$ and use the reconstructing rank-1 lattice with generating vector $\mathbf{z}$ as stated in (3.11) and size $M := 2\,040\,484\,044$. Similarly as in [43, Section 5.2], we perturb the samples of the trigonometric polynomial by additive complex (white) Gaussian noise $\varepsilon_j \in \mathbb{C}$ with zero mean and standard deviation $\sigma$. The noise is generated by $\varepsilon_j := \sigma/\sqrt{2}\,(\varepsilon_{1,j} + \mathrm{i}\varepsilon_{2,j})$ where $\varepsilon_{1,j}, \varepsilon_{2,j}$ are independent standard normal distributed. Since the signal-to-noise ratio (SNR) can be approximately computed by

$$\mathrm{SNR} \approx \frac{\sum_{j=0}^{M-1} |g(\mathbf{x}_j)|^2/M}{\sum_{j=0}^{M-1} |\varepsilon_j|^2/M} \approx \frac{\sum_{\mathbf{k}\in\mathrm{supp}(\hat{g})} |\hat{g}_\mathbf{k}|^2}{\sigma^2},$$

this leads to the choice $\sigma := \sqrt{\sum_{\mathbf{k}\in\mathrm{supp}(\hat{g})} |\hat{g}_\mathbf{k}|^2}/\sqrt{\mathrm{SNR}}$ for a targeted SNR value. The SNR is often expressed in the logarithmic decibel scale (dB), with $\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \mathrm{SNR}$ and $\mathrm{SNR} = 10^{\mathrm{SNR}_{\mathrm{dB}}/10}$, i.e., a linear $\mathrm{SNR} = 10^2$ corresponds to a logarithmic $\mathrm{SNR}_{\mathrm{dB}} = 20$ and $\mathrm{SNR} = 10^3$ corresponds to $\mathrm{SNR}_{\mathrm{dB}} = 30$. Here, our tests use sparsity $s = 100$ and signal-to-noise ratios $\mathrm{SNR}_{\mathrm{dB}} \in \{0, 5, 10, 15, 20, 25, 30\}$. Moreover, we only use the bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathcal{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$. Besides that, we choose the algorithm parameters as in Figure 3.4.




(a) success rates vs. noise level for $s = 100$

(b) relative $\ell^2$ errors vs. noise level for $s = 100$

Figure 3.6 Average success rates (all frequencies detected) and relative $\ell^2$ errors over 100 test runs for Algorithm 3.1 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 3.2 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials supported on the hyperbolic cross $\mathcal{I} := \mathcal{H}^{10}_{33}$, setting randomScale := rs $\in \{0.3, 0.5\}$ and using the bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathcal{I}} \|\mathbf{k}\|_1$ with approximate value $1.6 \cdot 10^{10}$.

In Figure 3.6a, we visualize the success rates depending on the noise level. For randomScale $\in \{0.3, 0.5\}$ and both algorithms, the success rates start at less than 0.12 for $\mathrm{SNR}_{\mathrm{dB}} = 0$ and grow for increasing signal-to-noise ratios until at least 0.90 for $\mathrm{SNR}_{\mathrm{dB}} = 30$. The success rates of Algorithm 3.2 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ ("2dim") are often higher than for Algorithm 3.1 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ ("phase"), which may be caused by the larger numbers of samples for Algorithm 3.2 and the noise model used. Note that the numbers of samples correspond to those in Figure 3.4b for $s = 100$ independent of the noise level. For Algorithm 3.2 with randomScale $= 0.3$, the increase of the success rate seems to stagnate at $\mathrm{SNR}_{\mathrm{dB}} = 20$, while this does not seem to be the case for randomScale $= 0.5$ or Algorithm 3.1. In particular, this behavior can also be observed in Figure 3.6b, where we plot the average relative $\ell^2$ error of the Fourier coefficients against the signal-to-noise ratio. Here, we observe that for randomScale $= 0.3$, the decrease of the errors for increasing $\mathrm{SNR}_{\mathrm{dB}}$ values almost stops once reaching $\mathrm{SNR}_{\mathrm{dB}} = 20$ for both algorithms. Initially, the average error of Algorithm 3.2 is smaller, but at $\mathrm{SNR}_{\mathrm{dB}} = 15$ and higher, the average error of Algorithm 3.1 is smaller. In case

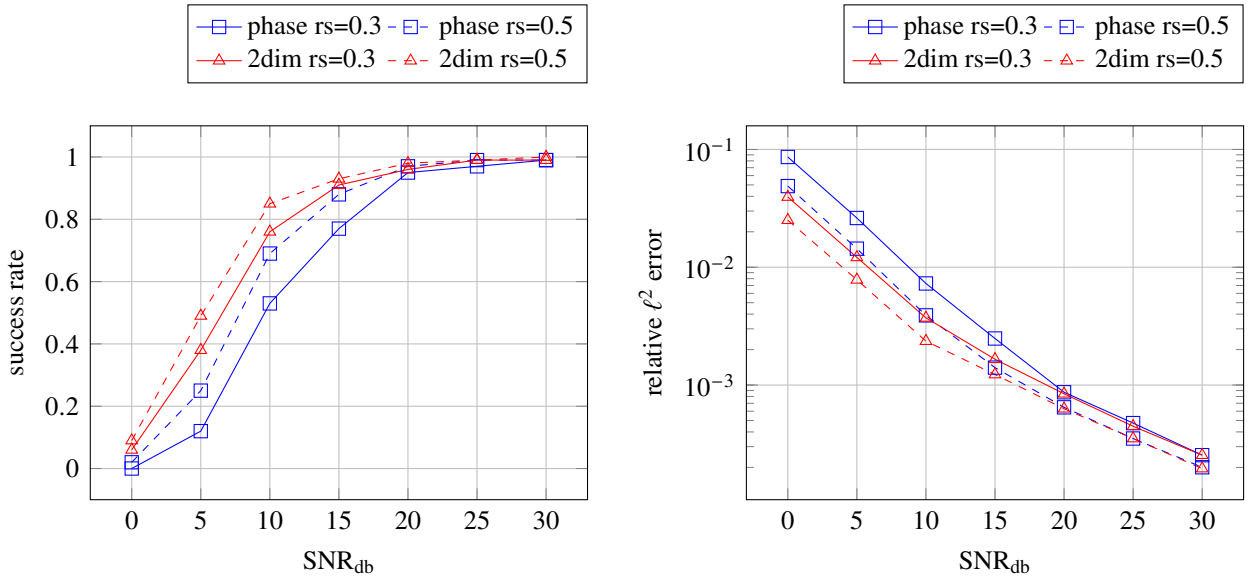
of randomScale $= 0.5$, we observe a distinct decrease for growing signal-to-noise ratios for both algorithms.

#### 3.4.1.3 Deterministic frequency set within 10-dimensional hyperbolic cross and noisy samples

Next, instead of randomly chosen frequencies, we now consider frequencies on a $d$-dimensional weighted hyperbolic cross

$$\mathcal{H}_K^{d,\alpha} := \left\{ \mathbf{k} \in \mathbb{Z}^d : \prod_{\ell \in [d]} \max(1, (\ell+1)^\alpha \, |k_\ell|) \leq \frac{K}{2} \quad \text{and} \quad \max_{\ell \in [d]} k_\ell < \frac{K}{2} \right\}.$$

Here, we use $d = 10$, $K = 33$, $\mathcal{I} := \mathcal{H}_{33}^{10}$, and $\alpha = 1.7$, which yields $s = |\mathcal{H}_{33}^{10,1.7}| = 101$. Again, the Fourier coefficients $\hat{g}_\mathbf{k}$ are randomly chosen from $[-1,1] + \mathrm{i}\,[-1,1]$, $|\hat{g}_\mathbf{k}| \geq 10^{-3}$. We use the same lattice and bandwidth parameter as in the last subsection as well as the same noise model and parameters.




(a) success rates vs. noise level for $s = 100$

(b) relative $\ell^2$ errors vs. noise level for $s = 100$

Figure 3.7 Average success rates (all frequencies detected) and relative $\ell^2$ errors over 100 test runs for Algorithm 3.1 with $\mathcal{A}_{2s,\tilde{M}}^{\mathrm{sub,MC}}$, denoted by "phase", and Algorithm 3.2 with $\mathcal{A}_{2s,\tilde{M}}^{\mathrm{sub,MC}}$, denoted by "2dim", on multivariate trigonometric polynomials with (deterministic) frequencies on weighted hyperbolic cross within hyperbolic cross $\mathcal{I} := \mathcal{H}_{33}^{10}$, setting randomScale := rs $\in \{0.3, 0.5\}$ and bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k} \in \mathcal{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$.

In Figure 3.7, we depict the obtained results. In particular, the results in Figure 3.7a are very

similar to the ones for randomly chosen frequencies in Figure 3.6a. For the case of deterministic frequencies in Figure 3.7a, the success rates are slightly better. Moreover, we do not observe the "stagnation" of the success rates for Algorithm 3.2 with randomScale = 0.3. Correspondingly, the relative $\ell^2$ errors, as shown in Figure 3.7b, decrease distinctly for growing signal-to-noise ratios. Algorithm 3.2 performs slightly better than Algorithm 3.1, but also requires more than one order of magnitude more samples, similar to the results shown in Figure 3.4b for $s = 100$.

### 3.4.2 Compressible case in 10 dimensions

In this section, we apply the methods on a test function which is not exactly sparse but compressible. In addition, we also consider noisy samples as in Section 3.4.1.2. We use the 10-variate periodic test function $g : \mathbb{T}^{10} \to \mathbb{R}$,

$$g(\mathbf{x}) := \prod_{\ell \in \{0,2,7\}} K_2(x_\ell) + \prod_{\ell \in \{1,4,5,9\}} K_4(x_\ell) + \prod_{\ell \in \{3,6,8\}} K_6(x_\ell), \tag{3.12}$$

from [59, Section 3.3] and [43, Section 5.3] which has infinitely many non-zero Fourier coefficients $\hat{g}_{\mathbf{k}}$, where $K_m : \mathbb{T} \to \mathbb{R}$ is the B-Spline of order $m \in \mathbb{N}$,

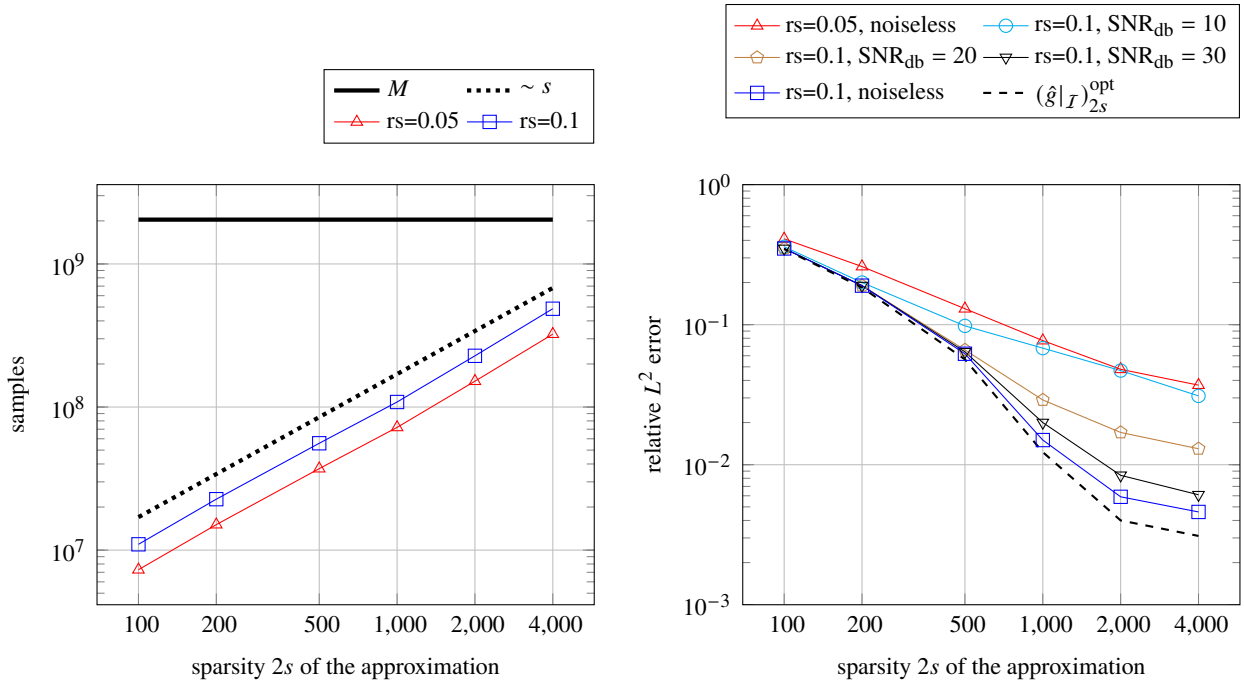
$$K_m(x) := C_m \sum_{k \in \mathbb{Z}} \operatorname{sinc}\left(\frac{\pi}{m} k\right)^m (-1)^k \, \mathrm{e}^{2\pi \mathrm{i} k x},$$

with a constant $C_m > 0$ such that $\|K_m\|_{L^2(\mathbb{T})} = 1$. We remark that each B-Spline $K_m$ of order $m \in \mathbb{N}$ is a piece-wise polynomial of degree $m - 1$. We apply Algorithm 3.1 with $\mathcal{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ and use the sparsity parameters $s \in \{50, 100, 250, 500, 1000, 2000\}$, which corresponds to $2s \in \{100, 200, 500, 1000, 2000, 4000\}$-many frequencies and Fourier coefficients for the output of Algorithm 3.1. We use the frequency set $\mathcal{I} := \mathcal{H}^{10}_{33}$ and randomScale := rs $\in \{0.05, 0.1\}$. Moreover, we work with the same rank-1 lattice as in Section 3.4.1.2.

The obtained basis index sets $\operatorname{supp}(\hat{\mathbf{g}}^s)$ should "consist of" the union of three lower dimensional manifolds, a three-dimensional hyperbolic cross in the dimensions 1, 3, 8; a four-dimensional hyperbolic cross in the dimensions 2, 5, 6, 10; and a three-dimensional hyperbolic cross in the dimensions 4, 7, 9. All tests are performed 100 times and the relative $L^2$ approximation error

$$\frac{\|g - g^s\|_{L^2}}{\|g\|_{L^2}} = \frac{\sqrt{\|g\|^2_{L^2} - \sum_{\mathbf{k} \in \operatorname{supp}(\hat{\mathbf{g}}^s)} |\hat{g}_{\mathbf{k}}|^2 + \sum_{\mathbf{k} \in \operatorname{supp}(\hat{\mathbf{g}}^s)} |\hat{g}^s_{\mathbf{k}} - \hat{g}_{\mathbf{k}}|^2}}{\|g\|_{L^2}}$$

is computed each time.




(a) samples vs. sparsity $2s$

(b) relative $L^2$ errors vs. sparsity $2s$

Figure 3.8 Average number of samples and relative $L^2$ errors over 100 test runs for Algorithm 3.1 with $\mathcal{A}^{\text{sub,MC}}_{2s,\tilde{M}}$ on 10-dimensional test function (3.12) consisting of tensor products of B-Splines of different order. Search space is unweighted hyperbolic cross $\mathcal{I} := \mathcal{H}^{10}_{33}$ with SFT parameters randomScale := rs $\in \{0.05, 0.1\}$ and $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathcal{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$.

In Figure 3.8a, we visualize the average number of samples against the sparsity $2s$ of the approximation. We observe an almost linear increase with respect to $2s$. In Figure 3.8b, we show the average relative errors for randomScale $\in \{0.05, 0.1\}$ in the noiseless case as well as randomScale $= 0.1$ for $\text{SNR}_{\text{db}} \in \{10, 20, 30\}$. In general, for increasing sparsity, the errors become smaller. For randomScale $= 0.05$ in the noiseless case and randomScale $= 0.1$ with $\text{SNR}_{\text{db}} = 10$, the average error is similar and stays above $3 \cdot 10^{-2}$ even for sparsity $2s = 4000$. For higher signal-to-noise ratio, the error decreases further. For $\text{SNR}_{\text{db}} = 30$, the obtained average error is $6.1 \cdot 10^{-3}$ for $2s = 4000$, which is only approximately twice as high as the best possible error when using the $2s$ largest (by magnitude) Fourier coefficients $\hat{g}_{\mathbf{k}}$ with the restriction $\mathbf{k} \in \mathcal{I} := \mathcal{H}^{10}_{33}$. The latter is plotted in Figure 3.8b as dashed line without markers.

# CHAPTER 4

# SPARSE FOURIER SPECTRAL METHODS FOR SOLVING PDE

As discussed in Section 1.1.3, this chapter focuses on a sparse spectral method for solving elliptic PDEs. We begin with a review of the literature on sparse spectral methods against which we motivate our results in Section 4.1. Section 4.2 gives the advection-diffusion-reaction PDE setup and Section 4.3 converts this problem to its Galerkin representation underpinning the spectral method approach. The following three sections provide the ingredients outlined by Strang's lemma, Lemma 1.1, in Section 1.1.3:

1. a Fourier series truncation method for the solution and the resulting error analysis (Section 4.4),
2. a (sparse) Fourier series approximation technique (Section 4.5), and
3. a version of Strang's lemma that ties everything together (Section 4.6).

We close with a numerics section, Section 4.7, describing the implementation of our technique and a variety of numerical experiments demonstrating the theory.

## 4.1 Overview of results and prior work

We now outline some of the previous literature on spectral methods with an emphasis on exploiting sparsity. Along the way, various shortcomings will arise, and we will use these as opportunities to motivate and explain our approach in the sequel.

### 4.1.1 Prior attempts to relieve dependence on bandwidth via SFT-type methods

A key work pioneering the use of SFTs in computing solutions to PDEs is due to Daubechies, et al. [21]. This work mostly focuses on time-dependent, one-dimensional problems where the spectral scheme is formulated as alternating Fourier-projections and time-steps. Thus, there is no need to impose an a priori Fourier basis truncation on the solution. The proposed projection step instead utilizes an SFT at each time step to adaptively retain the most significant frequencies throughout the time-stepping procedure. Time-independent problems like (1.3) can then be handled by stepping in time until a stationary solution is obtained.

A simplified form of this algorithm is shown to succeed numerically in [21], and it is also analyzed theoretically in the case where the diffusion coefficient consists of a known, fine-scale mode superimposed over lower frequency terms. There, the Fourier-projection step can be considered to be fixed. However, removing the known fine-scale assumption leads to many difficulties, including the possibility of sparsity-induced omissions in early time steps cascading into larger errors later on. In this chapter, on the other hand, we focus on the case of time-independent problems. This allows us to utilize SFTs only once initially. By doing so we avoid the possibility of SFT-induced error accumulation over many time steps. The main difficulty in our analysis then becomes determining how the Fourier-sparse representations of the PDE data discovered by high-dimensional SFTs can be used to rapidly find a suitable Fourier representation of the solution. This takes the form of mixing the Fourier supports the data into *stamping sets* (discussed in detail in Section 4.4) on which we can analyze the projection error of the solution. In fact, these stamping sets can be viewed as a modification and generalization of the techniques used in the one-dimensional and known fine-scale analysis from [21].

### 4.1.2 Attempts to relieve curse of dimensionality

Many attempts to overcome the curse of dimensionality in Fourier spectral methods for PDE have focused on using basis truncations which allow for an efficient high-dimensional Fourier transform. One of the most popular techniques is the sparse grid spectral method, which computes Fourier coefficients on the hyperbolic cross [47, 11, 29, 30, 63, 31, 20]. In general, a sparse grid method reduces the number of sampling points necessary to approximate the PDE data to $O(K \log^{d-1}(K))$, where $K$ acts as a type of bandwidth parameter. Algorithms to compute spectral representations using these sparse sampling grids run with similar complexity. When used in conjunction with spectral methods for solving PDE, these sparse grid Fourier transforms produce solution approximations with error estimates similar to the full $d$-dimensional FFT-versions reduced by factors only on the order of $1/\log^{d-1}(K)$.

In the context of sparse grid Fourier transforms, these methods compute Fourier coefficients with frequencies indexed on hyperbolic crosses of similar cardinality to the number of sampling

points. These hyperbolic crosses have intimate links with the space of bounded mixed derivative, in the sense that they are the optimal Fourier-approximation spaces for this class. Thus, sparse grid Fourier spectral methods are particularly apt for problems where the solution is of bounded mixed derivative, as this produces an optimal $u - u^{\text{truncation}}$ term in Lemma 1.1 above.

Though sparse-grid spectral methods can efficiently solve a variety of high-dimensional problems, there are clear downsides for the types of problems we target in this chapter. While many problems fit the bounded mixed derivative assumption [67, 68], and therefore have accurate Fourier representations on the hyperbolic cross, the multiscale, Fourier-sparse problems that we are interested in are especially problematic. In fact, since a hyperbolic cross of bandwidth $K$ contains only those frequencies $\mathbf{k} \in \mathbb{Z}^d$ with $\prod_{i\in[d]} |k_i| = O(K)$, $d$-dimensional frequencies active in all dimensions can have only $\|\mathbf{k}\|_\infty = O(K^{1/d})$. Thus, in a multiscale problem with even one frequency that interacts in all dimensions, a hyperbolic cross is required with a bandwidth exponential in $d$ to properly resolve the data. This then forces the traditionally curse-of-dimensionality-mitigating $\log^{d-1}(K)$ terms characteristic of sparse grid methods to be at least on the order of $d^{d-1}$.

### 4.1.3 More on high-dimensional Fourier transforms

As outlined in Section 4.1.1 above, this chapter uses sparse Fourier transforms to create an adaptive basis truncation suited to the PDE data. This mimics a similar evolution in the field of high-dimensional Fourier transforms from sparse grids to more flexible techniques [52, 22, 55, 49, 31, 50, 56, 34]. In particular, the rank-1 lattice based approaches for high-dimensional Fourier transforms discussed in Chapters 2 and 3 originate from a link between early high-dimensional quadrature techniques and Fourier approximations on the hyperbolic cross [49, 50].

Though many rank-1 lattice approaches take $\mathcal{I}$ to be the hyperbolic cross to leverage the well-studied regularity properties and cardinality bounds similarly enjoyed in the sparse-grid literature, rank-1 lattice results are available for arbitrary frequency sets. The computationally efficient extension of these techniques via sparse Fourier transforms in Chapter 3 as well as the randomization trick presented in Section 4.5 take this frequency set flexibility to its limit, allowing $\mathcal{I}$ to be the a priori unknown set of the most important Fourier coefficients of the function to be approximated.

This again suggests the applicability of these methods over sparse grid (or other non-sparsity exploiting) Fourier transforms in the context of multiscale problems involving even a small number of Fourier coefficients in extremely high dimensions.

### 4.1.4 Additional links to compressive sensing

As discussed above, the SFT literature overlaps considerably with the language and techniques of compressive sensing. As previously detailed in Chapter 3, the high-dimensional SFT we use herein provides error bounds with best $s$-term approximation, compressive-sensing-type error guarantees [19]. As a result, the Fourier coefficients of the PDE data are approximated with errors depending on the compressibility of their true Fourier series, and then the compressibility of the PDE's solution in the Fourier basis is inferred from the Fourier compressibility of the data in a direct and constructive fashion.

Another very successful line of work, however, aims to more directly apply standard compressive sensing reconstruction methods to the general spectral method framework for solving PDEs. Referred to as CORSING [9, 4, 10, 8, 6], these techniques use compressed sensing concepts to recover a sparse representation of the solution to the system of equations derived from the (Petrov-)Galerkin formulation of a PDE. These methods have been further extended to the case of pseudospectral methods in [5], in which a simpler-to-evaluate matrix equation is subsampled and used as measurements for a compressive sensing algorithm (as an aside, [5] and discussions with the author served as a primary inspiration for the results in this chapter). This compressive spectral collocation method works by finding the largest Fourier-sine coefficients of the solution with frequencies in the integer hypercube with bandwidth $K$ by applying Orthogonal Matching Pursuit (OMP) on a set of samples of the PDE data. By using OMP, the method is able to succeed with measurements on the order of $O(d\exp(d)s\log^3(s)\log(K))$ where $s$ is the imposed sparsity level of the solution's Fourier series. Thus, while the $O(K^d)$ dependence from a traditional Fourier (pseudo)spectral method is avoided and the method adapts well to large bandwidths, the curse of dimensionality is still apparent.

Recently, an improvement on [5] that addresses the curse of dimensionality was made avail-

able which is therefore well-suited for similar types of problems discussed in this chapter. In [66], the approach of approximating Fourier-sine coefficients on a full hypercube is replaced with approximating Fourier coefficients on a hyperbolic cross. This has the effect of converting the linear dependence on $d$ in the sampling complexity to a $\log(d)$ due to cardinality estimates of the hyperbolic cross. However, the $\exp(d)$ term is refined using a different technique. The key theoretical ingredient for being able to apply compressive sensing to these problems is bounding the Riesz constants of the basis functions that result after applying the differential operator [6]. A careful estimation of these constants on the Fourier basis indexed by a hyperbolic cross is able to entirely remove the exponential in $d$ dependence, leading to a sampling complexity on the order of $\mathcal{O}(C_a s \log(d) \log^3(s) \log(K))$, where $C_a$ involves terms depending on ellipticity and compressibility properties of $a$. Notably, this estimation procedure has connections to our stamping set techniques described in Section 4.4.

On the other hand, though focusing on the hyperbolic cross in compressive spectral collocation breaks the curse of dimensionality in the sampling complexity, the method still suffers from the inability to generalize to multiscale problems or generic frequency sets of interest like those described in Section 4.1.2. Additionally, as mentioned in Section 4.1.4, the compressive-sensing algorithm used for recovery (in this case OMP) suffers from a computational complexity on the order of the cardinality of the truncation set of interest. For the hyperbolic cross, this is still exponential in $\log(d)$. Finally, the error estimates are presented in terms of the compressibility of the Fourier series of the solution $u$, which may not be known a priori from the PDE data. We expect that there may be some way to link our stamping theory and convergence estimates with the compressive sensing theory to refine and generalize both approaches.

## 4.2 Elliptic PDE setup

We begin with a model elliptic partial differential equation.

**Definition 4.1.** For some $a : \mathbb{T}^d \to \mathbb{R}$, $\mathbf{b} : \mathbb{T}^d \to \mathbb{R}^d$, $c : \mathbb{T}^d \to \mathbb{R}$ sufficiently smooth, define the *advection-diffusion-reaction operator in divergence form* $\mathcal{L}$ by

$$\mathcal{L}u = -\nabla \cdot (a\nabla u) + \mathbf{b} \cdot \nabla u + cu.$$

If for some $f : \mathbb{T}^d \to \mathbb{R}$ sufficiently smooth, $u \in C^2$ satisfies

$$\mathcal{L}u = f, \tag{SF}$$

we say that *u solves the given PDE with periodic boundary conditions in the strong form*.

Now, after multiplying by the complex conjugate of a test function $v \in H^1(\mathbb{T}^d)$ and integrating the first term by parts, we define the sesquilinear form associated to $\mathcal{L}$ as $\mathfrak{L} : H^1 \times H^1 \to \mathbb{C}$ with

$$\mathfrak{L}(u, v) := \int_{\mathbb{T}^d} a(\mathbf{x})\nabla u(\mathbf{x}) \cdot \overline{\nabla v}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) \cdot \nabla u(\mathbf{x})\overline{v}(\mathbf{x}) + c(\mathbf{x})u(\mathbf{x})\overline{v}(\mathbf{x})\, d\mathbf{x},$$

and we say that $u \in H^1$ *solves the given PDE with periodic boundary conditions in the weak form* if

$$\mathfrak{L}(u, v) = \langle f, v\rangle_{L^2} \quad \text{for all } v \in H^1. \tag{WF}$$

For our purposes, we will take $a, c \in L^\infty(\mathbb{T}^d; \mathbb{R})$, $\mathbf{b} \in L^\infty(\mathbb{T}^d; \mathbb{R})^d$ (i.e., each coordinate of the advection field is in $L^\infty$), and $f \in L^2(\mathbb{T}^d; \mathbb{R})$.

By the conditions specified in the Lax-Milgram theorem (see, e.g., [23]), we are guaranteed that a unique solution to (WF) exists. We use the formulation as stated in [8, Proposition 2.1] and proven in [7].

**Proposition 4.1.** *For $a, c \in L^\infty(\mathbb{T}^d; \mathbb{R})$, $\mathbf{b} \in L^\infty(\mathbb{T}^d; \mathbb{R})^d$, $\mathfrak{L}$ is continuous with continuity constant*

$$\beta \le \max\left\{\|a\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \|\mathbf{b}(\mathbf{x})\|_2, \|c\|_{L^\infty}\right\},$$

*that is*

$$|\mathfrak{L}(u, v)| \le \beta\|u\|_{H^1}\|v\|_{H^1} \qquad \text{for all } u, v \in H^1. \tag{4.1}$$

*Additionally, assuming $\mathbf{b} \in H^1(\mathbb{T}^d; \mathbb{R})^d$, if $a(\mathbf{x}) \ge a_{\min} > 0$ and $-\frac{1}{2}\nabla \cdot \mathbf{b}(\mathbf{x}) + c(\mathbf{x}) \ge d_{\min} > 0$ a.e. on $\mathbb{T}^d$, then $\mathfrak{L}$ is also coercive with coercivity constant*

$$\alpha \ge \min\{a_{\min}, d_{\min}\},$$

*that is*

$$|\mathfrak{L}(u, u)| \ge \alpha\|u\|_{H^1}^2 \qquad \text{for all } u \in H^1. \tag{4.2}$$

*Under conditions* (4.1) *and* (4.2)*, if $f \in L^2(\mathbb{T}^d; \mathbb{R})$ then* (WF) *has unique solution $u \in H^1$ satisfying*

$$\|u\|_{H^1} \le \frac{\|f\|_{L^2}}{\alpha}. \tag{4.3}$$

### 4.3 Galerkin spectral methods

By Theorem 1.1, it is equivalent to replace the weak PDE (WF) by

$$\mathfrak{L}(u, \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}) = \langle f, \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}\rangle_{L^2} =: \hat{f}_{\mathbf{k}} \quad \text{for all } \mathbf{k} \in \mathbb{Z}^d.$$

Rewriting the sesquilinear form on the left-hand side and using the Fourier series representations of $a$, $\mathbf{b}$ (where we collect all coordinates' Fourier coefficients at a given frequency $\mathbf{k} \in \mathbb{Z}^d$ into the vectors $\hat{\mathbf{b}}_{\mathbf{k}} \in \mathbb{C}^d$), $c$, and $u$, we obtain

$$\begin{aligned}
\mathfrak{L}(u, \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}) &= \sum_{\mathbf{l}_1,\mathbf{l}_2\in\mathbb{Z}^d} \hat{a}_{\mathbf{l}_1}\hat{u}_{\mathbf{l}_2} \int_{\mathbb{T}^d} \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_1\cdot\mathbf{x}} \nabla \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_2\cdot\mathbf{x}} \cdot \overline{\nabla \mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\mathbf{x}}}\, d\mathbf{x} \\
&\quad + \sum_{\mathbf{l}_1,\mathbf{l}_2\in\mathbb{Z}^d} \hat{u}_{\mathbf{l}_2} \int_{\mathbb{T}^d} \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_1\cdot\mathbf{x}} \hat{\mathbf{b}}_{\mathbf{l}_1} \cdot \nabla \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_2\cdot\mathbf{x}} \overline{\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\mathbf{x}}}\, d\mathbf{x} \\
&\quad + \sum_{\mathbf{l}_1,\mathbf{l}_2\in\mathbb{Z}^d} \hat{c}_{\mathbf{l}_1}\hat{u}_{\mathbf{l}_2} \int_{\mathbb{T}^d} \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_1\cdot\mathbf{x}} \mathrm{e}^{2\pi\mathrm{i}\mathbf{l}_2\cdot\mathbf{x}} \overline{\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\mathbf{x}}}\, d\mathbf{x} \\
&= \sum_{\mathbf{l}_1,\mathbf{l}_2\in\mathbb{Z}^d} \delta_{\mathbf{l}_1,\mathbf{k}-\mathbf{l}_2} \left[ (2\pi)^2 (\mathbf{l}_2 \cdot \mathbf{k}) \hat{a}_{\mathbf{l}_1} + 2\pi\mathrm{i} \left( \hat{\mathbf{b}}_{\mathbf{l}_1} \cdot \mathbf{l}_2 \right) + \hat{c}_{\mathbf{l}_1} \right] \hat{u}_{\mathbf{l}_2} \\
&= \sum_{\mathbf{l}\in\mathbb{Z}^d} \left[ (2\pi)^2 (\mathbf{l} \cdot \mathbf{k}) \hat{a}_{\mathbf{k}-\mathbf{l}} + 2\pi\mathrm{i} \left( \hat{\mathbf{b}}_{\mathbf{k}-\mathbf{l}} \cdot \mathbf{l} \right) + \hat{c}_{\mathbf{k}-\mathbf{l}} \right] \hat{u}_{\mathbf{l}} \\
&=: (L\hat{u})_{\mathbf{k}},
\end{aligned}$$

where $L$ is an operator in $\ell^2$. This leads to the *Galerkin form* of our PDE,

$$L\hat{u} = \hat{f}. \tag{GF}$$

The computational advantages of (GF) are clear. By numerically approximating $\hat{a}$, $\hat{\mathbf{b}}$, $\hat{c}$ and $\hat{f}$ (which automatically truncates $L$), we arrive at a discretized, finite system of equations that can be solved for the Fourier coefficients of our solution.

We will use a fast sparse Fourier transform (SFT) for functions of many dimensions to approximate our PDE data which then leads to a sparse system of equations that we can quickly solve to approximate $\hat{u}$. This SFT will use the values of $a$, $\mathbf{b}$, $c$ and $f$ at equispaced nodes on a randomized rank-1 lattice in $\mathbb{T}^d$, and therefore, our technique is effectively a pseudospectral method where the discretization of the solution space $\{\hat{u} \mid u \in h\}$ is adapted to the PDE data.

Before we move to the detailed discussion of this SFT, we provide a more detailed analysis of the Galerkin operator in Section 4.4 to help us analyze the resulting spectral method. But first, we note that $L$ also captures the behavior of $\mathfrak{L}$ as a sesquilinear form.

**Proposition 4.2.** *For $\hat{u}, \hat{v} \in \ell^2$ with $u, v \in H^1$,*

$$\mathfrak{L}(u, v) = \langle L\hat{u}, \hat{v} \rangle_{\ell^2}.$$

*Proof.* By the Fourier series representation of $v$,

$$\mathfrak{L}(u, v) = \sum_{\mathbf{k} \in \mathbb{Z}^d} \mathfrak{L}(u, \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ}) \overline{\hat{v}}_{\mathbf{k}} = \sum_{\mathbf{k} \in \mathbb{Z}^d} (L\hat{u})_{\mathbf{k}} \, \overline{\hat{v}}_{\mathbf{k}} = \langle L\hat{u}, \hat{v} \rangle_{\ell^2}. \qquad \square$$

### 4.4 Stamping sets and truncation analysis

Notably, (GF) gives us insight into the frequency support of $\hat{u}$. The structure outlined in the following proposition is crucial in constructing a fast spectral method that exploits Fourier-sparsity.

**Proposition 4.3.** *Given $\hat{a}$, $\hat{\mathbf{b}}$, and $\hat{c}$, the Fourier coefficients of the diffusion coefficient, the advection field, and the reaction coefficient of an ADR equation respectively, denote the set of "active" frequencies*

$$\mathcal{A} := \operatorname{supp}(\hat{a}) \cup \left( \bigcup_{j \in [d]} \operatorname{supp}\left(\hat{b}_j\right) \right) \cup \operatorname{supp}(\hat{c}) \subset \mathbb{Z}^d.$$

*For any set $\mathcal{F} \subset \mathbb{Z}^d$ and $N \in \mathbb{N}_0$, recursively define the sets*

$$\begin{aligned} \mathcal{S}^N[\mathcal{A}](\mathcal{F}) &:= \begin{cases} \mathcal{F} & \text{if } N = 0 \\ \mathcal{S}^{N-1}[\mathcal{A}](\mathcal{F}) + \mathcal{A} & \text{if } N > 0 \end{cases}, \\ \mathcal{S}^{\infty}[\mathcal{A}](\mathcal{F}) &:= \bigcup_{N=0}^{\infty} \mathcal{S}^N[\mathcal{A}](\mathcal{F}), \end{aligned} \tag{4.4}$$

*where here, addition is defined as the Minkowski sum of sets. Under the conditions of Proposition 4.1,* $\operatorname{supp}(\hat{u}) \subset \mathcal{S}^{\infty}[\mathcal{A}](\operatorname{supp}(\hat{f}))$.

*Proof.* Note first that the fact that $a$, $\mathbf{b}$, and $c$ are real imply the supports of their Fourier series are "rotationally" symmetric in $\mathbb{Z}^d$, e.g., $\operatorname{supp}(\hat{a}) = -\operatorname{supp}(\hat{a})$. Now, we show that $L_{\mathbf{k},\mathbf{k}} \neq \mathbf{0}$ for all $\mathbf{k} \in \mathbb{Z}^d$. Recall that

$$L_{\mathbf{k},\mathbf{k}} := (2\pi)^2 (\mathbf{k} \cdot \mathbf{k}) \hat{a}_{\mathbf{0}} + 2\pi \mathrm{i} \left( \hat{\mathbf{b}}_{\mathbf{0}} \cdot \mathbf{k} \right) + \hat{c}_{\mathbf{0}}.$$

It suffices to show that $\hat{a}_{\mathbf{0}}$ and $\hat{c}_{\mathbf{0}}$ are strictly positive as the middle term will always be purely imaginary. Since $a$ is always strictly positive under the assumptions of Proposition 4.1, its mean $\hat{a}_{\mathbf{0}}$ is necessarily strictly positive. As for $c$, the conditions of Proposition 4.1 require

$$-\frac{1}{2}\nabla \cdot \mathbf{b} + c > 0$$

which implies

$$\hat{c}_{\mathbf{0}} > \frac{1}{2}\int_{\mathbb{T}^d} \nabla \cdot \mathbf{b}(\mathbf{x})\, d\mathbf{x}.$$

However, the divergence theorem implies that the right hand side is zero, and therefore $\hat{c}_{\mathbf{0}}$ is positive as desired.

Now, since $\mathbf{L}_{\mathbf{k},\mathbf{k}}$ is nonzero, we may rearrange the equality $(L\hat{u})_{\mathbf{k}} = \hat{f}_{\mathbf{k}}$ to obtain

$$\hat{u}_{\mathbf{k}} = \frac{\hat{f}_{\mathbf{k}} - \sum_{\mathbf{l}\in(\{\mathbf{k}\}+\mathcal{A})\setminus\{\mathbf{k}\}} L_{\mathbf{k},\mathbf{l}}\hat{u}_{\mathbf{l}}}{L_{\mathbf{k},\mathbf{k}}},$$

where we have restricted the summation to only those frequencies where the entries of row $\mathbf{k}$ of $L$ are nonzero, that is, the active frequencies of the PDE data translated by $\mathbf{k}$. Thus, $\hat{u}_{\mathbf{k}}$ explicitly depends only on the values of $\hat{u}$ on $\mathcal{S}^1[\mathcal{A}](\{\mathbf{k}\}) \setminus \{\mathbf{k}\}$, which themselves then depend only on values of $\hat{u}$ on $\mathcal{S}^2[\mathcal{A}](\{\mathbf{k}\})$, and so on. This decouples the system of equations $L\hat{u}$ into a disjoint collection of systems of equations, one for each class of frequencies $\mathcal{S}^\infty[\mathcal{A}](\{\mathbf{k}\})$. Since Proposition 4.1 implies that $\hat{v} = 0$ is the unique solution of $L\hat{v} = 0$, the unique solution of the system of equations for $\hat{u}$ on $\mathcal{S}^\infty[\mathcal{A}](\{\mathbf{k}\})$ for any $\mathbf{k} \notin \operatorname{supp}(\hat{f})$ is $\hat{u}|_{\mathcal{S}^\infty[\mathcal{A}](\{\mathbf{k}\})} = 0$. Therefore, $\operatorname{supp}(\hat{u}) \subset \mathcal{S}^\infty[\mathcal{A}](\operatorname{supp}(\hat{f}))$ as desired. □

In what follows, when the set $\mathcal{F}$ (often $\operatorname{supp}(\hat{f})$) and set of active frequencies $\mathcal{A}$ are clear from context, we suppress them in the notation given by (4.4) so that $\mathcal{S}^N := \mathcal{S}^N[\mathcal{A}](\mathcal{F})$. Intuitively, we can imagine constructing $\mathcal{S}^N$ by first creating a "rubber stamp" in the shape of $\mathcal{A}$. This rubber stamp is then stamped onto every frequency in $\mathcal{F} =: \mathcal{S}^0$ to construct $\mathcal{S}^1$. Then, this process is repeated, stamping each element of $\mathcal{S}^1$ to produce $\mathcal{S}^2$, and so on. For this reason, we will colloquially refer to these as "stamping sets." Figure 4.1 gives an example of this stamping procedure for $d = 2$.

A key approach of our further analysis will be analyzing the decay of $\hat{u}$ on successive stamping levels. The stamping level will become the driving parameter in the spectral method rather than

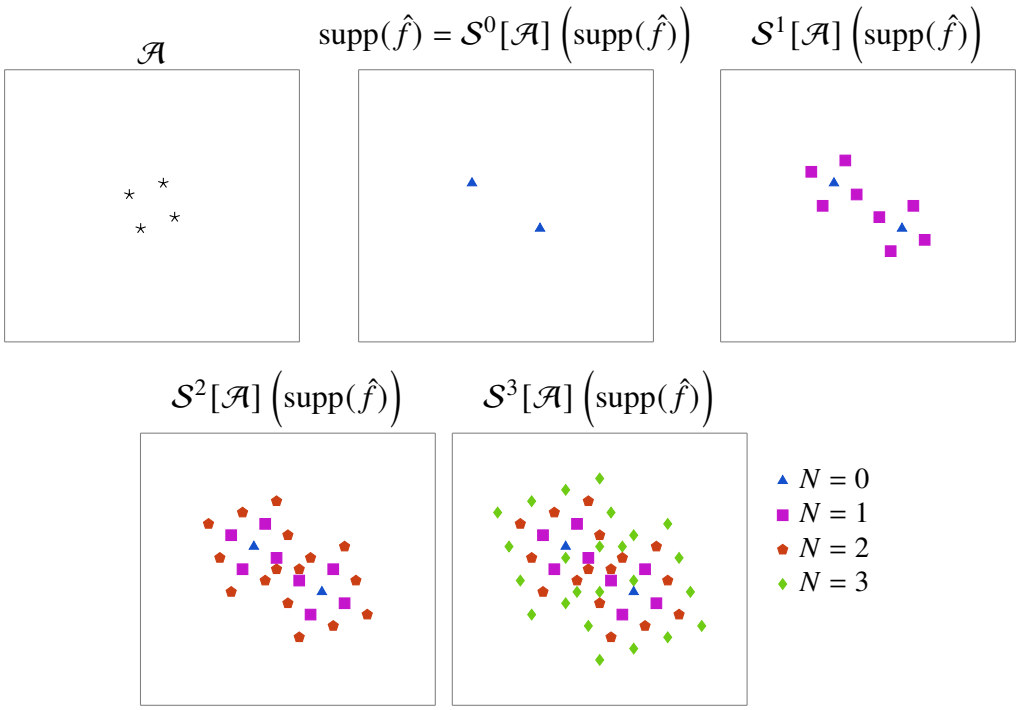



Figure 4.1 New frequencies in each stamping level up to $N = 3$ where $N = 0$ is $\text{supp}(\hat{f})$.

bandwidth in a traditional spectral method. Before moving onto this analysis however, we provide an upper bound for the cardinality of the stamping sets. This will ultimately be used to upper bound the computational complexity of our technique.

**Lemma 4.1.** *Suppose that $\mathcal{A} = -\mathcal{A}$ with $\mathbf{0} \in \mathcal{A}$, and $\left|\text{supp}(\hat{f})\right| \leq |\mathcal{A}|$. Then*

$$\left|\mathcal{S}^N[\mathcal{A}](\text{supp}(\hat{f}))\right| \leq 7\max(|\mathcal{A}|, 2N+1)^{\min(|\mathcal{A}|,2N+1)}.$$

We prove this by first providing the following combinatorial upper bound for the cardinality of a stamp set.

**Lemma 4.2.** *Suppose that $\mathcal{A} = -\mathcal{A}$ with $\mathbf{0} \in \mathcal{A}$. Then*

$$\left|\mathcal{S}^N[\mathcal{A}](\text{supp}(\hat{f}))\right| \leq \left|\text{supp}(\hat{f})\right| \sum_{n=0}^{N} \sum_{t=0}^{\min(n,(|\mathcal{A}|-1)/2)} 2^t \binom{(|\mathcal{A}|-1)/2}{t} \binom{n-1}{t-1}. \tag{4.5}$$

*Proof.* We begin by separating $\mathcal{S}^N$ into the disjoint pieces

$$\mathcal{S}^N = \bigsqcup_{n=0}^{N} \left(\mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)\right)$$

and computing the cardinality of each of these sets (where we take $S^{-1} = \emptyset$). If $\mathbf{k} \in \mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)$, then we are able to write $\mathbf{k}$ as

$$\mathbf{k} = \mathbf{k}_f + \sum_{m=1}^{n} \mathbf{k}_{\mathcal{A}}^m \tag{4.6}$$

where $\mathbf{k}_f \in \operatorname{supp}(\hat{f})$ and $\mathbf{k}_{\mathcal{A}}^m \in \mathcal{A} \setminus \{\mathbf{0}\}$ for all $m = 1, \ldots, n$. Additionally, since $\mathbf{k}$ is not in any earlier stamping sets, this is the smallest $n$ for which this is possible. In particular, it is not possible for any two frequencies in the sum to be negatives of each other resulting in pairs of cancelled terms.

With this summation in mind, arbitrarily split $\mathcal{A} \setminus \{\mathbf{0}\}$ into $A \sqcup -A$ (i.e., place all frequencies which do not negate each other into $A$ and their negatives in $-A$). By collecting like frequencies that occur as a $\mathbf{k}_{\mathcal{A}}^m$ term in (4.6), we can rewrite this sum as

$$\mathbf{k} = \mathbf{k}_f + \sum_{\mathbf{k}_{\mathcal{A}} \in A} s(\mathbf{k}, \mathbf{k}_{\mathcal{A}}) m(\mathbf{k}, \mathbf{k}_{\mathcal{A}}) \mathbf{k}_{\mathcal{A}}, \tag{4.7}$$

where the sign function $s(\mathbf{k}, \mathbf{k}_{\mathcal{A}})$ is given by

$$s(\mathbf{k}, \mathbf{k}_{\mathcal{A}}) := \begin{cases} 1 & \text{if } \mathbf{k}_{\mathcal{A}} \text{ is a term in the summation (4.6)} \\ -1 & \text{if } -\mathbf{k}_{\mathcal{A}} \text{ is a term in the summation (4.6)} \\ 0 & \text{otherwise} \end{cases}$$

and the multiplicity function $m(\mathbf{k}, \mathbf{k}_{\mathcal{A}})$ is defined as the number of times that $\mathbf{k}_{\mathcal{A}}$ or $-\mathbf{k}_{\mathcal{A}}$ appears as a $\mathbf{k}_{\mathcal{A}}^m$ term in (4.6). Letting $\mathbf{s}(\mathbf{k}) := (s(\mathbf{k}, \mathbf{k}_{\mathcal{A}}))_{\mathbf{k}_{\mathcal{A}} \in A}$ and $\mathbf{m}(\mathbf{k}) := (m(\mathbf{k}, \mathbf{k}_{\mathcal{A}}))_{\mathbf{k}_{\mathcal{A}} \in A}$, we can then identify any $\mathbf{k} \in \mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)$ with the tuple

$$(\mathbf{k}_f, \mathbf{s}(\mathbf{k}), \mathbf{m}(\mathbf{k})) \in \operatorname{supp}(\hat{f}) \times \{-1, 0, 1\}^A \times \{0, \ldots, n\}^A.$$

Upper bounding the number of these tuples that can correspond to a value of $\mathbf{k} \in \mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)$ will then upper bound the cardinality of this set.

Since any $\mathbf{k}_f \in \operatorname{supp}(\hat{f})$ can result in a valid $\mathbf{k}$ value, we will focus on the pairs of sign and multiplicity vectors. Define by $T_n \subset \{-1, 0, 1\}^A \times \{0, \ldots, n\}^A$ the set of valid sign and multiplicity pairs that can correspond to a $\mathbf{k} \in \mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)$. In particular, for $(\mathbf{s}, \mathbf{m}) \in T_n$, $\|\mathbf{m}\|_1 = n$ and

$\operatorname{supp}(\mathbf{s}) = \operatorname{supp}(\mathbf{m})$. Thus, we can write

$$T_n \subset \bigsqcup_{t=0}^{\min(n,|A|)} \left\{(\mathbf{s},\mathbf{m}) \in \{-1,0,1\}^A \times \{0,\dots,n\}^A \mid \|\mathbf{m}\|_1 = n \text{ and } |\operatorname{supp}(\mathbf{s})| = |\operatorname{supp}(\mathbf{m})| = t\right\}.$$

This inner set then corresponds to the $t$-partitions of the integer $n$ spread over the $|A|$ entries of $\mathbf{m}$ where each non-zero term is assigned a sign $-1$ or 1. The cardinality is therefore $2^t\binom{|A|}{t}\binom{n-1}{t-1}$: the first factor is from the possible sign options, the second is the number of ways to choose the entries of $\mathbf{m}$ which are nonzero, and the last is the number of $t$-partitions of $n$ which will fill the nonzero entries of $\mathbf{m}$. Noting that $|A| = \frac{|\mathcal{A}|-1}{2}$, our final cardinality estimate is

$$\begin{aligned}
\left|\mathcal{S}^N\right| &= \sum_{n=0}^{N} \left|\mathcal{S}^n \setminus \left(\bigcup_{i=0}^{n-1} \mathcal{S}^i\right)\right| \\
&\le \sum_{n=0}^{N} \left|\operatorname{supp}(\hat{f})\right| |T_n| \\
&\le \left|\operatorname{supp}(\hat{f})\right| \sum_{n=0}^{N} \sum_{t=0}^{\min(n,(|\mathcal{A}|-1)/2)} 2^t \binom{(|\mathcal{A}|-1)/2}{t} \binom{n-1}{t-1}
\end{aligned}$$

as desired. □

Though this upper bound is much tighter than the one given in the main text, it is harder to parse. As such, we simplify it to the bound presented in Lemma 4.1.

*Proof of Lemma 4.1.* Let $r = (|\mathcal{A}|-1)/2$. We consider two cases:

**Case 1:** $r \ge N$ We estimate the innermost sum of (4.5). Since $r \ge N \ge n$, $\min(n, (|\mathcal{A}|-1)/2) = n$.

By upper bounding the binomial coefficients with powers of $r$, we obtain

$$\begin{aligned}
\sum_{t=0}^{n} 2^t \binom{r}{t}\binom{n-1}{t-1} &\le \sum_{t=0}^{n} 2^t (r^t)^2 \\
&\le 2(2r^2)^n
\end{aligned}$$

where the second estimate follows from the approximating the geometric sum. Again, bounding the next geometric sum by double the largest term, we have

$$\left|\mathcal{S}^N\right| \le \left|\operatorname{supp}(\hat{f})\right| \sum_{n=0}^{N} 2(2r^2)^n \le (2r+1)4(2r^2)^N \le 2(2r+1)^{2N+1} = |\mathcal{A}|^{2N+1}.$$

**Case 2:** $r < N$ Bounding the innermost sum of (4.5) proceeds much the same way as Case 1, but we must first split the outermost sum into the first $r+1$ terms and last $N-r$ terms. Working with the first terms, we find

$$\sum_{n=0}^{r}\sum_{t=0}^{n} 2^t \binom{r}{t}\binom{n-1}{t-1} \leq 4(2r^2)^r$$

using the argument in Case 1. Now, we bound

$$\begin{aligned}\sum_{n=r+1}^{N}\sum_{t=0}^{r} 2^t \binom{r}{t}\binom{n-1}{t-1} &\leq \sum_{n=r+1}^{N} 2(2(n-1)^2)^r \\ &\leq 2^{r+1}\int_r^N n^{2r}\,dn \\ &\leq \sqrt{2}\frac{(\sqrt{2}N)^{2r+1}}{2r+1}.\end{aligned}$$

Thus,

$$\left|\mathcal{S}^N\right| \leq \left|\mathrm{supp}(\hat{f})\right|\left[4(2r^2)^r + \sqrt{2}\frac{(\sqrt{2}N)^{2r+1}}{2r+1}\right] \leq 5\sqrt{2}\left(\sqrt{2}N\right)^{|\mathcal{A}|} \leq 7(2N+1)^{|\mathcal{A}|}.$$

Combining the two cases gives the desired upper bound. □

Proposition 4.3 gives us a natural way to consider truncations of the solution $u$ in frequency space. We will use these truncations to discretize the Galerkin formulation (GF) in Section 4.6 below. In order to analyze the error in the resulting spectral method algorithm, we will need quantitative bounds on how the solution decays outside of the frequency sets $\mathcal{S}^N := \mathcal{S}^N[\mathcal{A}](\mathrm{supp}(\hat{f}))$. For $\mathcal{S}^N$ to be finite, we assume in this section that $\mathcal{A}$ and $\mathrm{supp}(\hat{f})$ are finite. This assumption will be lifted later via Lemma 4.5.

We begin with a technical result regarding the interplay between $L$ and the supports of vectors that it acts on.

**Proposition 4.4.** *For any $\hat{v}$ with $\mathrm{supp}(\hat{v}) \subset \mathcal{S}^n \setminus \mathcal{S}^{n-1}$, $\mathrm{supp}(L\hat{v}) \subset \mathcal{S}^{n+1} \setminus \mathcal{S}^{n-2}$.*

*Proof.* For any $\mathbf{k} \in \mathbb{Z}^d$, recall that row $\mathbf{k}$ of $L$ is supported on $\{\mathbf{k}\} + \mathcal{A}$. Consider

$$(L\hat{v})_{\mathbf{k}} = \sum_{\mathbf{l}\in\mathbb{Z}^d} L_{\mathbf{k},\mathbf{l}}\hat{v}_{\mathbf{l}} = \sum_{\mathbf{l}\in(\{\mathbf{k}\}+\mathcal{A})\cap\mathrm{supp}(\hat{v})} L_{\mathbf{k},\mathbf{l}}\hat{v}_{\mathbf{l}} = \sum_{\mathbf{l}\in(\{\mathbf{k}\}+\mathcal{A})\cap(\mathcal{S}^n\setminus\mathcal{S}^{n-1})} L_{\mathbf{k},\mathbf{l}}\hat{v}_{\mathbf{l}}.$$

This sum is nonempty only if $\mathbf{k}$ is such that there exists $\mathbf{l} \in \mathcal{S}^n \setminus \mathcal{S}^{n-1}$ and $\mathbf{k}^*_{\mathcal{A}} \in \mathcal{A}$ with $\mathbf{k} = \mathbf{l} + \mathbf{k}^*_{\mathcal{A}}$. By definition of $\mathbf{l} \in \mathcal{S}^n \setminus \mathcal{S}^{n-1}$, $n$ is the minimal such number that

$$\mathbf{l} = \mathbf{k}_f + \sum_{m=1}^{n} \mathbf{k}^m_{\mathcal{A}}, \text{ where } \mathbf{k}_f \in \operatorname{supp}(\hat{f}),\ \mathbf{k}^m_{\mathcal{A}} \in \mathcal{A} \text{ for all } m = 1, \dots, n$$

holds. In particular, this implies that $\mathbf{k}^m_{\mathcal{A}} \neq \mathbf{0}$ for all $m = 1, \dots, n$.

There are now two cases. First, if $\mathbf{k}^*_{\mathcal{A}} = -\mathbf{k}^m_{\mathcal{A}}$ for any $m$, $\mathbf{k} = \mathbf{l} + \mathbf{k}^*_{\mathcal{A}} \in \mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}$, and the proposition is satisfied. On the other hand, we consider the case when $\mathbf{k}^*_{\mathcal{A}}$ does not negate any $\mathbf{k}^m_{\mathcal{A}}$ involved in the sum equalling $\mathbf{l}$. If $\mathbf{k}^*_{\mathcal{A}} = \mathbf{0}$, then clearly $\mathbf{k} = \mathbf{l} \in \mathcal{S}^n \setminus \mathcal{S}^{n-1}$. In any other case, we represent

$$\mathbf{k} = \mathbf{k}_f + \sum_{m=1}^{n} \mathbf{k}^m_{\mathcal{A}} + \mathbf{k}^*_{\mathcal{A}} =: \mathbf{k}_f + \sum_{m=1}^{n+1} \mathbf{k}^m_{\mathcal{A}},$$

where $n+1$ is the smallest number for which this holds. Thus, $\mathbf{k} \in \mathcal{S}^{n+1} \setminus \mathcal{S}^n$. Altogether then, the only possible $\mathbf{k}$ values such that the sum is nonzero are those in $\mathcal{S}^{n+1} \setminus \mathcal{S}^{n-2}$, completing the proof. □

Noting that $\operatorname{supp}(L\hat{u}) = \operatorname{supp}(\hat{f})$, we observe the following interesting relationship between the values of $\hat{u}$ on neighboring stamping levels. Below, to simplify notation, for all $m, n \in \mathbb{N}_0$, we set

$$d_{m,n} := \langle L\hat{u}_{\mathcal{S}^m \setminus \mathcal{S}^{m-1}}, \hat{u}_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \rangle_{\ell^2},$$

with the convention that $\mathcal{S}^{-1} = \emptyset$.

**Corollary 4.1.** *For all $n \in \mathbb{N}_0$,*

$$d_{n+1,n} + d_{n,n} + d_{n-1,n} = \begin{cases} \langle \hat{f}, \hat{u}|_{\mathcal{S}^0} \rangle_{\ell^2} & \text{if } n = 0 \\ 0 & \text{otherwise.} \end{cases}$$

*Proof.* By Proposition 4.4, $\hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}}$ is $\ell^2$-orthogonal to $L\hat{u}|_{\mathcal{S}^m \setminus \mathcal{S}^{m-1}}$ for all $m \notin \{n-1, n, n+1\}$. In our simplified notation, $d_{m,n} = 0$ for all $m \notin \{n-1, n, n+1\}$. Thus

$$\langle \hat{f}, \hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \rangle_{\ell^2} = \langle L\hat{u}, \hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \rangle_{\ell^2} = \sum_{m=0}^{\infty} d_{m,n} = d_{n+1,n} + d_{n,n} + d_{n-1,n}.$$

The proof is finished by noting that

$$\langle \hat{f}, \hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \rangle_{\ell^2} = \begin{cases} \langle \hat{f}, \hat{u}|_{\mathcal{S}^0} \rangle & \text{if } n = 0 \\ 0 & \text{otherwise.} \end{cases}$$ □

We are now ready to estimate $\hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}}$ in terms of its neighbors $\hat{u}|_{\mathcal{S}^{n+1} \setminus \mathcal{S}^n}$ and $\hat{u}|_{\mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}}$. The standard approach would be to use a combination of coercivity and continuity (see, e.g., the proof of Lemma 4.6 or [13, Section 6.4] for other examples): for $n > 0$,

$$\alpha \left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1}^2 \leq |d_{n,n}| \leq |d_{n+1,n}| + |d_{n-1,n}| \leq \beta \left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} \left( \left\| u|_{\mathcal{S}^{n+1} \setminus \mathcal{S}^n} \right\|_{H^1} + \left\| u|_{\mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}} \right\|_{H^1} \right),$$

and we obtain

$$\left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} \leq \frac{\beta}{\alpha} \left( \left\| u|_{\mathcal{S}^{n+1} \setminus \mathcal{S}^n} \right\|_{H^1} + \left\| u|_{\mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}} \right\|_{H^1} \right).$$

However, we will hope to iterate this bound, and the fact that $\beta \geq \alpha$ will not allow for us to show any decay as $n \to \infty$. Thus, we require a slightly subtler estimate than simply using continuity.

**Proposition 4.5.** *Define*

$$\beta_-^{\mathbf{0}} := \max \left\{ \|a - \hat{a}_{\mathbf{0}}\|_{L^\infty}, \sup_{\mathbf{x} \in \mathbb{T}^d} \left\| \mathbf{b}(\mathbf{x}) - \hat{\mathbf{b}}_{\mathbf{0}} \right\|_2, \|c - \hat{c}_{\mathbf{0}}\|_{L^\infty} \right\}.$$

*For $n > 0$, we have*

$$|d_{n\pm1,n}| \leq \beta_-^{\mathbf{0}} \left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} \left\| u|_{\mathcal{S}^{n\pm1} \setminus \mathcal{S}^{n\pm1-1}} \right\|_{H^1}.$$

*Proof.* Restricting all sums to the support of the vectors they index, we have

$$d_{n\pm1,n} = \sum_{\mathbf{k} \in \mathcal{S}^n \setminus \mathcal{S}^{n-1}} \sum_{\mathbf{l} \in (\{\mathbf{k}\} + \mathcal{A})) \cap (\mathcal{S}^{n\pm1} \setminus \mathcal{S}^{n\pm1-1})} L_{\mathbf{k},\mathbf{l}} \hat{u}_{\mathbf{l}} \overline{\hat{u}}_{\mathbf{k}}.$$

Clearly, choosing $\mathbf{l} = \mathbf{k} \in \mathcal{S}^n \setminus \mathcal{S}^{n-1}$ would not allow for $\mathbf{l} \in \mathcal{S}^{n\pm1} \setminus \mathcal{S}^{n\pm1-1}$. Thus, no term multiplying $L_{\mathbf{k},\mathbf{k}}$ will appear in the sum. This implies that there are no terms including the Fourier coefficients $\hat{a}_{\mathbf{0}}$, $\hat{\mathbf{b}}_{\mathbf{0}}$, or $\hat{c}_{\mathbf{0}}$. It is therefore equivalent to replace $L$ with a version $L^-$ defined using the Fourier coefficients $\hat{a} - \hat{a}_{\mathbf{0}}$, $\hat{\mathbf{b}} - \hat{\mathbf{b}}_{\mathbf{0}}$, and $\hat{c} - \hat{c}_{\mathbf{0}}$. We then have the equivalence

$$d_{n\pm1,n} = \langle L^- \hat{u}|_{\mathcal{S}^{n\pm1} \setminus \mathcal{S}^{n\pm1-1}}, \hat{u}|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \rangle_{\ell^2},$$

which by Proposition 4.2 and the standard argument to prove the continuity upper bound, implies

$$|d_{n\pm1,n}| \leq \beta_-^{\mathbf{0}} \left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} \left\| u|_{\mathcal{S}^{n\pm1} \setminus \mathcal{S}^{n\pm1-1}} \right\|_{H^1}.$$

as desired. □

The same argument preceding Proposition 4.5 then gives the desired "neighbor" estimate.

**Corollary 4.2.** *For all $n > 1$,*

$$\left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} \leq \frac{\beta_-^{\mathbf{0}}}{\alpha} \left( \left\| u|_{\mathcal{S}^{n+1} \setminus \mathcal{S}^n} \right\|_{H^1} + \left\| u|_{\mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}} \right\|_{H^1} \right).$$

We now have the pieces to state an estimate of the truncation error.

**Lemma 4.3.** *Let $a$, $\mathbf{b}$, $c$, $f$, and $u$ be as in Proposition 4.1. Assume*

$$3\beta_-^{\mathbf{0}} < \alpha. \tag{4.8}$$

*Then*

$$\|u - u|_{\mathcal{S}^N}\|_{H^1} \leq \left( \frac{\beta_-^{\mathbf{0}}}{\alpha - 2\beta_-^{\mathbf{0}}} \right)^{N+1} \frac{\|f\|_{L^2}}{\alpha}.$$

*Proof.* We begin by breaking $\operatorname{supp}(\hat{u}) \setminus \mathcal{S}^N$ into sets of new contributions $\bigcup_{n=N+1}^{\infty} \left( \mathcal{S}^n \setminus \mathcal{S}^{n-1} \right)$ (which holds due to Proposition 4.3). Thus

$$\|u - u|_{\mathcal{S}^N}\|_{H^1} \leq \sum_{n=N+1}^{\infty} \left\| u|_{\mathcal{S}^n \setminus \mathcal{S}^{n-1}} \right\|_{H^1} =: T_N.$$

Applying the neighbor bound, Corollary 4.2, (where we define $A := \beta_-^{\mathbf{0}}/\alpha$), we have

$$\begin{aligned} T_N &\leq A \left( \sum_{n=N+1}^{\infty} \left\| u|_{\mathcal{S}^{n+1} \setminus \mathcal{S}^n} \right\|_{H^1} + \sum_{n=N+1}^{\infty} \left\| u|_{\mathcal{S}^{n-1} \setminus \mathcal{S}^{n-2}} \right\|_{H^1} \right) \\ &= A \left( T_{N+1} + T_{N-1} \right) \\ &= 2AT_N + A \left( \left\| u|_{\mathcal{S}^N \setminus \mathcal{S}^{N-1}} \right\|_{H^1} - \left\| u|_{\mathcal{S}^{N+1} \setminus \mathcal{S}^N} \right\|_{H^1} \right). \end{aligned}$$

After rearranging, and ignoring the negative term, we find

$$T_N \leq \frac{A}{1 - 2A} \left\| u|_{\mathcal{S}^N \setminus \mathcal{S}^{N-1}} \right\|_{H^1}. \tag{4.9}$$

Noting that we always have

$$\left\| u|_{\mathcal{S}^N \setminus \mathcal{S}^{N-1}} \right\|_{H^1} \leq T_{N-1}, \tag{4.10}$$

iterating (4.9) and (4.10) in turn gives

$$\|u - u|_{\mathcal{S}^N}\|_{H^1} \leq T_N \leq \left(\frac{A}{1-2A}\right)^{N+1} \left\|u|_{\mathcal{S}^0}\right\|_{H^1} \leq \left(\frac{A}{1-2A}\right)^{N+1} \frac{\|f\|_{L^2}}{\alpha}. \qquad \square$$

### 4.5 Fully sublinear-time SFTs with randomized lattices

In Chapter 3, two methods for high-dimensional SFTs are presented, each with a deterministic and Monte Carlo variant. Below, we will be using the faster of the two algorithms (at the cost of slightly suboptimal error guarantees), the phase-encoding approach with the nonequispaced sublinear-time SFT discussed in Corollary 3.2. We focus on only the Monte Carlo variant as the improvements in this section require randomization.

To use the high-dimensional phase-encoding SFT given in Algorithm 3.1, we need to know a reconstructing rank-1 lattice in advance. Though component-by-component algorithms can deterministically construct a reconstructing rank-1 lattice given any frequency set $\mathcal{I}$, as previously discussed, these algorithms are superlinear in $|\mathcal{I}|$ as they effectively search the frequency space for collisions throughout construction.

This section presents an alternative based on choosing a random lattice. This lattice is chosen by drawing $\mathbf{z}$ from a uniform distribution over $\{1, \ldots, M-1\}^d$ for $M$ sufficiently large. Below, we provide probability estimates for when this lattice is reconstructing for a frequency set $\mathcal{I}$.

**Lemma 4.4.** *Let* $K_{\mathcal{I}} := \max_{j\in[d]}(\max_{\mathbf{k}\in\mathcal{I}} k_j - \min_{\mathbf{l}\in\mathcal{I}} l_j) + 1$ *be the expansion of the frequency set* $\mathcal{I} \subset \mathbb{Z}^d$. *Let* $\sigma \in (0, 1]$, *and fix* $M$ *to be the smallest prime greater than* $\max(K_{\mathcal{I}}, \frac{|\mathcal{I}|^2}{\sigma})$. *Then drawing each component of* $\mathbf{z}$ *i.i.d from* $\{1, \ldots M-1\}$ *gives that* $\Lambda(\mathbf{z}, M)$ *is a reconstructing rank-1 lattice for* $\mathcal{I}$ *with probability* $1 - \sigma$.

*Proof.* In order to show that $\Lambda(\mathbf{z}, M)$ is reconstructing for $\mathcal{I}$, it suffices to show that for any $\mathbf{k} \neq \mathbf{l} \in \mathcal{I}$, $\mathbf{k}\cdot\mathbf{z} \not\equiv \mathbf{l}\cdot\mathbf{z} \bmod M$ (cf. Definition 1.2). Thus, we are interested in showing that $\mathbb{P}[\exists \mathbf{k} \neq \mathbf{l} \in \mathcal{I} \text{ s.t. } (\mathbf{k}-\mathbf{l})\cdot\mathbf{z} \equiv \mathbf{0} \bmod M]$ is small.

If $\mathbf{k}, \mathbf{l} \in \mathcal{I}$ are distinct, at least one component $k_j - l_j$ is nonzero. Since $M > K_{\mathcal{I}}$, we therefore have that $k_j - l_j \not\equiv 0 \bmod M$, and since $M$ is prime, $k_j - l_j$ has a multiplicative inverse modulo $M$. Then $\mathbb{P}[(\mathbf{k}-\mathbf{l})\cdot\mathbf{z} \equiv \mathbf{0} \bmod M] = \mathbb{P}\left[z_j = (k_j - l_j)^{-1}\left(\sum_{i\in[d], i\neq j}(k_i - l_i)z_i \bmod M\right)\right]$. Since $z_j$

is uniformly distributed in $\{1, \dots M-1\}$, this probability is $\frac{1}{M-1}$. By the union bound,

$$\mathbb{P}[\exists \mathbf{k} \neq \mathbf{l} \in \mathcal{I} \text{ s.t. } (\mathbf{k}-\mathbf{l}) \cdot \mathbf{z} \equiv \mathbf{0} \bmod M] \leq \sum_{\mathbf{k} \neq \mathbf{l} \in \mathcal{I}} \mathbb{P}[(\mathbf{k}-\mathbf{l}) \cdot \mathbf{z} \equiv \mathbf{0} \bmod M] \leq \frac{|\mathcal{I}|^2}{M-1} \leq \sigma$$

as desired. □

One important consequence of Lemma 4.4 is that we no longer need to provide the frequency set of interest in Corollary 3.2. Having chosen $K$, the expansion, and $s$, the sparsity level, we can always take $\mathcal{I}$ to be the frequencies corresponding to the largest $s$ Fourier coefficients of the function $g$ in the hypercube $\mathcal{B}_K^d$. Lemma 4.4 then implies that a randomly generated lattice with length $\max(K, s^2/\sigma)$ will be reconstructing for these optimal frequencies with probability $\sigma$. We summarize this in the following corollary.

**Corollary 4.3.** *Given a multivariate bandwidth $K$, a sparsity level $s$, probability of failure $\sigma \in (0, 1]$, and sampling access to $g \in L^2$, there exists a fast, randomized SFT which will produce a $2s$-sparse approximation $\hat{\mathbf{g}}^s$ of $\hat{g}$ and function $g^s := \sum_{\mathbf{k} \in \mathrm{supp}(\hat{\mathbf{g}}^s)} \hat{g}^s_{\mathbf{k}} \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ}$ approximating $g$ satisfying*

$$\|g - g^s\|_{L^2} \leq \|\hat{g} - \hat{\mathbf{g}}^s\|_{\ell^2} \leq (25 + 3K)\sqrt{s} \left\| \hat{g} - (\hat{g}|_K)_s^{\mathrm{opt}} \right\|_{\ell^1}$$

*with probability $1 - \sigma$. If $g \in L^\infty$, then $g^s$ and $\hat{\mathbf{g}}^s$ satisfy the upper bound*

$$\|g - g^s\|_{L^\infty} \leq \|\hat{g} - \hat{\mathbf{g}}^s\|_{\ell^1} \leq (35 + 3K)s \left\| \hat{g} - (\hat{g}|_K)_s^{\mathrm{opt}} \right\|_{\ell^1}$$

*with the same probability estimate. The total number of samples of $g$ and computational complexity of the algorithm can be bounded above by*

$$O\left( ds \log^3(dK \max(K, s/\sigma)) \log\left( \frac{dK \max(K, s/\sigma)}{\sigma} \right) \right).$$

*If we fix $\sigma$ (say $\sigma = 0.95$), this reduces to a complexity of*

$$O\left( ds \log^4(dK \max(K, s)) \right).$$

### 4.6 A sparse spectral method via SFTs

Let $\hat{\mathbf{a}}^s$, $\hat{\mathbf{b}}^s$, $\hat{\mathbf{c}}^s$, and $\hat{\mathbf{f}}^s$ be $s$-sparse approximations of $\hat{a}$, $\hat{\mathbf{b}}$, $\hat{c}$, and $\hat{f}$ respectively, where each coordinate in $\hat{\mathbf{b}}$ is approximated separately. We will use these approximations to discretize the

Galerkin formulation (GF) of our PDE. The first step is to reduce to the case where the PDE data is Fourier-sparse which is motivated by the following lemma.

**Lemma 4.5.** *Let* $a' := a|_{\mathrm{supp}(\hat{\mathbf{a}}^s)}$, $b'_j := b_j|_{\mathrm{supp}(\hat{\mathbf{b}}^s_j)}$ *for* $j \in [d]$, $c' := c|_{\mathrm{supp}(\hat{\mathbf{c}}^s)}$, *and* $f' := f|_{\mathrm{supp}(\hat{\mathbf{f}}^s)}$. *Define*

$$\beta'_- := \max\left\{\|a - a'\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \|\mathbf{b} - \mathbf{b}'\|_2, \|c - c'\|_{L^\infty}\right\}.$$

*Suppose that* $a'$, $\mathbf{b}'$, $c'$, *and* $f'$ *satisfy the conditions of Proposition 4.1 and let* $u'$ *be the unique solution of the resulting elliptic PDE, which we write in Galerkin form as*

$$L'\hat{u}' = \hat{f}'. \tag{4.11}$$

*Then*

$$\|u - u'\|_{H^1} \leq \frac{\|f - f'\|_{L^2}}{\alpha} + \frac{\beta'_-\|f'\|_{L^2}}{\alpha\alpha'}.$$

*where* $\alpha'$ *is taken to be the coercivity coefficient of the differential operator defined using* $a'$, $\mathbf{b}'$, *and* $c'$.

*Proof.* We begin by observing

$$L(\hat{u} - \hat{u}') = L\hat{u} - L'\hat{u}' - (L - L')\hat{u}' = \hat{f} - \hat{f}' - (L - L')\hat{u}',$$

and therefore

$$|\langle L(\hat{u} - \hat{u}'), \hat{u} - \hat{u}'\rangle| \leq \left|\langle \hat{f} - \hat{f}', \hat{u} - \hat{u}'\rangle\right| + |\langle (L - L')\hat{u}', \hat{u} - \hat{u}'\rangle|.$$

After an application of Proposition 4.2 to convert the $\ell^2$ inner products into sesquilinear forms, we can make use of coercivity, (4.2), continuity, (4.1), and the Cauchy-Schwarz inequality to produce the $H^1$ approximation

$$\alpha\|u - u'\|_{H^1} \leq \left\|\hat{f} - \hat{f}'\right\|_{\ell^2} + \beta'_-\|u'\|_{H^1}.$$

An application of the stability estimate (4.3) gives the desired bound

$$\|u - u'\|_{H^1} \leq \frac{\|f - f'\|_{L^2}}{\alpha} + \frac{\beta'_-\|f'\|_{L^2}}{\alpha\alpha'}. \qquad \square$$

We can now replace the trial and test spaces in (WF) with finite dimensional approximations so as to convert (GF) to a matrix equation. Inspired by Proposition 4.3 and the truncation error analysis in Section 4.4, we use the space of functions whose Fourier coefficients are supported on $\mathcal{S}^N := \mathcal{S}^N[\mathcal{A}](\operatorname{supp} \hat{f})$. By doing so, we discretize the Galerkin formulation of the problem (GF) into the finite system of equations

$$(\mathbf{L}_N \hat{\mathbf{u}})_{\mathbf{k}} := \sum_{\mathbf{l} \in \mathcal{S}^N} \left[ (2\pi)^2 (\mathbf{l} \cdot \mathbf{k}) \hat{a}_{\mathbf{k}-\mathbf{l}} + 2\pi \mathrm{i} \left( \hat{\mathbf{b}}_{\mathbf{k}-\mathbf{l}} \cdot \mathbf{l} \right) + \hat{c}_{\mathbf{k}-\mathbf{l}} \right] \hat{u}_{\mathbf{l}} = \hat{f}_{\mathbf{k}} \quad \text{for all } \mathbf{k} \in \mathcal{S}^N. \tag{4.12}$$

However, in practice, we do not know $\hat{a}$, $\hat{\mathbf{b}}$, $\hat{c}$, and $\hat{f}$ exactly (and indeed, they may not be exactly sparse). Thus, we substitute the SFT approximations $\hat{\mathbf{a}}^s$, $\hat{\mathbf{b}}^s$, $\hat{\mathbf{c}}^s$, and $\hat{\mathbf{f}}^s$, defining the new finite-dimensional operator $\mathbf{L}_{s,N} : \mathbb{C}^{\mathcal{S}^N} \to \mathbb{C}^{\mathcal{S}^N}$ by

$$\left(\mathbf{L}_{s,N} \hat{\mathbf{u}}\right)_{\mathbf{k}} := \sum_{\mathbf{l} \in \mathcal{S}^N} \left[ (2\pi)^2 (\mathbf{l} \cdot \mathbf{k}) \hat{a}^s_{\mathbf{k}-\mathbf{l}} + 2\pi \mathrm{i} \left( \hat{\mathbf{b}}^s_{\mathbf{k}-\mathbf{l}} \cdot \mathbf{l} \right) + \hat{c}^s_{\mathbf{k}-\mathbf{l}} \right] \hat{u}_{\mathbf{l}} \quad \text{for all } \mathbf{k} \in \mathcal{S}^N.$$

Our new approximate solution will be $\hat{\mathbf{u}}^{s,N} \in \mathbb{C}^{\mathcal{S}^N}$ which solves

$$\mathbf{L}_{s,N} \hat{\mathbf{u}}^{s,N} = \hat{\mathbf{f}}^s. \tag{4.13}$$

We summarize our technique in Algorithm 4.1.

**Algorithm 4.1** Sparse spectral method

**Input:** PDE data $a$, $\mathbf{b}$, $c$, and $f$, a sparsity parameter $s$, a bandwidth parameter $K$, and stamping level $N$

**Output:** Fourier coefficients $\hat{\mathbf{u}}^{s,N}$ of approximate solution

1: $\hat{\mathbf{a}}^s \leftarrow \mathrm{SFT}[s, K](a)$ // SFT is Algorithm 3.1 using a random rank-1 lattice (cf. Section 4.5)
2: $\mathcal{A}^s \leftarrow \operatorname{supp}(\hat{\mathbf{a}}^s)$
3: **for** $j \in [d]$ **do**
4: $\quad \hat{\mathbf{b}}^s_j \leftarrow \mathrm{SFT}[s, K](b_j)$
5: $\quad \mathcal{A}^s \leftarrow \mathcal{A}^s \cup \operatorname{supp}\left(\hat{\mathbf{b}}^s_j\right)$
6: **end for**
7: $\hat{\mathbf{c}}^s \leftarrow \mathrm{SFT}[s, K](c)$
8: $\mathcal{A}^s \leftarrow \mathcal{A}^s \cup \operatorname{supp}(\hat{\mathbf{c}}^s)$
9: $\hat{\mathbf{f}}^s \leftarrow \mathrm{SFT}[s, K](f)$
10: Compute $\mathcal{S}^N[\mathcal{A}^s]\left(\operatorname{supp}\left(\hat{\mathbf{f}}^s\right)\right)$ // see, e.g., (4.4) or (4.7)
11: $(\mathbf{L}_{s,N})_{\mathbf{k} \in \mathcal{S}^N, \mathbf{l} \in \mathcal{S}^N} \leftarrow (2\pi)^2 (\mathbf{l} \cdot \mathbf{k}) \hat{a}^s_{\mathbf{k}-\mathbf{l}} + 2\pi \mathrm{i} \left( \hat{\mathbf{b}}^s_{\mathbf{k}-\mathbf{l}} \cdot \mathbf{l} \right) + \hat{c}^s_{\mathbf{k}-\mathbf{l}}$
12: $\hat{\mathbf{u}}^{s,N} \leftarrow \mathbf{L}_{s,N} \backslash \hat{\mathbf{f}}^s$ // using MATLAB backslash notation for matrix solve

Showing that $u^{s,N}$ converges to $u$ now relies on a version of Strang's lemma [13, Equation (6.4.46)]. We make the assumption here that all functions' Fourier coefficients are supported on the supports of the outputs of their respective SFTs so that our use of $\mathcal{S}^N$ is unambiguous. However, this assumption will be lifted by Lemma 4.5 in Corollary 4.4 below.

**Lemma 4.6** (Strang's Lemma)**.** *Suppose that* $\operatorname{supp}(\hat{a}) = \operatorname{supp}(\hat{\mathbf{a}}^s)$, $\operatorname{supp}\left(\hat{b}_j\right) = \operatorname{supp}\left(\hat{\mathbf{b}}^s_j\right)$ *for all* $j \in [d]$, $\operatorname{supp}(\hat{c}) = \operatorname{supp}(\hat{\mathbf{c}}^s)$, *and* $\operatorname{supp}(\hat{f}) = \operatorname{supp}(\hat{\mathbf{f}}^s)$. *Also suppose that* $a^s \geq a^s_{\min} > 0$ *and* $-\frac{1}{2}\nabla \cdot \mathbf{b}^s + c^s \geq d^s_{\min} > 0$ *on* $\mathbb{T}^d$, *with* $\alpha^s \geq \min\{a^s_{\min}, d^s_{\min}\}$. *Additionally, define*

$$\beta^s_- := \max\left\{\|a - a^s\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \|\mathbf{b} - \mathbf{b}^s\|_2, \|c - c^s\|_{L^\infty}\right\}.$$

*Let* $u$ *and* $u^{s,N}$ *be as above. Then*

$$\left\|u - u^{s,N}\right\|_{H^1} \leq \left(1 + \frac{\beta}{\alpha^s}\right)\left\|u|_{\mathbb{Z}^d\setminus\mathcal{S}^N}\right\|_{H^1} + \frac{\beta^s_-}{\alpha^s}\|u|_{\mathcal{S}^N}\|_{H^1} + \frac{\|f - f^s\|_{L^2}}{\alpha^s}.$$

*Proof.* Define $L^s$ as $L$ where $a$, $\mathbf{b}$, and $c$ are replaced by $a^s$, $\mathbf{b}^s$, and $c^s$. Note that $L^s$ is still an infinite dimensional operator and is not truncated to $\mathcal{S}^N$ like $\mathbf{L}_{s,N}$ is. We let $\hat{\mathbf{e}} := \hat{\mathbf{u}}^{s,N} - \hat{u}|_{\mathcal{S}^N}$, and consider

$$\begin{aligned}
\mathbf{L}_{s,N}\hat{\mathbf{e}} &= \mathbf{L}_{s,N}\hat{\mathbf{u}}^{s,N} - (L^s\hat{u}|_{\mathcal{S}^N})|_{\mathcal{S}^N} \\
&= \hat{\mathbf{f}}^s - \hat{f} + (L\hat{u})|_{\mathcal{S}^N} - (L^s\hat{u}|_{\mathcal{S}^N})|_{\mathcal{S}^N} \\
&= \hat{\mathbf{f}}^s - \hat{f} + (L\hat{u}|_{\mathbb{Z}^d\setminus\mathcal{S}^N})|_{\mathcal{S}^N} + ((L - L^s)\hat{u}|_{\mathcal{S}^N})|_{\mathcal{S}^N}.
\end{aligned}$$

Noting that $\mathbf{L}_{s,N}\hat{\mathbf{e}} = (L^s\hat{\mathbf{e}})|_{\mathcal{S}^N}$ and owing to coercivity of $L^s$, we have

$$\begin{aligned}
\alpha^s\|e\|^2_{H^1} &\leq \left|\langle \mathbf{L}_{s,N}\hat{\mathbf{e}}, \hat{\mathbf{e}}\rangle\right| \\
&\leq \|f^s - f\|_{L^2}\|e\|_{H^1} + \beta\left\|u|_{\mathbb{Z}^d\setminus\mathcal{S}^N}\right\|_{H^1}\|e\|_{H^1} + \beta^s_-\|u|_{\mathcal{S}^N}\|_{H^1}\|e\|_{H^1}.
\end{aligned}$$

The result then follows from rearranging to estimate $\|e\|_{H^1}$ and using the triangle inequality to estimate $\left\|u - u^{s,N}\right\|_{H^1} \leq \|u - u|_{\mathcal{S}^N}\|_{H^1} + \|e\|_{H^1}$. □

We can now thread all of our results together into a final convergence analysis. The first corollary below is a more direct application of Strang's lemma which is then followed by another corollary

which takes advantage of the SFT recovery results. We will also return to the setting where the PDE data are not necessarily Fourier sparse. Thus, we again employ intermediate, compactly Fourier-supported PDE data as in Lemma 4.5.

**Corollary 4.4.** *Let $a^s$, $\mathbf{b}^s$, $c^s$, and $f^s$ be Fourier sparse approximations of $a$, $\mathbf{b}$, $c$, and $f$. Let $a' = a|_{\mathrm{supp}(\hat{\mathbf{a}}^s)}$, $b'_j = b_j|_{\mathrm{supp}(\hat{\mathbf{b}}^s_j)}$ for all $j \in [d]$, $c' = c|_{\mathrm{supp}(\hat{\mathbf{c}}^s)}$, and $f' = f|_{\mathrm{supp}(\hat{\mathbf{f}}^s)}$. Suppose $a$, $\mathbf{b}$, $c$, $f$; $a'$, $\mathbf{b}'$, $c'$, $f'$; and $a^s$, $\mathbf{b}^s$, $c^s$, $f^s$ satisfy the conditions of Proposition 4.1 with coercivity constants $\alpha$, $\alpha'$, and $\alpha^s$ respectively. Define the three modified continuity constants*

$$\begin{aligned}
\beta'_- &:= \max\left\{\|a - a'\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \|\mathbf{b} - \mathbf{b}'\|_2, \|c - c'\|_{L^\infty}\right\},\\
\beta'^{,\mathbf{0}}_- &:= \max\left\{\|a' - \hat{a}_{\mathbf{0}}\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \left\|\mathbf{b}' - \hat{\mathbf{b}}_{\mathbf{0}}\right\|_2, \|c' - \hat{c}_{\mathbf{0}}\|_{L^\infty}\right\},\\
\beta'^{,s}_- &:= \max\left\{\|a' - a^s\|_{L^\infty}, \sup_{\mathbf{x}\in\mathbb{T}^d} \|\mathbf{b}' - \mathbf{b}^s\|_2, \|c' - c^s\|_{L^\infty}\right\}.
\end{aligned}$$

*Additionally, suppose that*

$$3\beta'^{,\mathbf{0}}_- < \alpha'. \tag{4.14}$$

*Then with $u$ the exact solution to* (WF) *and $u^{s,N}$ the output of Algorithm 4.1, we have*

$$\begin{aligned}
\left\|u - u^{s,N}\right\|_{H^1} \leq{}& \frac{\|f - f'\|_{L^2}}{\alpha} + \frac{\beta'_-\|f'\|_{L^2}}{\alpha\alpha'} + \frac{\beta'^{,s}_-\|f'\|_{L^2}}{\alpha^s\alpha'} + \frac{\|f' - f^s\|_{L^2}}{\alpha^s}\\
&+ \left(1 + \frac{\beta'_-}{\alpha^s}\right)\left(\frac{\beta'^{,\mathbf{0}}_-}{\alpha' - 2\beta'^{,\mathbf{0}}_-}\right)^{N+1}\frac{\|f'\|_{L^2}}{\alpha'}.
\end{aligned} \tag{4.15}$$

*Proof.* The condition (4.14) allows the use of Lemma 4.3, which upper bounds the truncation error in Lemma 4.6. Combining Lemma 4.5 with this bound from Lemma 4.6 and applying the stability estimate from Proposition 4.1 finishes the proof. □

This upper bound relies on the intermediate $a'$, $\mathbf{b}'$, $c'$, and $f'$. However, in practice, it is more likely that user of this algorithm will have knowledge regarding the well-posedness of the original problem (i.e., that $a$, $\mathbf{b}$, $c$, and $f$ satisfy Proposition 4.1) and will be able to verify the well-posedness of the sparse approximate problem (i.e., that $a^s$, $\mathbf{b}^s$, $c^s$, and $f^s$ satisfy Proposition 4.1) or at least increase the accuracy of the SFT so that the coercivity conditions of the original problem are not too far perturbed. The intermediate "prime" functions, on the other hand, are less accessible. Therefore, we rewrite this statement so the assumptions and error bounds can be quantified using only

errors between the original functions and the sparse approximations which Corollary 4.3 gives upper bounds for.

**Corollary 4.5.** *Assume that* $a, c \in L^\infty(\mathbb{T}^d;\mathbb{R})$, $\mathbf{b} \in H^1(\mathbb{T}^d;\mathbb{R})^d$, *and* $f \in L^2(\mathbb{T}^d;\mathbb{R})$ *with* $a(\mathbf{x}) \geq a_{\min} > 0$ *and* $-\frac{1}{2}\nabla\cdot\mathbf{b}(\mathbf{x}) + c(\mathbf{x}) \geq d_{\min} > 0$ *a.e. on* $\mathbb{T}^d$. *Let* $a^s$, $\mathbf{b}^s$, $c^s$, *and* $f^s$ *be Fourier sparse approximations supported in frequency on* $\mathcal{B}_K^d$ *of* $a$, $\mathbf{b}$, $c$, *and* $f$ *respectively with*

$$\begin{aligned} \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1} &< a_{\min}, \\ \|\hat{c} - \hat{\mathbf{c}}^s\|_{\ell^1} + \frac{\pi K}{2}\sum_{j\in[d]}\left\|\hat{b}_j - \hat{\mathbf{b}}_j^s\right\|_{\ell^1} - \|\nabla\cdot(\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty} &< d_{\min}. \end{aligned} \tag{4.16}$$

*Define*

$$\overline{\alpha} := \min\left\{a_{\min} - \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}, d_{\min} - \|\hat{c} - \hat{\mathbf{c}}^s\|_{\ell^1} - \frac{\pi K}{2}\sum_{j\in[d]}\left\|\hat{b}_j - \hat{\mathbf{b}}_j^s\right\|_{\ell^1} - \|\nabla\cdot(\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty}\right\} > 0,$$

$$\hat{\beta}_-^s := \max\left\{\|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}, \sqrt{\sum_{j\in[d]}\left\|\hat{b}_j - \hat{\mathbf{b}}_j^s\right\|_{\ell^1}^2}, \|\hat{c} - \hat{\mathbf{c}}^s\|_{\ell^1}\right\},$$

*and*

$$\hat{\beta}_-^{\mathbf{0}} := \max\left\{\|\hat{a} - \hat{a}_{\mathbf{0}}\|_{\ell^1}, \sqrt{\sum_{j\in[d]}\left\|\hat{b}_j - \left(\hat{b}_j\right)_{\mathbf{0}}\right\|_{\ell^1}^2}, \|\hat{c} - \hat{c}_{\mathbf{0}}\|_{\ell^1}\right\}.$$

*Additionally, suppose that*

$$3\hat{\beta}_-^{\mathbf{0}} \leq \overline{\alpha}.$$

*Then with* $u$ *the exact solution to* (WF) *and* $u^{s,N}$ *the output of Algorithm 4.1, we have*

$$\left\|u - u^{s,N}\right\|_{H^1} \leq 3\frac{\left\|\hat{f}\right\|_{\ell^2}}{\overline{\alpha}}\left(\frac{\left\|\hat{f} - \hat{\mathbf{f}}^s\right\|_{\ell^2}}{\left\|\hat{f}\right\|_{\ell^2}} + \frac{\hat{\beta}_-^s}{\overline{\alpha}} + \left(\frac{\hat{\beta}_-^{\mathbf{0}}}{\overline{\alpha} - 2\hat{\beta}_-^{\mathbf{0}}}\right)^{N+1}\right).$$

*Proof.* Since $\hat{a}' = \hat{a}|_{\mathrm{supp}(\hat{\mathbf{a}}^s)}$,

$$\begin{aligned} \|a - a'\|_{L^\infty} &\leq \|\hat{a} - \hat{a}'\|_{\ell^1} \leq \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}, \\ \|a' - a^s\|_{L^\infty} &\leq \|\hat{a}' - \hat{\mathbf{a}}^s\|_{\ell^1} \leq \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}, \end{aligned}$$

and analogously for $c$, $b_j$ for all $j \in [d]$, and $f$, where the latter uses $\ell^2$ norms. This allows for the replacement of $\beta'_-$ and $\beta'^{,s}_-$ in (4.15) by $\hat{\beta}_-^s$ as well as the replacement of $\|f - f'\|_{L^2}$ and $\|f' - f^s\|_{L^2}$ by $\left\|\hat{f} - \hat{\mathbf{f}}^s\right\|_{L^2}$. A similar argument allows the replacement of $\beta'^{,\mathbf{0}}_-$ by $\hat{\beta}_-^{\mathbf{0}}$.

Additionally,

$$a^s \geq a - \|a - a^s\|_{L^\infty} \geq a - \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1} \text{ and}$$

$$a' \geq a - \|a - a'\|_{L^\infty} \geq a - \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}$$

giving $\min(a^s_{\min}, a'_{\min}) \geq a_{\min} - \|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}$. We can bound $\min(d^s_{\min}, d'_{\min})$ from below similarly. In particular, e.g.,

$$c' - \frac{1}{2}\nabla \cdot \mathbf{b}' \geq c - \frac{1}{2}\nabla \cdot \mathbf{b} - \|c - c'\|_{L^\infty} - \frac{1}{2}\|\nabla \cdot (\mathbf{b} - \mathbf{b}')\|_{L^\infty}.$$

The $\|c - c'\|_{L^\infty}$ term can be bounded by $\|\hat{c} - \hat{\mathbf{c}}^s\|_{\ell^1}$. To bound the divergence term, we use

$$\begin{aligned}
\|\nabla \cdot (\mathbf{b} - \mathbf{b}')\|_{L^\infty} &\leq \|\nabla \cdot (\mathbf{b} - \mathbf{b}')|_K\|_{L^\infty} + \|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty} \\
&= \left\| \sum_{j \in [d]} \sum_{\mathbf{k} \notin \operatorname{supp}(\hat{\mathbf{b}}^s_j) \cap \mathcal{B}^d_K} \left(\hat{b}_j\right)_{\mathbf{k}} \partial_j \mathrm{e}^{2\pi \mathrm{i} \mathbf{k} \cdot \circ} \right\|_{L^\infty} + \|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty} \\
&\leq 2\pi \sum_{j \in [d]} \sum_{\mathbf{k} \notin \operatorname{supp}(\hat{\mathbf{b}}^s_j) \cap \mathcal{B}^d_K} \left| \left(\hat{b}_j\right)_{\mathbf{k}} k_j \right| + \|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty} \qquad (4.17) \\
&\leq K\pi \sum_{j \in [d]} \left\| \hat{b}_j - \hat{b}'_j \right\|_{\ell^1} + \|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty} \\
&\leq K\pi \sum_{j \in [d]} \left\| \hat{b}_j - \hat{\mathbf{b}}^s_j \right\|_{\ell^1} + \|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty}
\end{aligned}$$

Thus $\min(\alpha', \alpha^s) \geq \overline{\alpha}$ as stated, implying the satisfaction of Proposition 4.1 for the PDEs with $a'$, $\mathbf{b}'$, $c'$, $f'$ and $a^s$, $\mathbf{b}^s$, $c^s$, $f^s$ as data. This also allows the replacement of $\alpha$, $\alpha'$ and $\alpha^s$ in (4.15) by $\overline{\alpha}$.

The rest follows by upper bounding $\|f'\|_{L^2}$ by $\left\|\hat{f}\right\|_{\ell^2}$, combining like terms, and simplifying. □

*Remark* 4.1. Corollary 4.5, includes some overly cautious concessions in order to produce a fully unified result with cleaner error bounds. In particular, condition (4.16) and the resulting definition of $\overline{\alpha}$ are used to avoid the need to consider well-posedness of the approximate versions of the PDE as required in Corollary 4.4. In general, this condition is less important as the SFT approximations of the PDE data become more accurate. The pessimistic advection term bounding in (4.17) is a result of the fact that $C^1$ guarantees for the SFT algorithm are not available. Again, this step is unnecessary if it is known (or assumed) that the approximate PDEs are well-posed. However, note

that the truncation term $\|\nabla \cdot (\mathbf{b} - \mathbf{b}|_K)\|_{L^\infty}$ can be controlled via regularity results for multivariate Fourier truncation, e.g., [64, 47], so long as the regularity of the advection field is known a priori.

*Remark* 4.2. We can interpret this upper bound by focusing on the sum

$$\frac{\left\|\hat{f} - \hat{\mathbf{f}}^s\right\|_{\ell^2}}{\left\|\hat{f}\right\|_{\ell^2}} + \frac{\hat{\beta}^s_-}{\overline{\alpha}} + \left(\frac{\hat{\beta}^{\mathbf{0}}_-}{\overline{\alpha} - \hat{\beta}^{\mathbf{0}}_-}\right)^{N+1}. \tag{4.18}$$

The first term is controlled by the accuracy of the SFT approximation to $f$. As a reminder, using Algorithm 3.1 for this SFT produces a near optimal error, upper bounded in Corollary 4.3 by

$$\left\|\hat{f} - \hat{\mathbf{f}}^s\right\|_{\ell^2} \leq (25 + 3K)\sqrt{s}\left\|\hat{f} - \left(\hat{f}|_K\right)^{\text{opt}}_s\right\|_{\ell^1}.$$

The second term, $\hat{\beta}^s_-/\overline{\alpha}$ is controlled by the accuracy of the SFT approximations of the coefficients defining the differential operator, $a$, $\mathbf{b}$, and $c$. Again, recall that Algorithm 3.1 produces near optimal approximations with error upper bounded by, e.g.,

$$\|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1} \leq (25 + 3K)s\left\|\hat{a} - (\hat{a}|_K)^{\text{opt}}_s\right\|_{\ell^1}.$$

The final term is controlled by two factors: the properties of the PDE data and the stamping level chosen. We see that the error decays exponentially as the stamping level increases. The base of this exponent is controlled by the PDE data. In particular, convergence is accelerated as $a$ and $c$ approach large constants and $\mathbf{b}$ approaches a field with divergence zero and little deviation from its mean. Indeed $\beta^{\mathbf{0}}_-$ is reduced as the deviation of all three coefficients from their mean decreases. The other piece, $\overline{\alpha}$, (ignoring the SFT-dependent terms) increases as the minimums of $a$ and $c - \frac{1}{2}\nabla \cdot \mathbf{b}$ increase.

*Remark* 4.3. The computational complexity of Algorithm 4.1 is

$$O\left(ds\log^4(dK\max(K,s)) + \max(s, 2N+1)^{3\min(s,2N+1)}\right).$$

in the case of no advection field, and

$$O\left(d^2 s\log^4(dK\max(K,s)) + \max(ds, 2N+1)^{3\min(ds,2N+1)}\right).$$

when an advection field is present. This is due to the three or $d + 3$ SFTs respectively and a matrix solve of a $\left|\mathcal{S}^N\right| \times \left|\mathcal{S}^N\right|$ system. Note that computing the stamping set can be done by enumerating

the frequencies using the techniques in Lemma 4.2 and therefore is subject to the same upper bound as given in Lemma 4.1 for a stamp set's cardinality. Recall also that the SFT complexity can be tuned to produce SFT approximations satisfying the above bounds with higher probability.

We do not analyze the complexity of the matrix solve in depth, and instead resort to the upper bound given by Gaussian elimination on the dense matrix. However, $\mathbf{L}_{s,N}$ is relatively sparse for larger stamping levels. As the capabilities of sparse solvers depend strongly on analyzing the graph connecting interacting rows in $\mathbf{L}_{s,N}$ (cf. [28, Chapter 11]), we expect that the analysis of an efficient sparse solver could be carried out using much of the same analysis of stamping sets performed in Section 4.4.

## 4.7 Numerics

This section gives examples of the algorithm summarized above applied to various problems. We begin with an overview of our implementation as well as some techniques used evaluate the accuracy of our approximations. We then present solutions to univariate and very high-dimensional multiscale problems with both exactly sparse and Fourier-compressible data. For simplicity, all experiments presented except for the last discard the advection and reaction terms, solving only a stationary diffusion equation. In this setting, solutions are unique up to constant shifts, so we always consider solutions with mean zero, that is, $\hat{u}_{\mathbf{0}} = 0$.

### 4.7.1 Code and testing overview

We implement Algorithm 4.1 described above in MATLAB using an object-oriented approach, with all code publicly available.[1] All SFTs are computed using the rank-1 lattice sparse Fourier transforms from Chapter 3.[2]

In order to evaluate the quality of our approximations, we need to choose an appropriate metric. Letting $u^{s,N}$ be the approximation returned by our algorithm, the ideal choice would be to use $\left\|u - u^{s,N}\right\|_{H^1}$. However, for the types of problems we will be investigating, the true solution $u$ is unavailable to us. Instead, we will use a proxy that takes advantage of the stability result in

[1] https://gitlab.com/grosscra/SparseADR

[2] this code is publicly available at https://gitlab.com/grosscra/Rank1LatticeSparseFourier

Proposition 4.1.

**Lemma 4.7.** *Let $u$ be the true solution to* (GF) *and $u^{s,N}$ be the approximation returned by solving* (4.13). *Define $\hat{f}^{s,N} := L\hat{u}^{s,N}$ with $f^{s,N} = \mathcal{L}u^{s,N}$. Then*

$$\left\|u - u^{s,N}\right\|_{H^1} \leq \frac{\left\|f - f^{s,N}\right\|_{L^2}}{\alpha} = \frac{\left\|\hat{f} - \hat{f}^{s,N}\right\|_{\ell^2}}{\alpha}.$$

*Proof.* The result follows from the fact that $\hat{u} - \hat{u}^{s,N}$ solves $L\left(\hat{u} - \hat{u}^{s,N}\right) = \hat{f} - L\hat{u}^{s,N} = \hat{f} - \hat{f}^{s,N}$ and applying Proposition 4.1. □

In the sequel, we will ignore $\alpha$ since we are mostly interested in convergence properties in $s$ and $N$ and we will compute the relative error

$$\frac{\left\|f - f^{s,N}\right\|_{L^2}}{\|f\|_{L^2}} \text{ or } \frac{\left\|\hat{f} - \hat{f}^{s,N}\right\|_{\ell^2}}{\left\|\hat{f}\right\|_{\ell^2}}$$

as our proxy instead. Whenever the data are exactly Fourier-sparse, the numerator of the second of these proxies can be computed exactly due to the fact that $\operatorname{supp}(\hat{f}^{s,N})$ is known to be contained in $\mathcal{S}^{N+1}$ (cf. Proposition 4.4). However, in the non-sparse setting, even though $f - f^{s,N}$ can be evaluated pointwise, computing an accurate approximation of its norm on $\mathbb{T}^d$ is challenging for large $d$. For this reason, we approximate the norm via Monte Carlo sampling. We also furnish the cases where exactly computing $\left\|\hat{f} - \hat{f}^{s,N}\right\|_{\ell^2}$ is possible with the pointwise Monte Carlo estimates to show that in practice, Monte Carlo sampling does as well as the exact computation.

### 4.7.2 Univariate compressible

We begin by replicating the lone numerical example of solving an elliptic problem in [21, Section 5.1]. In this case, we solve the univariate problem

$$-(a(x)u'(x))' = f(x) \text{ for all } x \in \mathbb{T}, \text{ where}$$
$$a(x) = \frac{1}{10}\exp\left(\frac{0.6 + 0.2\cos(2\pi x)}{1 + 0.7\sin(256\pi x)}\right), \quad f(x) = \exp(-\cos(2\pi x)) - \int_{\mathbb{T}} \exp(-\cos(2\pi x))\,dx \tag{4.19}$$

(note that the only difference from [21] is that we use the domain $\mathbb{T} = [0, 1]$ rather than $[0, 2\pi]$). This data is not Fourier sparse, but is compressible. In the original paper, a bandwidth of $K = 1\,536$ is considered and approximations with 9 and 17 Fourier coefficients are used.

We first construct a high accuracy approximation of the solution to (4.19) by numerically integrating on an extremely fine mesh of 10 000 points. This allows us to forgo our proxy error described in Lemma 4.7. As in [21], the bandwidth of our SFT used is set to $K = 1\,536$. Due to our SFT returning a $2s$ sparse approximation, we use $s = 4$ and $s = 8$ to compare with the 9 and 17 terms respectively considered in the original paper, and also provide an example with $s = 12$. We set the stamping level to $N = 1$ throughout, which, as discussed in the introduction, is similar to the technique used in [21].

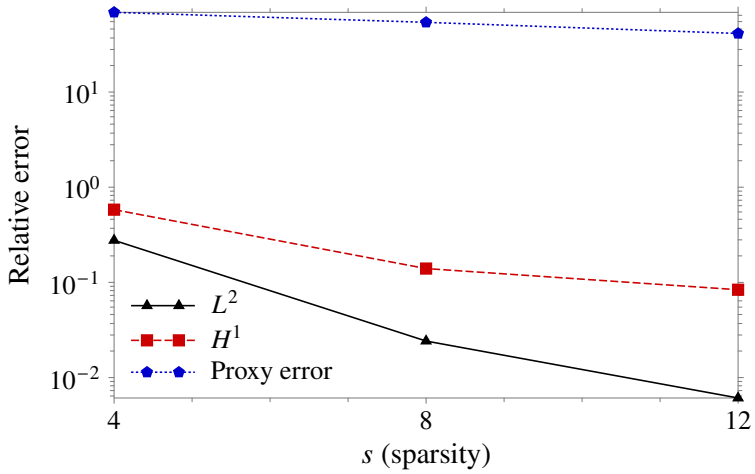
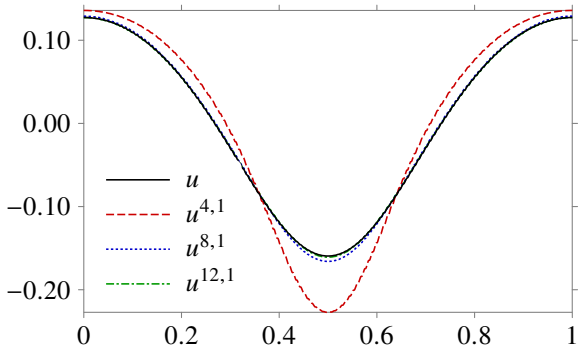
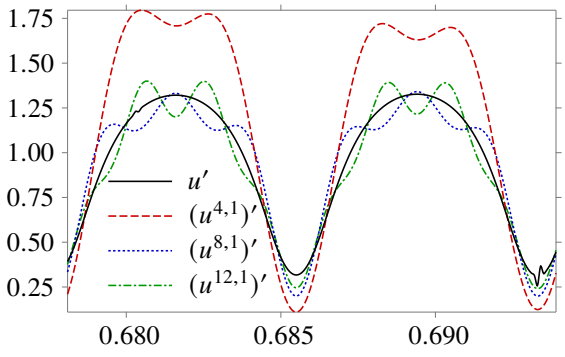



Figure 4.2 Errors in approximating the solution to (4.19).




(a) Approximate solutions of (4.19).




(b) Detail of approximate derivatives of (4.19).

Figure 4.3 Qualitative results.

The relative errors approximated in $L^2$ and $H^1$ are given in Figure 4.2. The original paper does not give numerical results, and instead, gives qualitative results, comparing the approximate

solutions and their derivatives with the true solution and its derivative. We have replicated this qualitative analysis in Figure 4.3 with similar results.

Figure 4.2 also shows the error computed via the proxy described by Lemma 4.7, and in particular, how pessimistic the proxy error can be. In this case, the small errors in the derivative (visualized in Figure 4.3b) are compounded by passing the approximate solution through the operator where $a'$ is often large relative to $a$. In future examples, we will see that the convergence of the proxy error is much more tolerable.

### 4.7.3 Multivariate exactly sparse

#### 4.7.3.1 Low sparsity

Moving to the multivariate case, we start with a simple example with exactly sparse data. Our goal is to solve

$$\begin{gathered} -\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathbb{T}^d, \text{ where} \\ a(\mathbf{x}) = \hat{a}_0 + c_a \cos(2\pi \mathbf{k}_a \cdot \mathbf{x}), \quad f(x) = \sin\big(2\pi \mathbf{k}_f \cdot \mathbf{x}\big). \end{gathered} \tag{4.20}$$

We draw $c_a \sim \text{Unif}([-1, 1])$, keep it constant for each dimension, and set $\hat{a}_{\mathbf{0}} = 4$ so that our problem remains elliptic (in the specific example below, $c_a \approx -0.6$). For dimensions varying from $d = 1$ to $d = 1\,024$, we then draw $\mathbf{k}_a, \mathbf{k}_f \sim \text{Unif}\left([-499, 500]^d \cap \mathbb{Z}^d\right)$. The PDE (4.20) is then solved for stamping levels $N = 1, \ldots, 5$. The bandwidth of the SFT is set to 1000 and the sparsity is set to 2. We then compute a Monte Carlo approximation of the proxy error choosing 200 points drawn uniformly from $\mathbb{T}^d$ and also compute the proxy error exactly by virtue of the sparsity of $a$ and $f$. The results are given in Figure 4.4.

We see that the results do not depend on the dimension of the problem. Since all dependence on $d$ is in the runtime of the SFT, we also observe that in practice, after the SFTs of the data have been computed, re-solving the problem on different stamping levels takes about the same amount of time for each $d$. The error also converges exponentially in the stamping level as suggested by the theoretical error guarantees. Notably, we also see that the Monte Carlo approximation with 200 points captures the same proxy error as the exact computation.

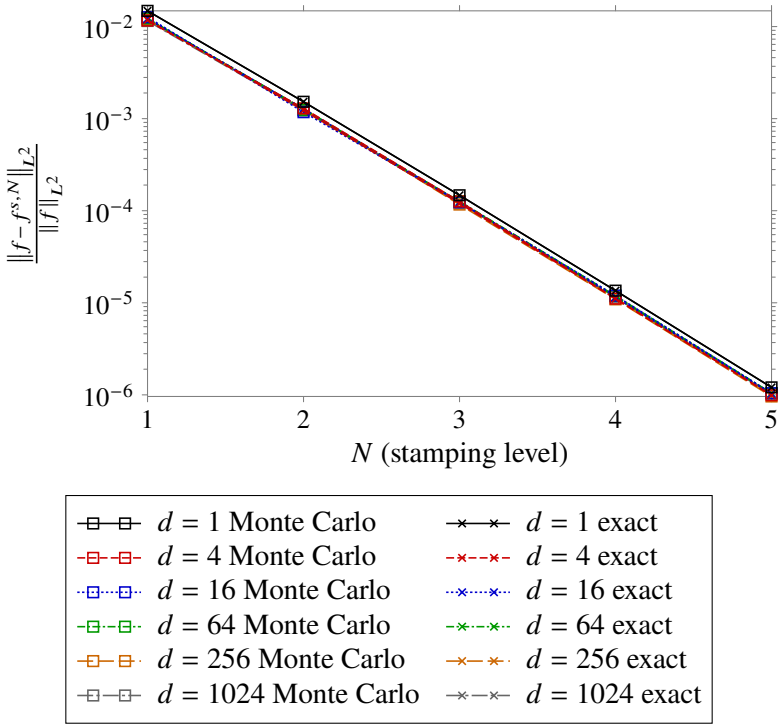



Figure 4.4 Proxy error solving (4.20) with $d = 1, 4, 16, 64, 256, 1\,024$ and $N = 1, \ldots, 5$.

#### 4.7.3.2 High sparsity

We expand on the exactly sparse case by testing a diffusion coefficient with much higher sparsity. Here, we solve (4.20) with

$$a(\mathbf{x}) = \hat{a}_{\mathbf{0}} + \sum_{\mathbf{k} \in \mathcal{I}_a} c_{\mathbf{k}} \cos(2\pi \mathbf{k} \cdot \mathbf{x}). \tag{4.21}$$

The vector of coefficients is drawn as $\mathbf{c} \sim \text{Unif}\left([-1, 1]^{25}\right)$ once and reused in each test. For every $d$, the frequencies $\mathbf{k} \in \mathcal{I}_a$ are each drawn uniformly from $[-499, 500]^d \cap \mathbb{Z}^d$ as before with $|\mathcal{I}_a| = 25$. Here $\hat{a}_0 = 4\lceil \|\mathbf{c}\|_2 \rceil$ to ensure ellipticity. Again, the bandwidth of the SFT algorithm is set to 1 000, but the sparsity is now fixed to 26. The results are given in Figure 4.5

Again, we see that the results do not depend on the spatial dimension except for the notable example of $d = 1$. The $d = 1$ case suffers from similar issues in a pessimistic proxy error as in Figure 4.2. Specifically, the right hand-side for this example was generated with frequency $k_f = -10$ and is therefore relatively low-frequency. Thus, the high-frequency modes leading to errors in the approximate solution are amplified by the high-frequencies in $a$ when computing $f^{s,N}$. Indeed, in further experiments (not pictured here), increasing the frequencies of $f$ or decreasing the frequen-

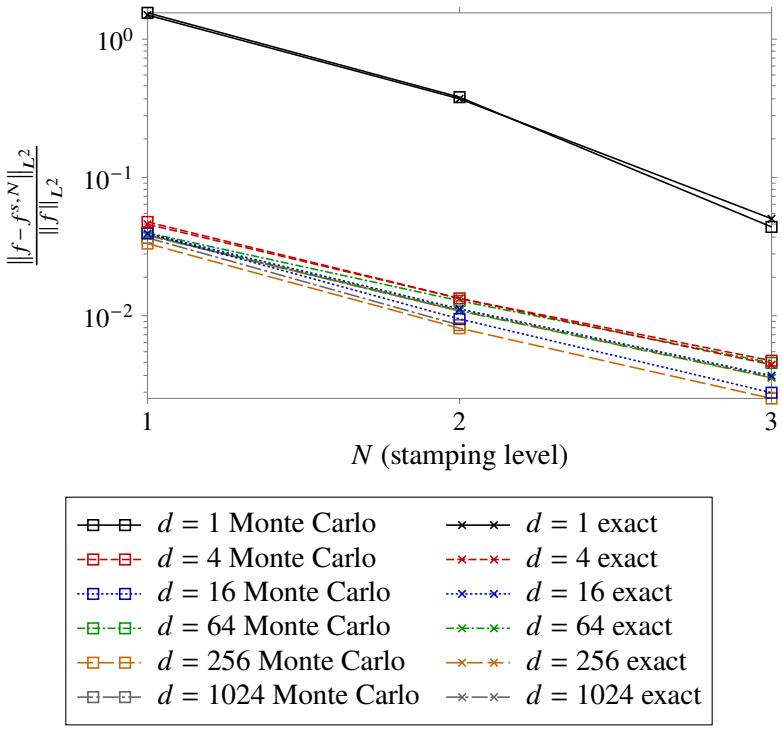



Figure 4.5 Proxy error solving (4.20) with diffusion coefficient (4.21) in dimensions $d = 1$, 4, 16, 64, 256, 1 024 and stamping levels $N = 1, \ldots, 3$.

cies of $a$ result in a lower proxy error.

For the other dimensions, the slight offsets in the exact proxy error can be attributed to the randomized frequencies as well as slight variations in the randomized SFT code. We do see slightly more variance in the proxy error computed using Monte Carlo sampling however. This is to be expected for data with more varied frequency content, and as such, in future experiments, we increase the number of sampling points.

Note that because we consider sparsity much larger than the stamping level, the computational and memory complexity of the stamping and solution step is much higher. As suggested by Lemma 4.1, the size of the resulting stamp set (and therefore the necessary matrix solve) in the largest case is at most $7 \cdot 52^7 \approx 7 \times 10^{12}$ which pushes the memory boundaries of our computational resources.

### 4.7.4 Multivariate compressible

In order to test Fourier-compressible data which is not exactly sparse, we use a series of tensorized, periodized Gaussians. Here, we present the only details necessary to demonstrate our

algorithm's effectiveness on Fourier-compressible data, but for a fuller treatment on the Fourier properties of periodized Gaussians, see e.g., [53, Section 2.1].

Here, we define the periodic Gaussian $G_r : \mathbb{T} \to \mathbb{R}$ by

$$G_r(x) = \frac{\sqrt{2\pi}}{r} \sum_{m=-\infty}^{\infty} \mathrm{e}^{-\frac{(2\pi)^2(x-m)}{2r^2}}$$

where the dilation-type parameter $r$ allows us to control the effective support of $\hat{G}_r$. In practice, we truncate the infinite sum to $m \in \{-10, \dots, 10\}$ as additional terms do not change the output up to machine precision. Note here that the nonstandard multiplicative factors help control the behavior of the function in frequency rather than space. Given a multivariate modulating frequency $\mathbf{k} \in \mathbb{Z}^d$, we define the modulated, tensorized, periodic Gaussian by

$$G_{r,\mathbf{k}}(\mathbf{x}) = \prod_{j \in [d]} \mathrm{e}^{2\pi \mathrm{i} k_i x_i} G_r(x_i).$$

Finally, given a set of frequencies $\mathcal{I} \subset \mathbb{Z}^d$, dilation parameters $\mathbf{r} \in \mathbb{R}_+^{\mathcal{I}}$, and coefficients $\mathbf{c} \in \mathbb{R}^{\mathcal{I}}$, we can define Gaussian series

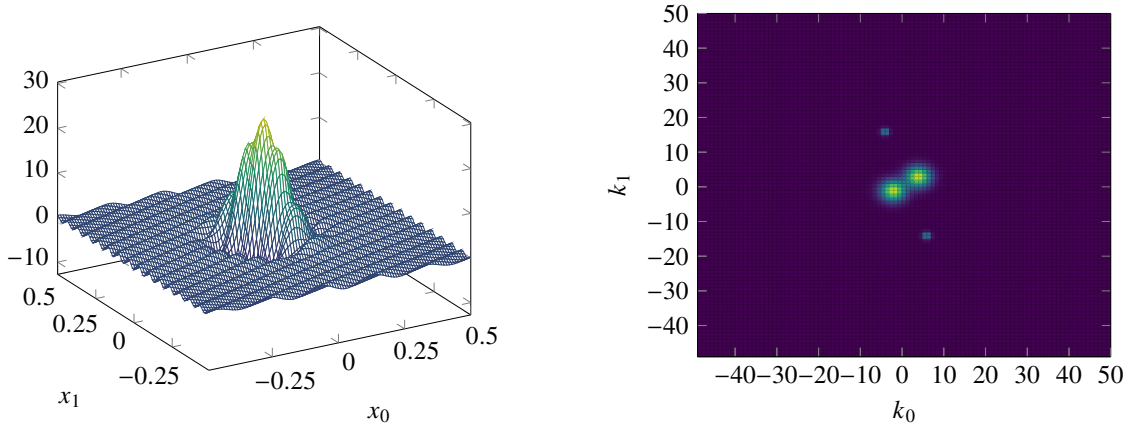
$$G_{\mathbf{c},\mathbf{r}}^{\mathcal{I}}(\mathbf{x}) := \sum_{\mathbf{k} \in \mathcal{I}} c_{\mathbf{k}} G_{r_{\mathbf{k}},\mathbf{k}}(\mathbf{x}).$$

Depending on the severity of the dilations chosen (i.e., $r_{\mathbf{k}} \gg 1$), this can well approximate a Fourier series with frequencies in $\mathcal{I}$. On the other hand, a less severe dilation results in Fourier coefficients with magnitudes forming less concentrated Gaussians centered around the "frequencies" $\mathbf{k} \in \mathcal{I}$ and $-\mathbf{k}$. An example of a series with its associated Fourier transform is given in Figure 4.6.

In our first experiment, we fix $d = 2$ and vary both stamp level and sparsity to again solve (4.20). The diffusion coefficient in (4.20) is replaced with a two-term Gaussian series $a = c_0 + G_{\mathbf{c},\mathbf{r}}^{\mathcal{I}}$, where

$$\mathcal{I} \sim \mathrm{Unif}\left(\left([-24, 25]^2 \cap \mathbb{Z}^2\right)^2\right), \quad \mathbf{c} \sim \mathrm{Unif}\left([-1, 1]^2\right), \quad \mathbf{r} = 1.1^2\mathbf{1}, \quad c_0 = 10\left\lceil \|\mathbf{c}\|_2 \right\rceil.$$

Note the increased constant factor from our previous examples to decrease the likelihood of sparse approximations of $a$ not satisfying the ellipticity property. The Fourier transform of the resulting $a$ used for the following test is depicted in Figure 4.7 below. The diffusion equation is then solved across various sparsities with increasing stamping level. The bandwidth parameter of the SFT is




(a) $c_1 G_{r_1,\mathbf{k}_1} + c_2 G_{r_2,\mathbf{k}_2}$ (b) $c_1 \hat{G}_{r_1,\mathbf{k}_1} + c_2 \hat{G}_{r_2,\mathbf{k}_2}$

Figure 4.6 An example Gaussian series with $c_1 = c_2 = 1$, $r_1 = 0.5$, $r_2 = 2$, $\mathbf{k}_1 = (3, 2)$, and $\mathbf{k}_2 = (-5, 15)$. The first term corresponds to the wider Gaussian shape and more spread out portions of the Fourier transform. The second term contributes to the highly oscillatory parts and the isolated spikes in the Fourier transform.

set to $K = 100$ to account for the wider effective support of $\hat{a}$. The Monte Carlo proxy error is computed with 1 000 samples and depicted in Figure 4.8.

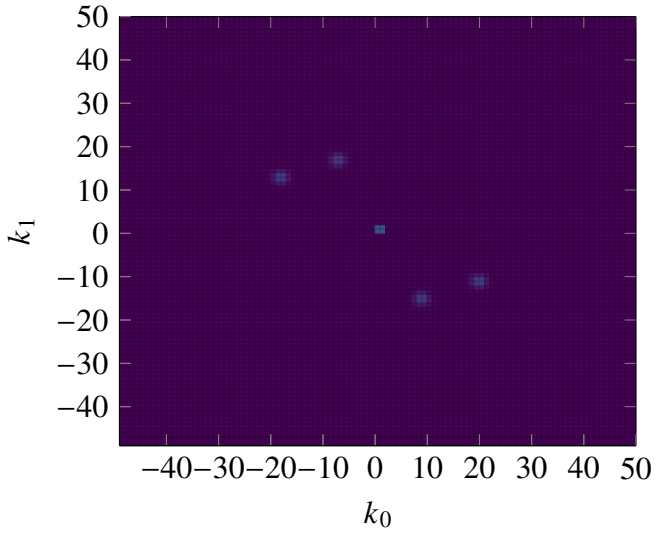



Figure 4.7 The specific $\hat{a}$ used in examples depicted in Figure 4.8.

Here, the stamping level does not affect convergence until the sparsity is above $s \geq 16$. This demonstrates the tradeoff between sparsity and stamping level in regards to the error bound (4.18). Until the SFT is able to capture enough useful information in $\hat{a}$, the $\|\hat{a} - \hat{\mathbf{a}}^s\|_{\ell^1}$ piece of the error bound dominates. Eventually, this factor is reduced far enough that the stamping term becomes

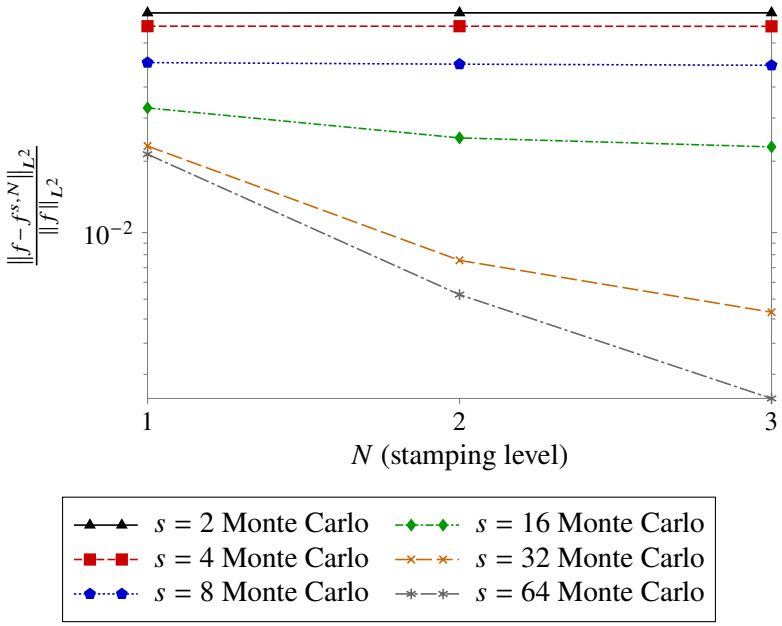



Figure 4.8 Proxy error solving (4.20) with Gaussian series diffusion coefficient with sparsity levels $s = 2, 4, 8, 16, 32, 64$, and stamping levels $N = 1, \ldots, 3$.

apparent.

We provide another example, where sparsity is fixed at $s = 16$, and dimension and stamping level are increased. Again we solve (4.20) with the diffusion coefficient replaced by the two-term Gaussian series $a = c_0 + G^{\mathcal{I}}_{\mathbf{c},\mathbf{r}}$, where

$$\mathcal{I} \sim \text{Unif}\left(\left([-249, 250]^d \cap \mathbb{Z}^d\right)^2\right), \quad \mathbf{c} \sim \text{Unif}\left([-1, 1]^2\right), \quad \mathbf{r} = 1.1^d\mathbf{1}, \quad c_0 = 10\left\lceil \|\mathbf{c}\|_2 \right\rceil,$$

and $\mathbf{c}$ and $c_0$ are not redrawn across test cases. The bandwidth of the SFT is set to 1 000 to again account for the potentially widened Fourier transform of $a$. With a 1 000 point Monte Carlo approximation of the proxy error, the results are given in Figure 4.9.

Here we observe much the same behavior as the previous test case. This is due to the fact that the dimension additionally drives the sparsity of the Gaussian Fourier transforms based on the choice of dilation $\mathbf{r} = 1.1^d\mathbf{1}$. In additional experiments performed at higher dimensions (not pictured here), this factor results in numerical instability and the approximation error blows up. We also see that the $d = 2$ and $d = 4$ examples are swapped from their assumed positions (and the $d = 2$ case even mildly benefits from increased stamping level). This is attributed to the random draw of the frequency locations affecting the proxy error as well as the SFT algorithm performing better in

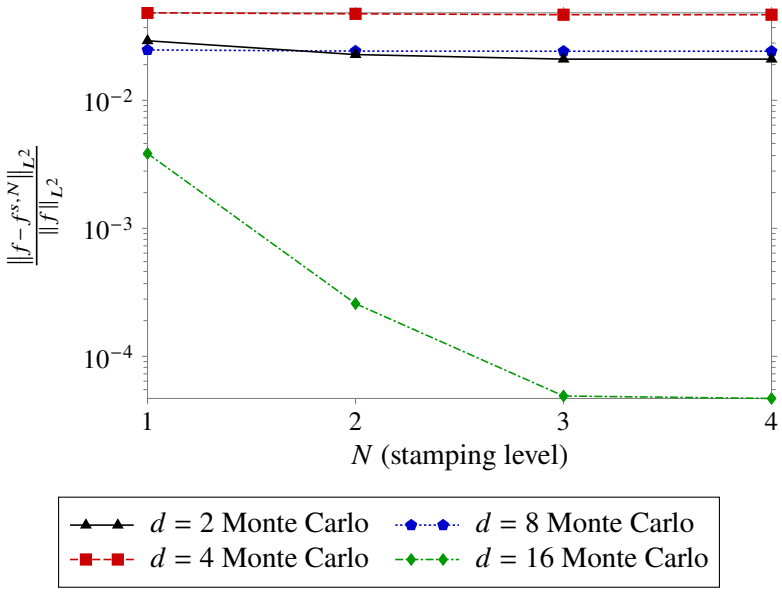



Figure 4.9 Approximate proxy error solving (4.20) with Gaussian series diffusion coefficient with $d = 2, 4, 8, 16$ and $N = 1, \ldots, 5$.

lower dimensions when all parameters are fixed.

### 4.7.5 Three-dimensional exactly sparse advection-diffusion-reaction equation

We now extend our numerical experiments to the situation of a three-dimensional advection-diffusion-reaction equation. We work with the PDE

$$-\nabla \cdot (a\nabla u) + \mathbf{b} \cdot \nabla u + cu = f \tag{4.22}$$

with exactly sparse data

$$\begin{aligned}
a(\mathbf{x}) &= \hat{a}_{\mathbf{0}} + \sum_{\mathbf{k} \in \mathcal{I}_a^{\text{sine}}} c_{a,\mathbf{k}}^{\text{sine}} \sin(2\pi \mathbf{k} \cdot \mathbf{x}) + \sum_{\mathbf{k} \in \mathcal{I}_a^{\text{cosine}}} c_{a,\mathbf{k}}^{\text{cosine}} \cos(2\pi \mathbf{k} \cdot \mathbf{x}) \\
b_j(\mathbf{x}) &= \sum_{\mathbf{k} \in \mathcal{I}_{b_j}^{\text{sine}}} c_{b_j,\mathbf{k}}^{\text{sine}} \sin(2\pi \mathbf{k} \cdot \mathbf{x}) + \sum_{\mathbf{k} \in \mathcal{I}_{b_j}^{\text{cosine}}} c_{b_j,\mathbf{k}}^{\text{cosine}} \cos(2\pi \mathbf{k} \cdot \mathbf{x}) \text{ for all } j \in [3] \\
c(\mathbf{x}) &= \hat{c}_{\mathbf{0}} + \sum_{\mathbf{k} \in \mathcal{I}_c^{\text{sine}}} c_{c,\mathbf{k}}^{\text{sine}} \sin(2\pi \mathbf{k} \cdot \mathbf{x}) + \sum_{\mathbf{k} \in \mathcal{I}_c^{\text{cosine}}} c_{c,\mathbf{k}}^{\text{cosine}} \cos(2\pi \mathbf{k} \cdot \mathbf{x}) \\
f(\mathbf{x}) &= \sum_{\mathbf{k} \in \mathcal{I}_f^{\text{sine}}} c_{f,\mathbf{k}}^{\text{sine}} \sin(2\pi \mathbf{k} \cdot \mathbf{x}) + \sum_{\mathbf{k} \in \mathcal{I}_f^{\text{cosine}}} c_{f,\mathbf{k}}^{\text{cosine}} \cos(2\pi \mathbf{k} \cdot \mathbf{x}),
\end{aligned} \tag{4.23}$$

where

$$\left|\mathcal{I}_a^{\text{sine}}\right| = \left|\mathcal{I}_a^{\text{cosine}}\right| = 2$$

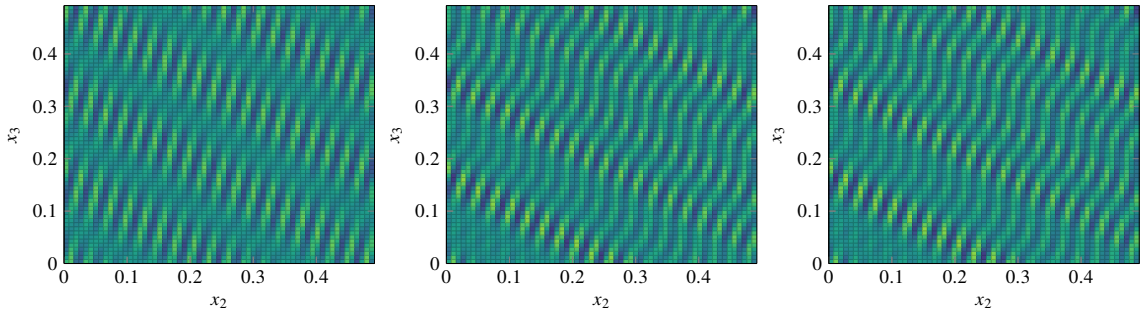
$$\left|\mathcal{I}_{b_j}^{\text{sine}}\right| = \left|\mathcal{I}_{b_j}^{\text{cosine}}\right| = \left|\mathcal{I}_c^{\text{sine}}\right| = \left|\mathcal{I}_{\mathbf{c}}^{\text{cosine}}\right| = 5 \text{ for all } j \in [3]$$

$$\left|\mathcal{I}_f^{\text{sine}}\right| = 2, \text{ and } \left|\mathcal{I}_f^{\text{cosine}}\right| = 3.$$

In total, there are 45 terms composing the differential operator, and 5 terms composing the forcing function. Each frequency is randomly drawn from $\text{Unif}([-49, 50]^3 \cap \mathbb{Z}^3)$ and each coefficient for $a$ and $f$ from $\text{Unif}([-1, 1])$. The coefficients for $\mathbf{b}$ and $c$ are drawn from $\text{Unif}([0, 1])$. To ensure well-posedness, $\hat{a}_{\mathbf{0}} = 4\left\lceil \sqrt{\left\|c_a^{\text{sine}}\right\|_2^2 + \left\|c_a^{\text{cosine}}\right\|_2^2} \right\rceil$, and $\hat{c}_{\mathbf{0}} = 4\left\lceil \sqrt{\left\|c_c^{\text{sine}}\right\|_2^2 + \left\|c_c^{\text{cosine}}\right\|_2^2} \right\rceil$. The bandwidth of the SFT is set to $K = 100$ and we consider sparsity levels $s = 2$ and $s = 5$. Due to the large size of the stamp, we only consider stamping levels $N = 1, 2$.

| | | $\left\|f - f^{s,N}\right\|_{L^2} / \|f\|_{L^2}$ | |
|---|---|---|---|
| $s$ | $N$ | exact | Monte Carlo |
| 2 | 1 | 0.518 | 0.518 |
| | 2 | 0.518 | 0.518 |
| 5 | 1 | 0.054 | 0.054 |
| | 2 | 0.031 | 0.031 |

Table 4.1 Error in approximating solution to ADR equation (4.22).




(a) Slice through $f^{2,1}$. (b) Slice through $f^{10,2}$. (c) Slice through $f$.

Figure 4.10 Samples of $f^{10,2}$ and $f$ on the $x_1 = 63/128$ plane.

The resulting true and Monte Carlo proxy error (sampled over 1 000 points) is given in Table 4.1. Additionally, Figure 4.10 shows a portion of a slice through $f$ as well as $f^{2,1}$ and $f^{10,2}$ which are computed by passing $u^{2,1}$ and $u^{10,2}$ through the differential operator.

We note that $f^{10,2}$ and $f$ appear qualitatively indistinguishable. However, since the sparsity level, $s = 2$, used to compute $u^{2,1}$ is lower than the sparsity of any term in (4.23), $f^{2,1}$ loses some of characteristics of the original source term. Though it captures some of the true behavior in both larger scales (e.g., the oscillations moving in the northeast direction) and finer scales (e.g., the oscillations moving in the southeast direction), some interfering modes which produce the "wavy" effect are left out. This is supported by the relative errors reported in Table 4.1. Note also that the stamping level affects the convergence in $s = 5$ case, but not the $s = 2$ case. This is due to the sparsity related errors in (4.18) overwhelming the stamping term until the SFT approximations of the data are accurate enough.

# BIBLIOGRAPHY


[1] Sina Bittens and Gerlind Plonka. Real sparse fast DCT for vectors with short support. *Linear Algebra Appl.*, 582:359–390, 2019.

[2] Sina Bittens and Gerlind Plonka. Sparse fast DCT for vectors with one-block support. *Numer. Algorithms*, 82(2):663–697, 2019.

[3] Sina Bittens, Ruochuan Zhang, and Mark A Iwen. A deterministic sparse FFT for functions with structured Fourier sparsity. *Advances in Computational Mathematics*, 45(2):519–561, 2019.

[4] Simone Brugiapaglia. *COmpRessed SolvING: Sparse Approximation of PDEs based on Compressed Sensing*. PhD thesis, Polytecnico Di Milano, Milan, Italy, January 2016.

[5] Simone Brugiapaglia. A compressive spectral collocation method for the diffusion equation under the restricted isometry property. In Marta D'Elia, Max Gunzburger, and Gianluigi Rozza, editors, *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*, Lecture Notes in Computational Science and Engineering, pages 15–40. Springer International Publishing, Cham, 2020.

[6] Simone Brugiapaglia, Sjoerd Dirksen, Hans Christian Jung, and Holger Rauhut. Sparse recovery in bounded Riesz systems with applications to numerical methods for PDEs. *Applied and Computational Harmonic Analysis*, 53:231–269, July 2021.

[7] Simone Brugiapaglia, Stefano Micheletti, Fabio Nobile, and Simona Perotto. Supplementary material to "Wavelet–Fourier CORSING techniques for multidimensional advection–diffusion–reaction equations", September 2020.

[8] Simone Brugiapaglia, Stefano Micheletti, Fabio Nobile, and Simona Perotto. Wavelet–Fourier CORSING techniques for multidimensional advection–diffusion–reaction equations. *IMA Journal of Numerical Analysis*, (draa036), September 2020.

[9] Simone Brugiapaglia, Stefano Micheletti, and Simona Perotto. Compressed solving: A numerical approximation technique for elliptic PDEs based on compressed sensing. *Computers & Mathematics with Applications*, 70(6):1306–1335, September 2015.

[10] Simone Brugiapaglia, Fabio Nobile, Stefano Micheletti, and Simona Perotto. A theoretical study of COmpRessed SolvING for advection-diffusion-reaction problems. *Mathematics of Computation*, 87(309):1–38, January 2018.

[11] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, May 2004. Publisher: Cambridge University Press.

[12] Glenn Byrenheid, Lutz Kämmerer, Tino Ullrich, and Toni Volkmer. Tight error bounds for rank-1 lattice sampling in spaces of hybrid mixed smoothness. *Numerische Mathematik*, 136(4):993–1034, August 2017.

[13] Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Scientific Computation. Springer-Verlag, Berlin Heidelberg, 2006.

[14] Bosu Choi, Andrew Christlieb, and Yang Wang. Multiscale High-Dimensional Sparse Fourier Algorithms for Noisy Data. *ArXiv e-prints*, 2019. arXiv:1907.03692.

[15] Bosu Choi, Andrew Christlieb, and Yang Wang. High-dimensional sparse Fourier algorithms. *Numerical Algorithms*, 87(1):161–186, May 2021.

[16] Bosu Choi, Mark Iwen, and Toni Volkmer. Sparse harmonic transforms ii: best s-term approximation guarantees for bounded orthonormal product bases in sublinear-time. *Numerische Mathematik*, 148(2):293–362, Jun 2021.

[17] Bosu Choi, Mark A. Iwen, and Felix Krahmer. Sparse harmonic transforms: A new class of sublinear-time algorithms for learning functions of many variables. *Found. Comput. Math.*, 2020.

[18] Andrew Christlieb, David Lawlor, and Yang Wang. A multiscale sub-linear time Fourier algorithm for noisy data. *Appl. Comput. Harmon. Anal.*, 40(3):553 – 574, 2016.

[19] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best $k$-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, January 2009.

[20] Dinh Dũng, Vladimir Temlyakov, and Tino Ullrich. *Hyperbolic Cross Approximation*. Advanced Courses in Mathematics - CRM Barcelona. Springer International Publishing, Cham, 2018.

[21] Ingrid Daubechies, Olof Runborg, and Jing Zou. A sparse spectral method for homogenization multiscale problems. *Multiscale Modeling & Simulation*, 6(3):711–740, January 2007. Publisher: Society for Industrial and Applied Mathematics.

[22] Michael Döhler, Stefan Kunis, and Daniel Potts. Nonequispaced hyperbolic cross fast Fourier transform. *SIAM Journal on Numerical Analysis*, 47(6):4415–4428, January 2010. Publisher: Society for Industrial and Applied Mathematics.

[23] Lawrence C. Evans. *Partial differential equations*. Number v. 19 in Graduate studies in mathematics. American Mathematical Society, Providence, R.I, second edition edition, 2010.

[24] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Springer, 2013.

[25] A. C. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse Fourier representations. In Manos Papadakis, Andrew F. Laine, and Michael A. Unser, editors, *Wavelets XI*, volume 5914, pages 398 – 412. International Society for Optics and Photonics, SPIE, 2005.

[26] Anna C Gilbert, Piotr Indyk, Mark Iwen, and Ludwig Schmidt. Recent developments in the sparse Fourier transform: A compressed Fourier transform for big data. *IEEE Signal Processing Magazine*, 31(5):91–100, 2014.

[27] Anna C. Gilbert, Martin J. Strauss, and Joel A. Tropp. A tutorial on fast Fourier sampling. *IEEE Signal Process. Mag.*, 25(2):57–66, 2008.

[28] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[29] V Gradinaru. Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation. *Computing (Wien. Print)*, 80(1):1–22, January 2007. Place: Wien Publisher: Springer.

[30] Michael Griebel and Jan Hamaekers. Sparse grids for the Schrödinger equation. *Special issue on molecular modelling*, 41(2):215–247, January 2007. Place: Les Ulis Publisher: EDP Sciences.

[31] Michael Griebel and Jan Hamaekers. Fast discrete Fourier transform on generalized sparse grids. In Jochen Garcke and Dirk Pflüger, editors, *Sparse Grids and Applications - Munich 2012*, volume 97, pages 75–107. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computational Science and Engineering.

[32] Craig Gross and Mark Iwen. Sparse spectral methods for solving high-dimensional and multiscale elliptic PDEs. *ArXiv e-prints*, 2023. arXiv:2302.00752.

[33] Craig Gross, Mark Iwen, Lutz Kämmerer, and Toni Volkmer. Sparse Fourier transforms on rank-1 lattices for the rapid and low-memory approximation of functions of many variables. *Sampling Theory, Signal Processing, and Data Analysis*, 20(1):1, December 2021.

[34] Craig Gross, Mark A Iwen, Lutz Kämmerer, and Toni Volkmer. A deterministic algorithm for constructing multiple rank-1 lattices of near-optimal size. *Advances in Computational Mathematics*, 47(6):1–24, 2021.

[35] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric 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.

[36] Mark A Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10(3):303–338, 2010.

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

[38] Lutz Kämmerer. *High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling*. Ph.D, Universitätsverlag Chemnitz, 2014.

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

[40] Lutz Kämmerer. Multiple rank-1 lattices as sampling schemes for multivariate trigonometric polynomials. *Journal of Fourier Analysis and Applications*, 24(17):17–44, 2018.

[41] Lutz Kämmerer. Constructing spatial discretizations for sparse multivariate trigonometric polynomials that allow for a fast discrete Fourier transform. *Applied and Computational Harmonic Analysis*, 47(3):702–729, 2019.

[42] Lutz Kämmerer, Felix Krahmer, and Toni Volkmer. A sample efficient sparse FFT for arbitrary frequency candidate sets in high dimensions. *Numerical Algorithms*, 89(4):1479–1520, Apr 2022.

[43] Lutz Kämmerer, Daniel Potts, and Toni Volkmer. High-dimensional sparse FFT based on sampling along multiple rank-1 lattices. *Appl. Comput. Harmon. Anal.*, 51:225–257, 2021.

[44] Lutz Kämmerer and Toni Volkmer. Approximation of multivariate periodic functions based on sampling along multiple rank-1 lattices. *Journal of Approximation Theory*, 246:1–27, 2019.

[45] Michael Kapralov. *Sparse Fourier Transform in Any Constant Dimension with Nearly-Optimal Sample Complexity in Sublinear Time*, page 264–277. Assoc. Comput. Mach., New York, NY, USA, 2016.

[46] Frances Kuo, Giovanni Migliorati, Fabio Nobile, and Dirk Nuyens. Function integration, reconstruction and approximation using rank-1 lattices. *Mathematics of Computation*, 90(330):1861–1897, July 2021.

[47] Friedrich Kupka. *Sparse grid spectral methods for the numerical solution of partial differential equations with periodic boundary conditions*. Ph.D., Universität Wien, Vienna, Austria, November 1997.

[48] Lutz Kämmerer. A fast probabilistic component-by-component construction of exactly integrating rank-1 lattices and applications. *ArXiv e-prints*, 2020. arXiv:2012.14263.

[49] Lutz Kämmerer, Stefan Kunis, and Daniel Potts. Interpolation lattices for hyperbolic cross trigonometric polynomials. *Journal of Complexity*, 28(1):76–92, February 2012.

[50] Lutz Kämmerer, Daniel Potts, and Toni Volkmer. Approximation of multivariate periodic functions by trigonometric polynomials based on rank-1 lattice sampling. *Journal of Complexity*, 31(4):543–576, August 2015.

[51] David Lawlor, Yang Wang, and Andrew Christlieb. Adaptive sub-linear time Fourier algorithms. *Adv. Adapt. Data Anal.*, 05(01):1350003, 2013.

[52] Dong Li and Fred J. Hickernell. Trigonometric spectral collocation methods on lattices. In *Recent advances in scientific computing and partial differential equations (Hong Kong, 2002)*, volume 330 of *Contemp. Math.*, pages 121–132. Amer. Math. Soc., Providence, RI, 2003.

[53] Sami Merhi, Ruochuan Zhang, Mark A. Iwen, and Andrew Christlieb. A new class of fully discrete sparse Fourier transforms: Faster stable implementations with guarantees. *Journal of Fourier Analysis and Applications*, 25(3):751–784, June 2019.

[54] Lucia Morotti. Explicit universal sampling sets in finite vector spaces. *Appl. Comput. Harmon. Anal.*, 43(2):354–369, 2017.

[55] Hans Munthe-Kaas and Tor Sørevik. Multidimensional pseudo-spectral methods on lattice grids. *Applied Numerical Mathematics*, 62(3):155–165, March 2012.

[56] Gerlind Plonka, Daniel Potts, Gabriele Steidl, and Manfred Tasche. *Numerical Fourier Analysis*. Applied and Numerical Harmonic Analysis. Springer International Publishing, Cham, 2018.

[57] Gerlind Plonka and Katrin Wannenwetsch. A sparse fast Fourier algorithm for real non-negative vectors. *J. Comput. Appl. Math.*, 321:532–539, 2017.

[58] Gerlind Plonka, Katrin Wannenwetsch, Annie Cuyt, and Wen-shin Lee. Deterministic sparse FFT for $M$-sparse vectors. *Numer. Algorithms*, 78(1):133–159, 2018.

[59] Daniel Potts and Toni Volkmer. Sparse high-dimensional FFT based on rank-1 lattice sampling. *Appl. Comput. Harmon. Anal.*, 41(3):713–748, 2016.

[60] J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64–94, 1962.

[61] A.D. Rubio, A. Zalts, and C.D. El Hasi. Numerical solution of the advection-reaction-diffusion equation at different scales. *Environmental Modelling & Software*, 23(1):90–95, January 2008.

[62] Ben Segal and MA Iwen. Improved sparse Fourier approximation results: faster implementations and stronger guarantees. *Numer. Algorithms*, 63(2):239–263, 2013.

[63] Jie Shen and Li-Lian Wang. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. *SIAM Journal on Numerical Analysis*, 48(3):1087–1109, January 2010. Publisher: Society for Industrial and Applied Mathematics.

[64] V. N. Temlyakov. *Approximation of periodic functions*. Comput. Math. Anal. Ser. Nova Sci. Publ., Inc., Commack, NY, 1993.

[65] Toni Volkmer. *Multivariate Approximation and High-Dimensional Sparse FFT Based on Rank-1 Lattice Sampling*. Ph.D, Universitätsverlag Chemnitz, 2017.

[66] Weiqi Wang and Simone Brugiapaglia. Compressive Fourier collocation methods for high-dimensional diffusion equations with periodic boundary conditions. *ArXiv e-prints*, 2022. arxiv:2206.01255.

[67] Harry Yserentant. On the regularity of the electronic Schrödinger equation in Hilbert spaces of mixed derivatives. *Numerische Mathematik*, 98(4):731–759, October 2004.

[68] Harry Yserentant. Sparse grid spaces for the numerical solution of the electronic Schrödinger equation. *Numerische Mathematik*, 101(2):381–389, August 2005.